UHH-LT at SemEval-2019 Task 6: Supervised vs. Unsupervised Transfer Learning for Offensive Language Detection

Gregor Wiedemann **Eugen Ruppert*** **Chris Biemann**
Language Technology Group / *Base.Camp
Department of Informatics
University of Hamburg, Germany
{gwiedemann, ruppert, biemann}@informatik.uni-hamburg.de

Abstract

We present a neural network based approach of transfer learning for offensive language detection. For our system, we compare two types of knowledge transfer: supervised and unsupervised pre-training. Supervised pre-training of our bidirectional GRU-3-CNN architecture is performed as multi-task learning of parallel training of five different tasks. The selected tasks are supervised classification problems from public NLP resources with some overlap to offensive language such as sentiment detection, emoji classification, and aggressive language classification. Unsupervised transfer learning is performed with a thematic clustering of 40M unlabeled tweets via LDA. Based on this dataset, pre-training is performed by predicting the main topic of a tweet. Results indicate that unsupervised transfer from large datasets performs slightly better than supervised training on small 'near target category' datasets. In the SemEval Task, our system ranks 14 out of 103 participants.

1 Introduction

The automatic detection of hate speech, cyberbullying, abusive, aggressive or offensive language has become a vital field of research in natural language processing (NLP) during recent years. Especially in social media, the tone of conversations escalates in a disturbing way that often threatens a free, democratic and argumentative discourse of users. Concerning the tremendous amount of digital texts posted on platforms such as Twitter, Facebook or in comments sections of online newspapers, automatic approaches to offensive language detection are of high relevancy for moderation and filtering of content as well as for studying the phenomenon of offensive language use in social media at large scale.

To take account of this development, a shared task on "offensive language detection" was conducted at the SemEval 2019 workshop. This paper describes our approach to the shared task 6 (OffensEval) as organized and described in detail by Zampieri et al. (2019b). The task contains three hierarchically ordered sub-tasks: Task A requires a classification of tweets into either offensive (OFF) or not offensive (NOT), Task B subdivides all offensive tweets into either targeted insults (TIN) or generally offensive expressions not targeted to any specific entity (UNT), and Task C finally asks to assign one out of three specific target labels to all targeted insults: groups (GRP, e.g. ethnic groups), individuals (IND, e.g. a politician or a specific Twitter user), or other (OTH, e.g. the media industry). The dataset consisting of 14,100 examples (13,240 in the training set, 860 in the test set) was annotated via crowdsourcing (Zampieri et al., 2019a). Each tweet was labeled by at least two annotators who must reach an agreement of at least 66% for including the tweet in the dataset. The dataset is characterized by a high imbalance of label distributions, especially for Tasks B and C.

There are several challenges for automatic offensive language detection that render simple dictionary-based approaches unusable. First, label distribution in the dataset is highly skewed for all sub-tasks. Although offensive language is a growing problem for social media communication, it still accounts for only a small fraction of all content posted. Second, language characteristics in social media pose a severe challenge to standard NLP tools. Misspellings, slang vocabulary, emoticons and emojis, as well as ungrammatical punctuation must be taken into account for a successful solution. Third, offensive language is highly context-dependent. For instance, swear words are often used to mark overly positive emotion ("This is fucking great!!!"), and actually neutral and descriptive sentences can be conceived as derogatory if they refer to a specific individual ("@Barack-

Obama he is a Muslim").

Our approach to the OffensEval shared task is based on two main contributions: First, we introduce a BiGRU-3CNN neural network architecture in combination with pre-trained sub-word embeddings that are able to handle social media language robustly. Second, we investigate two types of knowledge transfer: supervised and unsupervised pre-training. Supervised pre-training of our neural network architecture is performed as multi-task learning of parallel training of five different NLP tasks with some overlap to offensive language detection. Unsupervised transfer learning is performed with a thematic clustering of a large dataset of unlabeled tweets via LDA. After shortly referencing related work (Section 2), we introduce both approaches in detail in Section 3 and present the results in Section 4.

2 Related Work

Two recent survey papers, Schmidt and Wiegand (2017) and Fortuna and Nunes (2018), summarize the current state of the art in offensive language detection and related tasks such as hate speech or abusive language detection. Specifically for offensive language detection, the paper by Davidson et al. (2017) introduced a publicly available dataset which was reused in (Malmasi and Zampieri, 2017, 2018; ElSherief et al., 2018; Zhang et al., 2018) as well as in our approach of supervised pre-training.

A predecessor of our transfer learning approach has already successfully been applied at GermEval 2018 (Wiegand et al., 2018), a shared task on offensive language detection in German language tweets. In our paper (Wiedemann et al., 2018), we tested different types of knowledge transfer and transfer learning strategies. We further found that latent semantic clusters of user handles in tweets (e.g. user accounts run by media companies or politicians) are a very helpful feature to predict offensive language since they provide valuable context information how to interpret otherwise ambiguous tweets. Unfortunately, this feature cannot be used for the SemEval 2019 Task 6 since user mentions have all been unified to the token '@USER' in the training data. Thus, we base our approach on the best performing transfer learning strategy from Wiedemann et al. (2018) but implement several minor improvements, which will be described in detail in the following.

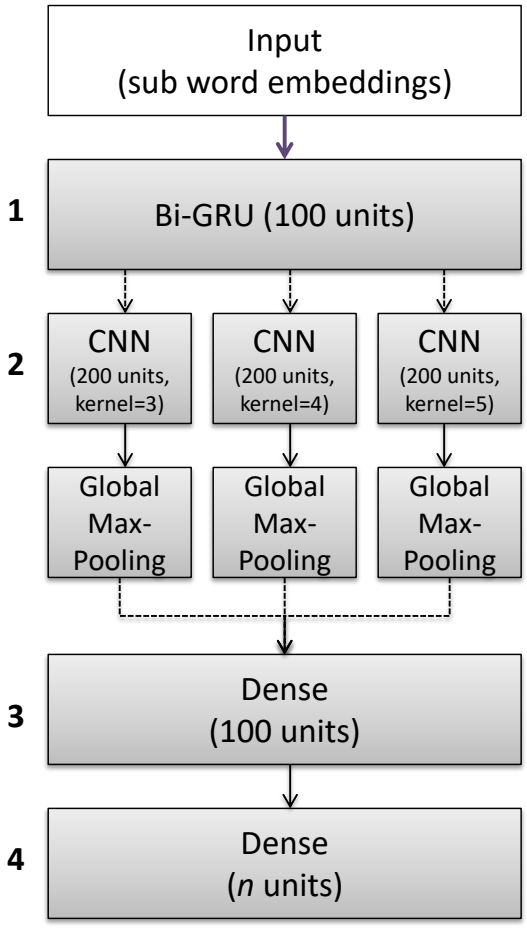

Figure 1: BiGRU-3-CNN model architecture. We use a combination of recurrent and convolutional cells for learning. As input, we rely on (sub-)word embeddings. Dashed lines indicate dropout with rate 0.5 between layers. The last dense layer contains n units for prediction of the probability of each of the n classification labels per sub-task.

3 Methodology

We utilize a neural network architecture for text classification with randomly initialized weights as a baseline, and together with two types of pre-training layer weights for transfer learning: supervised and unsupervised pre-training. Evaluation for model selection is performed via 10-fold cross-validation to determine submission candidates for the SemEval shared task.

Preprocessing: Tweets are tokenized with an extended version of the NLTK (Bird et al., 2009) tweet tokenizer. In addition to correct tokenization of emoticons and shortening of repeated character sequences (e.g. '!!!!!!') to a maximum length of three, we separate # characters as individual token

from hashtags. If hashtags contain camel casing, we split them into separate tokens at each uppercase letter occurrence (e.g. '#DemocratsForPeace' is tokenized into '# democrats for peace'). Finally, all tokens are reduced to lower case. In order to account for atypical language, we use sub-word embeddings to represent the input of token sequences to our model. FastText embeddings (Bojanowski et al., 2017) are derived from character n-grams and, thus, provide meaningful word vectors even for words unseen during training, misspelled words and words specifically used in the context of social media such as emojis. We utilize a pre-trained model for the English language published by Bojanowski et al. (2017).

Model architecture: We employ a neural network architecture implemented with the Keras framework for Python[1] as shown in Fig. 1. It combines a bi-directional Gated Recurrent Unit (GRU) layer (Cho et al., 2014) with 100 units followed by three parallel convolutional layers (CNN), each with a different kernel size $k \in 3, 4, 5$, and a filter size 200. The outputs of the three CNN blocks are reduced by global max-pooling and concatenated into a single vector. This vector is then fed into a dense layer with LeakyReLU activation producing a final feature vector of length 100, which is forwarded into the prediction layer (softmax activation). For regularization, dropout is applied to the recurrent layer and to each CNN block after global max-pooling (dropout rate 0.5). For training, we use categorical cross-entropy loss and the Nesterov Adam optimization with a learning rate of 0.002. To account for imbalance in the training set, we set class weights to pay more attention to samples from the under-represented class in the loss function.

Supervised Pre-training: Instead of end-to-end text classification based on a random initialization of the parameters weights of our model, we seek to increase performance from knowledge transfer. For the supervised approach, we pre-train the model weights in a multi-task learning setup with related semantic categories. Instead of one prediction layer (see layer 4 in Fig. 1), we use m prediction layers connected to layer 3 to train m tasks in parallel. The following four publicly available datasets were compiled into one training set: offensive language tweets by (Davidson et al., 2017), flamewar comments from the Yahoo news annotated corpus (Napoles et al., 2017), sentiments of tweets from (Go et al., 2009), aggressive tweets and Facebook comments from the TRAC shared task (Kumar et al., 2018). A fifth dataset was compiled from about 30,000 randomly sampled tweets in our unsupervised background collection (see next Section) containing either a happy or an angry emoji. The merged dataset contains ca. 115k partially labeled instances for pre-training from which a sample of 5k was used as validation set. Missing labels for the combined set were filled by training a separate model for each of the m individual tasks on the respective dataset and predict a label for each instance in the other four datasets. Multi-task pre-training is performed with a batch-size of 256 for 15 epochs.

Unsupervised Pre-training: For the unsupervised approach, we utilize a large background corpus of tweets that were collected from the Twitter streaming API in 2018. Since the API provides a random fraction of all tweets (1%), language identification is performed to filter for English tweets only. From this tweet collection, we sample 20 million non-duplicate tweets containing at least two non-URL tokens as our background corpus. As a pre-training task, we first compute a Latent Dirichlet Allocation (LDA) model (Blei et al., 2003) with $K = 1,000$ topics to obtain semantic clusters of our background corpus.[2] From the topic-document distribution of the resulting LDA model, we determine the majority topic id for each tweet as a target label for prediction during pre-training our neural model. Pre-training of the neural model is performed with a batch-size of 256 for 10 epochs.

Transfer learning: Once the neural model has been pre-trained, we can apply it for learning our actual target task. For this, we need to remove the final prediction layer of the pre-trained model (i.e. Layer 4 in Fig. 1) and add a new dense layer for prediction of one of the actual label sets. To prevent the effect of "catastrophic forgetting" of pre-trained knowledge during task-specific model training, we apply a specific layer weight freezing strategy as suggested in Wiedemann et al. (2018). First, the newly added final prediction layer is trained while all other model weights re-

[1] https://keras.io

[2] For LDA, we used Mallet (http://mallet.cs.umass.edu) with Gibbs Sampling for 1,000 iterations and priors $\alpha = 10/K$ and $\beta = 0.01$.

Task	No transfer	Supervised	Unsupervised
A	76.26	**77.46**	77.36
B	58.87	**61.24**	60.57
C	56.66	54.16	**58.26**

Table 1: Model selection (cross-validation, macro-F1)

main frozen. Training is conducted for 15 epochs. After each epoch performance is tested on the validation set. The best performing model state is then used in the next step of fine-tuning the pre-trained model layers. Employing a bottom-up strategy, we unfreeze the lowest layer (1) containing the most general knowledge first, then we continue optimization with the more specific layers (2 and 3) one after the other. During fine-tuning of every single layer, all other layers remain frozen and training is performed again for 15 epochs selecting the best performing model at the end of each layer optimization. In a final round of fine-tuning, all layers are unfrozen.

Ensemble: For each sub-task A, B and C, the cross-validation results in 10 best performing models from transfer learning per configuration. For submission to the shared task, we select the model with the highest average performance across all folds. Moreover, as a simple ensemble classification, predictions of these 10 models on the test set instances are combined via majority vote.

4 Results

Model selection: To compare different types of pre-training for knowledge transfer, we use the official shared task metric macro-averaged F1. Table 1 displays the averaged results of 10-fold cross-validation for all three tasks with no transfer as baseline compared to supervised transfer from multi-task learning and pre-training on unsupervised LDA clustering. The results indicate that transfer learning is able to improve performance for offensive language detection for all tasks. With the exception of supervised transfer for task C, the relative improvements are larger the smaller the training datasets get for each of the hierarchically ordered tasks. In general, for the lower level tasks B and C, a severe performance drop can be observed compared to task A.

The comparison between unsupervised and supervised pre-training delivers a mixed result. While the performance of the supervised transfer approach slightly exceeds the unsupervised approach for task A and B, for task C, containing only very small numbers of positive examples for each class in the training dataset, the unsupervised approach clearly beats the network pre-training by supervised near-target category tasks. Supervised transfer even fails to beat the baseline of no transfer learning at all. We assume that this type of pre-training tends to over-fit the model if there is only little training data to learn from. Unsupervised pre-training on very large datasets, in contrast, better captures generic language regularities which is beneficial for arbitrary categories.

System	F1 (macro)	Accuracy
Task A		
All NOT baseline	0.4189	0.7209
All OFF baseline	0.2182	0.2790
Supervised	**0.7887**	**0.8372**
Unsupervised	0.7722	0.8337
Task B		
All TIN baseline	0.4702	0.8875
All UNT baseline	0.1011	0.1125
Unsupervised	**0.6608**	**0.8917**
Task C		
All GRP baseline	0.1787	0.3662
All IND baseline	0.2130	0.4695
All OTH baseline	0.0941	0.1643
Unsupervised	**0.5752**	**0.6761**

Table 2: Official test set performance results

Shared task submissions: Table 2 displays our best official results of ensemble classifications submitted to the shared tasks A, B, and C. A systematic comparison between the two compared approaches of pre-training would have required submission of two classifications per sub-task, one for supervised and one for unsupervised pre-training. Unfortunately, the official shared task website only allowed for three submissions per sub-task[3]. This policy led to the decision to submit only variations / repeated runs of the best classifier we had until the task submission deadline.

Our supervised pre-training approach ranks 14 out of 103 for sub-task A. For sub-tasks B and C, only classifiers pre-training with the unsupervised approach have been submitted. They rank 21 out of 75 for B, and 13 out of 65 for C (see

[3] Also, the official test set was not released yet, so we cannot report a systematic comparison at this point.

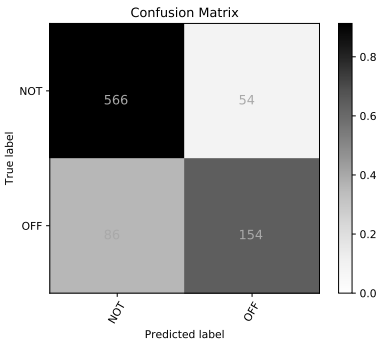

Figure 2: Sub-task A, supervised

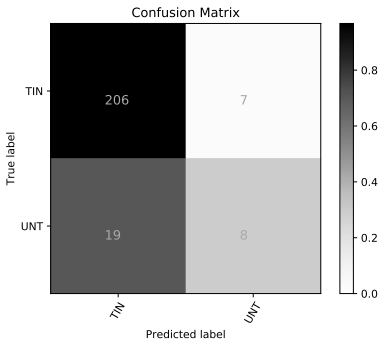

Figure 3: Sub-task B, unsupervised

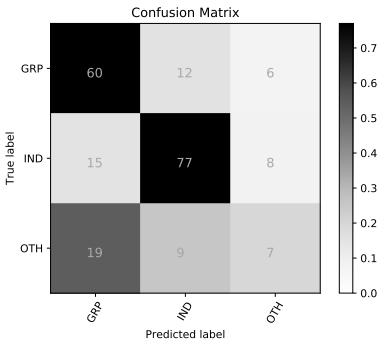

Figure 4: Sub-task C, unsupervised

Zampieri et al. (2019b) for a detailed comparison of all submissions). Fig. 2 to 4 show confusion matrices for the three best runs. The ratio between false positives and false negatives for sub-task A is fairly balanced. False positives mainly comprise hard cases where, for instance, swear words are used in a non-offensive manner. In the highly unbalanced annotations for sub-task B, more tweets were wrongly predicted as targeted insults than true yet unpredicted targeted insults. Here we observe many cases which contain offensive language and some mentioning of individuals or groups but both are not directly linked. A similar situation, where actually characteristics of two categories are contained in a tweet, can be observed for task C in which the classifier falsely predicts a group target instead of 'other'.

5 Conclusion

We systematically compared to types of knowledge transfer for offensive language detection: supervised and unsupervised pre-training of a BiGRU-3-CNN neural network architecture. The former uses a set of near-target category labeled short texts while the latter relies on a very large set of unlabeled tweets. On average, our system performs among the top 20% of all submissions of the OffensEval 2019 shared task. From our experiments, we can draw the following three main conclusions:

- Supervised pre-training with annotated near-target category data is beneficial if the target training data is fairly large.
- Unsupervised pre-training with unlabeled data from LDA clustering processes improves learning for arbitrary tasks even for fairly small target training datasets.
- For unsupervised pre-training, the benefit of transfer learning compared to the baseline without transfer is larger the smaller the target training dataset gets.

In future work, we plan to further investigate the differences between the two types of transfer learning by systematically investigating the influence of different near-target category datasets, and unsupervised topic clustering methods other than LDA for pre-training deep neural network architectures.

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. ACL.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media (ICWSM)*, pages 512–515, Montreal, Canada. AAAI.

Mai ElSherief, Vivek Kulkarni, Dana Nguyen, William Yang Wang, and Elizabeth Belding. 2018. Hate Lingo: A Target-based Linguistic Analysis of Hate Speech in Social Media. *arXiv preprint arXiv:1804.04257*.

Paula Fortuna and Sérgio Nunes. 2018. A Survey on Automatic Detection of Hate Speech in Text. *ACM Computing Surveys (CSUR)*, 51(4):85:1–85:30.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1(12):1–6.

Ritesh Kumar, Atul Kr. Ojha, Shervin Malmasi, and Marcos Zampieri. 2018. Benchmarking Aggression Identification in Social Media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbulling (TRAC)*, pages 1–11, Santa Fe, NM, USA.

Shervin Malmasi and Marcos Zampieri. 2017. Detecting Hate Speech in Social Media. In *Proceedings of the 2017 International Conference Recent Advances in Natural Language Processing (RANLP)*, pages 467–472, Varna, Bulgaria. ACL.

Shervin Malmasi and Marcos Zampieri. 2018. Challenges in Discriminating Profanity from Hate Speech. *Journal of Experimental & Theoretical Artificial Intelligence*, 30:1–16.

Courtney Napoles, Joel Tetreault, Enrica Rosata, Brian Provenzale, and Aasish Pappu. 2017. Finding Good Conversations Online: The Yahoo News Annotated Comments Corpus. In *Proceedings of The 11th Linguistic Annotation Workshop*, pages 13–23, Valencia, Spain. ACL.

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.*, pages 1–10, Valencia, Spain. ACL.

Gregor Wiedemann, Eugen Ruppert, Raghav Jindal, and Chris Biemann. 2018. Transfer Learning from LDA to BiLSTM-CNN for Offensive Language Detection in Twitter. In *Proceedings of GermEval Task 2018, 14th Conference on Natural Language Processing (KONVENS)*, pages 85–94, Vienna, Austria.

Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. 2018. Overview of the GermEval 2018 Shared Task on the Identification of Offensive Language. In *Proceedings of GermEval Task 2018, 14th Conference on Natural Language Processing (KONVENS)*, pages 1–10, Vienna, Austria.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the Type and Target of Offensive Posts in Social Media. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Minneapolis, MN, USA.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval)*, Minneapolis, MN, USA. ACL.

Ziqi Zhang, David Robinson, and Jonathan Tepper. 2018. Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network. In *The Semantic Web (ESWC 2018)*, pages 745–760, Iraklio, Greece. Springer.