Applied Intelligence (2018) 48:4730–4742
https://doi.org/10.1007/s10489-018-1242-y

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

Georgios K. Pitsilis[1] · **Heri Ramampiaro**[1] · **Helge Langseth**[1]

Published online: 26 July 2018

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

✉ Georgios K. Pitsilis
georgios.pitsilis@ntnu.no

Heri Ramampiaro
heri@ntnu.no

Helge Langseth
helge.langseth@ntnu.no

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

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

Despite the fact that the majority of the solutions for automated detection of offensive text rely on Natural Language Processing (NLP) approaches, there have lately been a tendency towards employing pure machine learning

techniques like neural networks for that task. NLP approaches have the drawback of being complex, and to a large extent dependent on the language used in the text. This provides a strong motivation for employing alternative machine learning models for the classification task. Moreover, the majority of the existing automated approaches depend on using pre-trained vectors (e.g. Glove, Word2Vec) as word embeddings to achieve good performance from the classification model. This makes the detection of hatred content unfeasible in cases where users have deliberately obfuscated their offensive terms with short slang words.

There is a plethora of unsupervised learning models in the existing literature to deal with hate-speech [21], as well as in detecting the sentiment polarity in tweets [2]. At the same time, the supervised learning approaches have still not been explored adequately. While the task of sentence classification seems similar to that of sentiment analysis, in hate-speech even negative sentiment could still provide useful insight. Our intuition is that the task of hate-speech detection can further benefit from the incorporation of other sources of information to be used as features into a supervised learning model. A simple statistical analysis on an existing annotated dataset of tweets [24], can easily reveal the existence of significant correlation between the user tendency in expressing opinions that belong to some offensive class (*Racism* or *Sexism*), and the annotation labels associated with that class. More precisely, the correlation coefficient value that describes such user tendency was found to be 0.71 for racism in the above dataset, while that value reached as high as 0.76 for sexism. In our opinion, utilizing such user-oriented behavioral data for reinforcing an existing solution is feasible, because such information is retrievable in real-world use-case scenarios like Twitter. This highlights the need to explore the user features more systematically to further improve the classification accuracy of a supervised learning system.

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

Our main contributions are: i) a deep learning architecture for text classification in terms of hateful content, which incorporates features derived form the users' behavioral data, ii) a language agnostic solution, due to no-use of pre-trained word embeddings, for detecting hate-speech, iii) an experimental evaluation of the model on a Twitter dataset, demonstrating the top performance achieved on the classification task. We put special focus on investigating how the additional features concerning the users' tendency to utter hate-speech, as expressed by their previous history, could leverage the performance. To the best of our knowledge, there has not been any previous study on exploring features related to the users tendency in hatred content that has used a deep learning model.

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

2 Problem statement

The problem we address in this work can be described as follows: We are given a set of posting written by a number of online users. Each posted short-text is associated with a class-label, where we consider the classes "Neutral" (N), "Racist" (R) and "Sexist" (S). From a training-set of labeled short-texts, we set out to train a classifier that when receiving a new posting from a given user can extract and combine information in the training-data about short-text messages in general and the posting-history of the active user in particular to successfully classify the new posting as either "N", "S" or "R". The research question we address in this work is thus:

> How to effectively identify the class of a new posting, given the identity of the posting user and the history of postings related to that user?

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

- To develop a novel method that can improve the state-of-the-art approaches within hate-speech classification, in terms of classification performance/accuracy.
- To investigate the impact of incorporating information about existing personalized labeled postings from users' past history on the classification performance/accuracy.

Note that existing solutions for automatic detection still fall short of effectively detecting abusive messages. Therefore there is a need for new algorithms, which would do the job of classification of such content more effectively and efficiently. Our work is a step in that direction.

3 Related work

Simple word-based approaches, if used for blocking the posting of text or blacklisting users, not only fail to identify

subtle offensive content, but they also affect freedom of speech and expression. The word ambiguity problem – that is, a word can have different meanings in different contexts – is mainly responsible for the high false positive rate in such approaches. Ordinary NLP approaches on the other hand, despite their popularity [21], are ineffective to detect unusual spelling, experienced in user-generated comment text. This is best known as the *spelling variation* problem, and it is caused either by unintentional or intentional replacement of single characters in a token, aiming to obfuscate the detectors. In general, the complexity of the natural language constructs renders the task quite challenging. Irrespective of the use of NLP approaches, we can distinguish two major categories in the existing solutions to the hate-speech problem: The Unsupervised learning and the Supervised learning.

Unsupervised learning approaches are quite common for detecting offensive messages in text, and essentially are applied concepts from NLP to exploit the lexical syntactic features of sentences [4], or used AI-solutions and bag-of-words-based text-representations [23]. The latter is known to be less effective for automatic detection, since hatred users apply various obfuscation tricks, such as replacing a single character in offensive words. For instance, applying a binary classifier onto a *paragraph2vec* representation of words has already been attempted on Amazon data in the past [7], but it only performed well on a binary classification problem. Another unsupervised learning based solution is the work in [25], in which the authors proposed a set of criteria that a tweet should exhibit in order to be classified as offensive. They also showed that differences in geographic distribution of users have only marginal effect on the detection performance. Despite the above observation, we explore other features that might be possible to improve the detection accuracy in the solution outlined below. In [24] is applied a crowd-sourced solution to tackle hate-speech, with the creation of an additional dataset of annotations to extend the existing corpus. They also investigated the impact of the experience of annotators in the classification performance.

As far as the supervised learning classification methods, their employment in the detection of hate-speech is not new. In [6] is described another way of distinguishing hate-speech from offensive language in tweets, based on a classifier mode that involves Naive Bayes, Decision Trees and SVM. Also, in [16] is attempted to discern abusive content with an NLP-based supervised model combining various linguistic and syntactic features in the text, considered at character uni-gram and bi-gram level, and tested on Amazon data. Jha and Mamidi [11] dealt with the classification problem of tweets, but their interest was on sexism alone, which they distinguished into 'Hostile', 'Benevolent' or 'Other'. While the authors used the dataset of tweets from [25], they treated the existing *'Sexism'* tweets as being of class *'Hostile'*, while they collected their own tweets for the *'Benevolent'* class, on which they finally applied the FastText classifier [12], and SVM.

The supervised learning models also include the Deep Neural Networks (DNNs). Their power comes from their ability to find data representations that are useful for classification and they are widely explored to handle NLP tasks. Convolution Neural Networks (CNN) [14] and Recurrent Neural Networks (RNN) [8] are the two main architectures of DNNs, which NLP has benefited from. CNNs are suited for multi-dimension input data sampled periodically, in which a number of adjacent inputs are convoluted into the next layer in the network. RNN can be thought of as the addition of loops to the architecture through back propagation in the training process, to update the network weights in every layer. LSTMs are special RNNs which allow arbitrary propagation of signals into the network, thus being sensitive to the order of values. In [22] is reported performance for a simple LSTM classifier not better than an ordinary SVM, when evaluated on a small sample of Facebook data for only 2 classes (*Hate, No-Hate*), and 3 different levels of strength of hatred. [1] approached the issue with a neural network-based model that uses LSTM, with features extracted by character *n-grams*, and assisted by Gradient Boosted Decision Trees. Their method achieved higher score over the same dataset of tweets than any unsupervised learning solution known so far. CNNs has also been explored as a potential solution in the hate-speech problem in tweets, with character *n-grams* and *word2vec* pre-trained vectors being the main tools. For example, in [19] classification is transformed into a 2-step problem, where abusive text is first distinguished from the non-abusive, and then the class of abuse (*Sexism* or *Racism*) is determined. In [9] is employed pre-trained CNN vectors in an effort to predict the four classes, and finally achieving a slightly higher *F-score* than character *n-grams*. A summary of the existing approaches in the problem of hate speech, along with their characteristics, is presented in Table 1.

In general, we can point out the main weaknesses of NLP-based models in their non-language agnostic nature and the low scores in detection. In spite of their high popularity [21], when used either in supervised or unsupervised learning models, we believe there is still a high potential for DNNs to further contribute to the issue. At this point it is also relevant to note the inherent difficulty of the hate-speech challenge itself, which can be clearly noted by the fact that no solution thus far has been able to obtain an F-score above 0.93.

Table 1 Cartography of existing research in hate-speech detection

Categ.	Citation	Model	Classes to detect	Classif. steps	Dataset used
Unsupervised	Chen et al. [4]	NLP	Continuous values	1	YouTube
	Warner and Hirschberg [23]	NLP	2 classes (Hate,Non-hate)	1	Yahoo
	Djuric et al. [7]	NLP	2 classes (Hate,Non-hate)	2	Yahoo
	Waseem [24]	Empirical	4 classes (Sexism,Racism,Both,None)	1	Twitter
	Waseem and Hovy [25]	Empirical	3 classes (Sexism,Racism,None)	1	Twitter
Supervised	Nobata et al. [16]	NLP	4 classes (Clean, Hate, Derogatory, Profanity)	1	Amazon
	Davidson et al. [6]	SVM Bayes Decision Trees Random Forest	3 classes (Hate,Offensive,Neither)	1	Twitter
	Jha and Mamidi [11]	SVM FastText	3 classes (Benevolant,Hostile,Other)	1	Twitter
	Vigna et al. [22]	LSTM	2 classes / 3 levels (Hate,No-hate) (Strong,Weak,No-hate)	1	Facebook
	Badjatiya et al. [1]	CNN LSTM FastText	3 classes (Sexism,Racism,Neither)	1	Twitter
	Park and Fung [19]	CNN	3 classes (Sexism,Racism,Neither)	1 / 2	Twitter
	Gambäck and Sikdar [9]	CNN	4 classes (Sexism,Racism,Both,Neither)	1	Twitter

4 Description of our recurrent neural network-based approach

In our experimentation we use a powerful type of RNN known as *Long Short-Term Memory Network* (LSTM). Inspired by the work in [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 behavior.
- An architecture, which combines the output by various LSTM classifiers to improve the classification ability.

4.1 Features

We first elaborate on 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 $t_{S,a} = |m_{S,a}|/|m_a|$.

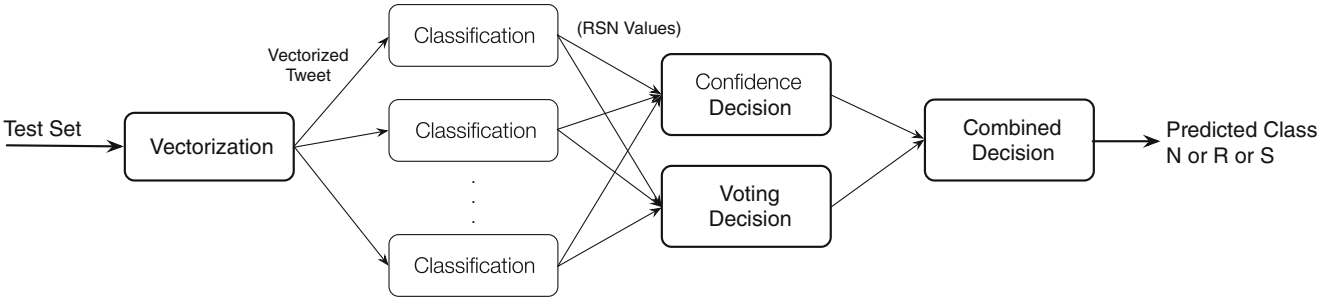


Fig. 1 High level view of the system with multiple classifiers

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. The employment of ensembles is a known technique used for improving the classification performance of a single model [17]. In this work, we apply the ensembles paradigm in our proposed solution to the hate-speech problem. 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 tweet, thereby providing an input vector for one LSTM classifier. A high level view of the architecture is shown in Fig. 1, with the multiple classifiers. The ensemble has two mechanisms for aggregating the classifications from the base classifiers; namely *Voting* and *Confidence*. Majority Voting is a known method to maximize the performance gain with the lowest number of classifiers [10, 18, 20]. In our work, we used a simpler rule for our specific needs. That is, the preferred method is majority voting, which is employed whenever at least two of the base classifiers agrees with respect to classification of a given tweet. When all classifiers disagree, the classifier with the strongest confidence in its prediction is given preference. The conflict resolution logic is implemented in the *Combined Decision* component.

Algorithm 1 Ensemble classifier

for $tw \in$ { tweets} **do**
  for $cl \in$ { classifiers } **do**
    $(N_{cl}, R_{cl}, S_{cl}) \leftarrow \text{classifier}_{cl}(tw)$
    $v_{cl} \leftarrow \max(N_{cl}, R_{cl}, S_{cl})$
    $id_{cl} \leftarrow \arg\max(N_{cl}, R_{cl}, S_{cl})$
  end for
  $m \leftarrow \text{mode}(id_1, id_2, id_3)$
  if $m \in$ {Neutral, Racist, Sexism} **then**
    decision $\leftarrow m$
  else
    decision $\leftarrow id_{\arg\max(v_1, v_2, v_3)}$
  end if
  print decision for tw
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 (*N*eutral, *R*acism, *S*exism).

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 symbols used in the language into account. This was done using the Moses[1] package for machine translation.

We chose 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

[1] http://www.statmt.org/moses/

Table 2 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

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's tendency. The details of each experiment, including the resulting size of each embedding can be found in Table 2, 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 Fig. 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 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.

In total we experimented with 11 different setups of the proposed scheme, each with a different ensemble of classifiers, as shown in Table 3.

5.3 Dataset

We experimented with an existing dataset of approximately 16k short messages from Twitter [25]. 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 as both 'Neutral' and 'Sexism', while six tweets were labeled as both 'Racism' and 'Neutral'. According to the dataset providers, the labeling was performed manually.[2]

The relatively small number of tweets in the dataset makes the task more challenging. As reported by several authors already, the dataset is imbalanced, with a majority of neutral tweets. Nevertheless, the size of the weakest class (Racism) being almost 5 times smaller than the size of the stronger class (Neutral), does not impose a strong level of imbalance in the dataset. Therefore, we chose to not apply any adjustments onto the data. Additionally, we used the public Twitter API to retrieve additional data associated with the user identity for each tweet in the original dataset.

5.4 Experimental setting

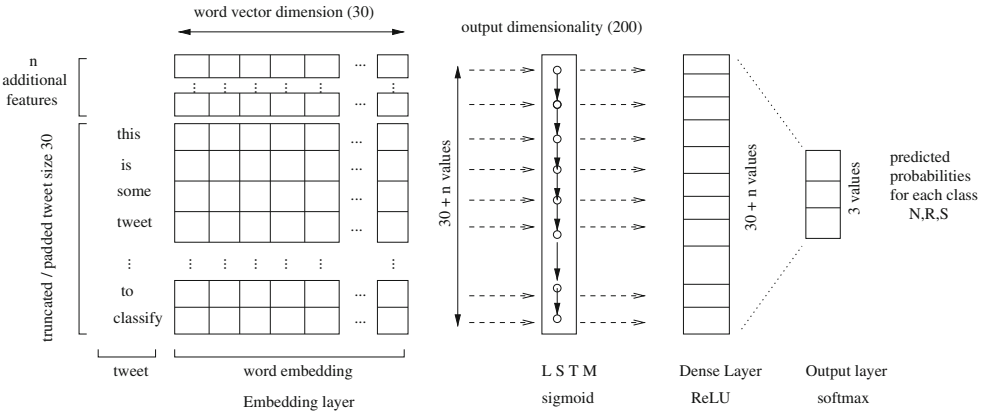
To produce results in a setup comparable with the current state of the art [1], we performed 10-fold cross validation and calculated the *Precision*, *Recall* and *F-Score* for every evaluated scheme. We randomly split each training fold into 15% validation and 85% training, while performance is evaluated over the remaining fold of unseen data. The model was implemented using Keras[3]. We used *categorical cross-entropy* as the learning objective, and selected the ADAM optimization algorithm [13]. Furthermore, the vocabulary size was set to 25000, and the batch-size during training was set to 500.

To avoid over-fitting, the model training was allowed to run for a maximum number of 100 epochs, out of which the optimally trained state was chosen for the model evaluation. An optimal epoch was identified, such that the validation accuracy was maximized, while at the same time the error remained within $\pm 1\%$ of the lowest ever figure within the

[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.

[3] https://github.com/fchollet/keras

Fig. 2 Our deep learning model



current fold. Throughout the experiment we observed that the optimal epochs typically occurred between the 30 and 40 epochs.

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

5.5 Results

We now present the most interesting results from our experiments.

Table 3 Evaluated ensemble schemes

	Classifier				
Tested scheme	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

For the evaluation, we used standard metrics for classification accuracy, suitable for studying problems such as sentiment analysis. In particular, we used *Precision* and *Recall*, with the former being calculated as the ratio of the number of tweets correctly classified to a given class over the total number of tweets classified to that class, while the latter measuring the ratio of messages correctly classified to a given class over the number of messages from that class. Additionally, the *F-score* is the harmonic mean of precision and recall, expressed as $F = \frac{2 \cdot P \cdot R}{P+R}$. For our particular case with three classes, P, R and 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 4, 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 4, the work in [25], 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 in [24] investigated the impact of the experience of the annotator in the performance, but still obtaining a lower *F-score* than ours. Furthermore,

Table 4 Evaluation results. (The values highlighted in bold indicate the best performance)

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 and Hovy [25]	Unsupervised List of Criteria	0.7290	0.7774	0.7391
Waseem [24]	Unsupervised Expert annotators only	0.9159	0.9292	0.9153
Park and Fung [19]	2 step HybridCNN (Word Vec. / Char Vec.)	0.8270	0.8270	0.8270

while the second part of the two step classification in [19] 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 4), achieves an *F-score* that is below 0.93, which is in line with other researches in the field [1]. Very interestingly, the incorporation of features related to user's behavior into the classification has provided a significant increase in the performance vs. using the textual content alone, ($F = 0.9295$ vs. $F = 0.9089$).

Another interesting finding is the observed performance improvement by using an ensemble instead of a single classifier; some ensembles outperform the best single

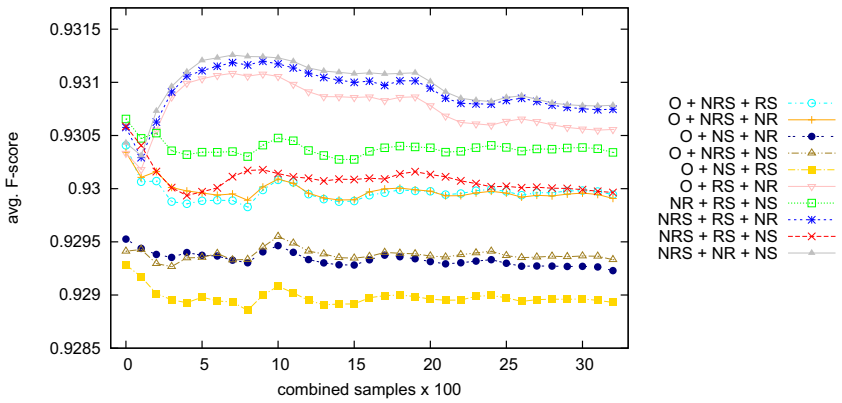


Fig. 3 Aggregated value for F-score vs the number of combined experiment runs

Table 5 Detailed results for every class label. (The values highlighted in bold indicate the best performance)

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

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 in [11], 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 in [1], as well as being inferior over CNN [19]. In general, through our ensemble schemes it 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 5. 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 in the literature [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 classifiers (xi), are provided in Table 6. The presented values is the sum over multiple runs.

To study the effect of the user's tendency in hate-speech in the F-score, we provide a break-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 Fig. 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 was 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 Fig. 4, we observe that the classification works quite effectively, detecting almost all cases of abusive content originated from the most

Table 6 Confusion Matrices of Results for the best performing ensembles with 3 and 5 classifiers

		Predicted label			
		Racism	Sexism	Neutral	Sum
Ensemble *(viii)*					
True label	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)*					
True label	Racism	9873754	991	19005	9893750
	Sexism	3034	4017687	2051154	6071875
	Neutral	4446	1252093	32771586	34028125
	Sum	9881234	5270771	34841745	49993750
					(15998×5^5)

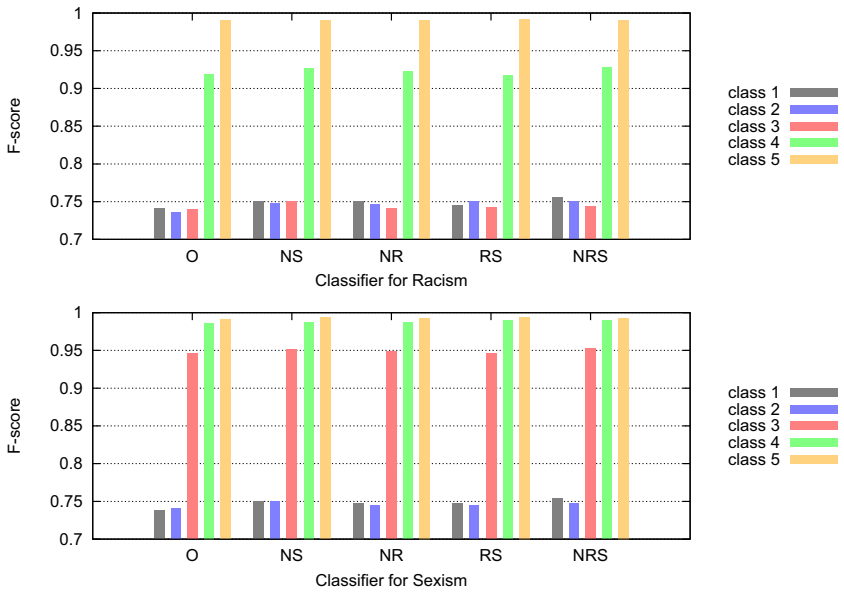


Fig. 4 F-score for various classes of users for single classifiers

tended users, something that is in line with our primary objective. 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 F-score for all the ensembles of classifiers for every particular users class in Figs. 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 Fig. 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 each class label value (Neutral, Sexism, Racism), a threshold to classify a tweet as belonging to that class. Next, we

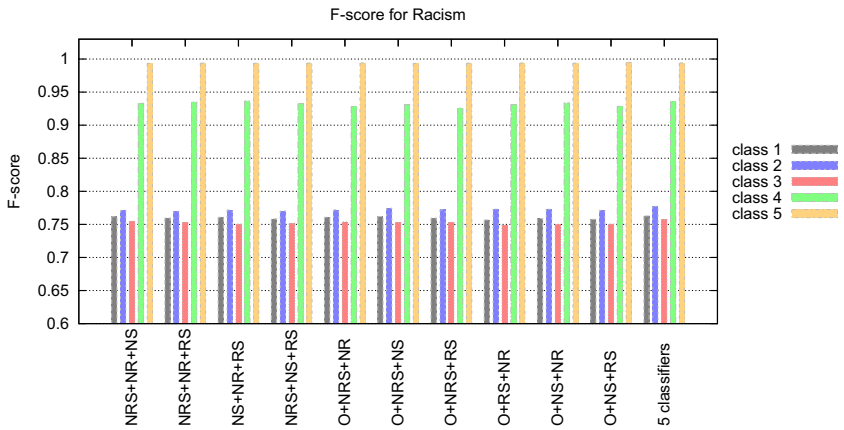


Fig. 5 F-score for various classes of users with tendency in racism for ensemble classifiers

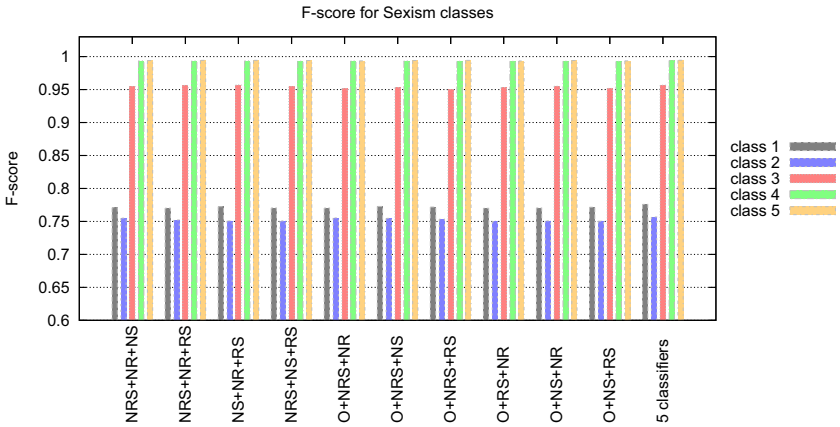


Fig. 6 F-score for various classes of users with tendency in sexism for ensemble classifiers

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 7). 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 a 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 regards to their tendency in sexism or racism (see Table 7).

The above results also confirm the optimal 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.

Finally, we need to point out that our approach does not rely on pre-trained vectors, which provides an 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.

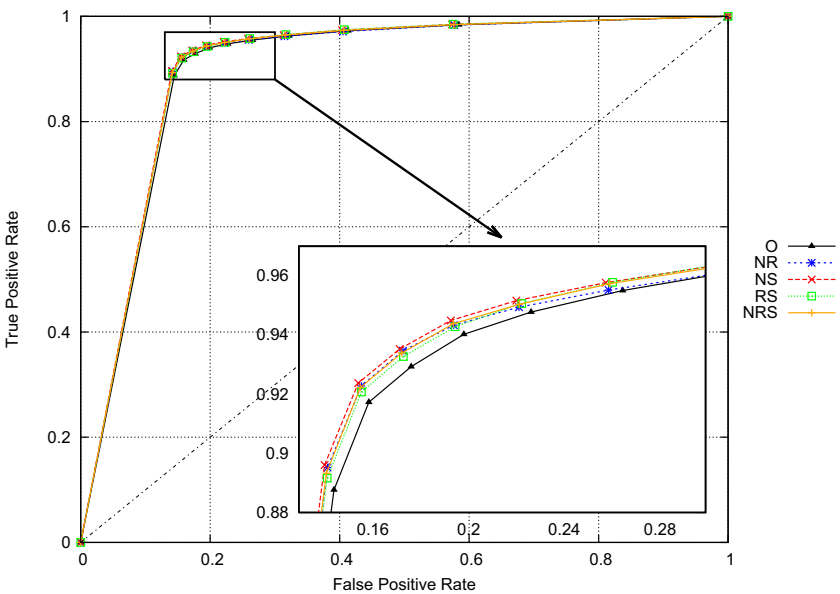


Fig. 7 ROC comparison for all single classifiers

Table 7 Area under the curve (AUC) of ROC for single classifiers. (The values highlighted in bold indicate the best performance)

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

6 Conclusions and future work

Automated detection of abusive language in on-line media has in 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.

In this section, we also discuss possible threats to vulnerability and limitations of our approach and give our perspectives on solving these issues. The stochastic behavior of the deep learning processes is the most important threat to construct validity, resulted in fluctuation in the F-score over the multiple runs (see Section 5.4). To overcome this, and thus ensure that our findings are valid, we ran every experiment multiple times and averaged the results. Concerning the generalizability of results, i.e. external validity, the experiment was performed on a single dataset of constant mixtures of labels and user profiles with a tendency towards a specific type of hatred language. However, our analysis of the dataset has shown that although we do not claim it is representative to all real-world data, its size and heterogeneity have been enough to test our method. Nevertheless, in our future study, we will further evaluate our approach over other datasets, including analyzing texts written in different languages. Further, we assumed that users behavior wouldn't change over time, which can also be considered as a threat to internal validity. That is, in a real use-case, users would normally be given access to the classifier output, so that upon the submission of a tweet they would become aware of the history in the labels given to the previous ones. This normally means that they would adapt their behavior to the classification criteria, and very likely avoid the inclusion of hateful content in their future postings. This is a general challenge in many applications of classification techniques, which could be solved through longer-lasting user studies or inclusion of other sources of information. For this reason, in our future work, we plan to investigate other sources of information that can be utilized to detect hateful messages.

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. International World Wide Web Conferences Steering Committee, pp 759–760
2. Barnaghi P, Ghaffari P, Breslin JG (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
3. Facebook, Google and Twitter agree german hate speech deal. Website. http://www.bbc.com/news/world-europe-35105003. Accessed 26 Nov 2016
4. Chen Y, Zhou Y, Zhu S, Xu H (2012) Detecting offensive language in social media to protect adolescent online safety. In: 2012 international conference on privacy, security, risk and trust (PASSAT 2012), and 2012 international confernece on social computing (SocialCom 2012), pp 71–80
5. Zuckerberg in Germany: No place for hate speech onFacebook. Website. http://www.dailymail.co.uk/wires/ap/article-3465562/Zuckerberg-no-place-hate-speech-Facebook.html. Accessed 26 Feb 2016
6. Davidson T, Warmsley D, Macy MW, Weber I (2017) Automated hate speech detection and the problem of offensive language. In: Proceedings of the 11th international conference on web and social media (ICWSM 2017), pp 512–515
7. Djuric N, Zhou J, Morris R, Grbovic M, Radosavljevic V, Bhamidipati N (2015) Hate speech detection with comment

embeddings. In: Proceedings of the 24th international conference on world wide web companion. ACM, pp 29–30
8. Elman J (1990) Finding structure in time. Cogn Sci 14:179–211
9. Gambäck B., Sikdar UK (2017) Using convolutional neural networks to classify hate-speech. In: Proceedings of the 1st workshop on abusive language online at ACL 2017
10. Gandhi I, Pandey M (2015) Hybrid ensemble of classifiers using voting. In: 2015 international conference on green computing and Internet of Things (ICGCIoT), pp 399–404. https://doi.org/10.1109/ICGCIoT.2015.7380496
11. Jha A, Mamidi R (2017) When does a compliment become sexist? analysis and classification of ambivalent sexism using twitter data. In: Proceedings of the second workshop on NLP and computational social science. Association for Computational Linguistics, pp 7–16
12. Joulin A, Grave E, Bojanowski P, Mikolov T (2016) Bag of tricks for efficient text classification. arXiv:1607.01759
13. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. In: Proceedings of the 3rd international conference on learning representations (ICLR 2014)
14. Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86:2278–2324. https://doi.org/10.1109/5.726791
15. NewYorkTimes (2017) Twitter must do more to block isis. Website. https://www.nytimes.com/2017/01/13/opinion/twitter-must-do-more-to-block-isis.html. Accessed 30 Sept 2017
16. Nobata C, Tetreault J, Thomas A, Mehdad Y, Chang Y (2016) Abusive language detection in online user content. In: Proceedings of the 25th international conference on world wide web (WWW 2016). International World Wide Web Conferences Steering Committee, pp 145–153
17. Omer S, Lior R (2018) Ensemble learning: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, p e1249. https://doi.org/10.1002/widm.1249. Published Online: Feb 27 2018
18. Orrite C, Rodríguez M, Martínez F, Fairhurst M (2008) Classifier ensemble generation for the majority vote rule. In: Ruiz-Shulcloper J, Kropatsch WG (eds) Progress in pattern recognition, image analysis and applications. Springer Berlin Heidelberg, Berlin Heidelberg, pp 340–347
19. Park JH, Fung P (2017) One-step and two-step classification for abusive language detection on twitter. In: Proceedings of the 1st workshop on abusive language online at ACL 2017
20. Saha S, Ekbal A (2013) Combining multiple classifiers using vote based classifier ensemble technique for named entity recognition. Data Knowl Eng 85:15–39. Natural Language for Information Systems, Communicating with Anything, Anywhere in Natural Language
21. Schmidt A, Wiegand M (2017) A survey on hate speech detection using natural language processing. In: Proceedings of the 5th international workshop on natural language processing for social media. Association for Computational Linguistics, pp 1–10)
22. Vigna FD, Cimino A, Dell'Orletta F, Petrocchi M, Tesconi M (2017) Hate me, hate me not: hate speech detection on facebook. In: Proceedings of the 1st Italian conference on cybersecurity (ITASEC17), pp 86–95. http://ceur-ws.org/Vol-1816/paper-09.pdf
23. Warner W, Hirschberg J (2012) Detecting hate speech on the world wide web. In: Proceedings of the 2nd workshop on language in social media (LSM2012) LSM '12. Association for Computational Linguistics, pp 19–26
24. Waseem Z (2016) Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter. In: Proceedings of the first workshop on NLP and computational social science. association for computational linguistics, pp 138–142
25. Waseem Z, Hovy D (2016) Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In: Proceedings of the NAACL student research workshop. Association for Computational Linguistics

Dr. Georgios K. Pitsilis holds a B.Eng. in engineering, a B.Eng. in informatics an MSc and Ph.D. in computer science. He has worked as postdoc researcher at the University of Luxembourg, Q2S - NTNU Norway, and Newcastle University. He is currently postdoc researcher at the Department of Computer Science, NTNU. His research interest are in the areas of trust management, recommender systems, information retrieval, data mining and distributed computing.

Dr. Heri Ramampiaro is an Associate Professor at the Department of Computer Science, NTNU. He is head of the Data and Artificial Intelligence (DART) group, Deputy Head of Department, and Scientific Coordinator of the Telenor – NTNU AI-Lab. Ramampiaro's current main research interests include information retrieval, data/text mining and machine learning.

Dr. Helge Langseth is a Professor at the Department of Computer Science, NTNU, affiliated with the Data and Artificial Intelligence (DART) group. His current research interests include Probabilistic Graphical Models, Normative Decision Support Systems, and Machine Learning in general.