RESCIENCE C

Replication / ML Reproducibility Challenge 2021

[¬Re] Hate Speech Detection based on Sentiment Knowledge Sharing

Matteo Brivio[1] and Çağrı Çöltekin[1]

[1]University of Tübingen, Tübingen, Germany

Edited by
Koustuv Sinha,
Sharath Chandra Raparthy

Reviewed by
Anonymous Reviewers

Received
04 February 2022

Published
23 May 2022

DOI
10.5281/zenodo.6574639

1 Reproducibility Summary

This report summarizes our efforts to reproduce the results presented in the ACL2021 paper Hate Speech Detection based on Sentiment Knowledge Sharing by Zhou et al. [1], as part of the ML Reproducibility Challenge 2021. We attempt to verify the main claims of the original study by reproducing the experiments comparing models with and without sentiment knowledge sharing. Although most scores in our replication study matches with the ones reported in the original paper, our experiments result in substantially lower scores for the full model with sentiment sharing. We also investigate variation in the scores, report additional scores (more suitable for the task), and discuss possible sources for the discrepancies observed.

1.1 Scope of Reproducibility

The main goal of this reproducibility attempt is to confirm the effectiveness of the hate speech detection framework proposed by Zhou et al. [1]. In particular, our efforts are directed at validating their main claim that sentiment knowledge sharing in a multi-task learning setup improves the performance of the model in predicting hate speech. Besides reproducing their main results, we perform repeated experiments to assess the variability of the scores and carry out a hyperparameter search.

1.2 Methodology

The authors provide a code-base which is available at https://github.com/1783696285/SKS. We reuse the available code, modifying it where necessary and integrating it with a few additional scripts for statistics computation and data preparation. Our code, data and results are available at https://github.com/matteobrv/repro-SKS.

1.3 Results

Our findings diverge substantially from the results reported in the original paper. In particular, in our reproduction experiments, including sentiment features appears to hurt the performance of the model in the hate speech detection task (approximately 0.5 to 2.0 F1-score) in the setting we could reproduce based on the description in the original paper and published source code (and limited contact with the authors, see Section 1.6).

Correspondence should be addressed to Çağrı Çöltekin (ccoltekin@sfs.uni-tuebingen.de)
The authors have declared that no competing interests exist.
Code is available at https://github.com/matteobrv/repro-SKS – DOI 10.5281/zenodo.6502870. – SWH swh:1:dir:d61b47330cc5c92d7ac4873269faa38a2e3c20bd.
Open peer review is available at https://openreview.net/forum?id=SSSGs3M7nRY.

Repeating the experiments with the different random initializations does not provide potential explanations for the differences.

1.4 What was easy

The paper provides some broad indications with respect to the training details and the code-base is publicly available. Similarly, the data-sets are also freely available and the authors provide links to them in their repository.

1.5 What was difficult

Like most 'research code', the code-base is rather convoluted. Following the instructions included in the authors' repository resulted in a number of exceptions caused by formatting issues, missing code snippets and hard-coded values. The lack of a clear and comprehensive documentation also contributed to an arduous code review and reproducibility effort.

1.6 Communication with original authors

We were able to resolve some of the problems with running the original code (e.g., missing code snippets) by contacting the authors through their public repository. Unfortunately, however, not all of our questions were answered, nor the issues were fixed (in a timely manner) and we had to resort to 'reasonable defaults' for some of the unspecified or unclear aspects of the original experiments. We also did not receive responses to our questions via emails sent to the email addresses provided on the paper.

2 Introduction

Being able to quickly and reliably detect hate speech in an automatic manner is an important task. Due to the growing number of regulations concerning the use of hate speech and other forms of offensive language online this topic has gained increasing interest, both in academia and industry [2, 3, 4, 5, 6].
As in any supervised learning task, the availability and the size of labeled data-sets pose significant challenges. The task is made even more arduous by its multilingual and multi-domain nature. One way to alleviate such problems is to make use of additional data-sets from potentially related tasks.
The study by Zhou et al. that we attempt to reproduce describes a multi-task learning framework for online hate speech detection that relies on the purportedly strong negative sentiment characterizing this threatening form of communication. The model presented in the original paper, Sentiment Knowledge Sharing (SKS), is a multi-head attention network that predicts whether the input text contains hate speech or not. The main claim of the paper revolves around the fact that the model is (optionally) trained in a multi-task setting also for sentiment analysis, and it can incorporate information from a dictionary of derogatory words through 'category embeddings' (see Section 4.1 for further details).
Based on experiments carried out on two benchmark data-sets, the original study claims that training a model relying both on sentiment information and category embeddings allows to boost its performance in the task of hate speech detection.

3 Scope of reproducibility

The work of Zhou et al. [1] is based on the intuition that hate speech detection and sentiment analysis are two highly correlated tasks and that hate speech is likely to arise from derogatory words. Our reproduction attempt aims to verify the following claims:

- A model relying both on Sentiment Knowledge Sharing (SKS) and a dictionary of derogatory words scores better than several strong baselines where sentiment features are not considered.
- Ablating the sentiment knowledge component (`-s`) results in a poorer performance, as the model relies solely on derogatory words features which, despite being likely indicators of hate speech, can make the model too sensitive too false positives (e.g., *I'm so fucking ready!*).
- A model where both sentiment knowledge and derogatory word features are ablated (`-sc`) scores the worst performance.

Besides trying to reproduce the original results (see Table 3 in [1]), we carry out a hyperparameter search to validate the values reported in the original paper. Since variation due to model initialization can be an important factor for irreproducibility [7, 8, 9], we run all experiments multiple times to check whether any observed differences stand when score variation is taken into consideration.

4 Methodology

4.1 Model description

The SKS model relies heavily both on the Mixture-of-Experts (MoE) approach as introduced by Shazeer et al. [10] and the Multi-gate Mixture-of-Experts (MMoE) presented by Ma et al. [11]. Its overall architecture consists mainly of three macro-components: an input layer, a sentiment knowledge sharing layer and a gated attention layer.

The input layer — In the input layer, word embeddings are used to encode words of each target sentence. Specifically, every token w_i of a given sentence $S = \{w_1, w_2, ..., w_i, ..., w_N\}$ is transformed into a real-valued vector $x_i \in \mathbb{R}^d$. Additionally, given that derogatory words represent a helpful marker of hate speech, each vector x_i is concatenated with a category embedding vector $c_i \in \mathbb{R}^{d^i}$, such that $x_i^{'} = x_i \oplus c_i$.
Category embeddings are created on the basis of a dictionary of derogatory words which allows to classify sentences into two categories, either containing derogatory words or not. The result of the classification is encoded as a vector c_i and appended to each word embedding x_i, such that the encoded sentence is $S^{'} = \{x_1^{'}, x_2^{'}, ..., x_i^{'}, ..., x_N^{'}\}$.

The sentiment knowledge sharing layer — The sentiment knowledge sharing component relies on a multi-task learning strategy which, according to the authors, would allow to take advantage of the high correlation between the two tasks of sentiment analysis and hate speech detection. In the proposed implementation, the two tasks share a bottom hidden layer based on the Mixture-of-Experts (MoE) approach. This layer is made up of multiple identical feature extraction units (Experts) each of which, in turn, is composed of a multi-head attention layer using 4 heads and two feed forward neural networks.
Each unit relies on the idea of multi-head attention introduced by Vaswani et al. [12], where the input matrix X is mapped to query $\mathrm{Q} \in \mathbb{R}^{(n_1 \times d_1)}$, key $\mathrm{K} \in \mathbb{R}^{(n_1 \times d_1)}$, and value $\mathrm{V} \in \mathbb{R}^{(n_1 \times d_1)}$ using linear transformations. Given these three matrices the attention parameters are computed as follows:

$$\mathrm{Attention}(\mathrm{Q}, \mathrm{K}, \mathrm{V}) = \mathrm{softmax}\left(\frac{\mathrm{QK}^\top}{d_1}\right)\mathrm{V}. \tag{1}$$

In the implementation proposed by Zhou et al. $\mathrm{K} = \mathrm{V}$ and d_1 corresponds to the number of hidden layer units. The ith output of the multi-head attention mechanism is:

$$\mathrm{M}_i = \mathrm{Attention}(\mathrm{QW}_i^Q, \mathrm{KW}_i^K, \mathrm{VW}_i^V), \tag{2}$$

where the parameter matrices are $\mathrm{W}_i^Q \in \mathbb{R}^{n_1 \times \frac{d_1}{l}}$, $\mathrm{W}_i^K \in \mathbb{R}^{n_1 \times \frac{d_1}{l}}$ and $\mathrm{W}_i^V \in \mathbb{R}^{n_1 \times \frac{d_1}{l}}$. All outputs are then concatenated and multiplied by W^O to get the final feature representation $\mathrm{H}^s = \mathrm{concat}(\mathrm{M}_1, \mathrm{M}_2, ..., \mathrm{M}_i, ..., \mathrm{M}_l)\mathrm{W}^O$.
Finally, the authors decide to use both maximum and average pooling [13] to fuse the feature representations, concatenating the two results:

$$\mathrm{P}_m = \mathrm{Pooling\_max}(\mathrm{H}^s), \tag{3}$$

$$\mathrm{P}_a = \mathrm{Pooling\_average}(\mathrm{H}^s), \tag{4}$$

$$\mathrm{P}_s = \mathrm{concat}(\mathrm{P}_m, \mathrm{P}_a). \tag{5}$$

The gated attention layer — The third macro-component is a gated attention mechanism which allows to select a subset of the feature extraction units from the previous layer. The output $g^k(x)$ of a specific gate k corresponds to the probability of selecting a specific unit. The subset of units selected through this process are then weighted and summed to get the final representation $f^k(x)$ for a given sentence, which is then passed to a feed-forward neural network to detect hate speech:

$$g^k(x) = \mathrm{softmax}(\mathrm{W}_{gn} * \mathrm{gate}(x)), \tag{6}$$

$$f^k(x) = \sum_{i=1}^{n} g^k(x)_i f_i(x), \tag{7}$$

$$y_k = h^k f^k(x). \tag{8}$$

4.2 Datasets

Following in the footsteps of Zhou et al. [1], we test the model and report results on two public hate speech data-sets: SemEval2019 Task-5 (SE) [4][1] and Davidson (DV) [2].[2] The former is freely available upon request, while the latter is openly distributed under an MIT License.

The SE data-set contains a total of $13,000$ tweets and is divided into training-, validation- and test-set, consisting of $9,000$, $1,000$ and $3,000$ samples, respectively. The training-set contains $3,783$ instances of hate speech and $5,217$ instances that are not. In the validation-set 427 samples are classified as hate speech and 573 as non-hate speech. The test-set is split into $1,260$ hate speech samples and $1,740$ non-hate speech ones.

The DV data-set contains a total of $24,783$ manually labeled tweets. Each tweet is assigned to either one of three classes: hate speech ($1,430$), offensive language ($19,190$) or neither ($4,163$). Zhou et al. merge the last two classes together and obtain $1,430$ tweets classified as hate speech and $23,353$ classified as non-hate speech.

Finally, the model relies also on a sentiment data-set freely available on Kaggle[3] under no specific license. Following the original study, we only use the training-set which contains $31,962$ tweets, $2,242$ of which are classified as having a negative sentiment, while the remaining $29,720$ a positive one.

4.3 Hyperparameters

We begin our reproducibility attempt, relying solely on the hyperparameters reported in the original paper. Our results are summarized in Table 1.

In the input layer, all word vectors are initialized using Glove Common Crawl Embeddings (840B Token) [14] with a dimension of 300, while category embeddings are randomly initialized and have a dimension of 100.

In the sentiment knowledge sharing layer, the multi-head attention mechanism is implemented using 4 heads. The two feed-forward networks in each expert unit have one layer with 400 units and two layers with 150 units, respectively. However, contrary to what we see in the implementation, it is worth noting that the original paper reports 200 units for the second network. After each layer a dropout rate of 0.1 is used.

The model is trained by mini-batches of 512 instances for 15 epochs, using the `RMSprop` optimizer and a learning rate of 0.001. The original study reports the use of learning rate decay and early stopping to avoid overfitting.

Hyperparameters tuning — The original work does not provide any details regarding hyperparameters tuning and upon contacting the authors to inquire about it we received no answer. We tune learning rate (10^{-6} to 10^{-1}, on a log scale), batch size (from 32 to 1024, on a $\log_2$ scale) and dropout rate (0.0 to 0.4 with increments of 0.1) on the SE data-set using grid-search with 60 epochs and find that the respective optimal values are 0.001, 256 and 0.0.

Despite discrepancies with the original values the model's performance remains similar. In this respect, considering the model variation (see Table 1 and Figure 1), any differences are likely due to random initialization.

[1] http://hatespeech.di.unito.it/hateval.html

[2] https://github.com/t-davidson/hate-speech-and-offensive-language/tree/master/data

[3] https://www.kaggle.com/dv1453/twitter-sentiment-analysis-analytics-vidya

4.4 Experimental setup and code

We try to reproduce the results presented in Table 3 of the original paper [1]. For both data-sets the authors train three models: SKS, which relies both on sentiment knowledge sharing and category embeddings; `-s`, a model where the sentiment knowledge sharing component is ablated; `-sc`, a model that does without both sentiment knowledge sharing and category embeddings. We rely largely on the TensorFlow [15] implementation made available by the authors, modifying it where necessary and integrating it with a few additional scripts for statistics computation and data preparation.
For each result reported in the original paper we repeat the corresponding experiment 10 times. Specifically, for each repetition the model is reinitialized and trained over 15 epochs. We keep the results from the best epoch of each repetition and then compute the average and the standard deviation for the originally employed measures i.e., accuracy and macro-F1 score for the SE data-set and accuracy and weighted-F1 score for the DV data-set.
Given that the DV data-set is highly unbalanced, the original study use a 5-fold cross-validation approach to measure the performance of each model. We follow in their footsteps and adopt the 10 times repetition strategy for each 5-fold experiment.
Our code, data as well as the final and intermediate per-iteration results are available at https://github.com/matteobrv/repro-SKS.

4.5 Computational requirements

We run all our experiments on an NVIDIA TITAN Xp with a 12 GB memory. Training the models on the SE data-set took approximately 24 minutes for the SKS model and 7 minutes both for the `-sc` and `-s` model. On the DV data-set the training took approximately 3 hours for the SKS model and 2 hours both for the `-sc` and `-s` model. The hyperparameters tuning step on the SE data-set took approximately 33 hours.

5 Results

In Table 1 we summarise the original results along the ones we obtained using the specified hyperparameters. Comparing our findings with those reported by the original study we observe a discrepancy in all three measures, accuracy, macro-F1 and weighted-F1 score, for both data-sets. In the SE data-set, the most notable differences concern the results of the SKS and `-s` models. In the DV data-set, there are some noteworthy discrepancies only with respect to the SKS model.
Looking at the mean scores we obtain on the SE data-set, the SKS model does not outperform both ablated versions `-s` and `-sc`, thus contradicting the first and second claim in Section 3. In fact, while SKS obtains an accuracy of 61.04 and a macro-F1 score of 60.88, the `-s` model outperforms it, reaching an accuracy and a macro-F1 score of 64.17 and 63.05, respectively. On the other hand, the third claim appears to hold. With an accuracy of 60.52 and a macro-F1 score of 60.47 the `-sc` model is the one registering the worst performance.
Turning to the DV data-set, none of the claims appear to be substantiated by our findings. The SKS model scores the lowest with an accuracy of 93.63 and a weighted-F1 score of 93.62, while the ablated versions `-s` and `-sc` register similar values for both metrics, with an accuracy of 93.99 and 93.98 and a weighted-F1 score of 94.11 and 94.12, respectively.
For a visual inspection of the results presented in Table 1 we also plot box plots of the scores obtained in multiple reproduction attempts in Figure 1. Despite some overlap in the range of the obtained scores, the median scores of the SKS model is lower than those of the ablated versions. The figure also shows that the scores reported in the original paper fall within the range ± 1.5 standard deviation from the mean of the scores of the

Model	DV				SE			
	Acc		F1 (weighted)		Acc		F1 (macro)	
	Orig.	Repro.	Orig.	Repro.	Orig.	Repro.	Orig.	Repro.
-sc	94.0	93.98 (±1.61)	94.0	94.12 (±1.73)	59.6	60.52 (±1.44)	59.3	60.47 (±1.40)
-s	94.5	93.99 (±1.49)	94.3	94.11 (±1.58)	61.3	64.17 (±0.99)	61.3	63.05 (±0.63)
SKS	95.1	93.63 (±2.09)	96.3	93.62 (±2.37)	65.9	61.04 (±1.81)	65.2	60.88 (±1.64)

Table 1. For each data-set and performance measure we report each model's original (Orig.) results on the left and the reproduced (Repro.) ones on the right, including the standard deviation of the reproduced score.

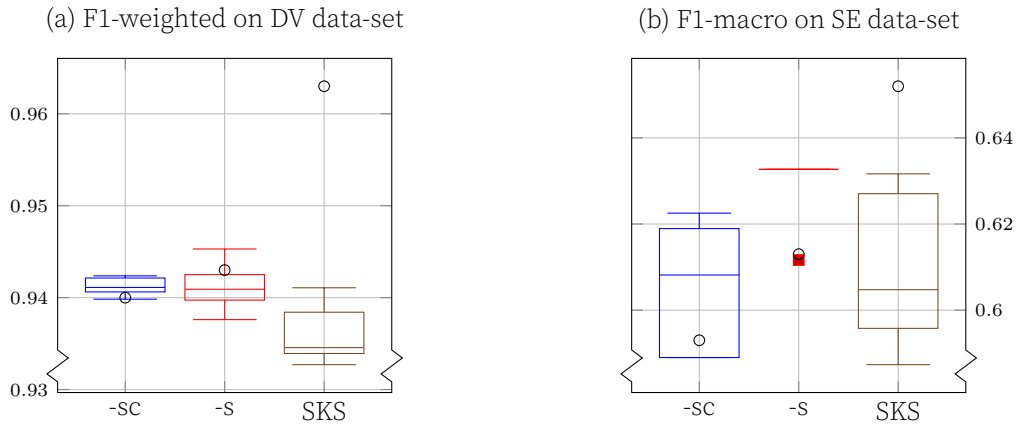

Figure 1. Box plots of (a) F1-weighted on the DV data-set and (b) F1-macro on the SE data-set, from repeated experiments with different initializations. Circles represent the scores reported in the original article. The red square in (b) indicates the single outlier for the `-s` option on this data-set. The rest of the scores are equal to the median. Note that the y-axes do not have the same scale.

multiple reproduction experiments. However, for both data-sets, the original scores of SKS is substantially above this range.

5.1 Alternative metrics

The original paper reports macro- or weighted-averaged F1 scores, with the motivation of comparability to earlier research on these data-sets. However, the task at hand is a binary classification task with a clear positive class. Incorporating the negative class score through averaging does not allow assessing the success of the classifier on the task. Furthermore, relying on weighted averaging without having a justified set of weights, but using weights proportional to the support of each class, rewards classifiers with majority bias even further.

To present a more interpretable impression of the success of each model and provide further insight into the differences based on model ablation and alternations, Figure 2 depicts the distribution of precision, recall and F1-scores for the different reproduction experiments carried out on the two data-sets.

Although there is a large overlap in the range of the scores, the plots indicate that jointly learning sentiment analysis (SKS) improves the precision of the hate speech detection on the DV data-set. Despite having a negative impact on the recall, it also yields a slightly better median F1 score. However, the effect of the sentiment task appears to be mostly negative on the SE data-set.

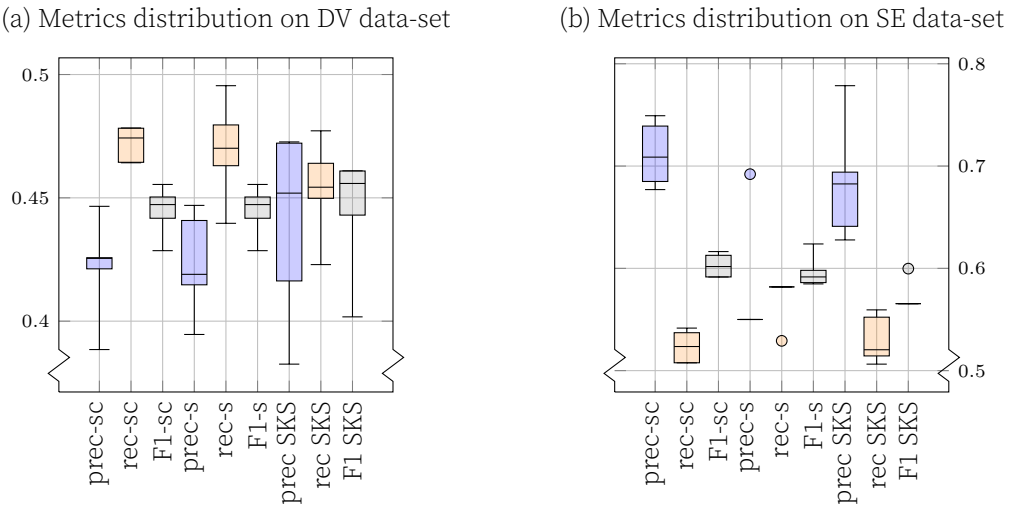

Figure 2. Box plots of binary precision (blue), recall (orange), and F1-scores (grey) on (a) the DV data-set and (b) the SE data-set, from repeated experiments with different initializations. Note that the y-axes do not have the same scale.

6 Discussion

The reproducibility results from Section 5 do not fully support the claims outlined in Section 3 for either data-sets. In particular, our findings seem to suggest that the multi-task learning approach implemented by the authors to allow the SKS model to extract sentiment features and apply them to hate speech detection does not yield the expected results. However, considering the lack of a comprehensive documentation, the convoluted structure of the code-base and the insufficient communication with the original authors it is hard to draw definitive conclusions. In fact, there are a number of plausible explanations as to why our findings diverge from those reported in the original paper.

For instance, considering the slight difference between the optimal hyperparameters we found and those reported by Zhou et al., as well as the large variation in the model's scores, one could speculate that, at least for part of the experiments, the study employed some parameters which have not been reported. This would also explain the difference between some of the values indicated in the paper and those used in the provided implementation.

Another explanation could lie in the fact that we inadvertently deviated from the original implementation while trying to fix some of the issues we faced in running the code-base. Whenever information was missing or not completely clear assumptions had to be made, and we tried to approximate the original results by trial and error. This was the case for the `-sc` model where the procedure to ablate the category embeddings component was not given and the answer we received from the authors did not help us to overcome the problem.

Yet another explanation revolves around the data. The main intuition behind the original study is the fact that hate speech typically carries a negative sentiment. Hence, the relation between these two tasks would help the model to better identify hate speech (arguably by increasing recall). However, a manual inspection of the data-sets suggests that their content may actually be surprising for a classifier informed by sentiment analysis. Both data-sets are collected using keywords that are likely to contain hate speech, and the negative class (i.e., non-hate speech one) contains posts that are either offensive (but not hate speech), or content generated by people counteracting earlier offensive content. That being the case, the sentiment of this class is not necessarily positive and helpful for discriminating hate speech *in these data-sets*. However, in a more realis-

tic environment, the original proposal may be promising. Given more 'normal' negative class instances, learning sentiment analysis jointly is likely to inform the hate speech detection task. The binary evaluation metrics presented in Figure 2 indicate that at least on the DV data-set, the addition of sentiment may have some positive effects. Understanding the reasons for these differences, and improving the joint learning model is a possible direction for future research.

6.1 What was easy

The paper provides some broad indications with respect to the training details and both the data-sets and the code-base are open-sourced.

6.2 What was difficult

The lack of a comprehensive documentation, the convoluted structure of the code-base and the insufficient communication with the original authors contributed to an arduous code review and reproducibility effort.

6.3 Communication with original authors

We first tried to review and run the provided code-base by ourselves. However, after encountering some issues related to how the data-sets were being processed and how to run the `-sc` model ablating category embeddings, we decided to reach out to the authors through GitHub. One of the corresponding authors provided some indications which, unfortunately, did not help us overcome the problems at hand.
We also tried to contact the authors per email twice, inquiring about some aspects of the model implementation as well as the procedure they followed to tune the hyperparameters. However, we never received an answer.

References

1. X. Zhou, Y. Yong, X. Fan, G. Ren, Y. Song, Y. Diao, L. Yang, and H. Lin. "Hate Speech Detection Based on Sentiment Knowledge Sharing." In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, 2021, pp. 7158–7166. DOI: 10.18653/v1/2021.acl-long.556. URL: https://aclanthology.org/2021.acl-long.556.
2. T. Davidson, D. Warmsley, M. Macy, and I. Weber. "Automated Hate Speech Detection and the Problem of Offensive Language." In: *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 11. 2017, pp. 512–515. URL: https://ojs.aaai.org/index.php/ICWSM/article/view/14955.
3. A. Schmidt and M. Wiegand. "A Survey on Hate Speech Detection using Natural Language Processing." In: *SocialNLP@EACL*. 2017.
4. V. Basile, C. Bosco, E. Fersini, D. Nozza, V. Patti, F. M. Rangel Pardo, P. Rosso, and M. Sanguinetti. "SemEval-2019 Task 5: Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter." In: *Proceedings of the 13th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, 2019, pp. 54–63. DOI: 10.18653/v1/S19-2007. URL: https://aclanthology.org/S19-2007.
5. W. Yin and A. Zubiaga. "Towards generalisable hate speech detection: a review on obstacles and solutions." In: *PeerJ Computer Science* 7 (2021), e598.
6. M. Zampieri, P. Nakov, S. Rosenthal, P. Atanasova, G. Karadzhov, H. Mubarak, L. Derczynski, Z. Pitenis, and Ç. Çöltekin. "SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020)." In: *Proceedings of the Fourteenth Workshop on Semantic Evaluation*. Barcelona (online): International Committee for Computational Linguistics, Dec. 2020, pp. 1425–1447. DOI: 10.18653/v1/2020.semeval-1.188. URL: https://aclanthology.org/2020.semeval-1.188.
7. N. Reimers and I. Gurevych. "Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging." In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, 2017, pp. 338–348. DOI: 10.18653/v1/D17-1035. URL: https://aclanthology.org/D17-1035.

8. Ç. Çöltekin. "Verification, Reproduction and Replication of NLP Experiments: a Case Study on Parsing Universal Dependencies." In: *Proceedings of the Fourth Workshop on Universal Dependencies (UDW 2020)*. Barcelona, Spain (Online): Association for Computational Linguistics, 2020, pp. 46–56. URL: https://aclanthology.org/2020.udw-1.6.
9. O. E. Gundersen, K. Coakley, and C. Kirkpatrick. "Sources of Irreproducibility in Machine Learning: A Review." In: *arXiv preprint arXiv:2204.07610* (2022).
10. N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer." In: (2017). arXiv:1701.06538 [cs.LG].
11. J. Ma, Z. Zhao, X. Yi, J. Chen, L. Hong, and E. H. Chi. "Modeling Task Relationships in Multi-Task Learning with Multi-Gate Mixture-of-Experts." In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '18. London, United Kingdom: Association for Computing Machinery, 2018, pp. 1930–1939. DOI: 10.1145/3219819.3220007. URL: https://doi.org/10.1145/3219819.3220007.
12. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. "Attention Is All You Need." In: *arXiv* (2017).
13. D. Shen, G. Wang, W. Wang, M. R. Min, Q. Su, Y. Zhang, R. Henao, and L. Carin. *On the Use of Word Embeddings Alone to Represent Natural Language Sequences*. 2018. URL: https://openreview.net/forum?id=Sy5OAyZC-.
14. J. Pennington, R. Socher, and C. D. Manning. "GloVe: Global Vectors for Word Representation." In: *Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 1532–1543. URL: http://www.aclweb.org/anthology/D14-1162.
15. Martín Abadi et al. "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems." In: (2015). Software available from tensorflow.org. URL: https://www.tensorflow.org/.