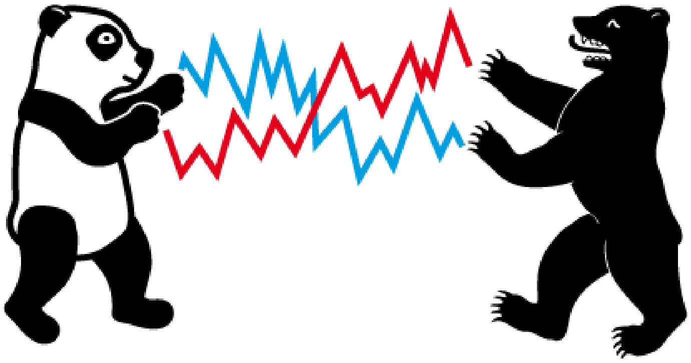
IRTG 1792 Discussion Paper 2019-029

Antisocial Online Behavior Detection Using Deep Learning

Elizaveta Zinovyeva [*]
Wolfgang Karl Härdle [*] [*2] [*3]
Stefan Lessmann [*]



[*] Humboldt-Universität zu Berlin, Germany
[*2] Singapore Management University, Singapore
[*3] Xiamen University, China

This research was supported by the Deutsche Forschungsgesellschaft through the International Research Training Group 1792 "High Dimensional Nonstationary Time Series".

http://irtg1792.hu-berlin.de
ISSN 2568-5619

Antisocial Online Behavior Detection Using Deep Learning[⋆]

Elizaveta Zinovyeva[a,*], Wolfgang Karl Härdle[a,b,c], Stefan Lessmann[a]

[a] *School of Business and Economics, Humboldt-Universität zu Berlin, Berlin, Germany*
[b] *Sim Kee Boon Institute for Financial Economics, Singapore Management University, Singapore, Singapore*
[c] *W.I.S.E. - Wang Yanan Institute for Studies in Economics, Xiamen University, Fujian, China*

Abstract

The shift of human communication to online platforms brings many benefits to society due to the ease of publication of opinions, sharing experience, getting immediate feedback and the opportunity to discuss the hottest topics. Besides that, it builds up a space for antisocial behavior such as harassment, insult and hate speech.

This research is dedicated to detection of antisocial online behavior detection (AOB) - an umbrella term for cyberbullying, hate speech, cyberaggression and use of any hateful textual content. First, we provide a benchmark of deep learning models found in the literature on AOB detection. Deep learning has already proved to be efficient in different types of decision support: decision support from financial disclosures, predicting process behavior, text-based emoticon recognition. We compare methods of traditional machine learning with deep learning, while applying important advancements of natural language processing: we examine bidirectional encoding, compare attention mechanisms with simpler reduction techniques, and investigate whether the hierarchical representation of the data and application of attention on differ-

[⋆] Financial support from the Deutsche Forschungsgemeinschaft via the IRTG 1792 “High Dimensional Non Stationary Time Series”, Humboldt-Universität zu Berlin, is gratefully acknowledged.

[*] Corresponding author

Email addresses: elizaveta.zinovyeva@hu-berlin.de (Elizaveta Zinovyeva), haerdle@hu-berlin.de (Wolfgang Karl Härdle), stefan.lessmann@hu-berlin.de (Stefan Lessmann)

ent layers might improve the predictive performance. As a partial contribution of the final hierarchical part, we introduce pseudo-sentence hierarchical attention network, an extension of hierarchical attention network – a recent advancement in document classification.

Keywords: Deep Learning, Cyberbullying, Antisocial Online Behavior, Attention Mechanism, Text Classification

1. Introduction

The shift of human communication to online platforms brings many benefits to society due to the ease of publication of opinions, sharing experience, getting immediate feedback and the opportunity to discuss the hottest topics. Besides that, it builds up a space for antisocial behavior such as harassment, insult and hate speech.

Detection of such antisocial behavior is of higher importance for social welfare due to financial, legislative and social reasons. According to the annual data of the Cyberbullying Research Center in 2016, 33.8 % of young people aged 12-17 in the US have experienced cyberbullying in their lifetime [1]. July 2, 2019, Facebook gets a $2.3 million fine because of violating German hate speech law [2]. According to this law, social media providers like Facebook, Google, Microsoft are obliged in Germany to remove hate speech posts within 24 hours and report on their progress every six months [3]. A few days after the government of France also considers announcing a law similar in application [4]. This emphasizes the importance for online platforms to find a solution on how to identify AOB. Manual detection and monitoring of online content can be very costly, this is why we need an automatic procedure to detect such behavior.

In this paper, we elaborate on the detection of antisocial online behavior (AOB) using DL methods. The term AOB is an umbrella term that describes any malicious behavior that can be found in the textual content on online communications platforms such as insult, threat, personal attack, usage of harmful, rude or offensive language, cyberbullying and abuse.

Early academic research on detection of AOB in online communication was mostly concentrated on the use of traditional machine learning methods (TML) such as logistic regressions, support vector machines and decision trees [e.g., 5], as well as lexicon-based approaches [e.g., 6]. These methods heavily rely on extensive feature engineering and performance highly depends on the representation of the data, deep learning (DL) methods automate the procedure of feature engineering by learning the representations of the data through non-linear transformations. Such representations often achieve much better performance than handcrafted features [7]. DL has proved to be efficient in different types of decision support: decision support from financial disclosures [8], predicting process behavior [9], text-based emoticon recognition [10] and many others. Also in the area of AOB detection DL modelling have gained a lot of popularity: within the last two years, the amount of academic research on deep learning models in the cyberbullying detection has grown exponentially. Later research shows a tendency of comparing the more sophisticated DL models from text classification with rather simpler architectures.

The main contribution of this work is the following: we provide a benchmark of DL structures found in the literature on AOB detection. We compare methods of TML with DL, while applying important advancements of natural language processing: we examine bidirectional encoding - a strategy to incorporate dependency of a future input to the model. Bidirectional processing have proved to be successful in the area of text-based emoticon recognition [10]. Moreover, we compare attention mechanisms, a way to reduce data while retaining information of intermediate hidden states and not only of the last hidden state, with simpler reduction techniques – global pooling layers. Finally, we investigate whether the hierarchical representation of the data and application of attention on different layers, a popular approach in document classification, might improve the predictive performance. As a partial contribution of the final hierarchical part, we introduce pseudo-sentence hierarchical attention network – an extension of hierarchical attention network – a recent advancement in document classification. The code is available on Github `https://github.com/QuantLet/AOBDL_code`.

2. Deep Learning Architectures

To fully appreciate the technical content, the reader might benefit from the following section where we describe the technology used in the previous academic literature on AOB detection: bidirectional recurrent neural networks, attention and pooling mechanisms, hierarchical learning approaches, as well as the model proposed in this work - pseudo-sentence hierarchical attention network.

On figure 1, we summarize our motivation on what machine learning approaches to include. The figure shows different types of methods and their drawbacks, which are simultaneously strengths of more complex models at the end of the arrow.

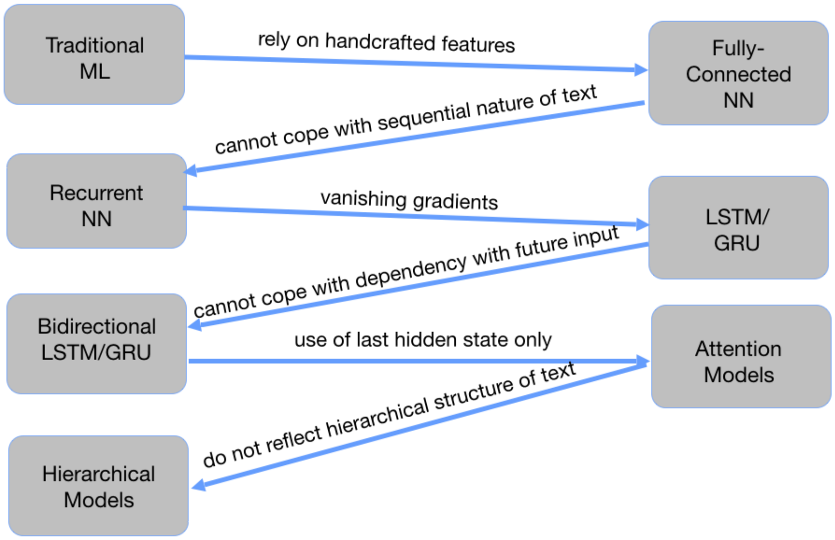

Traditional ML
rely on handcrafted features
Fully-Connected NN
cannot cope with sequential nature of text
Recurrent NN
vanishing gradients
LSTM/ GRU
cannot cope with dependency with future input
Bidirectional LSTM/GRU
use of last hidden state only
Attention Models
do not reflect hierarchical structure of text
Hierarchical Models

Figure 1: Overview of models

We start with methods of TML, and as mentioned in the introduction, these methods heavily rely on handcrafted features, whereas DL models help to learn abstract data representations and extract features automatically. A traditional fully connected neural network (NN) cannot cope with sequential data and introduces separate parameters for each time step separately, resulting in an insufficient generalization on sequences with length not seen during the training [7]. Networks with loops [11], based on a directed graph along the temporal component of data, recurrent neural networks (RNN) are able to work with sequences. Traditional RNNs, though, have impaired ability to deal with very long-term dependencies, the gradients propagated through the net might either vanish or explode [12, 13]. The models which can deal with such a “vanishing gradient” problem are long short-term memory (LSTM) [14] and gated recurrent unit (GRU) [15]. The recurrent neural networks usually have a causal structure: the state at time step t does depend only on the past information [7]. However, some applications require knowledge of the whole input sequence. In text classification, a word meaning might, in some cases, be dependent on other words at the end of the sentence. Bidirectional LSTM (BLSTM) and GRU (BGRU) proposed by Schuster and Paliwal [16] are models which can deal with such dependency of the future input. In the case of AOB detection, a bidirectional model may help us to differentiate between "idiot like me" and "idiot like you", where one of the sentences is meant to be insulting and other not. Attention models introduced by Bahdanau et al. [17] and other reduction techniques allow a model to memorize longer sequences, whereas hierarchical models help in reflecting the hierarchical structure of a text [18], which is advantageous in long text classification. By using attention, pooling layers, and hierarchical structure in AOB detection we aim to improve performance on longer posts. DL models that are included in the research will be described in greater detail in the following section.

2.1. Bidirectionality

Regular RNNs have a causal structure, the state at time t is trained only on the past information [7]. Some problems though, require information from the future or the whole input sentence. For example, in German language some prefixes of verbs are moved to the end of the sentence. To understand the meaning of the verb, whether we "switch on" or "switch off" the light for instance, the model needs to know the input from the end of the sentence.

Bidirectional RNN (BRNN) is a construct of two RNNs proposed by Schuster and Paliwal [16] in order to incorporate future temporal dynamics. The idea is to combine forward pass - one RNN going from the first to the last state with backward pass which goes vice versa. Outputs from forward states are not connected with those of backward pass. This extension of RNN allows to train the network in both time directions simultaneously [16]. BRNN is trained similarly as a regular RNN. Only if back-propagation through time is used, the forward and backward procedure are more complicated, since update of a state and output cannot be done simultaneously, special actions are required at the beginning and the end of training data. The forward state at $t = 1$ and $t = T$ are unknown, as well as local state derivatives, and set to 0.5 and 0 respectively [16]. The general structure can be see on the figure 2.

2.2. Dimensionality reduction with attention and pooling

The original Attention Mechanism was developed as a memory extension for the encoder-decoder (E-D) architecture, whereas the E-D architecture was

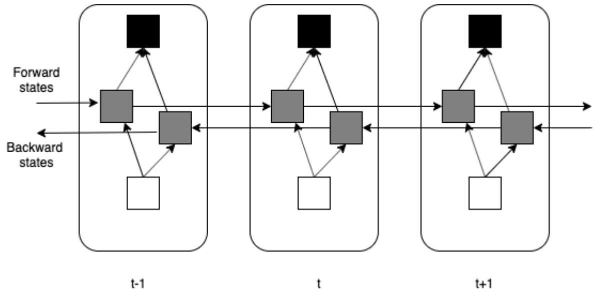


Figure 2: General structure of BRNN proposed by Schuster and Paliwal [16]

first proposed in two different papers, submitted almost at the same time, "Sequence to Sequence Learning with Neural Networks" by Sutskever et al. [19] and in "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation" by Cho et al. [15]. This structure is designed to tackle sequence-to-sequence modeling, where the input, as well as the output, are sequences. The domains that use such models are machine translation [e.g., 20], video captioning [e.g., 21], and speech recognition [e.g., 22].

Before the attention mechanism was invented, machine translation relied on encoding the whole input sequence into a one hidden state representation. And encoding data into only one hidden state representation might result in the loss of information. With an attention mechanism proposed by Bahdanau et al. [17], the model should be able to cope with a limitation of the classical E-D model – difficulty with decoding of long sentences. "This may make it difficult for the neural network to cope with long sentences, especially those that are longer than the sentences in the training corpus." [17]. Attentive model no longer encodes the full input sentence into a fixed-length vector – hidden state, yet, it enables the decoder to zoom or concentrate on different parts of the initial sentence while producing different elements of the output generation. In other words, instead of encoding the input sequence into a single fixed context vector, the attention model trains context vectors that identify inputs that are relevant for each output time step.

The attention mechanism that is used for text classification has a slightly different structure and acts as a reduction technique, such as average and max-pooling. By using the attention mechanism we introduce an additional context vector that summarizes the input on the different time steps and is co-trained with the model. By using the scalar product of the hidden representation of the input at different time steps, obtained through an additional one-layer fully-connected network, with context vector state, we measure their similarity. This similarity score is used to weight the input [18]. As compared to pooling techniques, we introduce additional trainable parameters to the model and a "smart" way to get our model to concentrate only on the important input.

Global Pooling Layers can be seen as a simpler alternative to the attention mechanism. Pooling layers are usually used in the context of convolutional neural networks (CNNs) and computer vision. Similarly to the attention layer, they also act as a reduction technique by taking the feature maps and transforming them into one vector [23]. Nonetheless, pooling layers can also be used in combination with recurrent neural networks, such as LSTMs and GRUs. They take the sentence matrix produced through embedding and encoding and reduces it to one vector. The most common approaches are averaging and taking the maximum along the embedding dimension, which subsequently results in the names Global Average Pooling and Global Max Pooling. They are less complex, as compared to the attention layer due to the fact they do not require additional parameters to be trained, which leads to the faster training process.

2.3. HAN and PsHAN

One of the models using attention mechanism on multiple levels is the hierarchical attention network (HAN). HAN was first proposed by Yang et al. [18] for document classification tasks. The authors have tested it on six different data sets of different sizes, reflecting the hierarchical structure of the text shown a positive effect on the models' performances. This multi-leveled structure should enable the model to pay more or less attention to different parts of content when constructing a representation of the document. Different words are differently important by depicting the whole essence of one sentence. Moreover, sentences are differently important when we describe the meaning of the whole post or a document. Moreover, the same word can have different meanings, which depend on the context. The hierarchical attention network is depicted in figure 3.

HAN is designed to classify documents from a corpus to different categories. The first part of the model is the word encoder, every document is broken down into sentences, where each of the sentences is encoded separately: first each word will be embedded into an embedding matrix, afterward using bidi-

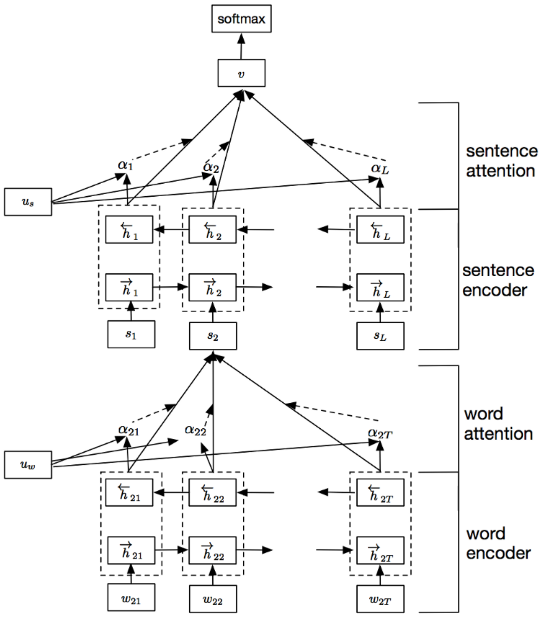


Figure 3: HAN Architecture. Proposed by Yang et al. [18].

rectional GRU (BGRU) words are encoded by summarizing context within the sentence for both directions, so each word is represented by its surrounding context. The next part of the mechanism is the word attention. Not each word is equally important for the meaning of the sentence. Therefore, word annotations obtained in the previous step are fed into a one-layer feed-forwarded neural network to generate hidden-state representation, which is then compared to the context vector through the softmax function resulting in normalized importance weights. These weights, afterward, are used to compute sentence vectors by weighting the sum of word annotations by their importance scores.

A similar procedure is done on the sentence level since not every sentence within a document contributes equally to the meaning of the document. Through BGRU each sentence within a document is encoded by its surrounding context, then hidden-representation is generated through a feed-forwarded neural network, which is then compared to a sentence level context vector. Obtained importance scores are used to weigh encoded sentences, the sum of which results in the document representation. Classification into categories is done in the very last step though a softmax one-layer feed-forwarded network.

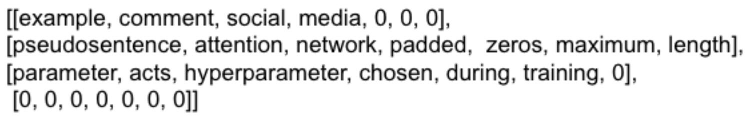
In the AOB detection though the length of posts varies extensively and rarely posts are very long documents, which HAN is normally used for.

In the next step, we propose the pseudo-sentence HAN (psHAN) algorithm - the extension of HAN. To the best of our knowledge, this adjustment of HAN has not been issued in the previous academic research. The motivation to adjust the HAN algorithm is twofold: In the classical HAN, the pre-processing step requires an additional split of the posts into sentences according to punctuation marks, which enforces to maintain all other pre-processing steps for each sentence independently after the split, otherwise, some pre-processing steps are not feasible. Whereas psHAN does not require additional splitting beforehand, which simplifies and accelerates the pre-processing procedure. Moreover, the technical implementation of HAN can be connected to an essential drawback: the length of a sentence, the number of sentences per post can have high variability, which is usually the case in AOB detection. And even though in theory using recurrent networks we can encode sequences of variable length, in practice during the implementation one has to set up a threshold on the length of a sequence, and HAN has even two such thresholds. Having a positively skewed distribution of the sentence length and the number of sentences, taking the maximum values for the threshold can lead to very long computational times. Whilst cutting off twice, on the sentence and post level, can remove a lot of essential for prediction information: for some sentences which are longer than the threshold we would remove parts of the text while adding zeros to the shorter sentences. Therefore, we propose pseudo-sentence hierarchical attention network psHAN – networks where the input documents are split not into real sentences but a set of sequences of an arbitrarily set length, so the input forms a list of sequences without separation into real sentences. Pseudo-sentence – a sequence of words from a sentence but not necessarily the whole sentence, the length of pseudo-sentence is treated as a meta-parameter, i.e. we choose the length during model selection and tuning.

Hence, having psHAN we are setting the cutoff only once - on the number of words per post, we will call it N, so padding procedure is done also only once. Further, we introduce two additional hyper-parameters: length of a sentence and the number of sentences per input document, as in the HAN architecture T and L respectively. In the case of psHAN, the following equality holds $L \times T = N$, which is not necessarily true for the HAN model.

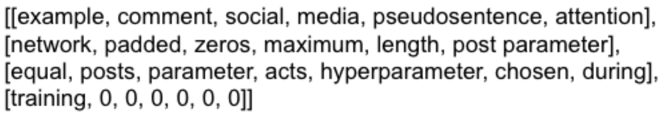
By rearranging data, we can gain the profit of the hierarchical structure at the same time not removing any information, every post is just split into several sequences of a specific word amount.

This can be exemplified in the following artificial case. Imagine having the following input sentence: *This is an example of a comment on social media. In pseudo-sentence attention network, it will be padded with zeros to the maximum length of a post parameter, which is equal for all the posts. This parameter acts as a hyper-parameter and is chosen during the training.* Assuming we set $T = 4$ and $L = 7$, so i.e., for psHAN $N = 28$. On the figures 4 and 5 we can see how the input will be tokenized for HAN and psHAN, respectively:



```
[[example, comment, social, media, 0, 0, 0],
[pseudosentence, attention, network, padded,  zeros, maximum, length],
[parameter, acts, hyperparameter, chosen, during, training, 0],
 [0, 0, 0, 0, 0, 0, 0]]
```

Figure 4: Tokenized Input for HAN



```
[[example, comment, social, media, pseudosentence, attention],
[network, padded, zeros, maximum, length, post parameter],
[equal, posts, parameter, acts, hyperparameter, chosen, during],
[training, 0, 0, 0, 0, 0, 0]]
```

Figure 5: Tokenized Input for psHAN

In the case of HAN, we had to introduce more zeros for shorter sentences while removing some of the words for longer sentences. In this work, we expect that the effect of having less reduced input will overweight the effect of having semantically completed input sequences. The rest of the psHAN's architecture remains the same as of HAN's, with the only difference that instead of real sentences we have pseudo-sentences.

3. Related Work

The amount of literature on DL for AOB detection has exponentially grown within the last two years. DL techniques can be used on different

Table 1: Deep learning in AOBD - literature review

		Algorithm								Other Details	
Study (in chronological order)	Details	RNN	LSTM/GRU	MLP/DNN	CNN	BLSTM/BGRU	ATTENTION	CNN/LSTM Hybrid	OTHER	Tokenization[1]	Text Features Solely Hierarchy
Potha and Maragoudakis [24]	Time-series modeling with Singular Value Decomposition and SVM. Comparison to MLP. Perverted ustice data			x						w	1
Mehdad and Tetreault [25]	Comparison of features on character and word level, distributional representation of comments compared to Recurrent Neural Network Language Model and Naive Bayes SVM	x								c, w	1
Zhang et al. [26]	Pronunciation-based CNN to deal with misspellings which do not affect words' pronunciation. Aim - practical, robust, universal method with high performance. Max Pooling				x					w	1
Badjatiya et al. [27]	Compare DL with TML: RF, SVM, GBDT, usage of pre-trained embeddings		x		x					c[2], w	1
Gao and Huang [28]	Incorporate contextual information, Fox News User Comments		x			x	x			c, w	0
Pavlopoulos et al. [29]	Automatic comment moderation. Camparison of different GRU based attention models with CNNs, and detox - model based on MLP and LR		x	x	x		x			w	1
Ptaszynski et al. [30]	Convolutional neural networks used on data in Japanese from unofficial school websites and fora. Part of speech POS, named entity recognition NER features				x					w	1
da Silveira Marciano et al. [31]	Extreme learning machine for cyberbullying detection in Portuguese language. Data from FB, Twitter, Brazilian news sites, collaborative tool for typical cyberbullying phrases identification								x	w	1
Vishwamitra et al. [32]	Two-level cyberbullying detection for mobile devices - Android appplication. Pronunciation-based CNN				x					w	1
Agrawal and Awekar [33]	Multiple social media platforms, transfer learning classification, usage of pre-trained embeddings		x		x	x	x			c, w	1
Al-Ajlan and Ykhlef [34]	CNN use of pretrained embeddings, Glove, Twitter data, comparison with SVM, Max Pooling. Literature divided into detection types				x					w	1
Aroyehun and Gelbukh [35]	Usage of data augmentation, pseudo-labelling techniques, pre-trained embeddings, Facebook data in English and Hindi, DL models vs NBSVM as a non-DL baselineLSTM,		x		x	x		x		c[3], w	1
Bu and Cho [36]	Ensemble of character level CNN vs word-level LRCN, usage of word embeddings for LCRN, max pooling for CNN		x		x			x		c, w	1

[1] w - word-level, c - character-level, [2], [3] character level is used only for traditional ML method

Table 2: Deep learning in AOBD - literature review (cont.)

Study (in chronological order)	Details	Algorithm								Other Details	
		RNN	LSTM/GRU	MLP/DNN	CNN	BLSTM/BGRU	ATTENTION	CNN/LSTM Hybrid	OTHER	Hierarchy Tokenization[4]	Text Features Solely
Chen et al. [37]	2D TF-IDF vs 1D TF-IDF and pre-trained embeddings, Twittert data, Max Pooling		x		x					w	1
Dadvar and Eckert [38]	Reproducibility study, Wikipedia, Twitter, and Formspring, YouTube datasets. Random, GloVe and SSWE embeddings		x		x	x	x			w	1
Fortuna et al. [39]	Italian language, Facebook and Twitter data			x						w	1
Founta et al. [40]	RNN-based networks with attention, additional meta-data features. Twitter		x							c, w	0
Georgakopoulos et al. [41]	CNN vs. different TML models on Wikipedia talk pages data				x					w	1
Ibrahim et al. [42]	Imbalanced data, data augmentation, Wikipedia data, Ensembles CNN				x	x				w	1
Khatua et al. [43]	Exploration of sexual violence on Twitter, gender-based violence, #MeToo movement, types of assault classification		x	x	x	x				w	1
Kumar et al. [44]	Benchmarking previous literature on AOBD. Mixed Hindi-English and other languages, e.g. German. Only descriptions of models used by other authors, partially reflected in other literature										
Pitsilis et al. [45]	Use of user-behavioral characteristics through text, knowledge of previous user's behavior, pre-trained embeddings, ensembles of LSTMs, Twitter data		x							w	0
Polignano and Basile [46]	Twitter, Facebook with Italian data. MLP compared to SVM, KNN, RF and other models of TML, usage of pre-trained embeddings word2vec			x						w	1
Raisi and Huang [47]	Weak-supervision weekly, ensemble of two co-trained learners, graph-node representation, pre-trained embeddings used		x							w, d	0
Risch and Krestel [48]	Bidirectional GRU with average pooling, English and Hindi FB data set and English Twitter, data augmentation and usage of word embeddings, ensemble with gradient boosting trees					x				c[5], w	1
Rosa et al. [49]	Differemt CNN-based models, usage of word2vec, DL vs DVM and LR, balanced and unbalanced data				x			x	x	w	1

[4] w - word-level, c - character-level, d - document-level, [5] character level used only for logistic regression not for DL models

Table 3: Deep learning in AOBD - literature review (cont.)

		Algorithm								Other Details		
Study (in chronological order)	Details	RNN	LSTM/GRU	MLP/DNN	CNN	BLSTM/BGRU	ATTENTION	CNN/LSTM Hybrid	OTHER	Tokenization[6]	Hierarchy	Text Features Solely
Tommasel et al. [50]	Pre-trained embeddings, sentiment features using SentiWordNet corpus, composed features TF-IDF. sentiment and punctuation related features, Gaussian noise after input layer		x							c, w		1
van Aken et al. [51]	Error-Analysis, DL models compared with LR, pre-trained embeddings, common challenges, ensemble learning, multi-label classification, Twitter and Facebook data		x		x	x	x			c[7], w		1
Zhong et al. [52]	Cyberaggression detection using convolutional networks, differentiation between cyberbullying and cyberaggression, session level bully incident, text and image features, Instagram data max pooling layer, comparison with LR, comment level, usage of word embeddings				x					w		0
Zimmerman et al. [53]	Ensembles of convolutional NNs, Twitter data				x					w		1
Cheng et al. [54]	HAN, attention on word and comment level, usage of word-embeddings. Compare to KNN, NB, LR, RF, XGBoost, Instagram data		x		x		x		x	c, w	1	1
Fagni et al. [55]	Data from Facebook and Twitter in Italian, limit feature engineering phase, Dl models and ensemble. Compare with SVM		x		x			x		w		1
Pandey et al. [56]	Sexual Assault intent detection, Twitter data, convolutional networks, usage of Part-of-Speech tags and pre-trained embeddings				x					w		1
Santosh and Aravind [57]	Hierarchical attention model, only one attention layer, word and syllable encoder, compare to SVM, RF, Twitter data		x				x		x	w, s	1	1

[6] w - word-level, c - character-level, s - syllable-level, [7] character level used only for logistic regression not for DL models

steps of the modeling process. One can use it solely for feature extraction to learn abstract representations of data and build another TML classifier on the top, e.g., Zhong et al. [58], or one can use deep learning networks for the whole training process, the feature extraction as well as classification. In this research, we concentrate on academic work that uses DL for the entire training process.

In the tables 1, 2 and 3 we depict papers on DL in AOB detection. We identify some details, algorithm, type of tokenization for textual input as well as whether the researchers train their models solely on textual features.

Earlier research is dedicated to architectures such as CNN and LSTM, as well as fully connected DNN and RNN models [24, 25]. In their work, da Silveira Marciano et al. [31] have used extreme learning machines – a class of neural networks, proposed by Huang et al. [59], where the parameter weights are specified on random and only upper layer weights are trained, to classify data whether it contains cyberbullying in the Portuguese language.

Later research on DL models in AOB detection tends to use more complex models and constructs, such as bidirectional recurrent networks, attention mechanism and even hierarchically structured data in combination with attention. Agrawal and Awekar [33] compare the performance of CNN, LSTM, bidirectional LSTM (BLSTM), and BLSTM with attention for training and validation for task-specific as well as for transfer learning. Santosh and Aravind [57] use a hierarchical model, where data is aggregated first on the syllable level and then on word level. The attention mechanism is applied further on word level. In another work, Cheng et al. [54] applied hierarchical attention model HAN on Instagram data and compared it to the performance of another TML classifiers, KNN, NB, LR, RF, and XGBoost. For their research, van Aken et al. [51] used LSTM, BLSTM, BGRU, BGRU with attention and CNN on Twitter and Wikipedia data. Furthermore, they identified common challenges in cyberbullying detection, such as doubtful labels and rhetorical questions, and performed in-depth error-analysis.

Interestingly, almost all the papers use tokenization on the word level and do not add additional variables apart from the text itself. Moreover, the majority of research is connected to classification the data in English language, nonethelsess, there are examples in other languages: Portuguese [31], Japanese [30], Italian [39], mix of Hindi and English [44, 48]. The most popular data sets used are coming from Twitter [e.g., 39, 46, 48, 51], Facebook [e.g., 55, 31, 39], Wikipedia [e.g., 42, 33, 41].

Current research uses sophisticated techniques such as attention conventionally compares it with either TML models or with DL models without other dimensionality reduction. To the best of our knowledge, in AOB detection, the pooling layers stacked onto recurrent layers have never been compared with the attention layer. Max pooling layers have been used only in combinations with CNN so far [e.g., 37]. Therefore, in our work, as one of the contributions, we want to identify whether we need the attention mechanism leading to the introduction of additional trainable parameters or whether global pooling can perform as good, keeping the model simpler in terms of the amount of co-trainable weights.

4. Experimental Design

In this section, we describe our experimental design including datasets, models, pre-processing and evaluation procedure we used.

4.1. Dataset

Data used for the experiments comes from a Kaggle competition "Toxic Comment Classification" [60]. The data is provided by Jigsaw, a project of Google — a technology incubator where researchers try to improve online communication by preventing cyberbullying, protection of the speech right, offering services preventing DDoS attacks on the websites about media, elections, and human-rights content, etc. [61]. One of the main areas of investigation is so-called "toxic comments, content that can be classified as "rude, disrespectful, or unreasonable comment that is likely to make people leave a discussion" [62]. This competition is dedicated to the identification of different levels of toxicity in the Wikipedia Talk Pages. The problem definition is organized as a multi-label classification problem with the following classes: toxic, severe toxic, obscene, insult, threat, identity hate. We decided to binarize the data to the *malicious* and *non-malicious* classes since the focus of our work lies on providing a benchmark in general rather than on multi-label classification.

Data contains 223,549 labeled data points. 29% of the labeled data were drawn for the test data, stratified sampling was used to maintain the same ratio of the malicious to non-malicious classes in the data. The percentage of the malicious content in the training set and the test set is around 10%. For the training and validation we have 158,719 and for the testing 64,830 observations.

4.2. Pre-processing

First, the data was transformed into the lower case. Short versions of negative contractions with apostrophes have been substituted with their full version. The negation "not" has usually shown good predictive performance, and if they would not have been substituted, in the step of deletion of additional punctuation this negation would have got lost. Moreover, we substituted the emojis with their semantic equivalent. Such information has shown to be very helpful in the tasks of sentiment analysis and classification of malicious content. After substitution was completed, the set of following stopwords, the words that are either used very often – have an equal likelihood to occur in all documents relevant for the task [63] or do not hold any additional semantic meaning for this task, was removed: [a, the, an, are, as, did, do, is, has, have, had, was, were, will, would, am, it, for, on, it, of]. The pronounce "you" showed importance in the identification of malicious content, therefore it was not removed.

Besides, we deleted repeating parts of the text, URLs, IP-addresses, usernames. Finally, the removal of all non-alphanumeric characters was performed.

4.2.1. Pre-processing for TML

For TML classifiers the TF-IDF tokenization was performed on the n-gram wording level, where $n = 1$. TF-IDF, short for term frequency—inverse document frequency, reflects how important a word is to a document in a collection or corpus [64]. The final meta-parameter of the maximum amount of the features was set to 40,000.

4.2.2. Pre-processing for DL

For all DL models tokenization was performed on the word level. The maximal amount of features have been also set to 40,000 in order to maintain compatibility with TML methods. As opposed to the TML, we did not apply the TF-IDF features weighting scheme. Distinct integer token IDs were assigned to the individual words. The aim of not deploying the TF-IDF is connected to the fact that for DL models embedding techniques will be applied. Finally, individual posts were padded to the maximum amount of words in the post, which was most frequently 400. Padding sequences with zeros is required if the sentence is shorter than the parameter – the maximum amount of elements in a post. Hence, additional zeros added to achieve the same length for all posts. The maximum amount of words can also be seen as a meta-parameter.

4.3. Evaluation

Cross-validation with $k = 5$, where k is the number of folds, was used for model evaluation to achieve a more reliable result and greater generalization rather than if we used the train-validation-test split. Stratified sampling was used.

For all models, manual hyper-parameter tuning was performed in order to ensure that difference in the performance arises through architecture characteristics and not through differentiation of hyper-parameters.

While dealing with AOB detection, we usually have to deal with a very small portion of the positive class observation – i.e., we have the data imbalance problem. Often the area under the receiver operating characteristic curve ROC, AUC ROC, is used for evaluation in the situation where we have to deal with data imbalance. Nonetheless, we decided to use the area under the precision-recall curve, AUC PR-C. The reason for using AUC under PR-C instead of AUC ROC is the fact that even high changes in the number of false positives can lead to a small change in the false positive rate used in ROC analysis [65], if we have a sufficiently large amount of true negatives. In order to avoid this problem, we should use the metric that does not include the amount of the true negative into the calculation of the score, e.g. AUC PR-C.

Thus, in the analysis we report metrics AUC PR-C, AUC ROC, as well as the F1 score. However for parameter tuning and model selection, PR-C space was used as the primary criterion. The threshold for the F-Score was set to 0.3, after selecting it using cross-validation for multiple models like LSTM, GRU, BGRU, BLSTM.

5. Experiments

5.1. TML vs. DL

As a first sub-experiment, we are going to compare methods of traditional machine learning with LSTM, CNN and GRU - the basic architectures the most common DL architecture are based on. For that purpose we decided to take one the most popular models from AOB detection with TML: logistic regression with l2 regularization LR, random forest RF [e.g., 33, 27, 5]. Moreover, we chose gradient boosting trees LightGBM as a model, its ensembles are often used in data science competitions for structured data as winning model.

In table 4, we depict the results in terms of AUC PR-C, AUC ROC and the F1 Score. As we expected, DL models outperform TML models, by approximately 0.02 in terms of average precision. The difference in the performance is higher in AUC PR-C and F1 Score than in AUC ROC.

Interestingly, the decision-tree ensemble-based models, which are seen to perform strongly with structured data, showed the worst performance in terms of AUC PR-C while having the smallest gap between CV and test values and having a high performance according to AUC ROC. Among all TML models logistic regression achieved the best performance.

GRU model has shown the best performance among DL models. The difference in the performance of different types of recurrent neural networks, GRU over LSTM, is quite marginal. Surprisingly, CNN has shown the most unsatisfactory result. One of the possible reasons could be a higher amount of meta-parameters and not so natural way of representing the text as a block, compared to the typical application domain of CNN networks - computer vision.

Our results support the idea that neural network-based architectures are more powerful in AOB detection, as a type of text classification than the traditional machine learning algorithms. The highlighted numbers represent the best performance in each category.

5.2. Bidirectionality

In the second experiment, we added bidirectionality to our recurrent networks, in order to understand, whether in AOB detection classification process depends on the future input. The results can be found in the table 5. Compared to the results in experiment 1, we can see a marginal improvement of the performance in terms of AUC PR-C of the GRU model. Nonetheless, the improvement is marginal that if using the models in production, one could consider using a model with fewer parameters in order to reduce computational costs. Lower F1 Score of the bidirectional model even supports the idea of preferring a rather simpler model. GRU-gate based models still show better performance than LSTM-based, which emphasizes that a simpler GRU unit can outperform LSTM that requires a greater amount of trainable parameters. Adding bidirectionality to the LSTM has even marginally

Table 4: Results of Experiment 1: TML vs. DL

Model	Folds	AUC Precision/Recall		AUC ROC		F1 Score	
		CV	Test	CV	Test	CV	Test
LR (Ridge)	CV Fold 1	0.869		0.969		0.760	
	CV Fold 2	0.866		0.968		0.760	
	CV Fold 3	0.867		0.968		0.766	
	CV Fold 4	0.814		0.965		0.723	
	CV Fold 5	0.776		0.963		0.687	
	CV Average	0.838	0.833	0.967	0.967	0.739	0.745
RF	CV Fold 1	0.846		0.960		0.766	
	CV Fold 2	0.854		0.964		0.773	
	CV Fold 3	0.847		0.961		0.766	
	CV Fold 4	0.787		0.960		0.720	
	CV Fold 5	0.737		0.961		0.652	
	CV Average	0.814	0.814	0.961	0.963	0.737	0.741
SVM	CV Fold 1	0.862		0.964		0.720	
	CV Fold 2	0.862		0.962		0.729	
	CV Fold 3	0.864		0.963		0.726	
	CV Fold 4	0.812		0.963		0.715	
	CV Fold 5	0.781		0.963		0.698	
	CV Average	0.836	0.827	0.963	0.962	0.718	0.722
LightGBM	CV Fold 1	0.851		0.960		0.751	
	CV Fold 2	0.854		0.959		0.759	
	CV Fold 3	0.850		0.958		0.761	
	CV Fold 4	0.798		0.956		0.721	
	CV Fold 5	0.764		0.958		0.680	
	CV Average	0.830	0.819	0.961	0.958	0.735	0.736
LSTM	CV Fold 1	0.886		0.976		0.800	
	CV Fold 2	0.884		0.973		0.800	
	CV Fold 3	0.888		0.973		0.814	
	CV Fold 4	0.835		0.970		0.719	
	CV Fold 5	0.806		0.970		0.689	
	CV Average	**0.860**	0.855	0.972	0.971	**0.764**	0.767
GRU	CV Fold 1	0.887		0.975		0.804	
	CV Fold 2	0.887		0.973		0.794	
	CV Fold 3	0.894		0.977		0.812	
	CV Fold 4	0.834		0.971		0.724	
	CV Fold 5	0.799		0.970		0.665	
	CV Average	**0.860**	**0.859**	**0.973**	**0.973**	0.760	**0.768**
CNN	CV Fold 1	0.882		0.975		0.783	
	CV Fold 2	0.881		0.974		0.798	
	CV Fold 3	0.882		0.974		0.799	
	CV Fold 4	0.829		0.970		0.725	
	CV Fold 5	0.787		0.967		0.641	
	CV Average	0.852	0.850	0.972	0.971	0.749	0.762

Table 5: Results of Experiment 2: Bidirectionality

Model	Folds	AUC Precision/Recall		AUC ROC		F1 Score	
		CV	Test	CV	Test	CV	Test
BLSTM	CV Fold 1	0.882		0.971		0.797	
	CV Fold 2	0.845		0.956		0.739	
	CV Fold 3	0.889		0.974		0.804	
	CV Fold 4	0.836		0.970		0.740	
	CV Fold 5	0.795		0.969		0.665	
	CV Average	0.850	0.852	0.968	0.971	0.749	0.748
BGRU	CV Fold 1	0.890		0.977		0.805	
	CV Fold 2	0.887		0.975		0.805	
	CV Fold 3	0.892		0.974		0.813	
	CV Fold 4	0.833		0.970		0.737	
	CV Fold 5	0.805		0.971		0.695	
	CV Average	**0.861**	**0.860**	**0.973**	**0.973**	**0.771**	**0.758**

worsen its performance on the test data and led to the increase of discrepancy of AUC PR-C between CV and test. It indicates that models of higher complexity for AOB detection on Wikipedia data might lead to overfitting.

5.3. Attention vs. Pooling

In the introductory and literature parts, we mentioned that AOB research quite often compares models with sophisticated reduction techniques with models without reduction techniques at all. Further, we introduce reduction techniques: we compare the attention mechanism with global maximum and average pooling by using BGRU and BLSTM + different reduction techniques. Therefore, we want to scrutinize, whether attention mechanism is needed for our recurrent based model or we can achieve sufficient performance using just global pooling. For that purpose, we take the best performing bidirectional GRU-based model and add attention, average global pooling, and maximum global pooling to it. The results are depicted in the table 6. Bidirectional GRU model with maximum pooling outperformed the other two reduction techniques, attention, and average pooling. The difference in the performance between all three reduction techniques is marginal. Compared to the previous experiments reduction techniques have decreased the performance.

5.4. Hierarchical Attention Models

In the last step, we dive into hierarchical attention models to investigate whether reflecting hierarchical structure that benefits in the document classification might also improve the performance of machine learning models in

Table 6: Results of Experiment 3: Attention vs. Pooling

Model	Folds	AUC Precision/Recall		AUC ROC		F1 Score	
		CV	Test	CV	Test	CV	Test
BGRU + Att.	CV Fold 1	0.887		0.972		0.805	
	CV Fold 2	0.878		0.969		0.787	
	CV Fold 3	0.877		0.971		0.789	
	CV Fold 4	0.822		0.968		0.718	
	CV Fold 5	0.792		0.968		0.689	
	CV Average	0.851	0.856	0.970	**0.972**	0.758	0.757
BGRU + Avg.	CV Fold 1	0.890		0.976		0.808	
	CV Fold 2	0.879		0.970		0.797	
	CV Fold 3	0.895		0.976		0.817	
	CV Fold 4	0.836		0.970		0.742	
	CV Fold 5	0.793		0.966		0.702	
	CV Average	**0.859**	0.855	0.972	0.971	**0.773**	**0.769**
BGRU + Max	CV Fold 1	0.887		0.976		0.794	
	CV Fold 2	0.883		0.973		0.799	
	CV Fold 3	0.889		0.975		0.802	
	CV Fold 4	0.834		0.969		0.714	
	CV Fold 5	0.803		0.970		0.658	
	CV Average	**0.859**	**0.857**	**0.972**	**0.972**	0.754	0.764

the case of AOB detection. Therefore we use HAN and psHAN proposed in the Deep Learning Architectures section.

In other words, we investigate whether there is a need to separate social media comments into sentences and use this hierarchical structure to improve predictive performance. As we can see in table 7, the use of such hierarchy only reduces area under precision recall and under the ROC curves, i.e., HAN performs worse than just GRU or BGRU. Interestingly, the use of pseudo-sentences in psHAN shows better results than the original HAN, nonetheless, still loosing to the best models GRU and BGRU.

5.5. Best Models

In table 8, we depicted all the best performing models throughout experiments. As we can see the BGRU outperformed all other models, following by the GRU. Moreover, GRU as well as BGRU has no gap between training and testing performance according to AUC ROC. As stated before bidirectionality has marginally improved the performance of a simpler GRU and in production we might tend to choose the model of lower complexity.

All the reduction techniques have decreased the performance, compared to the model without any reduction. Therefore, this type of increasing complexity of the model might even en-worsen the result. Our results emphasize the fact that it is not always recommendable to use reduction over different

Table 7: Results of Experiment 3: Hierarchical models

Model	Folds	AUC Precision/Recall		AUC ROC		F1 Score	
		CV	Test	CV	Test	CV	Test
HAN	CV Fold 1	0.870		0.965		0.791	
	CV Fold 2	0.863		0.966		0.777	
	CV Fold 3	0.866		0.964		0.782	
	CV Fold 4	0.812		0.958		0.721	
	CV Fold 5	0.784		0.963		0.666	
	CV Average	0.839	0.833	0.963	0.964	0.747	0.748
psHAN	CV Fold 1	0.879		0.971		0.788	
	CV Fold 2	0.880		0.971		0.795	
	CV Fold 3	0.886		0.972		0.804	
	CV Fold 4	0.829		0.969		0.752	
	CV Fold 5	0.791		0.968		0.708	
	CV Average	**0.853**	**0.853**	**0.970**	**0.971**	**0.769**	**0.768**

time-steps of our encoder but sometimes having just the last hidden-state is more efficient.
Reflecting hierarchical structure for AOB detection has also decreased the performance. A probable explanation for this fact is the higher variability of the length in the posts and in general shorter texts, as compared to the HAN domain – document classification. Nonetheless, psHAN, among all DL models, shows the smallest discrepancy between cross-validation and test performance, which indicates the robustness of the model.

6. Conclusion and Further work

The first aim of this work is to present a benchmark of DL models for AOB detection used in the existing literature. We showed that GRU-based models perform the best and all DL models outperform methods of TML. Further, we were able to show that predictive performance slightly improves when we introduce a bidirectional recurrent layer, as the future input is important to understand the meaning of the word now. Additionally, we concluded that in the case of Wikipedia data we do not need any reduction techniques, neither hierarchical structure is required. A rather simple GRU and bidirectional GRU outperform models with additional structures. Moreover, the proposed psHAN model outperforms the original HAN and also shows the smallest discrepancy in the performance between cross-validation and test sets among all DL models. For further work, we are planning to address the problem of noisy labels and use similar techniques as in the semi-supervised learning

Table 8: Best models

Model	Folds	AUC Precision/Recall		AUC ROC		F1 Score	
		CV	Test	CV	Test	CV	Test
GRU	CV Fold 1	0.887		0.975		0.804	
	CV Fold 2	0.887		0.973		0.794	
	CV Fold 3	0.894		0.977		0.812	
	CV Fold 4	0.834		0.971		0.724	
	CV Fold 5	0.799		0.970		0.665	
	CV Average	0.860	0.859	**0.973**	**0.973**	0.760	**0.768**
BGRU	CV Fold 1	0.890		0.977		0.805	
	CV Fold 2	0.887		0.975		0.805	
	CV Fold 3	0.892		0.974		0.813	
	CV Fold 4	0.833		0.970		0.737	
	CV Fold 5	0.805		0.971		0.695	
	CV Average	**0.861**	**0.860**	**0.973**	**0.973**	**0.771**	0.758
BGRU + Max	CV Fold 1	0.887		0.976		0.794	
	CV Fold 2	0.883		0.973		0.799	
	CV Fold 3	0.889		0.975		0.802	
	CV Fold 4	0.834		0.969		0.714	
	CV Fold 5	0.803		0.970		0.658	
	CV Average	0.859	0.857	0.972	0.972	0.754	0.764
psHAN	CV Fold 1	0.879		0.971		0.788	
	CV Fold 2	0.880		0.971		0.795	
	CV Fold 3	0.886		0.972		0.804	
	CV Fold 4	0.829		0.969		0.752	
	CV Fold 5	0.791		0.968		0.708	
	CV Average	0.853	0.853	0.970	0.971	0.769	**0.768**

to correct wrongly assigned labels, extend the research to different data sets and investigate whether the use of pre-trained transformer models might outperform models trained only on the domain data set.

References

[1] J. W. Patchin, 2016 cyberbullying data, https://cyberbullying.org/2016-cyberbullying-data, 2016. Accessed at July 5, 2019.

[2] Q. Wong, Facebook fined $2.3 million for violating germany's hate speech law, https://www.cnet.com/news/facebook-fined-2-3-million-for-violating-germanys-hate-speech-law/, 2019. Accessed at July 5, 2019.

[3] Deutsche Welle, Germany fines facebook for underreporting hate speech complaints, https://www.dw.com/en/germany-fines-facebook-for-underreporting-hate-speech-complaints/a-49447820-0, 2019. Accessed at July 10, 2019.

[4] M. Kelly, France wants to fine facebook over hate speech, https://www.theverge.com/2019/7/4/20682513/french-parliament-facebook-google-social-network-hate-speech-removal, 2019. Accessed at July 5, 2019.

[5] T. Davidson, D. Warmsley, M. Macy, I. Weber, Automated hate speech detection and the problem of offensive language, in: Eleventh international aaai conference on web and social media, 2017.

[6] U. Bretschneider, R. Peters, Detecting cyberbullying in online communities, in: ECIS, 2016, p. ResearchPaper61.

[7] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, Deep learning, volume 1, MIT press Cambridge, 2016.

[8] M. Kraus, S. Feuerriegel, Decision support from financial disclosures with deep neural networks and transfer learning, Decision Support Systems 104 (2017) 38–48.

[9] J. Evermann, J.-R. Rehse, P. Fettke, Predicting process behaviour using deep learning, Decision Support Systems 100 (2017) 129–140.

[10] B. Kratzwald, S. Ilić, M. Kraus, S. Feuerriegel, H. Prendinger, Deep learning for affective computing: Text-based emotion recognition in decision support, Decision Support Systems 115 (2018) 24–35.

[11] D. E. Rumelhart, G. E. Hinton, R. J. Williams, et al., Learning representations by back-propagating errors, Cognitive modeling 5 (1988) 1.

[12] S. Hochreiter, Untersuchungen zu dynamischen neuronalen netzen, Diploma, Technische Universität München 91 (1991).

[13] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, et al., Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.

[14] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997) 1735–1780.

[15] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, arXiv preprint arXiv:1406.1078 (2014).

[16] M. Schuster, K. K. Paliwal, Bidirectional recurrent neural networks, IEEE Transactions on Signal Processing 45 (1997) 2673–2681.

[17] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473 (2014).

[18] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, E. Hovy, Hierarchical attention networks for document classification, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 1480–1489.

[19] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to sequence learning with neural networks, in: Advances in neural information processing systems, 2014, pp. 3104–3112.

[20] M.-T. Luong, H. Pham, C. D. Manning, Effective approaches to attention-based neural machine translation, arXiv preprint arXiv:1508.04025 (2015).

[21] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, T.-S. Chua, Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 5659–5667.

[22] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, Y. Bengio, Attention-based models for speech recognition, in: Advances in neural information processing systems, 2015, pp. 577–585.

[23] M. Lin, Q. Chen, S. Yan, Network in network, arXiv preprint arXiv:1312.4400 (2013).

[24] N. Potha, M. Maragoudakis, Cyberbullying detection using time series modeling, in: 2014 IEEE International Conference on Data Mining Workshop, IEEE, 2014, pp. 373–382.

[25] Y. Mehdad, J. Tetreault, Do characters abuse more than words?, in: Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2016, pp. 299–303.

[26] X. Zhang, J. Tong, N. Vishwamitra, E. Whittaker, J. P. Mazer, R. Kowalski, H. Hu, F. Luo, J. Macbeth, E. Dillon, Cyberbullying detection with a pronunciation based convolutional neural network, in: 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, 2016, pp. 740–745.

[27] P. Badjatiya, S. Gupta, M. Gupta, V. Varma, 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, 2017, pp. 759–760.

[28] L. Gao, R. Huang, Detecting online hate speech using context aware models, arXiv preprint arXiv:1710.07395 (2017).

[29] J. Pavlopoulos, P. Malakasiotis, I. Androutsopoulos, Deep learning for user comment moderation, arXiv preprint arXiv:1705.09993 (2017).

[30] M. Ptaszynski, J. K. K. Eronen, F. Masui, Learning deep on cyberbullying is always better than brute force, in: IJCAI 2017 3rd Workshop on Linguistic and Cognitive Approaches to Dialogue Agents (LaCATODA 2017), Melbourne, Australia, 2017, pp. 3—-10.

[31] J. J. da Silveira Marciano, E. M. A. M. Mendes, M. F. S. Barroso, Cyberbullying classification using extreme learning machine applied to portuguese language, in: Latin American Workshop on Computational Neuroscience, Springer, 2017, pp. 109–117.

[32] N. Vishwamitra, X. Zhang, J. Tong, H. Hu, F. Luo, R. Kowalski, J. Mazer, Mcdefender: Toward effective cyberbullying defense in mobile online social networks, in: Proceedings of the 3rd ACM on International Workshop on Security And Privacy Analytics, ACM, 2017, pp. 37–42.

[33] S. Agrawal, A. Awekar, Deep learning for detecting cyberbullying across multiple social media platforms, in: Advances in Information Retrieval, Springer International Publishing, Cham, 2018, pp. 141–153.

[34] M. A. Al-Ajlan, M. Ykhlef, Deep learning algorithm for cyberbullying detection, International Journal of Advanced Computer Science and Applications 9 (2018) 199–205.

[35] S. T. Aroyehun, A. Gelbukh, Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling, in: Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018), 2018, pp. 90–97.

[36] S.-J. Bu, S.-B. Cho, A hybrid deep learning system of CNN and LRCN to detect cyberbullying from SNS comments, Springer, 2018, pp. 561–572.

[37] J. Chen, S. Yan, K.-C. Wong, Verbal aggression detection on twitter comments: Convolutional neural network for short-text sentiment analysis, Neural Computing and Applications (2018) 1–10.

[38] M. Dadvar, K. Eckert, Cyberbullying Detection in Social Networks Using Deep Learning Based Models; A Reproducibility Study, arXiv preprint arXiv:.08046 (2018).

[39] P. Fortuna, I. Bonavita, S. Nunes, Merging datasets for hate speech classification in italian., in: EVALITA@ CLiC-it, 2018.

[40] A. M. Founta, D. Chatzakou, N. Kourtellis, J. Blackburn, A. Vakali, I. Leontiadis, A unified deep learning architecture for abuse detection,

in: Proceedings of the 10th ACM Conference on Web Science, ACM, 2019, pp. 105–114.

[41] S. V. Georgakopoulos, S. K. Tasoulis, A. G. Vrahatis, V. P. Plagianakos, Convolutional neural networks for toxic comment classification, in: Proceedings of the 10th Hellenic Conference on Artificial Intelligence, ACM, 2018, p. 35.

[42] M. Ibrahim, M. Torki, N. El-Makky, Imbalanced toxic comments classification using data augmentation and deep learning, in: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, 2018, pp. 875–878.

[43] A. Khatua, E. Cambria, A. Khatua, Sounds of silence breakers: Exploring sexual violence on twitter, in: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), IEEE, 2018, pp. 397–400.

[44] R. Kumar, A. K. Ojha, S. Malmasi, M. Zampieri, Benchmarking aggression identification in social media, in: Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018), 2018, pp. 1–11.

[45] G. K. Pitsilis, H. Ramampiaro, H. Langseth, Effective hate-speech detection in Twitter data using recurrent neural networks, Applied Intelligence 48 (2018) 4730–4742.

[46] M. Polignano, P. Basile, Hansel: Italian hate speech detection through ensemble learning and deep neural networks, EVALITA Evaluation of NLP and Speech Tools for Italian 12 (2018) 224.

[47] E. Raisi, B. Huang, Weakly supervised cyberbullying detection using co-trained ensembles of embedding models, in: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), IEEE, 2018, pp. 479–486.

[48] J. Risch, R. Krestel, Aggression identification using deep learning and data augmentation, in: Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018), 2018, pp. 150–158.

[49] H. Rosa, D. Matos, R. Ribeiro, L. Coheur, J. P. Carvalho, A "deeper" look at detecting cyberbullying in social networks, in: 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, 2018, pp. 1–8.

[50] A. Tommasel, J. M. Rodriguez, D. Godoy, Textual aggression detection through deep learning, in: Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018), 2018, pp. 177–187.

[51] B. van Aken, J. Risch, R. Krestel, A. Löser, Challenges for toxic comment classification: An in-depth error analysis, in: Proceedings of the 2nd Workshop on Abusive Language Online (ALW2), Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 33–42.

[52] H. Zhong, D. J. Miller, A. Squicciarini, Flexible inference for cyberbully incident detection, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2018, pp. 356–371.

[53] S. Zimmerman, U. Kruschwitz, C. Fox, Improving hate speech detection with deep learning ensembles, in: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018), 2018.

[54] L. Cheng, R. Guo, Y. Silva, D. Hall, H. Liu, Hierarchical attention networks for cyberbullying detection on the instagram social network, in: Proceedings of the 2019 SIAM International Conference on Data Mining, SIAM, 2019, pp. 235–243.

[55] T. Fagni, L. Nizzoli, M. Petrocchi, M. Tesconi, Six things i hate about you (in italian) and six classification strategies to more and more effectively find them., in: Italian Conference on Cybersecurity ITASEC, 2019.

[56] R. Pandey, H. Purohit, B. Stabile, A. Grant, Distributional semantics approach to detect intent in twitter conversations on sexual assaults, in: 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI), IEEE, 2018, pp. 270–277.

[57] T. Y. S. S. Santosh, K. V. S. Aravind, Hate speech detection in hindi-english code-mixed social media text, in: Proceedings of the ACM

India Joint International Conference on Data Science and Management of Data, ACM, 2019, pp. 310–313.

[58] H. Zhong, H. Li, A. C. Squicciarini, S. M. Rajtmajer, C. Griffin, D. J. Miller, C. Caragea, Content-driven detection of cyberbullying on the instagram social network, in: IJCAI, 2016, pp. 3952–3958.

[59] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (2006) 489–501.

[60] Jigsaw, Toxic comment classification challenge, https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge, 2018. Accessed at August 31, 2018.

[61] Jigsaw, Jigsaw project, https://jigsaw.google.com/, 2018.

[62] Jigsaw, What if technology could help improve conversations online?, https://www.perspectiveapi.com/, 2017. Accessed on August 31, 2018.

[63] W. J. Wilbur, K. Sirotkin, The automatic identification of stop words, Journal of information science 18 (1992) 45–55.

[64] J. D. Ullman, Mining of massive datasets, Cambridge University Press, 2011.

[65] J. Davis, M. Goadrich, The relationship between precision-recall and roc curves, in: Proceedings of the 23rd international conference on Machine learning, ACM, 2006, pp. 233–240.

IRTG 1792 Discussion Paper Series 2019

For a complete list of Discussion Papers published, please visit http://irtg1792.hu-berlin.de.

001 ”Cooling Measures and Housing Wealth: Evidence from Singapore” by Wolfgang Karl Härdle, Rainer Schulz, Taojun Xie, January 2019.

002 ”Information Arrival, News Sentiment, Volatilities and Jumps of Intraday Returns” by Ya Qian, Jun Tu, Wolfgang Karl Härdle, January 2019.

003 ”Estimating low sampling frequency risk measure by high-frequency data” by Niels Wesselhöfft, Wolfgang K. Härdle, January 2019.

004 ”Constrained Kelly portfolios under alpha-stable laws” by Niels Wesselhöfft, Wolfgang K. Härdle, January 2019.

005 ”Usage Continuance in Software-as-a-Service” by Elias Baumann, Jana Kern, Stefan Lessmann, February 2019.

006 ”Adaptive Nonparametric Community Detection” by Larisa Adamyan, Kirill Efimov, Vladimir Spokoiny, February 2019.

007 ”Localizing Multivariate CAViaR” by Yegor Klochkov, Wolfgang K. Härdle, Xiu Xu, March 2019.

008 ”Forex Exchange Rate Forecasting Using Deep Recurrent Neural Networks” by Alexander J. Dautel, Wolfgang K. Härdle, Stefan Lessmann, Hsin-Vonn Seow, March 2019.

009 ”Dynamic Network Perspective of Cryptocurrencies” by Li Guo, Yubo Tao, Wolfgang K. Härdle, April 2019.

010 ”Understanding the Role of Housing in Inequality and Social Mobility” by Yang Tang, Xinwen Ni, April 2019.

011 ”The role of medical expenses in the saving decision of elderly: a life cycle model” by Xinwen Ni, April 2019.

012 ”Voting for Health Insurance Policy: the U.S. versus Europe” by Xinwen Ni, April 2019.

013 ”Inference of Break-Points in High-Dimensional Time Series” by Likai Chen, Weining Wang, Wei Biao Wu, May 2019.

014 ”Forecasting in Blockchain-based Local Energy Markets” by Michael Kostmann, Wolfgang K. Härdle, June 2019.

015 ”Media-expressed tone, Option Characteristics, and Stock Return Predictability” by Cathy Yi-Hsuan Chen, Matthias R. Fengler, Wolfgang K. Härdle, Yanchu Liu, June 2019.

016 ”What makes cryptocurrencies special? Investor sentiment and return predictability during the bubble” by Cathy Yi-Hsuan Chen, Roméo Després, Li Guo, Thomas Renault, June 2019.

017 ”Portmanteau Test and Simultaneous Inference for Serial Covariances” by Han Xiao, Wei Biao Wu, July 2019.

018 ”Phenotypic convergence of cryptocurrencies” by Daniel Traian Pele, Niels Wesselhöfft, Wolfgang K. Härdle, Michalis Kolossiatis, Yannis Yatracos, July 2019.

019 ”Modelling Systemic Risk Using Neural Network Quantile Regression” by Georg Keilbar, Weining Wang, July 2019.

IRTG 1792, Spandauer Strasse 1, D-10178 Berlin
http://irtg1792.hu-berlin.de

This research was supported by the Deutsche Forschungsgemeinschaft through the IRTG 1792.

IRTG 1792 Discussion Paper Series 2019

For a complete list of Discussion Papers published, please visit http://irtg1792.hu-berlin.de.

020 ”Rise of the Machines? Intraday High-Frequency Trading Patterns of Cryptocurrencies” by Alla A. Petukhina, Raphael C. G. Reule, Wolfgang Karl Härdle, July 2019.

021 ”FRM Financial Risk Meter” by Andrija Mihoci, Michael Althof, Cathy Yi-Hsuan Chen, Wolfgang Karl Härdle, July 2019.

022 ”A Machine Learning Approach Towards Startup Success Prediction” by Cemre Ünal, Ioana Ceasu, September 2019.

023 ”Can Deep Learning Predict Risky Retail Investors? A Case Study in Financial Risk Behavior Forecasting” by A. Kolesnikova, Y. Yang, S. Lessmann, T. Ma, M.-C. Sung, J.E.V. Johnson, September 2019.

024 ”Risk of Bitcoin Market: Volatility, Jumps, and Forecasts” by Junjie Hu, Weiyu Kuo, Wolfgang Karl Härdle, October 2019.

025 ”SONIC: SOcial Network with Influencers and Communities” by Cathy Yi-Hsuan Chen, Wolfgang Karl Härdle, Yegor Klochkov, October 2019.

026 ”Affordable Uplift: Supervised Randomization in Controlled Exprtiments” by Johannes Haupt, Daniel Jacob, Robin M. Gubela, Stefan Lessmann, October 2019.

027 ”VCRIX - a volatility index for crypto-currencies” by Alisa Kim, Simon Trimborn, Wolfgang Karl Härdle, November 2019.

028 ”Group Average Treatment Effects for Observational Studies” by Daniel Jacob, Wolfgang Karl Härdle, Stefan Lessmann, November 2019.

029 ”Antisocial Online Behavior Detection Using Deep Learning” by Elizaveta Zinovyeva, Wolfgang Karl Härdle, Stefan Lessmann, November 2019.

IRTG 1792, Spandauer Strasse 1, D-10178 Berlin
http://irtg1792.hu-berlin.de

This research was supported by the Deutsche Forschungsgemeinschaft through the IRTG 1792.