Offensive Comment Classification on German Language Microposts

Matthias Bachfischer* **Uchenna Akujuobi †** **Xiangliang Zhang‡**
Computer, Electrical and Mathematical Sciences and Engineering Division
King Abdullah University for Science and Technology (KAUST)

Abstract

In this paper, we present two deep-learning based classifier systems for the identification of offensive comments in German Language microposts: A bidirectional LSTM and a CNN model. We compare the performance of our systems with a traditional, machine-learning based SVM classifier and evaluate our approach on Task 1 (binary classification) of the GermEval 2018 shared task where our best model is able to reach an F-1 score of 90.49%

1 Introduction

Modern communication devices and social media play an increasingly important role in our daily lives and the Internet has created opportunities for exchanging information with people from all over the globe in real-time. Unfortunately however, this freedom gets frequently abused, and hate speech and toxic comments are present in virtually all online communities. A 2017 report by Pew Research came to the conclusion that up to 41% of all adults have personally experienced online harassment (Duggan, 2017).

Automated detection routines to identify and block toxic messages have proven to be viable methods in shielding online communities from harassment (Wulczyn et al., 2017). Training a computer to understand the emotions and opinions expressed in a document is a common task in the area of Natural Language Processing (NLP), and the results from previous publications (Georgakopoulos et al., 2018) as well as a Kaggle competition [1] sponsored by Google Jigsaw have already shown promising results for the identification of toxic content in online messages.

The intention of this paper at hand is to create a series of deep-learning based neural network models to compete in Task 1 (binary classification) of the GermEval 2018[2] competition. The GermEval 2018 competition is a shared task for the identification of offensive comments in German language microposts. For our research, we choose a simple Support Vector Machine (SVM) model as a baseline and compare its performance against our implementations of a bidirectional Long Short-Term Memory (LSTM) and a Convolutional Neural Network (CNN) model.

2 Related Works

So far, most of the research in the area of toxic comment classification has been focused on English language, and a variety of machine-learning and deep-learning models have been produced to tackle this problem (Schmidt and Wiegand, 2017). Amongst others, Georgakopoulos et al. (2018) used a deep-learning based CNN model to detect toxic language in online content. Nobata et al. (2016), on the other hand, developed a machine-learning model using a variety of feature classes (N-grams, Syntactic Semantics etc.) and were able to outperform existing deep-learning based approaches. Other research by Razavi et al. (2010) used multi-level classification to detect offensive comments, mainly in Usenet messages.

The identification of toxicity in German language messages has received less attention by the research community so far, and comparable research results are sparse. In the related domain of sentiment analysis for tweets, Cieliebak et al. (2017) created a corpus consisting of 10.000 tweets in German language and provided benchmarks for the classi-

*bachfischer.matthias@googlemail.com
†uchenna.akujuobi@kaust.edu.sa
‡xiangliang.zhang@kaust.edu.sa

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

[2] Germeval 2018 - Shared Task on the Identification of Offensive Language - https://projects.cai.fbi.h-da.de/iggsa

fication of these tweets into sentiment classes of either *positive, negative* or *neutral* using a CNN.

3 Data

Before training our systems, we first obtain the training set from the GermEval 2018 competition mailing list. The training set contains a total of 5009 messages which have been labeled either *OFFENSE* or *OTHER*. A detailed breakdown of the class distribution in the dataset is presented in Table 1.

Dataset	Offense	Other	Total
Training set	1688	3321	5009

Table 1: Class distribution - GermEval 2018 dataset

The dataset is imbalanced, and the majority of the tweets (66%) belong to the neutral class, whereas the remaining data (34%) belongs to the offensive class. The microposts within the dataset were extracted exclusively from Twitter[3] because the conference organizers regarded tweets "as a prototypical type of micropost"[2].

4 Experimental Setup

We present two classification systems in our research: a bidirectional LSTM model and a CNN model. Both models were implemented in Python and make use of the Keras library (Chollet, 2015) for training the classifier. The experiments were performed on a workstation running Ubuntu 16.04 with 64 cores and 128 GB of RAM.
For this research we use the word vectors published by Deriu et al.(2017). These vectors were trained on a total of 200 million tweets and have a dimensionality of $d = 200$.
Preprocessing: Before extracting features, we first preprocess the data according to the following procedure:

1. Replace URLs, usernames and retweets with replacement tokens *URLTOK*, *USRTOK* and *rt*
2. Convert tweet text to lowercase
3. Convert categorical classification variables into an One-Hot encoded vector
4. Tokenize tweets (using Keras's builtin Tokenizer) and create a list of word indexes with length $l = 100$ (comments shorter than 100 are padded with 0)

For further reference, a preprocessed tweet is presented in Example 4.1.

Example 4.1:
Original: *@salzmanufaktur @Renft1964* Jetzt bekommt Merkel noch Grüne Untergangs-Beschleuniger dabei!
Preprocessed: *USRTOK USRTOK* jetzt bekommt merkel noch grüne untergangs-beschleuniger dabei!

5 System Description

After preprocessing, we feed the data into our classification models: a bidirectional LSTM and a CNN model. By using the word vectors from Deriu et al., we create an embedding matrix where we randomly initialize the words that are not in the word embeddings with the arithmetic mean and standard deviation obtained from the embeddings. The resulting embedding matrix has the size of $|\vec{w}_1;...;\vec{w}_L| \in \mathrm{IR}^{\mathrm{L}\times 200}$ with L being the number of unique words in our training set.
While training our models, we try to minimize the binary cross entropy loss on the training set given per the formula below:

$$-\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)] \qquad (1)$$

The final outputs of the models are connected to a softmax regression layer which returns the class $\hat{y} \in [1,K]$ with the largest propability

$$\hat{y} = \arg\max_j \; P(y = j \mid \mathbf{x}) = \frac{e^{\mathbf{x}^\top \mathbf{w}_j}}{\sum_{k=1}^{K} e^{\mathbf{x}^\top \mathbf{w}_k}} \qquad (2)$$

where w_j denotes the weight vector for class j. For the optimization step, we choose the *Adam* optimizer (Kingma and Ba, 2015) with a learning rate $lr = 0.001$, $\beta_1 = 0.99$, $\beta_2 = 0.999$ and $\varepsilon = 1^{-8}$.

5.1 LSTM Model

The LSTM model in this research was derived from the works of Hochreiter and Schmidhuber (1997). So far, LSTM networks have been successfully applied in a variety of tasks such as Machine Translation (Sutskever et al., 2014) and Image Captioning (Vinyals et al., 2015). Recent research however shows that LSTM models also perform well when applied to NLP tasks such as text classification

[3]Twitter social network - `https://twitter.com`

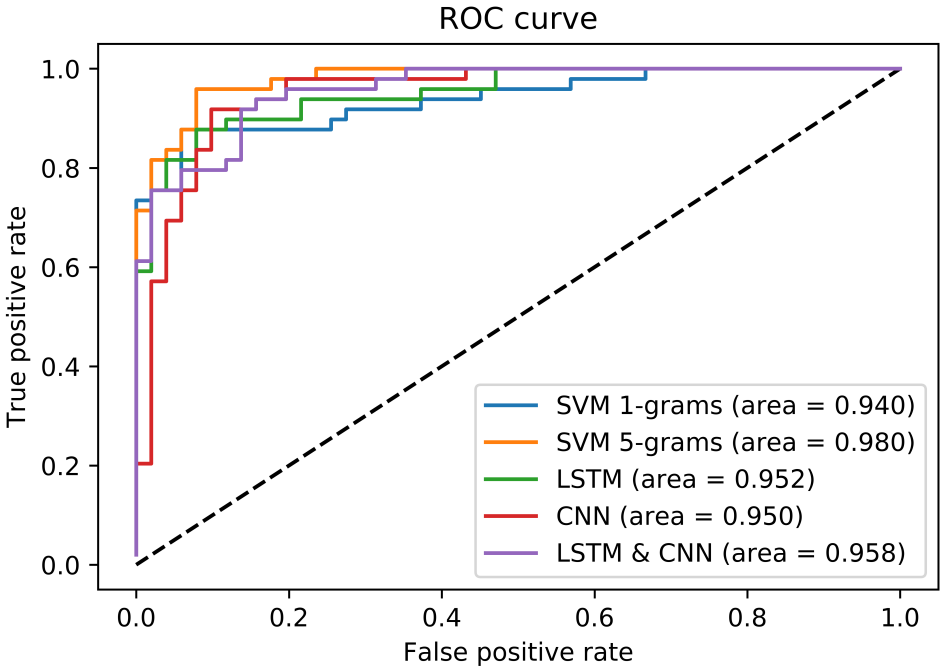

Figure 1: Comparison of ROC metric for our models (PLACEHOLDER)

(Zhang et al., 2015; Zhou et al., 2016).
In this paper, we employ a bidirectional LSTM model with 64 units. The output is passed to two fully connected layers with 64 and 2 units respectively. To prevent our model from overfitting, we apply the early-stopping technique (Prechelt, 1998) in combination with a dropout of $d = 0.5$ after the first dense layer as well as on the recurrent input signal of the LSTM units (Gal and Ghahramani, 2016). We furthermore use Rectified Linear Unit (RELU) as the activation function of the first hidden layer (Srivastava et al., 2014; Glorot et al., 2011).

5.2 CNN Model

The CNN model used in this research builds on the work of Kim (2014) who proposed to use a 2 layered CNN to perform sentence classification in NLP tasks. We create our model by using 1 convolutional layer with 64 filters (one layer consists of a convolution and a pooling layer). The output of the convolutional layers is then fed into a sequence of 2 dense layers. As in the previous model, we again make use of the RELU function for the activation of the layers and apply early-stopping in combination with a dropout of $d = 0.5$.

5.3 Ensemble

To further improve the overall performance, we create an ensemble by combining the results from the LSTM and the CNN model. To create the ensemble, we average the probability outputs from both models and apply the *argmax* function to obtain the binary outputs. We then use these predictions for the final submission run of the GermEval 2018 task.

5.4 SVM Model

As a baseline for the evaluation of our results, we use a simple SVM classifier trained on Term-Frequency times Inverse Document-Frequency (TF-IDF) vectors (Ramos, 2003) of the tweets in the dataset. The TF-IDF scores were calculated by using the count matrices of 1-grams and 5-grams where the tweet texts serve as input tokens. The classifier uses Stochastic Gradient Descent (SGD) with the logistic regression loss function where we multiply the regularization term with a constant $\alpha = 1^{-5}$.

6 Results

Since the official test set will only be published by the task organizers on July 17, 2018, the authors of this paper used a sample set with size $n = 100$ obtained from the conference website to conduct their preliminary experiments. Once the official test data is published, this section will be updated and the final results will be submitted to the shared task.

For the evaluation of our models, we use the script provided by the competition organizers to calculate the F-1 score on the test set (size $n = 100$). In Table 2, we present the F1-scores for each category. We marked the best result in bold face.

	Offense	Other	Average
SVM (1-grams)	85.71	88.07	87.33
SVM (5-grams)	88.89	90.91	90.49
LSTM	88.89	90.91	**90.49**
CNN	80.00	85.22	84.10
LSTM & CNN	81.40	85.96	84.95

Table 2: F1-Scores per Category

To evaluate the output of our classifiers, we further included the Receiver Operating Characteristic (ROC) metric in Figure 1. As we can see, the LSTM model clearly outperforms the CNN model on the GermEval 2018 dataset.
Once the official test data is released, we will update this section with an in-depth discussion of the results.

7 Conclusion

The objective of our participation in the GermEval 2018 shared task was to evaluate the performance of deep-learning based models on the classification of offensive language in German microposts. We presented two systems and evaluated their results on the given dataset: a bidirectional LSTM model and a CNN model. As a result from this research, we have shown that the LSTM model outperforms the CNN model on the given GermEval dataset.
We hope that online social network platforms can use our results to build systems that can successfully detect and combat toxicity in online conversations. Services such as the Perspective API [4] are taking a step into the right direction, and we expect to see more fascinating research for making the Internet a more friendlier and welcoming place.

References

François Chollet. 2015. Keras library. `https://keras.io` (accessed June 19, 2018).

Mark Cieliebak, Jan Milan Deriu, Dominic Egger, and Fatih Uzdilli. 2017. A twitter corpus and benchmark resources for german sentiment analysis. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, Valencia, Spain, April 3-7, 2017*, pages 45–51. Association for Computational Linguistics.

Jan Deriu, Aurelien Lucchi, Valeria De Luca, Aliaksei Severyn, Simon Müller, Mark Cieliebak, Thomas Hofmann, and Martin Jaggi. 2017. Leveraging large amounts of weakly supervised data for multi-language sentiment classification. In *Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, April 3–7, 2017*, pages 1045–1052. International World Wide Web Conferences Steering Committee.

Maeve Duggan. 2017. Online harassement 2017. Report, Pew Research Center.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48, New York, USA, June 19 - 24, 2016*, pages 1050–1059. JMLR.org.

Spiros V. Georgakopoulos, Sotiris K. Tasoulis, Aristidis G. Vrahatis, and Vassilis P. Plagianakos. 2018. Convolutional neural networks for toxic comment classification. *arXiv preprint arXiv:1802.09957*.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, USA, April 11-13 2011*, volume 15 of *Proceedings of Machine Learning Research*, pages 315–323.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Lstm can solve hard long time lag problems. *Neural Computation*, 9(8):1735–1780.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, October 25-29, 2014*, pages 1746–1751. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR), San Diego, USA, May 7-9, 2015*.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th International Conference on World Wide Web, Montreal, Quebec, Canada, April 11-15 2016*, pages 145–153. International World Wide Web Conferences Steering Committee.

[4] Perspective API - `https://www.perspectiveapi.com/`

Lutz Prechelt. 1998. *Early stopping-but when?* Neural Networks: Tricks of the trade. Springer.

Juan Ramos. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the First instructional Conference on Machine Learning, Piscataway, USA, December 3-8, 2003*, volume 242, pages 133–142.

Amir H. Razavi, Diana Inkpen, Sasha Uritsky, and Stan Matwin. 2010. Offensive language detection using multi-level classification. In *Proceedings of the 23rd Canadian Conference on Advances in Artificial Intelligence, Ottawa, Canada, May 31 - June 02, 2010*, pages 16–27. Springer.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, Valencia, Spain, April 3-7, 2017*, pages 1–10.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, Montreal, Canada, December 8-13, 2014*, pages 3104–3112.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, USA, June 7-12, 2018*, pages 3156–3164. IEEE.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, April 3-7, 2017*, pages 1391–1399. International World Wide Web Conferences Steering Committee.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, Montreal Canada, December 7-12, 2015*, pages 649–657.

Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. 2016. Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, Osaka, Japan, December 11-17, 2016*, page 3485–3495.