

Applied Intelligence

Effective hate-speech detection in Twitter data using recurrent neural networks

--Manuscript Draft--

| | |
|--|--|
| Manuscript Number: | |
| Full Title: | Effective hate-speech detection in Twitter data using recurrent neural networks |
| Article Type: | Original Submission |
| Keywords: | text classification, micro-blogging, hate-speech, deep learning, recurrent neural networks, Twitter |
| Corresponding Author: | Georgios K. Pitsilis, Ph.D. Norwegian University of Science and Technology (NTNU) Trondheim, NORWAY |
| Corresponding Author Secondary Information: | |
| Corresponding Author's Institution: | Norwegian University of Science and Technology (NTNU) |
| Corresponding Author's Secondary Institution: | |
| First Author: | Georgios K. Pitsilis, Ph.D. |
| First Author Secondary Information: | |
| Order of Authors: | Georgios K. Pitsilis, Ph.D. Heri Ramampiaro, PhD Helge Langseth, PhD |
| Order of Authors Secondary Information: | |
| Funding Information: | |
| Abstract: | <p>This paper addresses the important problem of discerning hateful content in social media. We propose a detection scheme that is an ensemble of Recurrent Neural Network (RNN) classifiers, and it incorporates various features associated with user-related information, such as the users' tendency towards racism or sexism. These data are fed as input to the above classifiers along with the word frequency vectors derived from the textual content. We evaluate our approach on a publicly available corpus of 16k tweets, and the results demonstrate its effectiveness in comparison to existing state-of-the-art solutions. More specifically, our scheme can successfully distinguish racism and sexism messages from normal text, and achieve higher classification quality than current state-of-the-art algorithms.</p> |

Noname manuscript No.
(will be inserted by the editor)

Effective hate-speech detection in Twitter data using recurrent neural networks

Georgios K. Pitsilis · Heri Ramampiaro ·
Helge Langseth

Received: date / Accepted: date

Abstract This paper addresses the important problem of discerning hateful content in social media. We propose a detection scheme that is an ensemble of *Recurrent Neural Network (RNN)* classifiers, and it incorporates various features associated with user-related information, such as the users' tendency towards *racism* or *sexism*. These data are fed as input to the above classifiers along with the word frequency vectors derived from the textual content. We evaluate our approach on a publicly available corpus of 16k tweets, and the results demonstrate its effectiveness in comparison to existing state-of-the-art solutions. More specifically, our scheme can successfully distinguish racism and sexism messages from normal text, and achieve higher classification quality than current state-of-the-art algorithms.

Keywords text classification · micro-blogging · hate-speech · deep learning · recurrent neural networks · Twitter

1 Introduction

Social media is a very popular way for people to express their opinions publicly and to interact with others online. In aggregation, social media can provide a reflection of public sentiment on various events. Unfortunately, any user engaging online, either on social media, forums or blogs, will always have the risk of

Georgios K. Pitsilis
E-mail: georgios.pitsilis@ntnu.no

Heri Ramampiaro
E-mail: heri.ramampiaro@ntnu.no

Helge Langseth
E-mail: helge.langseth@ntnu.no

Department of Computer Science,
Norwegian University of Science and Technology (NTNU),
NO-7491 Trondheim, Norway

1 being targeted or harassed via abusive language, expressing hate in the form of
2 racism or sexism, with possible impact on his/her on-line experience, and the
3 community in general. The existence of social networking services creates the
4 need for detecting user-generated hateful messages prior to publication. Any
5 published text that is used to express hatred towards some particular group
6 with the intention to humiliate its members is considered a hateful message.
7

8 Although hate speech is protected under the free speech provisions in some
9 countries, e.g. the United States, there are other countries, such as Canada,
10 France, United Kingdom, and Germany, where there are laws prohibiting it
11 as being promoting violence or social disorder. Social media services such as
12 Facebook and Twitter have been criticized for not having done enough to pro-
13 hibit the use of their services for attacking people belonging to some specific
14 race, minority etc. [12]. They have announced though that they would seek
15 to battle against racism and xenophobia [5]. Nevertheless, the current solu-
16 tions deployed by, e.g., Facebook and Twitter have so far been to address the
17 problem with manual effort, relying on users to report offensive comments [3].
18 This not only requires a huge effort by human annotators, but it also has
19 the risk of applying discrimination under subjective judgment. Moreover, a
20 non-automated task by human annotators would have strong impact on sys-
21 tem response time, since a computer-based solution can accomplish this task
22 much faster than humans. The massive rise in the user-generated content in
23 the above social media services, with manual filtering not being scalable, high-
24 lights the need for automating the process of on-line hate-speech detection.
25

26 Despite the fact that the majority of the solutions for automated detec-
27 tion of offensive text rely on Natural Language Processing (NLP) approaches,
28 there have lately been a tendency towards employing pure machine learning
29 techniques like neural networks for that task. NLP approaches have the draw-
30 back of being complex, and to a large extent dependent on the language used
31 in the text. This provides a strong motivation for employing alternative ma-
32 chine learning models for the classification task. Moreover, the majority of
33 the existing automated approaches depend on using pre-trained vectors (e.g.
34 Glove, Word2Vec) as word embeddings to achieve good performance from the
35 classification model. That makes the detection of hatred content unfeasible in
36 cases where users have deliberately obfuscated their offensive terms with short
37 slang words.
38

39 There is a plethora of unsupervised learning models in the existing litera-
40 ture to deal with hate-speech [15], as well as in detecting the sentiment polarity
41 in tweets [2]. At the same time, the supervised learning approaches have still
42 not been explored adequately. While the task of sentence classification seems
43 similar to that of sentiment analysis, in hate-speech even negative sentiment
44 could still provide useful insight. Our intuition is that the task of hate-speech
45 detection can be further benefited by the incorporation of other sources of
46 information to be used as features into a supervised learning model. A simple
47 statistical analysis on an existing annotated dataset of tweets by Waseem [18],
48 can easily reveal the existence of significant correlation between the user ten-
49 dency in expressing opinions that belong to some offensive class (*Racism* or
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

1 *Sexism*), and the annotation labels associated with that class. More precisely,
2 the correlation coefficient value that describes such user tendency was found
3 to be 0.71 for racism in the above dataset, while that value reached as high
4 as 0.76 for sexism. In our opinion, utilizing such user-oriented behavioural
5 data for reinforcing an existing solution is feasible, because such information
6 is retrievable in real-world use-case scenarios like Twitter. This highlights the
7 need to explore the user features more systematically to further improve the
8 classification accuracy of a supervised learning system.
9

10 Our approach employs a neural network solution composed of multiple
11 classifiers based on Long-Short-Term-Memory (LSTM), and utilizes user be-
12 havioral characteristics such as the tendency towards racism or sexism to boost
13 performance. Although our technique is not necessarily revolutionary in terms
14 of the deep learning models used, we show in this paper that it is quite effec-
15 tive.

16 Our main contributions are: *i*) a deep learning architecture for text classi-
17 fication in terms of hateful content, which incorporates features derived from
18 the users' behavioural data, *ii*) a language agnostic solution, due to no-use of
19 pre-trained word embeddings, for detecting hate-speech, *iii*) an experimental
20 evaluation of the model on a Twitter dataset, demonstrating the top perfor-
21 mance achieved on the classification task. Special focus is given to investi-
22 gating how the additional features concerning the users' tendency to utter
23 hate-speech, as expressed by their previous history, could leverage the perfor-
24 mance. To the best of our knowledge, there has not been done any previous
25 study on exploring features related to the users tendency in hatred content
26 that has used a deep learning model.
27

28 The rest of the paper is organized as follows. In Section 2, we describe
29 the problem of hate speech in more detail. In Section 3, we discuss existing
30 related work. In Section 4, we present our proposed model. In Section 5, after
31 presenting the dataset, we describe our experimental evaluation and discuss
32 the results from the experiments. Finally, in Section 6, we conclude the paper
33 and outline possible future work.
34
35

36 2 Problem Statement

37

38 The problem we address in this work can be formally described as follows: Let
39 p be an unlabeled short sentence composed of a number of words, posted by
40 a user u . Let N, S, R be three classes denoting Neutrality, Sexism and Racism
41 in a textual content, respectively. Members of these classes are those postings
42 with content classified as belonging to the corresponding class, for which the
43 following holds: $N \cap S \cap R = \emptyset$. Further, given that user u has a previous history
44 of message postings $P_u : p \notin P_u$, we assume that any previous posting $p_u \in P_u$
45 by that user is already labeled as belonging to any of the classes N, S, R .
46 Similarly, other postings by other users have also been labeled accordingly,
47 forming up their previous history. Based on these facts, the problem is to
48 identify the class, which the unlabeled sentence p by user u belongs to.
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

1 The research question that we address in this work is:

2
3 *How to effectively identify the class of a new posting, given the identity*
4 *of the posting user and the history of postings related to that user?*
5

6 To answer this question, our main goals can be summarized as follows:

- 7
- 8 – To develop a novel method that can improve the state-of-art approaches
 - 9 within hate-speech classification, in terms of classification performance/accuracy.
 - 10 – To investigate the impact of incorporating information about existing per-
 - 11 sonalized labeled postings from users’ past history on the classification
 - 12 performance/accuracy.
 - 13

14 Note that existing solutions for automatic detection are still falling short
15 to effectively detect abusive messages. Therefore, there is a need for new algo-
16 rithms which would do the job of classification of such content more effectively
17 and efficiently. Our work is towards that direction.
18

19 20 **3 Related Work**

21
22 Simple word-based approaches, if used for blocking the posting of text or black-
23 listing users, not only fail to identify subtle offensive content, but they also
24 affect the freedom of speech and expression. The word ambiguity problem –
25 that is, a word can have different meanings in different contexts – is mainly
26 responsible for the high false positive rate in such approaches. Ordinary NLP
27 approaches on the other hand, are ineffective to detect unusual spelling, ex-
28 perimented in user-generated comment text. This is best known as the *spelling*
29 *variation* problem, and it is caused either by unintentional or intentional re-
30 placement of single characters in a token, aiming to obfuscate the detectors.
31 In general, the complexity of the natural language constructs renders the task
32 quite challenging.
33

34 The employment of supervised learning classification methods for hate
35 speech detection is not new. Vigna et al. [16] reported performance for a sim-
36 ple LSTM classifier not better than an ordinary SVM, when evaluated on a
37 small sample of Facebook data for only 2 classes (*Hate*, *No-Hate*), and 3 dif-
38 ferent levels of strength of hatred. Davidson et al. [6] described another way of
39 detecting offensive language in tweets, based on some supervised model. They
40 differentiate hate speech from offensive language, using a classifier that involves
41 naïve Bayes, decision trees and SVM. Also, Nobata et al. [13] attempted to
42 discern abusive content with a supervised model combining various linguistic
43 and syntactic features in the text, considered at character uni-gram and bi-
44 gram level, and tested on Amazon data. In general, we can point out the main
45 weaknesses of NLP-based models in their non-language agnostic nature and
46 the low scores in detection.
47

48 Unsupervised learning approaches are quite common for detecting offensive
49 messages in text by applying concepts from NLP to exploit the lexical syntactic
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

1 features of sentences [4], or using AI-solutions and bag-of-words based text-
2 representations [17]. The latter is known to be less effective for automatic
3 detection, since hatred users apply various obfuscation tricks, such as replacing
4 a single character in offensive words. For instance, applying a binary classifier
5 onto a *paragraph2vec* representation of words has already been attempted on
6 Amazon data in the past by Djuric et al. [7], but it only performed well on
7 a binary classification problem. Another unsupervised learning based solution
8 is the work by Waseem & Hovy [19], in which the authors proposed a set
9 of criteria that a tweet should exhibit in order to be classified as offensive.
10 They also showed that differences in geographic distribution of users have only
11 marginal effect on the detection performance. Despite the above observation,
12 we explore other features that might be possible to improve the detection
13 accuracy in the solution outlined below.
14
15

16
17 The work by Waseem [18] applied a crowd-sourced solution to tackle hate-
18 speech, with the creation of an additional dataset of annotations to extend the
19 existing corpus. The impact of the experience of annotators in the classification
20 performance was investigated. The work by Jha & Mamidi [9] dealt with the
21 classification problem of tweets, but their interest was on sexism alone, which
22 they distinguished into ‘Hostile’, ‘Benevolent’ or ‘Other’. While the authors
23 used the dataset of tweets from Waseem & Hovy [19], they treated the existing
24 ‘*Sexism*’ tweets as being of class ‘*Hostile*’, while they collected their own tweets
25 for the ‘*Benevolent*’ class, on which they finally applied the FastText by Joulin
26 et al. [10], and SVM classification.
27
28

29
30 Badjatiya et al. [1] approached the issue with a supervised learning model
31 that is based on a neural network. Their method achieved higher score over the
32 same dataset of tweets than any unsupervised learning solution known so far.
33 That solution uses an LSTM model, with features extracted by character *n*-
34 *grams*, and assisted by Gradient Boosted Decision Trees. Convolution Neural
35 Networks (CNN) has also been explored as a potential solution in the hate-
36 speech problem in tweets, with character *n*-*grams* and *word2vec* pre-trained
37 vectors being the main tools. For example, Park & Fung [14] transformed the
38 classification into a 2-step problem, where abusive text first is distinguished
39 from the non-abusive, and then the class of abuse (*Sexism* or *Racism*) is deter-
40 mined. Gambäck & Sikdar [8] employed pre-trained CNN vectors in an effort
41 to predict four classes. They achieved slightly higher *F-score* than character
42 *n*-*grams*.
43
44

45
46 In spite of the high popularity of NLP approaches in hate-speech classifi-
47 cation [15], we believe there is still a high potential for deep learning models
48 to further contribute to the issue. At this point it is also relevant to note the
49 inherent difficulty of the challenge itself, which can be clearly noted by the
50 fact that no solution thus far has been able to obtain an F-score above 0.93.
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

4 Description of our Recurrent Neural Network-based Approach

The power of neural networks comes from their ability to find data representations that are useful for classification. Recurrent Neural Networks (RNN) are a special type of neural network, which can be thought of as the addition of loops to the architecture. RNNs use back propagation in the training process to update the network weights in every layer. In our experimentation we used a powerful type of RNN known as *Long Short-Term Memory Network* (LSTM). Inspired by the work by Badjatiya et al. [1], we experiment with combining various LSTM models enhanced with a number of novel features in an ensemble. More specifically we introduce:

- A number of additional features concerned with the users’ tendency towards hatred behaviour.
- An architecture, which combines the output by various LSTM classifiers to improve the classification ability.

4.1 Features

We first elaborate into the details of the features derived to describe each user’s tendency towards each class (*Neutral*, *Racism* or *Sexism*), as captured in their tweeting history. In total, we define the three features t_{Na} , t_{Ra} , t_{Sa} , representing a user’s tendency towards posting *Neutral*, *Racist* and *Sexist* content, respectively. We let m_a denote the set of tweets by user a , and use $m_{N,a}$, $m_{R,a}$ and $m_{S,a}$ to denote the subsets of those tweets that have been labeled as *Neutral*, *Racist* and *Sexist* respectively. Now, the features are calculated as $t_{N,a} = |m_{N,a}|/|m_a|$, $t_{R,a} = |m_{R,a}|/|m_a|$, and $t_{S,a} = |m_{S,a}|/|m_a|$.

Furthermore, we choose to model the input tweets in the form of vectors using word-based frequency vectorization. That is, the words in the corpus are indexed based on their frequency of appearance in the corpus, and the index value of each word in a tweet is used as one of the vector elements to describe that tweet. We note that this modelling choice provides us with a big advantage, because the model is independent of the language used for posting the message.

4.2 Classification

To improve classification ability we employ an ensemble of LSTM-based classifiers. In total the scheme comprises a number of classifiers (3 or 5), each receiving the vectorized tweets together with behavioural features (see Section 4.1) as input.

The choice of various characteristics was done with the purpose to train the neural network with any data associations existing between the attributes for each tweet and the class label given to that tweet. In each case, the characteristic feature is attached to the already computed vectorized content for a

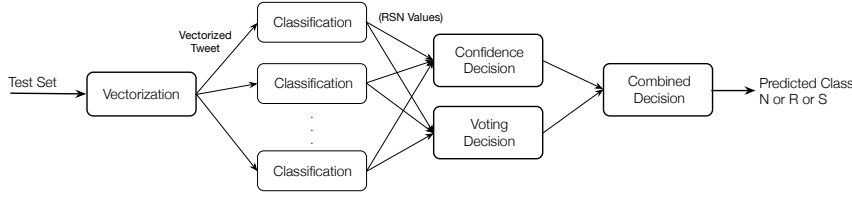


Fig. 1 High level view of the system with multiple classifiers

tweet, thereby providing an input vector for one LSTM classifier. A high level view of the architecture is shown in Figure 1, with the multiple classifiers. The ensemble has two mechanisms for aggregating the classifications from the base classifiers; namely *Voting* and *Confidence*. The preferred method is majority voting, which is employed whenever at least two of the base classifiers agrees wrt. classification of a given tweet. When all classifiers disagree, the classifier with strongest confidence in its prediction is given preference. The conflict resolution logic is implemented in the *Combined Decision* component.

Algorithm 1 Ensemble classifier

```

1: for  $tw \in \{ \text{tweets} \}$  do
2:   for  $cl \in \{ \text{classifiers} \}$  do
3:      $(N_{cl}, R_{cl}, S_{cl}) \leftarrow \text{classifier}_{cl}(tw)$ 
4:      $v_{cl} \leftarrow \max(N_{cl}, R_{cl}, S_{cl})$ 
5:      $id_{cl} \leftarrow \arg \max(N_{cl}, R_{cl}, S_{cl})$ 
6:   end for
7:    $m \leftarrow \text{mode}(id_1, id_2, id_3)$ 
8:   if  $m \in \{ \text{Neutral, Racist, Sexism} \}$  then
9:     decision  $\leftarrow m$ 
10:  else
11:    decision  $\leftarrow id_{\arg \max(v_1, v_2, v_3)}$ 
12:  end if
13:  print decision for  $tw$ 
14: end for
  
```

We present the above process in Algorithm 1. Here *mode* denotes a function that provides the dominant value within the inputs classes id_1, id_2, id_3 and returns NIL if there is a tie, while *classifier* is a function that returns the classification output in the form of a tuple (*Neutral*, *Racism*, *Sexism*).

5 Evaluation setup - Results

5.1 Data Preprocessing

Before training the neural network with the labeled tweets, it is necessary to apply the proper tokenization to every tweet. In this way, the text corpus is split into word elements, taking white spaces and the various punctuation

Table 1 Combined features in the proposed schemes

| Combination | Additional features | Features | Input Dimension |
|-------------|----------------------------|-----------------------------|-----------------|
| O | No additional features | - | 30 |
| NS | Neutral and Sexism | $t_{N,a}, t_{S,a}$ | 32 |
| NR | Neutral and Racism | $t_{N,a}, t_{R,a}$ | 32 |
| RS | Racism and Sexism | $t_{R,a}, t_{S,a}$ | 32 |
| NRS | Neutral, Racism and Sexism | $t_{N,a}, t_{R,a}, t_{S,a}$ | 33 |

symbols used in the language into account. This was done using the Moses¹ package for machine translation.

We choose to limit the maximum size of each tweet to be considered during training to 30 words, and padded tweets of shorter size with zeros. Next, tweets are converted into vectors using word-based frequency, as described in Section 4.1. To feed the various classifiers in our evaluation, we attach the feature values onto every tweet vector.

In this work we experimented with various combinations of attached features $t_{N,a}$, $t_{R,a}$, and $t_{S,a}$ that express the user tendency. The details of each experiment, including the resulting size of each embedding can be found in Table 1, with the latter denoted ‘input dimension’ in the table.

5.2 Deep learning model

In our evaluation of the proposed scheme, each classifier is implemented as a deep learning model having four layers, as illustrated in Figure 2, and is described as follows:

- *The Input (a.k.a Embedding) Layer.* The input layer’s size is defined by the number of inputs for that classifier. This number equals the size to the word vector plus the number of additional features. The word vector dimension was set to 30 so that to be able to encode every word in the vocabulary used.
- *The hidden layer.* The sigmoid activation was selected for the the hidden LSTM layer. Based on preliminary experiments the dimensionality of the output space for this layer was set to 200. This layer is fully connected to both the Input and the subsequent layer.
- *The dense layer.* The output of the LSTM was run through an additional layer to improve the learning and obtain more stable output. The *ReLU* activation function was used. Its size was selected equal to the size of the input layer.
- *The output layer.* This layer has 3 neurons to provide output in the form of probabilities for each of the three classes *Neutral*, *Racism*, and *Sexism*. The softmax activation function was used for this layer.

¹ <http://www.statmt.org/moses/>

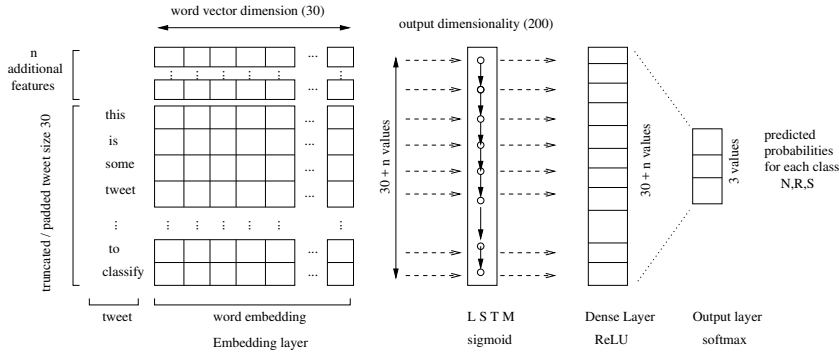


Fig. 2 Our deep learning model

In total we experimented with 11 different setups of the proposed scheme, each with a different ensemble of classifiers, see Table 2.

Table 2 Evaluated ensemble schemes

| Tested Scheme | Classifier | | | | |
|---------------|------------|-----|-----|----|-----|
| | I | II | III | IV | V |
| (i) | O | NRS | NR | - | - |
| (ii) | O | NRS | NS | - | - |
| (iii) | O | NRS | RS | - | - |
| (iv) | O | NS | RS | - | - |
| (v) | O | NS | NR | - | - |
| (vi) | O | RS | NR | - | - |
| (vii) | NRS | NR | RS | - | - |
| (viii) | NRS | NR | NS | - | - |
| (ix) | NRS | NS | RS | - | - |
| (x) | NS | RS | NR | - | - |
| (xi) | O | NS | RS | NR | NRS |

5.3 Dataset

We experimented with a dataset of approximately 16k short messages from Twitter, that was made available by Waseem & Hovy [19]. The dataset contains 1943 tweets labeled as *Racism*, 3166 tweets labeled as *Sexism* and 10889 tweets labeled as *Neutral* (i.e., tweets that neither contain sexism nor racism). There is also a number of dual labeled tweets in the dataset. More particularly, we found 42 tweets labeled both as both ‘Neutral’ and ‘Sexism’, while six tweets were labelled as both ‘Racism’ and ‘Neutral’. According to the dataset providers, the labeling was performed manually.²

² The small discrepancy observed in the class quantities with regard to those mentioned in the original dataset is due to fact that, at the time we performed the evaluation, a number of tweets were not retrievable.

1 The relatively small number of tweets in the dataset makes the task more
2 challenging. As reported by several authors already, the dataset is imbalanced,
3 with a majority of neutral tweets. Additionally, we used the public Twitter
4 API to retrieve additional data associated with the user identity for each tweet
5 in the original dataset.
6

7 8 5.4 Experimental Setting 9

10 To produce results in a setup comparable with the current state of the art
11 [1], we performed 10-fold cross validation and calculated the *Precision*, *Recall*
12 and *F-Score* for every evaluated scheme. We randomly split each training fold
13 into 15% validation and 85% training, while performance is evaluated over the
14 remaining fold of unseen data. The model was implemented using Keras³. We
15 used *categorical cross-entropy* as learning objective, and selected the ADAM
16 optimization algorithm by Kingma & Ba [11]. Furthermore, the vocabulary
17 size was set to 25000, and the batch-size during training was set to 500.
18

19 To avoid over-fitting, the model training was allowed to run for a maximum
20 number of 100 epochs, out of which the optimally trained state was chosen
21 for the model evaluation. An optimal epoch was identified so, such that the
22 validation accuracy was maximized, while at the same time the error remained
23 within $\pm 1\%$ of the lowest ever figure within the current fold. Throughout
24 the experiment we observed that the optimal epochs typically occurred after
25 between the 30 and 40 epochs.
26

27 To achieve stability in the results produced, we ran every single classifier
28 for 15 times and the output values were aggregated. In addition, the output
29 from each single classifier run was combined with the output from another
30 two single classifiers to build the input of an ensemble, producing 15^3 combi-
31 nations. For the case of the ensemble that incorporates all five classifiers we
32 restricted to using the input by only the first five runs of the single classifiers
33 (5^5 combinations). That was due to the prohibitively very large number of
34 combinations that were required.
35

36 37 5.5 Results 38

39 We now present the most interesting results from our experiments.

40 For the evaluation we used standard metrics for classification accuracy,
41 suitable for studying problems such as sentiment analysis. In particular we
42 used *Precision* and *Recall*, with the former calculated as the ratio of the num-
43 ber of tweets correctly classified to a given class over the total number of
44 tweets classified to that class, while the latter measures the ratio of messages
45 correctly classified to a given class over the number of messages from that
46 class. Additionally, the *F-score* is the harmonic mean of precision and recall,
47 expressed as $F = \frac{2 \cdot P \cdot R}{P + R}$. For our particular case with three classes, P , R and
48

49 ³ <https://github.com/fchollet/keras>
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

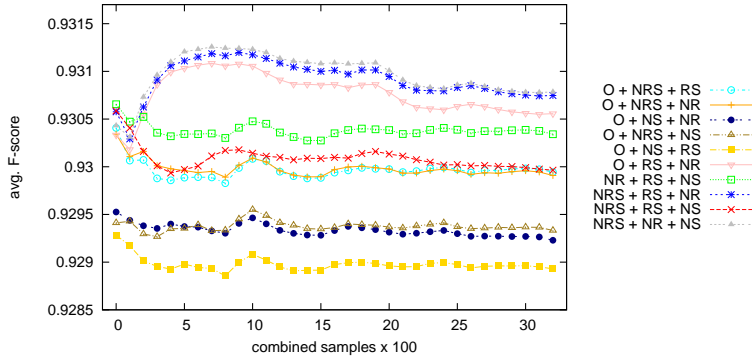


Fig. 3 Aggregated value for F-score vs the number of combined experiment runs

F are computed for each class separately, with the final F value derived as the weighted mean of the separate F -scores: $F = \frac{F_N \cdot N + F_R \cdot R + F_S \cdot S}{N + R + S}$; recall that $N = 10889$, $S = 3166$ and $R = 1943$. The results are shown in Table 3, along with the reported results from state of the art approaches proposed by other researchers in the field. Note that the performance numbers P, R and F of the other state of the art approaches are based on the authors' reported data in the cited works. Additionally, we report the performance of each individual LSTM classifier as if used alone over the same data (that is, without the ensemble logic). The F -score for our proposed approaches shown in the last column, is the weighted average value over the 3 classes (Neutral, Sexism, Racism). Moreover, all the reported values are average values produced for a number of runs of the same tested scheme over the same data. Figure 3 shows the F -Score as a function of the number of training samples for each ensemble of classifiers. We clearly see that the models converge. For the final run the F -score has standard deviation value not larger than 0.001, for all classifiers.

As can be seen in Table 3, the work by Waseem & Hovy [19], in which character n -grams and gender information were used as features, obtained the quite low F -score of 0.7391. Later work by the same author [18] investigated the impact of the experience of the annotator in the performance, but still obtaining a lower F -score than ours. Furthermore, while the second part of the two step classification by Park & Fung [14] performs quite well (reported an F -score of 0.9520) in detecting the particular class the abusive text belongs to, it nevertheless falls short in distinguishing hatred from non-hatred content in general. Finally, we observe that applying a simple LSTM classification in our approach, with no use of additional features (denoted 'single classifier (i)' in Table 3), achieves an F -score that is below 0.93, something that is in line with other researchers in the field, see [1]. Very interestingly, the incorporation of features related to user's behaviour into the classification has provided a significant increase in the performance vs. using the textual content alone, ($F = 0.9295$ vs. $F = 0.9089$).

Table 3 Evaluation Results

| Approach | Characteristics | Precision | Recall | F-Score |
|-------------------------|--|-----------|--------|---------------|
| single classifier (i) | O | 0.9175 | 0.9218 | 0.9196 |
| single classifier (ii) | NS | 0.9246 | 0.9273 | 0.9260 |
| single classifier (iii) | NR | 0.9232 | 0.9259 | 0.9245 |
| single classifier (iv) | RS | 0.9232 | 0.9264 | 0.9248 |
| single classifier (v) | NRS | 0.9252 | 0.9278 | 0.9265 |
| ensemble (i) | O + NRS + NR | 0.9283 | 0.9315 | 0.9298 |
| ensemble (ii) | O + NRS + NS | 0.9288 | 0.9319 | 0.9303 |
| ensemble (iii) | O + NRS + RS | 0.9283 | 0.9315 | 0.9299 |
| ensemble (iv) | O + NS + RS | 0.9277 | 0.9310 | 0.9293 |
| ensemble (v) | O + NS + NR | 0.9276 | 0.9308 | 0.9292 |
| ensemble (vi) | O + RS + NR | 0.9273 | 0.9306 | 0.9290 |
| ensemble (vii) | NRS + NR + RS | 0.9292 | 0.9319 | 0.9306 |
| ensemble (viii) | NRS + NR + NS | 0.9295 | 0.9321 | 0.9308 |
| ensemble (ix) | NRS + NS + RS | 0.9294 | 0.9321 | 0.9308 |
| ensemble (x) | NS + RS + NR | 0.9286 | 0.9314 | 0.9300 |
| ensemble (xi) | O + NS + RS + NR + NRS | 0.9305 | 0.9334 | 0.9320 |
| Badjatiya et al. [1] | LSTM + Random Embedding + GBDT | 0.9300 | 0.9300 | 0.9300 |
| Waseem & Hovy [19] | Unsupervised List of Criteria | 0.7290 | 0.7774 | 0.7391 |
| Waseem [18] | Unsupervised Expert annotators only | 0.9159 | 0.9292 | 0.9153 |
| Park & Fung [14] | 2 step HybridCNN (Word Vec. / Char Vec.) | 0.8270 | 0.8270 | 0.8270 |

Another interesting finding is the observed performance improvement by using an ensemble instead of a single classifier; some ensembles outperform the best single classifier. Furthermore, the NRS classifier, which produces the best score in relation to other single classifiers, is the one included in the best performing ensemble.

In comparison to the approach by Jha & Mamidi [9], which focuses on various classes of Sexism, the results show that our deep learning model is doing better as far as detecting Sexism in general, outperforming the FastText algorithm they have included in their experimental models (F=0.87). The inferiority of FastText over LSTM is also reported in the work by Badjatiya et al. [1], as well as being inferior over CNN by Park & Fung [14]. In general, through our ensemble schemes is confirmed that deep learning can outperform any NLP-based approaches known so far in the task of abusive language detection.

We also present the performance of each of the tested models per class label in Table 4. Results by other researchers have not been included, as these figures are not reported in the existing literature. As can be seen, *sexism* is quite easy to classify in hate-speech, while *racism* seems to be harder; similar results were reported by Davidson et al. [6]. This result is consistent across all ensembles.

For completion, the confusion matrices of the best performing approach that employs 3 classifiers (ensemble *viii*) as well as of the ensemble of all 5

Table 4 Detailed Results for every Class Label

| Proposed Approach | Class | Precision | Recall | F-Score |
|-------------------|----------------|-----------|--------|---------------|
| ensemble (viii) | <i>Neutral</i> | 0.9409 | 0.9609 | 0.9508 |
| | <i>Racism</i> | 0.7522 | 0.6646 | 0.7057 |
| | <i>Sexism</i> | 0.9991 | 0.9972 | 0.9981 |
| ensemble (ix) | <i>Neutral</i> | 0.9407 | 0.9612 | 0.9508 |
| | <i>Racism</i> | 0.7533 | 0.6627 | 0.7051 |
| | <i>Sexism</i> | 0.9986 | 0.9972 | 0.9979 |
| ensemble (vii) | <i>Neutral</i> | 0.9405 | 0.9611 | 0.9507 |
| | <i>Racism</i> | 0.7522 | 0.6616 | 0.7040 |
| | <i>Sexism</i> | 0.9990 | 0.9975 | 0.9983 |
| ensemble (xi) | <i>Neutral</i> | 0.9406 | 0.9631 | 0.9517 |
| | <i>Racism</i> | 0.7623 | 0.6617 | 0.7084 |
| | <i>Sexism</i> | 0.9992 | 0.9980 | 0.9986 |

Table 5 Confusion Matrices of Results for the best performing ensembles with 3 and 5 classifiers.

| ensemble (viii) | | <i>Predicted Label</i> | | | |
|-------------------|----------------|------------------------|---------------|----------------|-------------------------------------|
| <i>True Label</i> | | Racism | Sexism | Neutral | sum |
| | Racism | 10655320 | 5635 | 24295 | 10685250 |
| | Sexism | 3943 | 4357971 | 2195711 | 6557625 |
| | Neutral | 5929 | 1430030 | 35314416 | 36750375 |
| | sum | 10665192 | 5793636 | 37534422 | 53993250 (15998×15^3) |
| ensemble (xi) | | <i>Predicted Label</i> | | | |
| <i>True Label</i> | | Racism | Sexism | Neutral | sum |
| | Racism | 9873754 | 991 | 19005 | 9893750 |
| | Sexism | 3034 | 4017687 | 2051154 | 6071875 |
| | Neutral | 4446 | 1252093 | 32771586 | 34028125 |
| | sum | 9881234 | 5270771 | 34841745 | 49993750 (15998×5^5) |

classifiers (xi), are provided in Table 5. The presented values is the sum over multiple runs.

To study the effect of the user’s tendency in haste-speech in the F-score, we provide a brake-down of the computed values over five classes of users. Therefore, we divided the complete set of users into five subsets of equal size, wrt. their tendency on sexism or racism, and computed the F-score independently for each user class. We present the results for each individual classifier as well as for all the ensembles of classifiers we tested.

In Figure 4 we present the F-score achieved for each classifier over the five classes of users. In class 1 belong those users having the lowest tendency, while class 5 contains the users with the highest tendency. As can be seen in the above figure, tweets by users who are more tempted to hate speech are easier to detect by our algorithm, than the less tempted ones. Very interestingly, this characteristic works better for sexism rather for racism. Quite impressive is the fact that the F-Score for the most tempted users can reach as high as 0.995, no matter which classifier used. In addition, classifier (O), which does

not make use of user features, performs slightly worse, for the full range of classes of tendency.

From the output shown in Figure 4 we observe that the classification works quite effectively, detecting almost all cases of abusive content originated from the most tended users, something that is inline with our primary objective. In overall, the above observations confirm the original hypothesis of the classification accuracy being improved with the employment of additional user-based features into the prediction mechanism. For completion we also report the

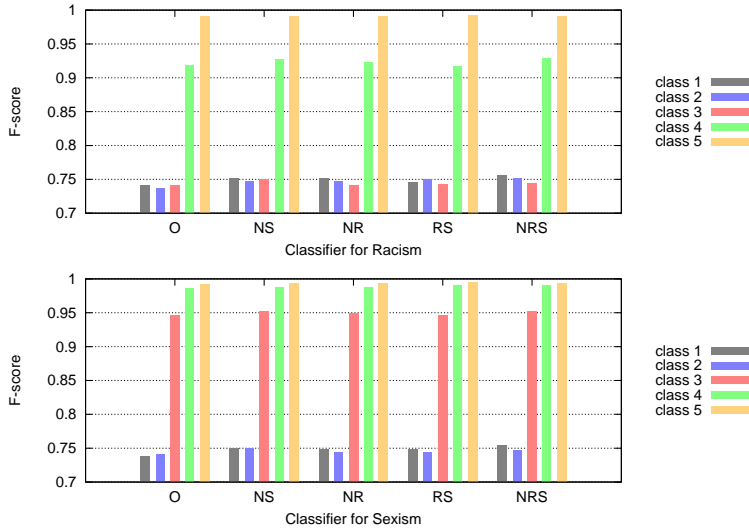


Fig. 4 F-score for various classes of users for single classifiers

F-score for all the ensembles of classifiers for every particular users class in Figures 5 and 6. As can be seen, for the case of ensembles, our approach has similar and equally good performance with that achieved by the use of individual classifiers. We also observe that, for the classes of users who are less tending towards sexism or racism, the 5-classifiers ensemble achieves the best performance in comparison to the other schemes.

Another interesting result is presented in Figure 7. It shows the Receiver Operative Characteristic (ROC) curves of all single classifiers we introduced. ROC values gives the ability to assess a classifier's performance over its entire operating range of the chosen thresholds used for separating one class from another. Also, it provides visualization of the trade-offs between sensitivity and specificity, so that finally an optimal model can be selected. To compute the ROC curves for 3-class label output, we applied the following rationale: For each classifier scheme, we firstly take each prediction that is essentially the output of the softmax activation function, and then we apply, in separate for

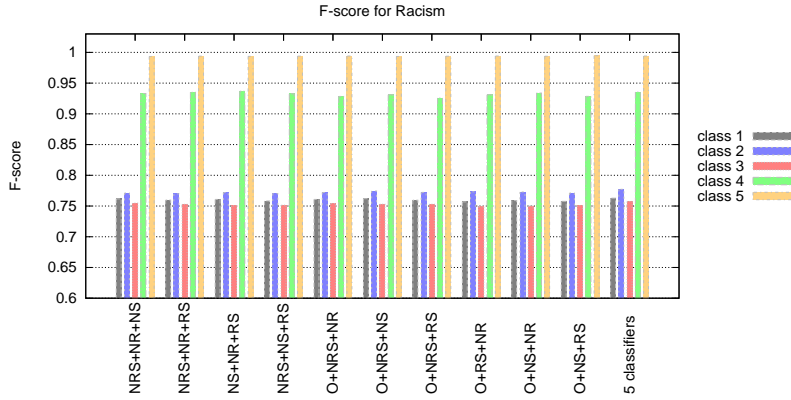


Fig. 5 F-score for various classes of users with tendency in racism for ensemble classifiers

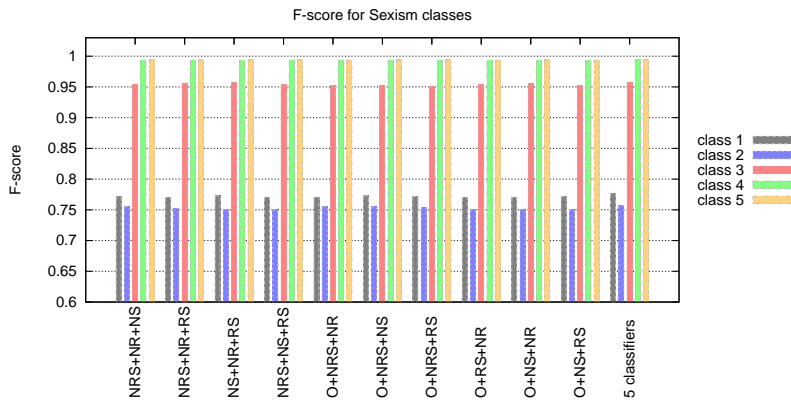


Fig. 6 F-score for various classes of users with tendency in sexism for ensemble classifiers

each class label value (Neutral, Sexism, Racism), a threshold to classify a tweet as belonging to that class. Next, we compute the True Positive Ratio and False Positive Ratio as a function of $tpr = \frac{tp}{tp+fn}$ and $fpr = \frac{fp}{fp+tn}$ respectively; and finally, the resulting values are averaged over the 3 classes of Neutral, Sexism and Racism. The above steps are repeated for a range of threshold values between (0.0 and 1.0) to produce the output finally demonstrated in the ROC curve for that classifier.

To express the resulting performance of a classifier in the form of numerical score we compute the Area Under Curve (AUC) value for each one (see Table 6). The figures show that NS is the best performing classifier achieving AUC value of 0.8406. While all the other single classifiers performed slightly worse, they still achieved high score that falls within the range between 0.8 and 0.9, which is characteristic of a good performing model. Also computed the AUC values for each of the 5 classes of users with regard their tendency in sexism or racism (see Table 6). The above results also confirm the opti-

mal performance achieved by the model in the task of separating the hateful content from the non-hateful one, when the posting is originated from users belonging to a class of high tendency towards sexism or racism.

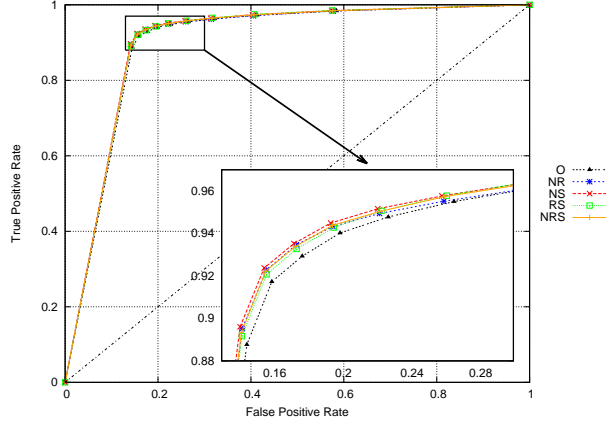


Fig. 7 ROC comparison for all single classifiers

Table 6 Area Under the Curve (AUC) of ROC for single classifiers

| Classifier | O | NR | NS | RS | NRS |
|-------------------|--------|---------------|---------------|---------------|--------|
| user class | | | | | |
| overall | 0.8354 | 0.8382 | 0.8406 | 0.8395 | 0.8395 |
| rac-1 | 0.6888 | 0.6934 | 0.6974 | 0.6956 | 0.6973 |
| rac-2 | 0.6864 | 0.6909 | 0.6951 | 0.6917 | 0.6899 |
| rac-3 | 0.6858 | 0.6874 | 0.6953 | 0.6889 | 0.6898 |
| rac-4 | 0.8364 | 0.8378 | 0.8401 | 0.8383 | 0.8404 |
| rac-5 | 0.8855 | 0.8859 | 0.8894 | 0.8897 | 0.8894 |
| sex-1 | 0.6868 | 0.6918 | 0.6958 | 0.6957 | 0.6927 |
| sex-2 | 0.6869 | 0.6885 | 0.6953 | 0.6879 | 0.6912 |
| sex-3 | 0.8556 | 0.8581 | 0.8597 | 0.8585 | 0.8595 |
| sex-4 | 0.8818 | 0.8817 | 0.8855 | 0.8855 | 0.8867 |
| sex-5 | 0.8808 | 0.8853 | 0.8881 | 0.8873 | 0.8846 |

Finally, we need to point out that our approach does not rely on pre-trained vectors, which provides a important advantage when dealing with short messages of this kind. More specifically, users will often prefer to obfuscate their offensive terms using shorter slang words or create new words by ‘inventive’ spelling and word concatenation. For instance, the word ‘Islamolunatic’ is not available in the popular pre-trained word embeddings (Word2Vec or GloVe), even though it appears with a rather high frequency in racist postings. Hence,

word frequency vectorization is preferable to the pre-trained word embeddings used in prior works, in order to build a language-agnostic solution.

6 Conclusions and Future Work

Automated detection of abusive language in on-line media has in the recent years become a key challenge. In this paper we have presented an ensemble classifier to detect hate-speech in short text, such as tweets. Our classifier uses deep learning and incorporates a series of features associated with users' behavioral characteristics, such as the tendency to post abusive messages, as input to the classifier. In summary, this paper has made several main contributions in order to advance the state-of-the-art. First, we have developed a deep learning architecture that uses word frequency vectorisation for implementing the above features. Second, we have proposed a method that, due to no-use of pre-trained word embeddings, is language independent. Third, we have done thorough evaluation of our model using a public dataset of labeled tweets, an open-sourced implementation built on top of Keras. This evaluation also includes an analysis of the performance of the proposed scheme for various classes of users. The experimental results have shown that our approach outperforms the current state-of-the-art approaches, and to the best of our knowledge, no other model has achieved better performance in classifying short messages. Also, the results have confirmed the original hypothesis of improving the classifier's performance by employing additional user-based features into the prediction mechanism.

For future work, we plan to investigate other sources of information that can be utilized to detect hateful messages. In addition, we intend to generalize the output gained in the current experiment, with evaluation over other datasets, including analyzing texts written in different languages.

Acknowledgements This work has been supported by Telenor Research, Norway, through the collaboration project between NTNU and Telenor. It has been carried out at the Telenor – NTNU AI-Lab.

References

1. Badjatiya, P., Gupta, S., Gupta, M., & Varma, V. (2017). Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion* (pp. 759–760). International World Wide Web Conferences Steering Committee.
2. Barnaghi, P., Ghaffari, P., & Breslin, J. G. (2016). Opinion mining and sentiment polarity on twitter and correlation between events and sentiment. In *2nd IEEE International Conference on Big Data Computing Service and Applications (BigDataService)* (pp. 52–57).

- 1 3. BBC (2016). Facebook, Google and Twitter agree german hate speech
2 deal. Website. <http://www.bbc.com/news/world-europe-35105003> Ac-
3 cessed: on 26/11/2016.
- 4 4. Chen, Y., Zhou, Y., Zhu, S., & Xu, H. (2012). Detecting offensive language
5 in social media to protect adolescent online safety. In *2012 International*
6 *Conference on Privacy, Security, Risk and Trust (PASSAT 2012)*, and
7 *2012 International Conference on Social Computing (SocialCom 2012)*
8 (pp. 71–80).
- 9 5. DailyMail (2016). Zuckerberg in Germany: No place for hate speech
10 on Facebook. Website. [http://www.dailymail.co.uk/wires/ap/](http://www.dailymail.co.uk/wires/ap/article-3465562/Zuckerberg-no-place-hate-speech-Facebook.html)
11 [article-3465562/Zuckerberg-no-place-hate-speech-Facebook.](http://www.dailymail.co.uk/wires/ap/article-3465562/Zuckerberg-no-place-hate-speech-Facebook.html)
12 [html](http://www.dailymail.co.uk/wires/ap/article-3465562/Zuckerberg-no-place-hate-speech-Facebook.html) Accessed: on 26/02/2016.
- 13 6. Davidson, T., Warmlesley, D., Macy, M. W., & Weber, I. (2017). Auto-
14 mated hate speech detection and the problem of offensive language. In
15 *Proceedings of the 11th International Conference on Web and Social Me-*
16 *dia (ICWSM 2017)* (pp. 512–515).
- 17 7. Djuric, N., Zhou, J., Morris, R., Grbovic, M., Radosavljevic, V., &
18 Bhamidipati, N. (2015). Hate speech detection with comment embeddings.
19 In *Proceedings of the 24th International Conference on World Wide Web*
20 *Companion* (pp. 29–30). ACM.
- 21 8. Gambäck, B., & Sikdar, U. K. (2017). Using convolutional neural networks
22 to classify hate-speech. In *Proceedings of the 1st Workshop on Abusive*
23 *Language Online at ACL 2017*.
- 24 9. Jha, A., & Mamidi, R. (2017). When does a compliment become sexist?
25 analysis and classification of ambivalent sexism using twitter data. In
26 *Proceedings of the Second Workshop on NLP and Computational Social*
27 *Science* (pp. 7–16). Association for Computational Linguistics.
- 28 10. Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks
29 for efficient text classification. *arXiv preprint arXiv:1607.01759*, .
- 30 11. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic opti-
31 mization. In *Proceedings of the 3rd International Conference on Learning*
32 *Representations (ICLR 2014)*.
- 33 12. NewYorkTimes (2017). Twitter must do more to block isis.
34 Website. [https://www.nytimes.com/2017/01/13/opinion/](https://www.nytimes.com/2017/01/13/opinion/twitter-must-do-more-to-block-isis.html)
35 [twitter-must-do-more-to-block-isis.html](https://www.nytimes.com/2017/01/13/opinion/twitter-must-do-more-to-block-isis.html) Accessed: on 30/09/2017.
- 36 13. Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., & Chang, Y. (2016).
37 Abusive language detection in online user content. In *Proceedings of the*
38 *25th International Conference on World Wide Web (WWW 2016)* (pp.
39 145–153). International World Wide Web Conferences Steering Commit-
40 tee.
- 41 14. Park, J. H., & Fung, P. (2017). One-step and two-step classification for
42 abusive language detection on twitter. In *Proceedings of the 1st Workshop*
43 *on Abusive Language Online at ACL 2017*.
- 44 15. Schmidt, A., & Wiegand, M. (2017). A survey on hate speech detection
45 using natural language processing. In *Proceedings of the 5th International*
46 *Workshop on Natural Language Processing for Social Media* (pp. 1–10).
- 47
- 48
- 49
- 50
- 51
- 52
- 53
- 54
- 55
- 56
- 57
- 58
- 59
- 60
- 61
- 62
- 63
- 64
- 65

- 1 Association for Computational Linguistics.
- 2
- 3 16. Vigna, F. D., Cimino, A., Dell’Orletta, F., Petrocchi, M., & Tesconi, M.
- 4 (2017). Hate me, hate me not: Hate speech detection on facebook. In
- 5 *Proceedings of the 1st Italian Conference on Cybersecurity (ITASEC17)*
- 6 (pp. 86–95). URL: <http://ceur-ws.org/Vol-1816/paper-09.pdf>.
- 7 17. Warner, W., & Hirschberg, J. (2012). Detecting hate speech on the world
- 8 wide web. In *Proceedings of the 2nd Workshop on Language in Social*
- 9 *Media (LSM 2012)* LSM ’12 (pp. 19–26). Association for Computational
- 10 Linguistics.
- 11 18. Waseem, Z. (2016). Are you a racist or am i seeing things? annotator
- 12 influence on hate speech detection on twitter. In *Proceedings of the First*
- 13 *Workshop on NLP and Computational Social Science* (pp. 138–142). As-
- 14 sociation for Computational Linguistics.
- 15 19. Waseem, Z., & Hovy, D. (2016). Hateful symbols or hateful people? pre-
- 16 dictive features for hate speech detection on twitter. In *Proceedings of*
- 17 *the NAACL Student Research Workshop*. Association for Computational
- 18 Linguistics.
- 19
- 20
- 21
- 22
- 23
- 24
- 25
- 26
- 27
- 28
- 29
- 30
- 31
- 32
- 33
- 34
- 35
- 36
- 37
- 38
- 39
- 40
- 41
- 42
- 43
- 44
- 45
- 46
- 47
- 48
- 49
- 50
- 51
- 52
- 53
- 54
- 55
- 56
- 57
- 58
- 59
- 60
- 61
- 62
- 63
- 64
- 65

Noname manuscript No.
(will be inserted by the editor)

Effective hate-speech detection in Twitter data using recurrent neural networks

Georgios K. Pitsilis · Heri Ramampiaro ·
Helge Langseth

Received: date / Accepted: date

Abstract This paper addresses the important problem of discerning hateful content in social media. We propose a detection scheme that is an ensemble of *Recurrent Neural Network (RNN)* classifiers, and it incorporates various features associated with user-related information, such as the users' tendency towards *racism* or *sexism*. These data are fed as input to the above classifiers along with the word frequency vectors derived from the textual content. We evaluate our approach on a publicly available corpus of 16k tweets, and the results demonstrate its effectiveness in comparison to existing state-of-the-art solutions. More specifically, our scheme can successfully distinguish racism and sexism messages from normal text, and achieve higher classification quality than current state-of-the-art algorithms.

Keywords text classification · micro-blogging · hate-speech · deep learning · recurrent neural networks · Twitter

1 Introduction

Social media is a very popular way for people to express their opinions publicly and to interact with others online. In aggregation, social media can provide a reflection of public sentiment on various events. Unfortunately, any user engaging online, either on social media, forums or blogs, will always have the risk of

Georgios K. Pitsilis
E-mail: georgios.pitsilis@ntnu.no

Heri Ramampiaro
E-mail: heri.ramampiaro@ntnu.no

Helge Langseth
E-mail: helge.langseth@ntnu.no

Department of Computer Science,
Norwegian University of Science and Technology (NTNU),
NO-7491 Trondheim, Norway

1 being targeted or harassed via abusive language, expressing hate in the form of
2 racism or sexism, with possible impact on his/her on-line experience, and the
3 community in general. The existence of social networking services creates the
4 need for detecting user-generated hateful messages prior to publication. Any
5 published text that is used to express hatred towards some particular group
6 with the intention to humiliate its members is considered a hateful message.
7

8 Although hate speech is protected under the free speech provisions in some
9 countries, e.g. the United States, there are other countries, such as Canada,
10 France, United Kingdom, and Germany, where there are laws prohibiting it
11 as being promoting violence or social disorder. Social media services such as
12 Facebook and Twitter have been criticized for not having done enough to pro-
13 hibit the use of their services for attacking people belonging to some specific
14 race, minority etc. [12]. They have announced though that they would seek
15 to battle against racism and xenophobia [5]. Nevertheless, the current solu-
16 tions deployed by, e.g., Facebook and Twitter have so far been to address the
17 problem with manual effort, relying on users to report offensive comments [3].
18 This not only requires a huge effort by human annotators, but it also has
19 the risk of applying discrimination under subjective judgment. Moreover, a
20 non-automated task by human annotators would have strong impact on sys-
21 tem response time, since a computer-based solution can accomplish this task
22 much faster than humans. The massive rise in the user-generated content in
23 the above social media services, with manual filtering not being scalable, high-
24 lights the need for automating the process of on-line hate-speech detection.
25

26 Despite the fact that the majority of the solutions for automated detec-
27 tion of offensive text rely on Natural Language Processing (NLP) approaches,
28 there have lately been a tendency towards employing pure machine learning
29 techniques like neural networks for that task. NLP approaches have the draw-
30 back of being complex, and to a large extent dependent on the language used
31 in the text. This provides a strong motivation for employing alternative ma-
32 chine learning models for the classification task. Moreover, the majority of
33 the existing automated approaches depend on using pre-trained vectors (e.g.
34 Glove, Word2Vec) as word embeddings to achieve good performance from the
35 classification model. That makes the detection of hatred content unfeasible in
36 cases where users have deliberately obfuscated their offensive terms with short
37 slang words.
38

39 There is a plethora of unsupervised learning models in the existing litera-
40 ture to deal with hate-speech [15], as well as in detecting the sentiment polarity
41 in tweets [2]. At the same time, the supervised learning approaches have still
42 not been explored adequately. While the task of sentence classification seems
43 similar to that of sentiment analysis, in hate-speech even negative sentiment
44 could still provide useful insight. Our intuition is that the task of hate-speech
45 detection can be further benefited by the incorporation of other sources of
46 information to be used as features into a supervised learning model. A simple
47 statistical analysis on an existing annotated dataset of tweets by Waseem [18],
48 can easily reveal the existence of significant correlation between the user ten-
49 dency in expressing opinions that belong to some offensive class (*Racism* or
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

1 *Sexism*), and the annotation labels associated with that class. More precisely,
2 the correlation coefficient value that describes such user tendency was found
3 to be 0.71 for racism in the above dataset, while that value reached as high
4 as 0.76 for sexism. In our opinion, utilizing such user-oriented behavioural
5 data for reinforcing an existing solution is feasible, because such information
6 is retrievable in real-world use-case scenarios like Twitter. This highlights the
7 need to explore the user features more systematically to further improve the
8 classification accuracy of a supervised learning system.
9

10 Our approach employs a neural network solution composed of multiple
11 classifiers based on Long-Short-Term-Memory (LSTM), and utilizes user be-
12 havioral characteristics such as the tendency towards racism or sexism to boost
13 performance. Although our technique is not necessarily revolutionary in terms
14 of the deep learning models used, we show in this paper that it is quite effec-
15 tive.

16 Our main contributions are: *i*) a deep learning architecture for text classi-
17 fication in terms of hateful content, which incorporates features derived from
18 the users' behavioural data, *ii*) a language agnostic solution, due to no-use of
19 pre-trained word embeddings, for detecting hate-speech, *iii*) an experimental
20 evaluation of the model on a Twitter dataset, demonstrating the top perfor-
21 mance achieved on the classification task. Special focus is given to investi-
22 gating how the additional features concerning the users' tendency to utter
23 hate-speech, as expressed by their previous history, could leverage the perfor-
24 mance. To the best of our knowledge, there has not been done any previous
25 study on exploring features related to the users tendency in hatred content
26 that has used a deep learning model.
27

28 The rest of the paper is organized as follows. In Section 2, we describe
29 the problem of hate speech in more detail. In Section 3, we discuss existing
30 related work. In Section 4, we present our proposed model. In Section 5, after
31 presenting the dataset, we describe our experimental evaluation and discuss
32 the results from the experiments. Finally, in Section 6, we conclude the paper
33 and outline possible future work.
34
35

36 2 Problem Statement

37

38 The problem we address in this work can be formally described as follows: Let
39 p be an unlabeled short sentence composed of a number of words, posted by
40 a user u . Let N, S, R be three classes denoting Neutrality, Sexism and Racism
41 in a textual content, respectively. Members of these classes are those postings
42 with content classified as belonging to the corresponding class, for which the
43 following holds: $N \cap S \cap R = \emptyset$. Further, given that user u has a previous history
44 of message postings $P_u : p \notin P_u$, we assume that any previous posting $p_u \in P_u$
45 by that user is already labeled as belonging to any of the classes N, S, R .
46 Similarly, other postings by other users have also been labeled accordingly,
47 forming up their previous history. Based on these facts, the problem is to
48 identify the class, which the unlabeled sentence p by user u belongs to.
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

1 The research question that we address in this work is:

2
3 *How to effectively identify the class of a new posting, given the identity*
4 *of the posting user and the history of postings related to that user?*
5

6 To answer this question, our main goals can be summarized as follows:

- 7
8 – To develop a novel method that can improve the state-of-art approaches
9 within hate-speech classification, in terms of classification performance/accuracy.
10 – To investigate the impact of incorporating information about existing per-
11 sonalized labeled postings from users’ past history on the classification
12 performance/accuracy.
13

14 Note that existing solutions for automatic detection are still falling short
15 to effectively detect abusive messages. Therefore, there is a need for new algo-
16 rithms which would do the job of classification of such content more effectively
17 and efficiently. Our work is towards that direction.
18
19

20 **3 Related Work**

21
22 Simple word-based approaches, if used for blocking the posting of text or black-
23 listing users, not only fail to identify subtle offensive content, but they also
24 affect the freedom of speech and expression. The word ambiguity problem –
25 that is, a word can have different meanings in different contexts – is mainly
26 responsible for the high false positive rate in such approaches. Ordinary NLP
27 approaches on the other hand, are ineffective to detect unusual spelling, ex-
28 perimented in user-generated comment text. This is best known as the *spelling*
29 *variation* problem, and it is caused either by unintentional or intentional re-
30 placement of single characters in a token, aiming to obfuscate the detectors.
31 In general, the complexity of the natural language constructs renders the task
32 quite challenging.
33

34 The employment of supervised learning classification methods for hate
35 speech detection is not new. Vigna et al. [16] reported performance for a sim-
36 ple LSTM classifier not better than an ordinary SVM, when evaluated on a
37 small sample of Facebook data for only 2 classes (*Hate*, *No-Hate*), and 3 dif-
38 ferent levels of strength of hatred. Davidson et al. [6] described another way of
39 detecting offensive language in tweets, based on some supervised model. They
40 differentiate hate speech from offensive language, using a classifier that involves
41 naïve Bayes, decision trees and SVM. Also, Nobata et al. [13] attempted to
42 discern abusive content with a supervised model combining various linguistic
43 and syntactic features in the text, considered at character uni-gram and bi-
44 gram level, and tested on Amazon data. In general, we can point out the main
45 weaknesses of NLP-based models in their non-language agnostic nature and
46 the low scores in detection.
47

48 Unsupervised learning approaches are quite common for detecting offensive
49 messages in text by applying concepts from NLP to exploit the lexical syntactic
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

1 features of sentences [4], or using AI-solutions and bag-of-words based text-
2 representations [17]. The latter is known to be less effective for automatic
3 detection, since hatred users apply various obfuscation tricks, such as replacing
4 a single character in offensive words. For instance, applying a binary classifier
5 onto a *paragraph2vec* representation of words has already been attempted on
6 Amazon data in the past by Djuric et al. [7], but it only performed well on
7 a binary classification problem. Another unsupervised learning based solution
8 is the work by Waseem & Hovy [19], in which the authors proposed a set
9 of criteria that a tweet should exhibit in order to be classified as offensive.
10 They also showed that differences in geographic distribution of users have only
11 marginal effect on the detection performance. Despite the above observation,
12 we explore other features that might be possible to improve the detection
13 accuracy in the solution outlined below.
14

15
16
17 The work by Waseem [18] applied a crowd-sourced solution to tackle hate-
18 speech, with the creation of an additional dataset of annotations to extend the
19 existing corpus. The impact of the experience of annotators in the classification
20 performance was investigated. The work by Jha & Mamidi [9] dealt with the
21 classification problem of tweets, but their interest was on sexism alone, which
22 they distinguished into ‘Hostile’, ‘Benevolent’ or ‘Other’. While the authors
23 used the dataset of tweets from Waseem & Hovy [19], they treated the existing
24 ‘*Sexism*’ tweets as being of class ‘*Hostile*’, while they collected their own tweets
25 for the ‘*Benevolent*’ class, on which they finally applied the FastText by Joulin
26 et al. [10], and SVM classification.
27

28
29 Badjatiya et al. [1] approached the issue with a supervised learning model
30 that is based on a neural network. Their method achieved higher score over the
31 same dataset of tweets than any unsupervised learning solution known so far.
32 That solution uses an LSTM model, with features extracted by character *n*-
33 *grams*, and assisted by Gradient Boosted Decision Trees. Convolution Neural
34 Networks (CNN) has also been explored as a potential solution in the hate-
35 speech problem in tweets, with character *n*-*grams* and *word2vec* pre-trained
36 vectors being the main tools. For example, Park & Fung [14] transformed the
37 classification into a 2-step problem, where abusive text first is distinguished
38 from the non-abusive, and then the class of abuse (*Sexism* or *Racism*) is deter-
39 mined. Gambäck & Sikdar [8] employed pre-trained CNN vectors in an effort
40 to predict four classes. They achieved slightly higher *F-score* than character
41 *n*-*grams*.
42

43
44
45 In spite of the high popularity of NLP approaches in hate-speech classifi-
46 cation [15], we believe there is still a high potential for deep learning models
47 to further contribute to the issue. At this point it is also relevant to note the
48 inherent difficulty of the challenge itself, which can be clearly noted by the
49 fact that no solution thus far has been able to obtain an F-score above 0.93.
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

4 Description of our Recurrent Neural Network-based Approach

The power of neural networks comes from their ability to find data representations that are useful for classification. Recurrent Neural Networks (RNN) are a special type of neural network, which can be thought of as the addition of loops to the architecture. RNNs use back propagation in the training process to update the network weights in every layer. In our experimentation we used a powerful type of RNN known as *Long Short-Term Memory Network* (LSTM). Inspired by the work by Badjatiya et al. [1], we experiment with combining various LSTM models enhanced with a number of novel features in an ensemble. More specifically we introduce:

- A number of additional features concerned with the users’ tendency towards hatred behaviour.
- An architecture, which combines the output by various LSTM classifiers to improve the classification ability.

4.1 Features

We first elaborate into the details of the features derived to describe each user’s tendency towards each class (*Neutral*, *Racism* or *Sexism*), as captured in their tweeting history. In total, we define the three features t_{Na} , t_{Ra} , t_{Sa} , representing a user’s tendency towards posting *Neutral*, *Racist* and *Sexist* content, respectively. We let m_a denote the set of tweets by user a , and use $m_{N,a}$, $m_{R,a}$ and $m_{S,a}$ to denote the subsets of those tweets that have been labeled as *Neutral*, *Racist* and *Sexist* respectively. Now, the features are calculated as $t_{N,a} = |m_{N,a}|/|m_a|$, $t_{R,a} = |m_{R,a}|/|m_a|$, and $t_{S,a} = |m_{S,a}|/|m_a|$.

Furthermore, we choose to model the input tweets in the form of vectors using word-based frequency vectorization. That is, the words in the corpus are indexed based on their frequency of appearance in the corpus, and the index value of each word in a tweet is used as one of the vector elements to describe that tweet. We note that this modelling choice provides us with a big advantage, because the model is independent of the language used for posting the message.

4.2 Classification

To improve classification ability we employ an ensemble of LSTM-based classifiers. In total the scheme comprises a number of classifiers (3 or 5), each receiving the vectorized tweets together with behavioural features (see Section 4.1) as input.

The choice of various characteristics was done with the purpose to train the neural network with any data associations existing between the attributes for each tweet and the class label given to that tweet. In each case, the characteristic feature is attached to the already computed vectorized content for a

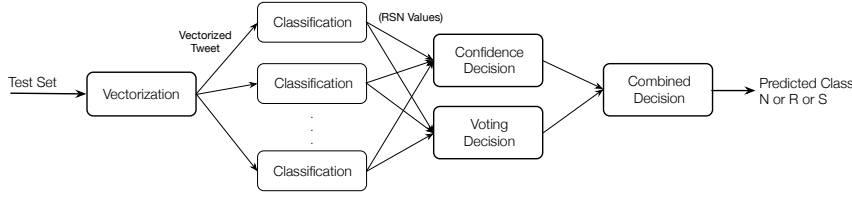


Fig. 1 High level view of the system with multiple classifiers

tweet, thereby providing an input vector for one LSTM classifier. A high level view of the architecture is shown in Figure 1, with the multiple classifiers. The ensemble has two mechanisms for aggregating the classifications from the base classifiers; namely *Voting* and *Confidence*. The preferred method is majority voting, which is employed whenever at least two of the base classifiers agrees wrt. classification of a given tweet. When all classifiers disagree, the classifier with strongest confidence in its prediction is given preference. The conflict resolution logic is implemented in the *Combined Decision* component.

Algorithm 1 Ensemble classifier

```

1: for  $tw \in \{ \text{tweets} \}$  do
2:   for  $cl \in \{ \text{classifiers} \}$  do
3:      $(N_{cl}, R_{cl}, S_{cl}) \leftarrow \text{classifier}_{cl}(tw)$ 
4:      $v_{cl} \leftarrow \max(N_{cl}, R_{cl}, S_{cl})$ 
5:      $id_{cl} \leftarrow \arg \max(N_{cl}, R_{cl}, S_{cl})$ 
6:   end for
7:    $m \leftarrow \text{mode}(id_1, id_2, id_3)$ 
8:   if  $m \in \{ \text{Neutral, Racist, Sexism} \}$  then
9:     decision  $\leftarrow m$ 
10:  else
11:    decision  $\leftarrow id_{\arg \max(v_1, v_2, v_3)}$ 
12:  end if
13:  print decision for  $tw$ 
14: end for
  
```

We present the above process in Algorithm 1. Here *mode* denotes a function that provides the dominant value within the inputs classes id_1, id_2, id_3 and returns NIL if there is a tie, while *classifier* is a function that returns the classification output in the form of a tuple (*Neutral*, *Racism*, *Sexism*).

5 Evaluation setup - Results

5.1 Data Preprocessing

Before training the neural network with the labeled tweets, it is necessary to apply the proper tokenization to every tweet. In this way, the text corpus is split into word elements, taking white spaces and the various punctuation

Table 1 Combined features in the proposed schemes

| Combination | Additional features | Features | Input Dimension |
|-------------|----------------------------|-----------------------------|-----------------|
| O | No additional features | - | 30 |
| NS | Neutral and Sexism | $t_{N,a}, t_{S,a}$ | 32 |
| NR | Neutral and Racism | $t_{N,a}, t_{R,a}$ | 32 |
| RS | Racism and Sexism | $t_{R,a}, t_{S,a}$ | 32 |
| NRS | Neutral, Racism and Sexism | $t_{N,a}, t_{R,a}, t_{S,a}$ | 33 |

symbols used in the language into account. This was done using the Moses¹ package for machine translation.

We choose to limit the maximum size of each tweet to be considered during training to 30 words, and padded tweets of shorter size with zeros. Next, tweets are converted into vectors using word-based frequency, as described in Section 4.1. To feed the various classifiers in our evaluation, we attach the feature values onto every tweet vector.

In this work we experimented with various combinations of attached features $t_{N,a}$, $t_{R,a}$, and $t_{S,a}$ that express the user tendency. The details of each experiment, including the resulting size of each embedding can be found in Table 1, with the latter denoted ‘input dimension’ in the table.

5.2 Deep learning model

In our evaluation of the proposed scheme, each classifier is implemented as a deep learning model having four layers, as illustrated in Figure 2, and is described as follows:

- *The Input (a.k.a Embedding) Layer.* The input layer’s size is defined by the number of inputs for that classifier. This number equals the size to the word vector plus the number of additional features. The word vector dimension was set to 30 so that to be able to encode every word in the vocabulary used.
- *The hidden layer.* The sigmoid activation was selected for the the hidden LSTM layer. Based on preliminary experiments the dimensionality of the output space for this layer was set to 200. This layer is fully connected to both the Input and the subsequent layer.
- *The dense layer.* The output of the LSTM was run through an additional layer to improve the learning and obtain more stable output. The *ReLU* activation function was used. Its size was selected equal to the size of the input layer.
- *The output layer.* This layer has 3 neurons to provide output in the form of probabilities for each of the three classes *Neutral*, *Racism*, and *Sexism*. The softmax activation function was used for this layer.

¹ <http://www.statmt.org/moses/>

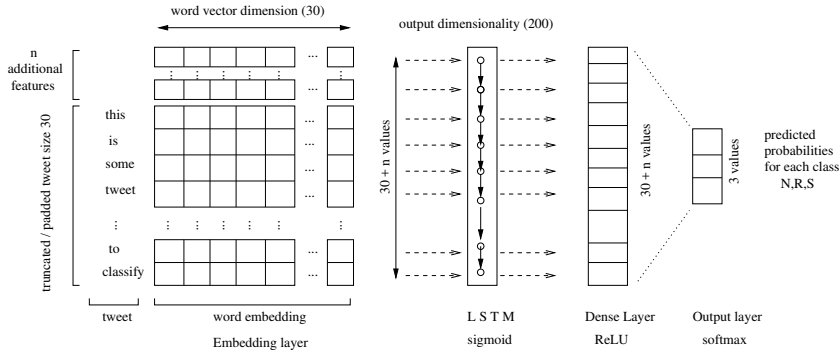


Fig. 2 Our deep learning model

In total we experimented with 11 different setups of the proposed scheme, each with a different ensemble of classifiers, see Table 2.

Table 2 Evaluated ensemble schemes

| Tested Scheme | Classifier | | | | |
|---------------|------------|-----|-----|----|-----|
| | I | II | III | IV | V |
| (i) | O | NRS | NR | - | - |
| (ii) | O | NRS | NS | - | - |
| (iii) | O | NRS | RS | - | - |
| (iv) | O | NS | RS | - | - |
| (v) | O | NS | NR | - | - |
| (vi) | O | RS | NR | - | - |
| (vii) | NRS | NR | RS | - | - |
| (viii) | NRS | NR | NS | - | - |
| (ix) | NRS | NS | RS | - | - |
| (x) | NS | RS | NR | - | - |
| (xi) | O | NS | RS | NR | NRS |

5.3 Dataset

We experimented with a dataset of approximately 16k short messages from Twitter, that was made available by Waseem & Hovy [19]. The dataset contains 1943 tweets labeled as *Racism*, 3166 tweets labeled as *Sexism* and 10889 tweets labeled as *Neutral* (i.e., tweets that neither contain sexism nor racism). There is also a number of dual labeled tweets in the dataset. More particularly, we found 42 tweets labeled both as both ‘Neutral’ and ‘Sexism’, while six tweets were labelled as both ‘Racism’ and ‘Neutral’. According to the dataset providers, the labeling was performed manually.²

² The small discrepancy observed in the class quantities with regard to those mentioned in the original dataset is due to fact that, at the time we performed the evaluation, a number of tweets were not retrievable.

1 The relatively small number of tweets in the dataset makes the task more
2 challenging. As reported by several authors already, the dataset is imbalanced,
3 with a majority of neutral tweets. Additionally, we used the public Twitter
4 API to retrieve additional data associated with the user identity for each tweet
5 in the original dataset.
6

7 8 5.4 Experimental Setting 9

10 To produce results in a setup comparable with the current state of the art
11 [1], we performed 10-fold cross validation and calculated the *Precision*, *Recall*
12 and *F-Score* for every evaluated scheme. We randomly split each training fold
13 into 15% validation and 85% training, while performance is evaluated over the
14 remaining fold of unseen data. The model was implemented using Keras³. We
15 used *categorical cross-entropy* as learning objective, and selected the ADAM
16 optimization algorithm by Kingma & Ba [11]. Furthermore, the vocabulary
17 size was set to 25000, and the batch-size during training was set to 500.
18

19 To avoid over-fitting, the model training was allowed to run for a maximum
20 number of 100 epochs, out of which the optimally trained state was chosen
21 for the model evaluation. An optimal epoch was identified so, such that the
22 validation accuracy was maximized, while at the same time the error remained
23 within $\pm 1\%$ of the lowest ever figure within the current fold. Throughout
24 the experiment we observed that the optimal epochs typically occurred after
25 between the 30 and 40 epochs.
26

27 To achieve stability in the results produced, we ran every single classifier
28 for 15 times and the output values were aggregated. In addition, the output
29 from each single classifier run was combined with the output from another
30 two single classifiers to build the input of an ensemble, producing 15^3 combi-
31 nations. For the case of the ensemble that incorporates all five classifiers we
32 restricted to using the input by only the first five runs of the single classifiers
33 (5^5 combinations). That was due to the prohibitively very large number of
34 combinations that were required.
35

36 37 5.5 Results 38

39 We now present the most interesting results from our experiments.

40 For the evaluation we used standard metrics for classification accuracy,
41 suitable for studying problems such as sentiment analysis. In particular we
42 used *Precision* and *Recall*, with the former calculated as the ratio of the num-
43 ber of tweets correctly classified to a given class over the total number of
44 tweets classified to that class, while the latter measures the ratio of messages
45 correctly classified to a given class over the number of messages from that
46 class. Additionally, the *F-score* is the harmonic mean of precision and recall,
47 expressed as $F = \frac{2 \cdot P \cdot R}{P + R}$. For our particular case with three classes, P , R and
48

49 ³ <https://github.com/fchollet/keras>
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

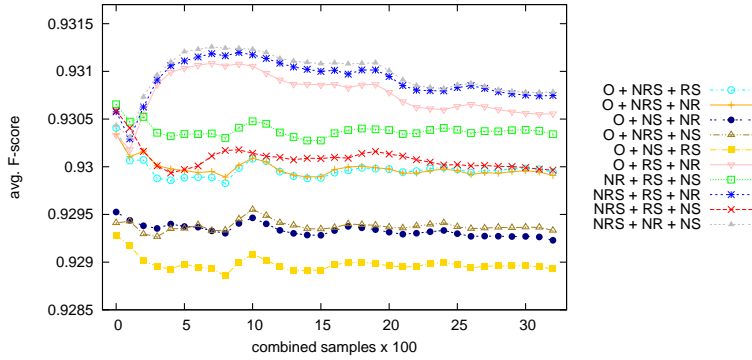


Fig. 3 Aggregated value for F-score vs the number of combined experiment runs

F are computed for each class separately, with the final F value derived as the weighted mean of the separate F -scores: $F = \frac{F_N \cdot N + F_R \cdot R + F_S \cdot S}{N + R + S}$; recall that $N = 10889$, $S = 3166$ and $R = 1943$. The results are shown in Table 3, along with the reported results from state of the art approaches proposed by other researchers in the field. Note that the performance numbers P, R and F of the other state of the art approaches are based on the authors' reported data in the cited works. Additionally, we report the performance of each individual LSTM classifier as if used alone over the same data (that is, without the ensemble logic). The F -score for our proposed approaches shown in the last column, is the weighted average value over the 3 classes (Neutral, Sexism, Racism). Moreover, all the reported values are average values produced for a number of runs of the same tested scheme over the same data. Figure 3 shows the F -Score as a function of the number of training samples for each ensemble of classifiers. We clearly see that the models converge. For the final run the F -score has standard deviation value not larger than 0.001, for all classifiers.

As can be seen in Table 3, the work by Waseem & Hovy [19], in which character n -grams and gender information were used as features, obtained the quite low F -score of 0.7391. Later work by the same author [18] investigated the impact of the experience of the annotator in the performance, but still obtaining a lower F -score than ours. Furthermore, while the second part of the two step classification by Park & Fung [14] performs quite well (reported an F -score of 0.9520) in detecting the particular class the abusive text belongs to, it nevertheless falls short in distinguishing hatred from non-hatred content in general. Finally, we observe that applying a simple LSTM classification in our approach, with no use of additional features (denoted 'single classifier (i)' in Table 3), achieves an F -score that is below 0.93, something that is in line with other researchers in the field, see [1]. Very interestingly, the incorporation of features related to user's behaviour into the classification has provided a significant increase in the performance vs. using the textual content alone, ($F = 0.9295$ vs. $F = 0.9089$).

Table 3 Evaluation Results

| Approach | Characteristics | Precision | Recall | F-Score |
|-------------------------|--|-----------|--------|---------------|
| single classifier (i) | O | 0.9175 | 0.9218 | 0.9196 |
| single classifier (ii) | NS | 0.9246 | 0.9273 | 0.9260 |
| single classifier (iii) | NR | 0.9232 | 0.9259 | 0.9245 |
| single classifier (iv) | RS | 0.9232 | 0.9264 | 0.9248 |
| single classifier (v) | NRS | 0.9252 | 0.9278 | 0.9265 |
| ensemble (i) | O + NRS + NR | 0.9283 | 0.9315 | 0.9298 |
| ensemble (ii) | O + NRS + NS | 0.9288 | 0.9319 | 0.9303 |
| ensemble (iii) | O + NRS + RS | 0.9283 | 0.9315 | 0.9299 |
| ensemble (iv) | O + NS + RS | 0.9277 | 0.9310 | 0.9293 |
| ensemble (v) | O + NS + NR | 0.9276 | 0.9308 | 0.9292 |
| ensemble (vi) | O + RS + NR | 0.9273 | 0.9306 | 0.9290 |
| ensemble (vii) | NRS + NR + RS | 0.9292 | 0.9319 | 0.9306 |
| ensemble (viii) | NRS + NR + NS | 0.9295 | 0.9321 | 0.9308 |
| ensemble (ix) | NRS + NS + RS | 0.9294 | 0.9321 | 0.9308 |
| ensemble (x) | NS + RS + NR | 0.9286 | 0.9314 | 0.9300 |
| ensemble (xi) | O + NS + RS + NR + NRS | 0.9305 | 0.9334 | 0.9320 |
| Badjatiya et al. [1] | LSTM + Random Embedding + GBDT | 0.9300 | 0.9300 | 0.9300 |
| Waseem & Hovy [19] | Unsupervised List of Criteria | 0.7290 | 0.7774 | 0.7391 |
| Waseem [18] | Unsupervised Expert annotators only | 0.9159 | 0.9292 | 0.9153 |
| Park & Fung [14] | 2 step HybridCNN (Word Vec. / Char Vec.) | 0.8270 | 0.8270 | 0.8270 |

Another interesting finding is the observed performance improvement by using an ensemble instead of a single classifier; some ensembles outperform the best single classifier. Furthermore, the NRS classifier, which produces the best score in relation to other single classifiers, is the one included in the best performing ensemble.

In comparison to the approach by Jha & Mamidi [9], which focuses on various classes of Sexism, the results show that our deep learning model is doing better as far as detecting Sexism in general, outperforming the FastText algorithm they have included in their experimental models (F=0.87). The inferiority of FastText over LSTM is also reported in the work by Badjatiya et al. [1], as well as being inferior over CNN by Park & Fung [14]. In general, through our ensemble schemes is confirmed that deep learning can outperform any NLP-based approaches known so far in the task of abusive language detection.

We also present the performance of each of the tested models per class label in Table 4. Results by other researchers have not been included, as these figures are not reported in the existing literature. As can be seen, *sexism* is quite easy to classify in hate-speech, while *racism* seems to be harder; similar results were reported by Davidson et al. [6]. This result is consistent across all ensembles.

For completion, the confusion matrices of the best performing approach that employs 3 classifiers (ensemble *viii*) as well as of the ensemble of all 5

Table 4 Detailed Results for every Class Label

| Proposed Approach | Class | Precision | Recall | F-Score |
|-------------------|----------------|-----------|--------|---------------|
| ensemble (viii) | <i>Neutral</i> | 0.9409 | 0.9609 | 0.9508 |
| | <i>Racism</i> | 0.7522 | 0.6646 | 0.7057 |
| | <i>Sexism</i> | 0.9991 | 0.9972 | 0.9981 |
| ensemble (ix) | <i>Neutral</i> | 0.9407 | 0.9612 | 0.9508 |
| | <i>Racism</i> | 0.7533 | 0.6627 | 0.7051 |
| | <i>Sexism</i> | 0.9986 | 0.9972 | 0.9979 |
| ensemble (vii) | <i>Neutral</i> | 0.9405 | 0.9611 | 0.9507 |
| | <i>Racism</i> | 0.7522 | 0.6616 | 0.7040 |
| | <i>Sexism</i> | 0.9990 | 0.9975 | 0.9983 |
| ensemble (xi) | <i>Neutral</i> | 0.9406 | 0.9631 | 0.9517 |
| | <i>Racism</i> | 0.7623 | 0.6617 | 0.7084 |
| | <i>Sexism</i> | 0.9992 | 0.9980 | 0.9986 |

Table 5 Confusion Matrices of Results for the best performing ensembles with 3 and 5 classifiers.

| ensemble (viii) | | <i>Predicted Label</i> | | | |
|-------------------|----------------|------------------------|---------------|----------------|-------------------------------------|
| <i>True Label</i> | | Racism | Sexism | Neutral | sum |
| | Racism | 10655320 | 5635 | 24295 | 10685250 |
| | Sexism | 3943 | 4357971 | 2195711 | 6557625 |
| | Neutral | 5929 | 1430030 | 35314416 | 36750375 |
| | sum | 10665192 | 5793636 | 37534422 | 53993250 (15998×15^3) |
| ensemble (xi) | | <i>Predicted Label</i> | | | |
| <i>True Label</i> | | Racism | Sexism | Neutral | sum |
| | Racism | 9873754 | 991 | 19005 | 9893750 |
| | Sexism | 3034 | 4017687 | 2051154 | 6071875 |
| | Neutral | 4446 | 1252093 | 32771586 | 34028125 |
| | sum | 9881234 | 5270771 | 34841745 | 49993750 (15998×5^5) |

classifiers (xi), are provided in Table 5. The presented values is the sum over multiple runs.

To study the effect of the user’s tendency in haste-speech in the F-score, we provide a brake-down of the computed values over five classes of users. Therefore, we divided the complete set of users into five subsets of equal size, wrt. their tendency on sexism or racism, and computed the F-score independently for each user class. We present the results for each individual classifier as well as for all the ensembles of classifiers we tested.

In Figure 4 we present the F-score achieved for each classifier over the five classes of users. In class 1 belong those users having the lowest tendency, while class 5 contains the users with the highest tendency. As can be seen in the above figure, tweets by users who are more tempted to hate speech are easier to detect by our algorithm, than the less tempted ones. Very interestingly, this characteristic works better for sexism rather for racism. Quite impressive is the fact that the F-Score for the most tempted users can reach as high as 0.995, no matter which classifier used. In addition, classifier (O), which does

not make use of user features, performs slightly worse, for the full range of classes of tendency.

From the output shown in Figure 4 we observe that the classification works quite effectively, detecting almost all cases of abusive content originated from the most tended users, something that is inline with our primary objective. In overall, the above observations confirm the original hypothesis of the classification accuracy being improved with the employment of additional user-based features into the prediction mechanism. For completion we also report the

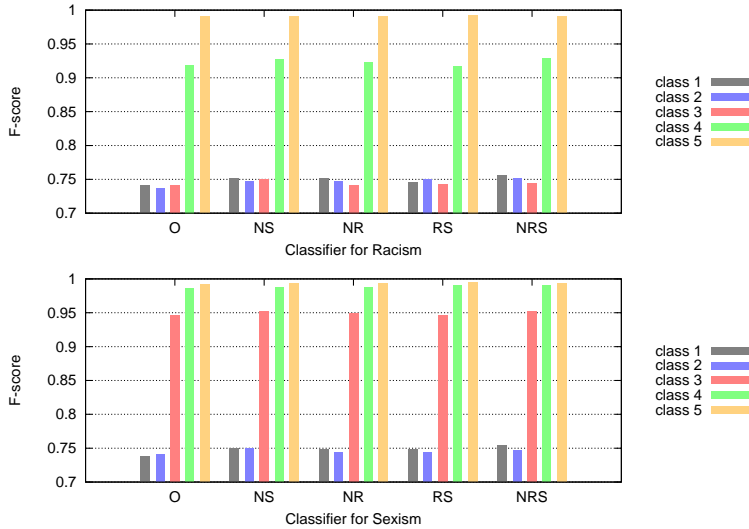


Fig. 4 F-score for various classes of users for single classifiers

F-score for all the ensembles of classifiers for every particular users class in Figures 5 and 6. As can be seen, for the case of ensembles, our approach has similar and equally good performance with that achieved by the use of individual classifiers. We also observe that, for the classes of users who are less tending towards sexism or racism, the 5-classifiers ensemble achieves the best performance in comparison to the other schemes.

Another interesting result is presented in Figure 7. It shows the Receiver Operative Characteristic (ROC) curves of all single classifiers we introduced. ROC values gives the ability to assess a classifier's performance over its entire operating range of the chosen thresholds used for separating one class from another. Also, it provides visualization of the trade-offs between sensitivity and specificity, so that finally an optimal model can be selected. To compute the ROC curves for 3-class label output, we applied the following rationale: For each classifier scheme, we firstly take each prediction that is essentially the output of the softmax activation function, and then we apply, in separate for

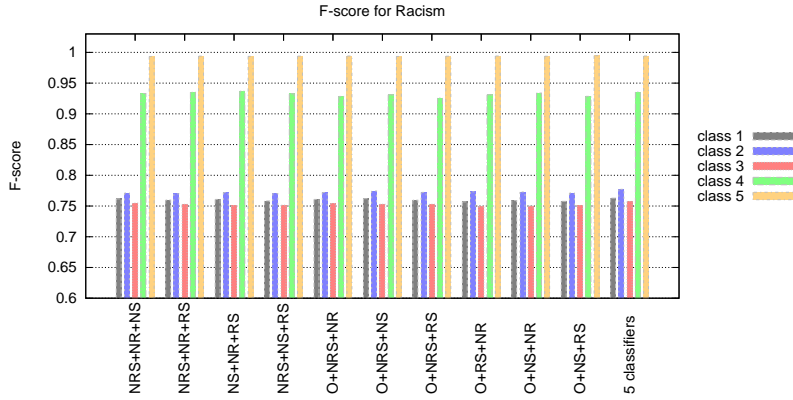


Fig. 5 F-score for various classes of users with tendency in racism for ensemble classifiers

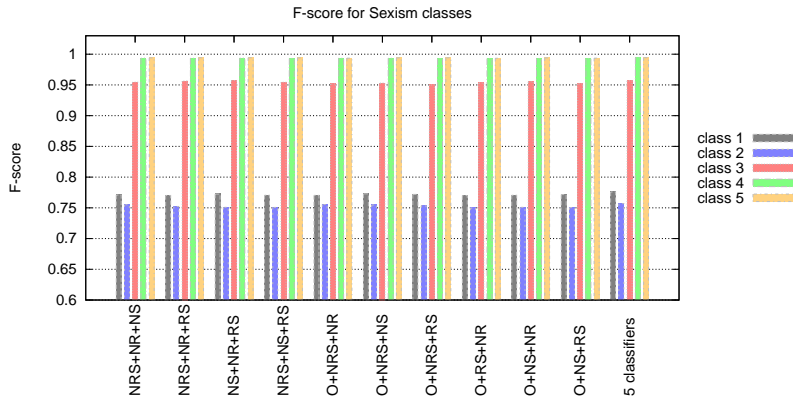


Fig. 6 F-score for various classes of users with tendency in sexism for ensemble classifiers

each class label value (Neutral, Sexism, Racism), a threshold to classify a tweet as belonging to that class. Next, we compute the True Positive Ratio and False Positive Ratio as a function of $tpr = \frac{tp}{tp+fn}$ and $fpr = \frac{fp}{fp+tn}$ respectively; and finally, the resulting values are averaged over the 3 classes of Neutral, Sexism and Racism. The above steps are repeated for a range of threshold values between (0.0 and 1.0) to produce the output finally demonstrated in the ROC curve for that classifier.

To express the resulting performance of a classifier in the form of numerical score we compute the Area Under Curve (AUC) value for each one (see Table 6). The figures show that NS is the best performing classifier achieving AUC value of 0.8406. While all the other single classifiers performed slightly worse, they still achieved high score that falls within the range between 0.8 and 0.9, which is characteristic of a good performing model. Also computed the AUC values for each of the 5 classes of users with regard their tendency in sexism or racism (see Table 6). The above results also confirm the opti-

mal performance achieved by the model in the task of separating the hateful content from the non-hateful one, when the posting is originated from users belonging to a class of high tendency towards sexism or racism.

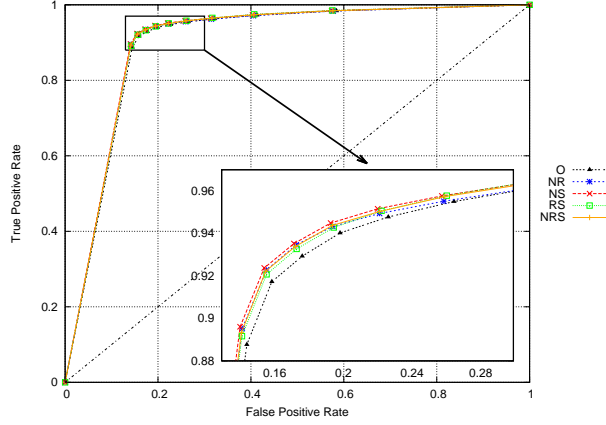


Fig. 7 ROC comparison for all single classifiers

Table 6 Area Under the Curve (AUC) of ROC for single classifiers

| Classifier | O | NR | NS | RS | NRS |
|-------------------|--------|---------------|---------------|---------------|--------|
| user class | | | | | |
| overall | 0.8354 | 0.8382 | 0.8406 | 0.8395 | 0.8395 |
| rac-1 | 0.6888 | 0.6934 | 0.6974 | 0.6956 | 0.6973 |
| rac-2 | 0.6864 | 0.6909 | 0.6951 | 0.6917 | 0.6899 |
| rac-3 | 0.6858 | 0.6874 | 0.6953 | 0.6889 | 0.6898 |
| rac-4 | 0.8364 | 0.8378 | 0.8401 | 0.8383 | 0.8404 |
| rac-5 | 0.8855 | 0.8859 | 0.8894 | 0.8897 | 0.8894 |
| sex-1 | 0.6868 | 0.6918 | 0.6958 | 0.6957 | 0.6927 |
| sex-2 | 0.6869 | 0.6885 | 0.6953 | 0.6879 | 0.6912 |
| sex-3 | 0.8556 | 0.8581 | 0.8597 | 0.8585 | 0.8595 |
| sex-4 | 0.8818 | 0.8817 | 0.8855 | 0.8855 | 0.8867 |
| sex-5 | 0.8808 | 0.8853 | 0.8881 | 0.8873 | 0.8846 |

Finally, we need to point out that our approach does not rely on pre-trained vectors, which provides a important advantage when dealing with short messages of this kind. More specifically, users will often prefer to obfuscate their offensive terms using shorter slang words or create new words by ‘inventive’ spelling and word concatenation. For instance, the word ‘Islamolunatic’ is not available in the popular pre-trained word embeddings (Word2Vec or GloVe), even though it appears with a rather high frequency in racist postings. Hence,

word frequency vectorization is preferable to the pre-trained word embeddings used in prior works, in order to build a language-agnostic solution.

6 Conclusions and Future Work

Automated detection of abusive language in on-line media has in the recent years become a key challenge. In this paper we have presented an ensemble classifier to detect hate-speech in short text, such as tweets. Our classifier uses deep learning and incorporates a series of features associated with users' behavioral characteristics, such as the tendency to post abusive messages, as input to the classifier. In summary, this paper has made several main contributions in order to advance the state-of-the-art. First, we have developed a deep learning architecture that uses word frequency vectorisation for implementing the above features. Second, we have proposed a method that, due to no-use of pre-trained word embeddings, is language independent. Third, we have done thorough evaluation of our model using a public dataset of labeled tweets, an open-sourced implementation built on top of Keras. This evaluation also includes an analysis of the performance of the proposed scheme for various classes of users. The experimental results have shown that our approach outperforms the current state-of-the-art approaches, and to the best of our knowledge, no other model has achieved better performance in classifying short messages. Also, the results have confirmed the original hypothesis of improving the classifier's performance by employing additional user-based features into the prediction mechanism.

For future work, we plan to investigate other sources of information that can be utilized to detect hateful messages. In addition, we intend to generalize the output gained in the current experiment, with evaluation over other datasets, including analyzing texts written in different languages.

Acknowledgements This work has been supported by Telenor Research, Norway, through the collaboration project between NTNU and Telenor. It has been carried out at the Telenor – NTNU AI-Lab.

References

1. Badjatiya, P., Gupta, S., Gupta, M., & Varma, V. (2017). Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion* (pp. 759–760). International World Wide Web Conferences Steering Committee.
2. Barnaghi, P., Ghaffari, P., & Breslin, J. G. (2016). Opinion mining and sentiment polarity on twitter and correlation between events and sentiment. In *2nd IEEE International Conference on Big Data Computing Service and Applications (BigDataService)* (pp. 52–57).

- 1 3. BBC (2016). Facebook, Google and Twitter agree german hate speech
2 deal. Website. <http://www.bbc.com/news/world-europe-35105003> Ac-
3 cessed: on 26/11/2016.
- 4 4. Chen, Y., Zhou, Y., Zhu, S., & Xu, H. (2012). Detecting offensive language
5 in social media to protect adolescent online safety. In *2012 International*
6 *Conference on Privacy, Security, Risk and Trust (PASSAT 2012)*, and
7 *2012 International Conference on Social Computing (SocialCom 2012)*
8 (pp. 71–80).
- 9 5. DailyMail (2016). Zuckerberg in Germany: No place for hate speech
10 on Facebook. Website. [http://www.dailymail.co.uk/wires/ap/](http://www.dailymail.co.uk/wires/ap/article-3465562/Zuckerberg-no-place-hate-speech-Facebook.html)
11 [article-3465562/Zuckerberg-no-place-hate-speech-Facebook.](http://www.dailymail.co.uk/wires/ap/article-3465562/Zuckerberg-no-place-hate-speech-Facebook.html)
12 [html](http://www.dailymail.co.uk/wires/ap/article-3465562/Zuckerberg-no-place-hate-speech-Facebook.html) Accessed: on 26/02/2016.
- 13 6. Davidson, T., Warmlesley, D., Macy, M. W., & Weber, I. (2017). Auto-
14 mated hate speech detection and the problem of offensive language. In
15 *Proceedings of the 11th International Conference on Web and Social Me-*
16 *dia (ICWSM 2017)* (pp. 512–515).
- 17 7. Djuric, N., Zhou, J., Morris, R., Grbovic, M., Radosavljevic, V., &
18 Bhamidipati, N. (2015). Hate speech detection with comment embeddings.
19 In *Proceedings of the 24th International Conference on World Wide Web*
20 *Companion* (pp. 29–30). ACM.
- 21 8. Gambäck, B., & Sikdar, U. K. (2017). Using convolutional neural networks
22 to classify hate-speech. In *Proceedings of the 1st Workshop on Abusive*
23 *Language Online at ACL 2017*.
- 24 9. Jha, A., & Mamidi, R. (2017). When does a compliment become sexist?
25 analysis and classification of ambivalent sexism using twitter data. In
26 *Proceedings of the Second Workshop on NLP and Computational Social*
27 *Science* (pp. 7–16). Association for Computational Linguistics.
- 28 10. Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks
29 for efficient text classification. *arXiv preprint arXiv:1607.01759*, .
- 30 11. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic opti-
31 mization. In *Proceedings of the 3rd International Conference on Learning*
32 *Representations (ICLR 2014)*.
- 33 12. NewYorkTimes (2017). Twitter must do more to block isis.
34 Website. [https://www.nytimes.com/2017/01/13/opinion/](https://www.nytimes.com/2017/01/13/opinion/twitter-must-do-more-to-block-isis.html)
35 [twitter-must-do-more-to-block-isis.html](https://www.nytimes.com/2017/01/13/opinion/twitter-must-do-more-to-block-isis.html) Accessed: on 30/09/2017.
- 36 13. Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., & Chang, Y. (2016).
37 Abusive language detection in online user content. In *Proceedings of the*
38 *25th International Conference on World Wide Web (WWW 2016)* (pp.
39 145–153). International World Wide Web Conferences Steering Commit-
40 tee.
- 41 14. Park, J. H., & Fung, P. (2017). One-step and two-step classification for
42 abusive language detection on twitter. In *Proceedings of the 1st Workshop*
43 *on Abusive Language Online at ACL 2017*.
- 44 15. Schmidt, A., & Wiegand, M. (2017). A survey on hate speech detection
45 using natural language processing. In *Proceedings of the 5th International*
46 *Workshop on Natural Language Processing for Social Media* (pp. 1–10).
- 47
- 48
- 49
- 50
- 51
- 52
- 53
- 54
- 55
- 56
- 57
- 58
- 59
- 60
- 61
- 62
- 63
- 64
- 65

- 1 Association for Computational Linguistics.
- 2
- 3 16. Vigna, F. D., Cimino, A., Dell’Orletta, F., Petrocchi, M., & Tesconi, M.
- 4 (2017). Hate me, hate me not: Hate speech detection on facebook. In
- 5 *Proceedings of the 1st Italian Conference on Cybersecurity (ITASEC17)*
- 6 (pp. 86–95). URL: <http://ceur-ws.org/Vol-1816/paper-09.pdf>.
- 7 17. Warner, W., & Hirschberg, J. (2012). Detecting hate speech on the world
- 8 wide web. In *Proceedings of the 2nd Workshop on Language in Social*
- 9 *Media (LSM 2012)* LSM ’12 (pp. 19–26). Association for Computational
- 10 Linguistics.
- 11 18. Waseem, Z. (2016). Are you a racist or am i seeing things? annotator
- 12 influence on hate speech detection on twitter. In *Proceedings of the First*
- 13 *Workshop on NLP and Computational Social Science* (pp. 138–142). As-
- 14 sociation for Computational Linguistics.
- 15 19. Waseem, Z., & Hovy, D. (2016). Hateful symbols or hateful people? pre-
- 16 dictive features for hate speech detection on twitter. In *Proceedings of*
- 17 *the NAACL Student Research Workshop*. Association for Computational
- 18 Linguistics.
- 19
- 20
- 21
- 22
- 23
- 24
- 25
- 26
- 27
- 28
- 29
- 30
- 31
- 32
- 33
- 34
- 35
- 36
- 37
- 38
- 39
- 40
- 41
- 42
- 43
- 44
- 45
- 46
- 47
- 48
- 49
- 50
- 51
- 52
- 53
- 54
- 55
- 56
- 57
- 58
- 59
- 60
- 61
- 62
- 63
- 64
- 65