Convolutional Neural Networks for Twitter Text Toxicity Analysis

Spiros V. Georgakopoulos, Sotiris K. Tasoulis, Aristidis G. Vrahatis, and Vassilis P. Plagianakos(✉)

Department of Computer Science and Biomedical Informatics, University of Thessaly, Lamia, Greece
{spirosgeorg,stasoulis,arisvrahatis,vpp}@uth.gr

Abstract. Toxic comment classification is an emerging research field with several studies that have address several tasks in the detection of unwanted messages on communication platforms. Although sentiment analysis is an accurate approach for observing the crowd behavior, it is incapable of discovering other types of information in text, such as toxicity, which can usually reveal hidden information. Towards this direction, a model for temporal tracking of comments toxicity is proposed using tweets related to the hashtag under study. More specifically, a classifier is trained for toxic comments prediction using a Convolutional Neural Network model. Next, given a hashtag all relevant tweets are parsed and used as input in the classifier, hence, the knowledge about toxic texts is transferred to a new dataset for categorization. In the meantime, an adapted change detection approach is applied for monitoring the toxicity trend changes over time within the hashtag tweets. Our experimental results showed that toxic comment classification on twitter conversations can reveal significant knowledge and changes in the toxicity are accurately identified over time.

Keywords: Convolutional neural networks · Toxic comments · Twitter conversations · Change detection

1 Introduction

Twitter sentiment analysis has covered a plethora of needs related to product opinion [1], stock market movement [22,24] and political influence on crowed [11,18,25]. Twitter has already shown its impact on politics [4,30], especially on the electoral body constituting it as an important tool for the popularity of a politician. As a sequence, twitter has been launched as the common communication platform for the politicians in the last decade [6]. Thus, an imperative need is created for more accurate text mining and Machine Learning methods towards twitter analysis.

The core of text mining and Machine Learning methods for text analysis is based on the detection of the message substance through the words (or words combination) that provide this information. The tendency of that field is the

L. Oneto et al. (Eds.): INNSBDDL 2019, INNS 1, pp. 370–379, 2020.
https://doi.org/10.1007/978-3-030-16841-4_38

sentiment analysis of text in order to analyze the message and to identify voters and crowd polarity. To achieve that, sentiment analysis tools based either on machine learning or lexical approaches are used. These tools estimate the sentiment content of the text [29]. The estimation is related to word count (binary - positive/negative, trinary - positive/neutral/negative, etc.) approaches using vocabularies with polarity influence.

Although the literature has been overwhelmed by such tools, there is a growing interest for online conversation analysis under different perspectives that may seem more appropriate for individual tasks. In this work, we focus on unveiling twitter comment toxicity based on a recent dataset provided within a Kaggle competition[1]. A toxic message is not only the regular scurrility, but also an aggressive message [33] or a message that occurs a personal attacks (such as treat and insult message) [32,34]. These types of behavior when they are appearing sequentially in time and not as outline message in the period of time (assuming that a regular voter do not present a such behaviour) may be malicious (bot, spam, etc.). The malicious messages may be part of politician rivals and have to be removed from the time series analysis of the text in order to extract accurate results of the voters behavior.

An indicative relative example is the project created by Google and Jigsaw, called Perspective, which uses machine learning to automatically detect toxic language [13]. More recently, toxic comment classification tools have been proposed using convolutional neural networks [9], recurrent neural networks [19] and deep learning approaches [26], but none of these have been applied on twitter data so far. Given the fact that twitter analysis has a major social impact and toxic comment classification is an emerging field for online conversations, identifying toxicity trend in twitter threads appears to be quite interesting. However, this task imposes two challenges that should be addressed. Characterization of twitter posts can be a hard task due to the short length of the documents, while the continuous stream of data necessitates the use of online methods for change detection. In this work, we propose a complete methodology that provides answers to both of these challenges.

2 Methodology

The proposed methodology is based on the construction of a classifier specifically trained for toxic comment classification, employing a recent and quite widespread labeled toxic comment dataset. The classifier is designed through a Convolutional Neural Network (CNN) model using the word2vec method [20] for mapping a physical text to a low-dimension representation. The gained knowledge of the classifier is exploited to classify a selection of twitter posts originated from a particular hashtag under study. The tweets are parsed online and a real time method for detecting trend changes is applied.

[1] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge.

Our methodology consists of three main steps, (i) design and implementation of CNN model for toxic comment classification, (ii) online parsing and characterization of tweets relevant to a pre-defined hashtag and (iii) real-time change detection analysis in the toxicity trend using an adapted Cumulative Sum (CUSUM) algorithm. A flowchart diagram of the methodology is presented in Fig. 1.

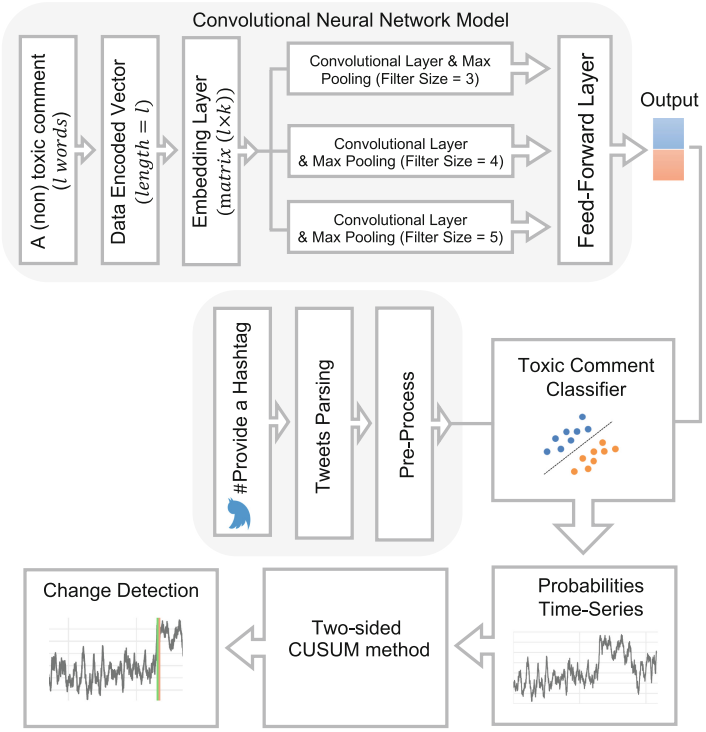

Fig. 1. Overview of the proposed methodology

2.1 CNN Model for Toxic Comments Classification

CNN models are appropriate for image classification, where the pixels of the images are represented by integer values within a specific range of values. On the other hand, the components of a sentence (the words) have to be encoded before being fed to the CNN [5]. To overcome this limitation, a vocabulary is applied as an index containing the words that appear in the set of document texts, mapping each word to an integer in $[0, 1]$. However, we have to deal also the variability in documents length (number of words in a document), since CNNs require a constant input dimensionality. To address it, we adopted the padding technique filling with zeros the document matrix to reach the maximum length amongst all documents in dimensionality.

In the next step, the encoded documents are transformed into matrices and each row corresponds to one word. The generated matrices pass through the embedding layer, where each word (row) is transformed into a low-dimension representation by a dense vector [7]. The procedure continues following the standard CNN methodology. In this work, we employ the fixed dense vectors for words, word2vec [20]. The word2vec embedding method has been trained on 100 billion words from Google News, producing a vocabulary of 3 million words. The embedding layer matches the input words with the fixed dense vector of the pre-trained embedding methods that have been selected. The values of these vectors do not change during the training process, unless there are words not already included in the vocabulary of the embedding method in which case they are randomly initialized.

The CNN model at hand is based on a selection of documents retrieved from Wikipedia by the Kaggle competition. The provided class labels were originally defined across different levels or types of toxicity for this study we consider the binary classification problem (document samples are characterized as toxic or non-toxic).

2.2 Tweets Streaming and Toxicity-Based Classification

Following the aforementioned training procedure, the toxic comment classifier is able to classify any set of tweets as toxic or non toxic. In our methodology, given a user-defined hashtag, all relevant tweets are streamed in real time using the Twitter stream Application Program Interfaces (API) proposed by Kalucki [14]. The streaming documents are consecutively cleaned removing noise, such as hashtags, spaces, numbers, punctuations, URLs etc. Following the procedure described in [28], before passed through the trained classifier, from which they receive a toxicity probability value. This is the value returned by the neuron of the output layer, that corresponds to the trained neuron on toxic comments indication. Subsequently, a time series of probability scores is generated, bounded by the softmax function performed on the output layer.

2.3 Toxicity Change Detection

To identify significant changes upon toxicity level, we utilize an online version of the Cumulative Sum (CUSUM) algorithm, focusing on a technique connected to a simple integration of signals with adaptive threshold [2,8,27]. To describe the change detection algorithm, we consider a sequence of independent random variables y_k, where y_k is a signal at the current time instant k (discrete time), with a probability density $p_\theta(y)$ depending only upon one scalar parameter θ. Before the unknown change time t, the parameter θ is equal to θ_0, and after the change it is equal to $\theta_1 \neq \theta_0$. Thus, the problem is to detect and estimate this parameter change. In this work, our goal is to detect the change assuming that the parameters θ_0 and θ_1 are known, which is a quite unrealistic assumption for many practical applications.

Usually parameters θ_0 and θ_1 can be experimentally estimated using test data, which we also do not consider available for the application at hand. However, for similar strongly bounded problems (signal values correspond to probabilities) parameter values can be empirically assumed. For example, we may consider θ_0 the state of toxicity level at the beginning of monitoring, which we may get by averaging over the first few samples. To this end, we are not interested in characterizing the current state (i.e. toxic, non toxic or neutral), but only in discovering negative or positive toxicity changes. Thus we only need to define the *the change magnitude*, which we arbitrary set to 0.3, taking into account that signal values lie between 0 and 1.

Since it is necessary to detect changes in each direction, discovering both increasing and decreasing toxicity of twitter posts, two one-sided algorithms were used for which $\theta_1^{pos} = \theta_0 + 0.3$ and $\theta_1^{neg} = \theta_0 - 0.3$, respectively. When a change is triggered for any of the two-sided functions at time t, the CUSUM algorithm is reset to zero and a re-initialization takes place. The algorithm restarts with a new value for θ_0 equal to the average of the few last observations ($[t - n \rightarrow t]$) and the new control state at time t is defined, while θ_1 is recalculated based on a fixed *change magnitude*.

To detect a change, as the samples (tweets) are arriving at each time instant, a decision rule is computed and compared against an adaptive threshold. This is the detection threshold h, a user-defined tuning parameter based on the average run length function that is defined as the expected number of samples before an action is taken [21]. More precisely, one has to set the mean time between false alarms ARL_0 and the mean detection delay ARL_1. These two specific values of the ARL function depend on the detection threshold h and can thus be used to set the performance of the CUSUM algorithm to the desired level for each particular application [10].

Under this perspective, we detect behavior changes under a dynamic estimation. This is crucial since a hashtag's toxicity could change several times across its lifetime and changes should be estimated based on its current state rather than its initial state. Towards this direction, our methodological framework can imprint on-line the change detections of a hashtag by updating its initial state.

3 Data Retrieval and Cleaning

The described methodological framework was applied on a Twitter data stream collected from 15-03-2018 to 24-03-2018 using the hashtag 'theresamay'. Obviously, this term refers to Theresa Mary May, Prime Minister of the United Kingdom and Leader of the Conservative Party since 2016. Our choice lies in the fact that politic-related Twitter hashtags offer a satisfying opportunity for testing toxicity changes. In addition, this hashtag was selected considering that during this period the 'Brexit' news topic attracted attention due to further discussions amongst high-level politicians regarding the relationships between the United Kingdom and European Union.

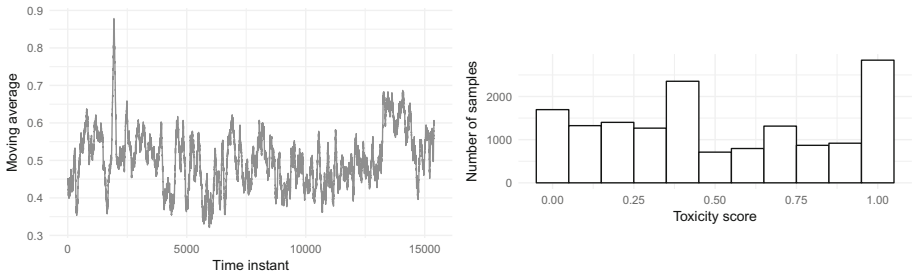

Fig. 2. The calculated moving average of window size 200 (left). Histogram of the toxicity score for the whole dataset (right).

Tweets are collected by the Twitter stream API, which is free to use and only requires a valid Twitter account for authentication. Withing the R-project, we used the "rtweet" package [15] to connect to the API and stream tweets filtered by keywords, such as Twitter hashtags (in our case 'theresamay'). Data are retrieved in JSON format and are parsed using the "rtweet" package. Then, the data cleaning needs to take place removing hashtags, spaces, numbers, punctuations, URLs etc. To achieve this, we employed functions from the "stringr" and "glue" packages respectively [12,31]. 15491 tweets have been streamed in total after discarding non-English language posts. Amongst them there are 11872 retweets (76%), which are not specially treated given that they still express an opinion. Then, generating a text corpus we may investigate most common words based on the Term Frequency-Inverse Document Frequency (TF-IDF) methodology [23], where the frequency of words is rescaled by how often they appear, penalizing most frequent words. The resulting words in descending order are: britain, look, russia, threat, strong, power.

4 Experimental Results

The first part of the proposed methodology is the training of a CNN model with the toxic comment dataset of Kaggle competition. The architecture that is used consist of three different convolutional layers simultaneously, with filter size width 128 and dense vector dimension 300. The width of the filters is equal to the vector dimension, while their height was 3, 4 and 5, for each convolutional layer, respectively. After each convolutional layer, a max-over-time pooling operation is applied. The output of the pooling layers is concatenated to a fully-connected layer, while the softmax function is applied on the final layer. The model is trained for 100 epochs using the Stochastic Gradient Descent algorithm [3] with mini-batches of 64 inputs and learning rate 0.005. The model achieved a classification accuracy 91.2%, when applied on unlabeled data from the same source [9].

In an attempt to visually inspect the toxicity trend changes, the corresponding moving average of the toxicity prediction probability scores retrieved by the CNN classifier is employed. Figure 2 (left) illustrates the calculated moving average for a window of total size 200. Subsequently, we investigate the histogram of tweet's distribution according to their toxicity level (see Fig. 2 (right)).

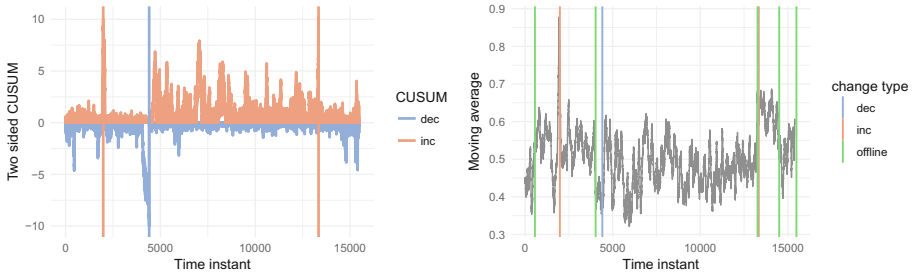

Fig. 3. The two sided CUSUM function along with the reported change points annotated by vertical lines corresponding to changes (positive and negative, respectively) (left). The reported change points with respect to the calculated moving average, where the additional green lines correspond to the change points reported by the off-line algorithm (right).

In what follows, we apply the change detection algorithm to the retrieved time series of probability scores and report the results. For the initialization of parameters values, we set $\theta_0 = 0.5$, while the *change magnitude* is set to 0.3; thus, we consider $\theta_1^{inc} = 0.8$ and $\theta_1^{dec} = 0.2$ for the two-sided CUSUM, respectively. In Fig. 3 (left) we show the CUSUM functions and the corresponding changes retrieved by the algorithm with vertical lines. Notice that each time a change is detected the algorithm restarts with an updated initialization. Selecting the value of the h parameter is usually subject to question and depends on the user requirements. For this particular topic of study in our analysis, we assume that a small number of reported changes per day are sufficient and thus we define $h = 10$. To visually investigate the reported change points we employed the calculated moving average presented in Fig. 2 (left). Vertical lines are depicted in the plot (see Fig. 3 (right)) with the corresponding colors from Fig. 3 (left). At this point, we may conclude that reported change points agree with a simple visual investigation of possible changes. To further verify this outcome, we employ a well established off-line methodology for change detection, which is capable to discover multiple change points [16,17] and estimate their number in the time series automatically if required. The penalty parameter of this algorithm is set to $1/2 * \log(n)$, where n is the total number of samples. The retrieved change points are also reported in Fig. 3 (right) with green vertical lines. We observe that this result accurately matches the change points retrieved by the adaptive CUSUM scheme, considering that the difference between reported changes mainly concerns the delay enforced by the online algorithm.

Fig. 4. Word-clouds of tweets belonging to different categories.

In the last part of our experimental analysis, we investigate the content of tweets according to the characterization by the classifier. For this purpose, we report word-clouds plotting the frequency of words appearing in the corresponding collection of Twitter posts. Again, the text is being preprocessed using the Term Frequency-Inverse Document Frequency (TF-IDF) (see Fig. 4). It appears that there exists a clear discrimination between the word-clouds. We observe that words like *coward* and *pathetic* appear in the toxic class validating the initial hypothesis. At this point we may also investigate example tweets from the two different categories that has been classified with strong confidence (see Fig. 5). The tweet on the right part of the Figure has been characterized as toxic with strong confidence and apparently is an ironic comment using also scurrility words that are not amongst the most common ones.

Remember when the *Brexiteers* made up the jokey lie about Cameron saying EURef result could lead to WWIII How we laughed Putin Trump Russia Brexit TheresaMay SalisburyAttack

You can thank you leaders and especially TheresaMay for the sad sorry state of your dingy *pathetic* little country or commonwealth or EuropeanUnion vassal state or whatever you are at this point *Coward* TheresaMay TheresaTheAppeaser

Fig. 5. A regular non-toxic tweet (left) and a toxic one (right) that contain some of the most frequent words from the two categories (annotated with red color).

5 Conclusion

Toxic comment classification is an emerging research field and although there are several approaches under this perspective, it remains at its infancy. Twitter's extensive data availability provides great potential for the development of Machine Learning research in social conversations. In this work, we study the comments toxicity under the perspective of a time series analysis to discover significant changes in real time. We may argue that this process increases the value of social media data by widening the understanding of community reaction in

certain circumstances supporting immediate decision making. The experimental analysis provided pieces of evidence that toxicity identification may unravel significant knowledge, which is hidden from other sentiment-based approaches.

Acknowledgment. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research. This project has received funding from the Hellenic Foundation for Research and Innovation (HFRI) and the General Secretariat for Research and Technology (GSRT), under grant agreement No 1901.

References

1. Anastasia, S., Budi, I.: Twitter sentiment analysis of online transportation service providers. In: 2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS), pp. 359–365, October 2016
2. Basseville, M., Nikiforov, I.V.: Detection of abrupt changes: theory and application (1993)
3. Bottou, L.: On-line learning and stochastic approximations. In: On-Line Learning in Neural Networks, pp. 9–42. Cambridge University Press, New York (1998). http://dl.acm.org/citation.cfm?id=304710.304720
4. Burgess, J., Bruns, A.: (Not) the Twitter election: the dynamics of the# ausvotes conversation in relation to the Australian media ecology. Journal. Pract. **6**(3), 384–402 (2012)
5. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. J. Mach. Learn. Res. **12**, 2493–2537 (2011)
6. Enli, G.S., Skogerbø, E.: Personalized campaigns in party-centred politics: Twitter and Facebook as arenas for political communication. Inf. Commun. Soc. **16**(5), 757–774 (2013)
7. Gal, Y., Ghahramani, Z.: A theoretically grounded application of dropout in recurrent neural networks. In: Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, Barcelona, Spain, 5–10 December 2016, pp. 1019–1027 (2016)
8. Georgakopoulos, S.V., Tasoulis, S.K., Plagianakos, V.P.: Efficient change detection for high dimensional data streams. In: 2015 IEEE International Conference on Big Data (Big Data), pp. 2219–2222, October 2015
9. Georgakopoulos, S.V., Tasoulis, S.K., Vrahatis, A.G., Plagianakos, V.P.: Convolutional neural networks for toxic comment classification. CoRR abs/1802.09957 (2018). http://arxiv.org/abs/1802.09957
10. Granjon, P.: The CUSUM algorithm a small review (2014)
11. Haselmayer, M., Jenny, M.: Sentiment analysis of political communication: combining a dictionary approach with crowdcoding. Qual. Quant. **51**(6), 2623–2646 (2017)
12. Hester, J.: glue: Interpreted String Literals (2017). https://CRAN.R-project.org/package=glue, r package version 1.2.0
13. Hosseini, H., Kannan, S., Zhang, B., Poovendran, R.: Deceiving Google's perspective API built for detecting toxic comments. arXiv preprint arXiv:1702.08138 (2017)

14. Kalucki, J.: Twitter streaming API (2010). http://apiwiki.twitter.com/Streaming-API-Documentation
15. Kearney, M.W.: rtweet: Collecting Twitter Data (2017). R package version 0.6.0
16. Killick, R., Fearnhead, P., Eckley, I.: Optimal detection of changepoints with a linear computational cost **107**, 1590–1598 (2012)
17. Killick, R., Haynes, K., Eckley, I.A.: changepoint: an R package for changepoint analysis (2016). https://CRAN.R-project.org/package=changepoint. R package version 2.2.2
18. Kušen, E., Strembeck, M.: Politics, sentiments, and misinformation: an analysis of the Twitter discussion on the 2016 Austrian presidential elections. Online Soc. Netw. Media **5**, 37–50 (2018)
19. Li, S.: Application of recurrent neural networks in toxic comment classification. Ph.D. thesis, UCLA (2018)
20. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Neural and Information Processing System (NIPS) (2013)
21. Page, E.S.: Continuous inspection schemes. Biometrika **41**(1/2), 100–115 (1954)
22. Pagolu, V.S., Reddy, K.N., Panda, G., Majhi, B.: Sentiment analysis of Twitter data for predicting stock market movements. In: 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), pp. 1345–1350, October 2016
23. Rajaraman, A., Ullman, J.D.: Mining of Massive Datasets. Cambridge University Press, Cambridge (2011)
24. Ranco, G., Aleksovski, D., Caldarelli, G., Grčar, M., Mozetič, I.: The effects of Twitter sentiment on stock price returns. PLoS One **10**(9), e0138441 (2015)
25. Ringsquandl, M., Petkovic, D.: Analyzing political sentiment on Twitter. In: AAAI Spring Symposium: Analyzing Microtext. AAAI Technical report, vol. SS-13-01. AAAI (2013)
26. Risch, J., Krestel, R.: Aggression identification using deep learning and data augmentation. In: Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC 2018), pp. 150–158 (2018)
27. Tasoulis, S., Doukas, C., Plagianakos, V., Maglogiannis, I.: Statistical data mining of streaming motion data for activity and fall recognition in assistive environments. Neurocomputing **107**, 87–96 (2013). Timely Neural Networks Applications in Engineering
28. Tasoulis, S.K., Vrahatis, A.G., Georgakopoulos, S.V., Plagianakos, V.P.: Real time sentiment change detection of Twitter data streams. CoRR abs/1804.00482 (2018)
29. Thelwall, M.: The heart and soul of the web? Sentiment strength detection in the social web with sentistrength, pp. 119–134. Springer, Cham (2017)
30. Wang, H., Can, D., Kazemzadeh, A., Bar, F., Narayanan, S.: A system for real-time Twitter sentiment analysis of 2012 US presidential election cycle. In: Proceedings of the ACL 2012 System Demonstrations, pp. 115–120. Association for Computational Linguistics (2012)
31. Wickham, H.: stringr: Simple, Consistent Wrappers for Common String Operations (2017). https://CRAN.R-project.org/package=stringr. R package version 1.2.0
32. Wulczyn, E., Thain, N., Dixon, L.: Ex machina: personal attacks seen at scale. In: Proceedings of the 26th International Conference on World Wide Web, WWW 2017, pp. 1391–1399. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva (2017)
33. Wulczyn, E., Thain, N., Dixon, L.: Wikipedia talk labels: aggression (2017)
34. Wulczyn, E., Thain, N., Dixon, L.: Wikipedia talk labels: personal attacks (2017)