BRUMS at HASOC 2019: Deep Learning Models for Multilingual Hate Speech and Offensive Language Identification

Tharindu Ranasinghe[1], Marcos Zampieri[2], and Hansi Hettiarachchi[3]

[1] Research Group in Computational Linguistics, University of Wolverhampton, UK
`T.D.RanasingheHettiarachchige@wlv.ac.uk`
[2] College of Liberal Arts, Rochester Institute of Technology, USA
`marcos.zampieri@rit.edu`
[3] School of Computing and Digital Technology, Birmingham City University, UK
`hansi.hettiarachchi@mail.bcu.ac.uk`

Abstract. In this paper, we describe the BRUMS entry to the Hate Speech and Offensive Content Identification in Indo-European Languages (HASOC) shared task 2019. The HASOC organizers provided participants with annotated datasets containing posts from social media in English, German, and Hindi (including code-mixing). We present a multilingual deep learning model to identify hate speech and offensive language in social media. Our best performing system was ranked 3rd among 79 entries in the English track of the HASOC sub-task 1.

Keywords: Offensive Language Identification · Hate Speech · Text Classification · Deep Learning.

1 Introduction

The various forms of abusive and offensive content online pose risks to the users of social media platforms. One such example is cyberbulling which has been linked to depression and suicide risk among teenagers [2]. As offensive language becomes pervasive in social media, scholars and companies have been working on developing systems capable of identifying offensive posts, which can be set aside for human moderation or permanently deleted [22]. The use of robust NLP methods in these systems is paramount to cope with the many ways such content can be presented. While direct abuse and insults containing profanity are fairly easy to be identified, recognizing indirect insults for example, which often include metaphors and sarcasm, are a challenge to human annotators and, as a consequence, to most state-of-the-art systems [16].

In light of these important challenges, a recent growing interest of the NLP community in detecting abusive and offensive content online has been observed. This is evidenced by previous work focusing on the identification of abusive content [19, 8], aggression [14], cyberbullying [4, 32], hate speech [6, 15, 17], and offensive language [30]. Along with these studies, a few shared tasks have been organized on these topics, such as HatEval [1] and OffensEval [34] co-located with SemEval 2019.

This paper revisits the problem of offensive language identification describing our submission to the sub-task A of the Hate Speech and Offensive Content Identification in Indo-European Languages (HASOC) shared task [18].[4] The remainder of this paper is structured as follows: Section 2 describes the system that was submitted, split into a description of the dataset (Section 2.1), how the data was processed (Section 2.2) and the architecture of the classifier that was used (Section 2.3). Section 3 presents an analysis of the results of our evaluation of the five different architectures (Section 3.1), as well as of the final submission (Section 3.2). Finally, Section 4 offers some final remarks and a conclusion.

2 Methods and Data

This section describes the shared task data, as well as the BRUMS system that was used in the competition. As HASOC is a multilingual competition, we use only minimal preprocessing methods to make the system portable to all languages in the dataset. For classification, we used and compared seven different neural network architectures suited to this task. Our implementation has been made available on Github.[5]

2.1 Dataset

The multilingual HASOC dataset has posts from Facebook and Twitter and it is distributed in tab separated format. The dataset contains posts written in 3 languages: German, English and code-mixed Hindi. The size of the training set is approximately 8,000 posts for each language and the test data contains approximately 1,000 posts for each language. The dataset is annotated using a three-level hierarchical model similar to the proposed in the annotation of the OLID datset [33]. In HASOC, each layer corresponds to one sub-task in the competition. We participate in sub-task A which focus on hate speech and offensive language identification. It is a binary classification task in which the goal is to develop systems able to classify tweets into two classes, namely: Hate and Offensive (HOF) and Non- Hate and offensive (NOT).

2.2 Text Preprocessing

As mentioned previously, the data preprocessing for this task was kept fairly minimal to make it portable for all the languages. More specifically, we perform

[4] https://hasoc2019.github.io/

[5] https://github.com/TharinduDR/HASOC-2019

only three specialised tasks for this data, followed by tokenisation. The tasks include removing usernames, removing *URL*s, and converting all tokens to lower case.

First, we completely remove all usernames from the texts, without inserting a placeholder. This is carried out by removing all strings beginning with the @ symbol, as this is how usernames are denoted on Twitter. The reasoning behind this step is mainly to remove noisy text, as it is highly unlikely that there would be any embeddings for the usernames. In addition to that, we believe that these usernames don't add much semantic meaning. Moreover, if, for instance, a majority of offensive tweets were written by one user, this could lead to bias in the system against one user.

After that, we remove all the *URL*s from the texts. All the tweets contain an *URL* which also refers to the *URL* of the particular tweet. All these *URL*s start with https://t.co/ followed by an unique ID. Therefore, we used a regular expression to remove all strings beginning with the *https://t.co/*. Similar to the usernames, *URL*s too do not add any semantic meaning and can be considered as noisy.

Final prepossessing step is only applied to the architectures that used character embeddings. The fastText pretrained character embedding models that we used only contain lower-cased letters. Therefore, we convert the text to lower case letters. However, the BERT models we used are cased. This preprocessing step was not used to the BERT based architecture.

2.3 Neural Network Architectures

For the first six architectures, after text processing we used fastText character embeddings [3] to encode text. The encoded tweets are then classified by one of the neural network architectures. We evaluated six different neural network architectures for the classification tasks: pooled Gated Recurrent Unit (GRU) (Section 2.3.1), Long Short-Term Memory (LSTM) and GRU with Attention (Section 2.3.3), 2D Convolution with Pooling (Section 2.3.4), GRU with Capsule (Section 2.3.5) and LSTM with Capsule and Attention (Section 2.3.6). The parameters of each architecture were optimised using 5-fold cross-validation considering binary cross entropy loss function and using adam optimiser [12]. We used the reducing learning rate on plateau technique when a deep learning architecture stopped improving. Deep learning architectures often benefit from reducing the learning rate by a factor once learning stagnates [21]. We monitored validation macro F1 score and if no improvement was seen for 2 epochs, the learning rate was reduced by a factor of 0.6, since this value seemed to offer the best improvement. These architectures were successfully applied to a number of classification tasks such as GRU with pooling for sequence labeling [5], GRU with capsule for toponym detection [20], and their success in these tasks inspired us to use them for the task at hand.

As the last architecture we fine-tuned BERT [7] model. For the English task, we used whole word masking variant of *BERT-Large* [6]. For the German and Hindi tasks, we used *BERT-Base, Multilingual Cased* model, which has been trained with 104 languages including German and Hindi. *BERT-Large* model has been widely used in text classification tasks like sentiment analysis [26] which motivated us to use it for aggression detection task.

2.3.1 Pooled GRU In this architecture, after the embedding layer, embedding vectors are fed to the bi-directional GRU [5] at their respective timestep. The bi-directional GRU-layer has 80 units. The final timestep output is fed into a max pooling layer and an average pooling layer in parallel [23]. After this, the outputs of the two pooling layers are concatenated and connected to a dense layer [11] activated with a sigmoid function. Additionally, there is a spatial dropout [27] between the embedding layer and the bi-directional GRU layer to avoid over-fitting. This architecture has been discussed in [13] as a common architecture to perform text classification tasks.

2.3.2 Stacked LSTM with Attention In this architecture, each of the embedding vectors are fed into a bi-directional LSTM-layer [24]. The output of this layer is again fed into a bi-directional LSTM-layer [24] with self attention [28]. Each of the bi-directional LSTM-layers has 64 units. Finally, the output is connected to two dense layers that are [11] activated first with a relu function, and then with a sigmoid function. We adopted this architecture from the *Toxic Comment Classification Challenge* in Kaggle[7].

2.3.3 LSTM and GRU with Attention With this architecture, the output of the embedding layer goes through a spatial dropout [27] and is then fed in parallel to a bi-directional LSTM-layer [24] with self attention and a bi-directional GRU-layer [5] with self attention [28]. Both the bi-directional LSTM-layer and the bi-directional GRU-layer have 40 units. The output from the bi-directional GRU-layer is fed into an average pooling layer and a max pooling layer. The output from these layers and the output of the bi-directional LSTM-layer are concatenated and connected to a dense layer with ReLU activation. After that, a dropout [25] is applied to the output and connected to a dense layer activated with a sigmoid function.

2.3.4 2D Convolution with Pooling The fourth architecture takes a different approach than the previous architectures by using 2D convolution layers [31], rather than LSTM or GRU layers. The outputs of the embedding layer are connected to four 2D convolution layers [31], each with max pooling layers. All the 2D convolution layers were initialised with normal kernel initialiser. The

[6] https://github.com/google-research/bert
[7] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

outputs of these are concatenated and connected to a dense layer activated with a sigmoid function after applying a dropout [25]. This architecture has been used in the *Quora Insincere Questions Classification* Kaggle competition[8].

2.3.5 GRU with Capsule Most of the previous architectures rely on a pooling layer. However, this architecture uses a capsule layer [10] rather than pooling layers. After applying a spatial dropout [27] the output of the embedding layer is fed into a bi-directional GRU-layer [5]. The bi-directional GRU-layer has 100 units and was initialised with the Glorot normal kernel initialiser and orthogonal recurrent initialiser with 1.0 gain. The output is then connected to a capsule layer [10]. The output of the capsule layer is flattened and connected to a dense layer with ReLU activation, a dropout [25] and batch normalisation applied, and re-connected to a dense layer with sigmoid activation. This architecture has been used to detect locations within word windows [9].

2.3.6 LSTM with Capsule and Attention This architecture uses combination of a capsule layer [10] and a self attention layer [28]. After the embedding layer a spatial dropout [27] is applied to the output, which is then fed into a bi-directional LSTM-layer [24] with 80 units. The layer is initialised with the Glorot normal kernel initialiser and orthogonal recurrent initialiser with 1.0 gain. The output of the bi-directional LSTM-layer is fed into a capsule layer and to a self attention layer in parallel. Then each output of both capsule layers and the self attention layer goes through a DropConnect [29]. They are concatenated before connecting to a dense layer with sigmoid activation. This architecture has been used in the *Jigsaw Unintended Bias in Toxicity Classification* competition.[9]

2.3.7 BERT for Text Classification For this method we used a different approach than the above architectures [7]. All of the above architectures use fastText character embeddings [3] feed in to neural network. In more details, a pre-trained neural network produces character embeddings which are then used as features in NLP models. In this approach we use a different approach with transfer learning - pre-training a neural network model on a known task, and then performing fine-tuning — using the trained neural network as the basis of a new purpose-specific model. The known task that the BERT uses is next sentence prediction. In the BERT training process, the model receives pairs of sentences as input and learns to predict if the second sentence in the pair is the subsequent sentence in the original document. Classification tasks can be done similarly to Next Sentence classification, by adding a classification layer on top of the Transformer output for the token. We used this particular method for this aggression detection task.

[8] https://www.kaggle.com/c/quora-insincere-questions-classification
[9] https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification

3 Results

This section presents the results of the evaluation of the six architectures, as well as the evaluation of the final submissions. We compare the performance of above neural network architectures in order to select submissions for each language.

3.1 Architecture Evaluation

This section describes how we selected the architecture for the final submission in each language. To evaluate the architectures, we used 20% of the available training data, and used the rest of the data for training. Table 1 shows the evaluation results of each architecture. We used two evaluation metrics: macro-averaged F1-score and weighted F1-score score as denoted in the table 1.

Table 1. Results of the architectures.

Architecture	English		German		Hindi	
	Macro F1	Weighted F1	Macro F1	Weighted F1	Macro F1	Weighted F1
Pooled GRU	0.7421	0.7979	0.5741	0.7662	0.7991	0.7886
Stacked LSTM with Attention	0.7005	0.7504	0.5411	0.7389	0.7666	0.7742
LSTM and GRU with Attention	0.7205	0.7675	0.5456	0.7451	0.7436	0.7656
2D Convolution with Pooling	0.7516	0.8116	0.5772	0.7775	0.8011	0.7996
GRU with Capsule	0.7218	0.7781	0.5725	0.7552	0.7882	0.7776
LSTM with Capsule and Attention	0.6865	0.7279	0.5332	0.7210	0.7333	0.7542
BERT	**0.7891**	**0.8418**	**0.5881**	**0.7871**	**0.8025**	**0.8030**

As shown in the table 1, BERT based architecture had the best F1 scores from the seven experimented architectures for all the languages. Therefore, we submitted output from the BERT architecture as our final submission.

3.2 Submission Results

This section presents the results of the evaluation of our submission. The evaluation was carried out by the task organisers, and at the time of writing the paper the GOLD standards of the test set is not available. Therefore, we report only the evaluation provided to us by the task organisers which is solely based on F1-scores. Our submission had macro-averaged F1-scores 0.7694, 0.5464 and 0.8025 for English, German and Hindi respectively. Furthermore, our BERT based submission had weighted F1 0.8379, 0.7870, 0.8030 for English, German and Hindi respectively. According to the results presented in [18], our best performing system was ranked 3rd among 79 entries in the English track of the HASOC sub-task 1.

4 Conclusion

In this paper, we have presented the BRUMS system for identifying offensive language in tweets and Facebook posts in German, English, and Hindi. The system uses minimal preprocessing, and relies on word and context embeddings. We experimented with different deep neural network architectures in order to determine the most suitable for this task. According to our evaluation, and the results provided by the task organisers, it is clear that fine tuning BERT architecture scores highest overall.

Due to non language-specific preprocessing, our system performs well in all the three languages. Compared to the performance obtained by the other participants of the HASOC sub-task 1, our model achieves its best results on English data ranking 3rd among 79 entries. We are interested in investigating the performance of our system in other sub-tasks too. In the future, we would like to see how to extend this system to other languages as well as similar tasks and datasets.

Acknowledgements

We would like to thank the HASOC shared task organizers for organizing this interesting shared task and for replying promptly to all our inquiries. We further thank the anonymous reviewers for their feedback.

References

1. Basile, V., Bosco, C., Fersini, E., Nozza, D., Patti, V., Pardo, F.M.R., Rosso, P., Sanguinetti, M.: 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. pp. 54–63 (2019)
2. Bauman, S., Toomey, R.B., Walker, J.L.: Associations among bullying, cyberbullying, and suicide in high school students. Journal of adolescence **36**(2), 341–350 (2013)
3. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606 (2016)
4. Chelmis, C., Zois, D.S., Yao, M.: Mining patterns of cyberbullying on twitter. In: 2017 IEEE International Conference on Data Mining Workshops (ICDMW). pp. 126–133. IEEE (2017)
5. Chung, J., Çaglar Gülçehre, Cho, K., Bengio, Y.: Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. CoRR (2014)
6. Davidson, T., Warmsley, D., Macy, M., Weber, I.: Automated Hate Speech Detection and the Problem of Offensive Language. In: Proceedings of ICWSM (2017)

7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (2018)
8. Founta, A.M., Djouvas, C., Chatzakou, D., Leontiadis, I., Blackburn, J., Stringhini, G., Vakali, A., Sirivianos, M., Kourtellis, N.: Large scale Crowdsourcing and Characterization of Twitter Abusive Behavior. In: Twelfth International AAAI Conference on Web and Social Media (2018)
9. Hettiarachchi, H., Ranasinghe, T.: Emoji powered capsule network to detect type and target of offensive posts in social media. In: Proc. of the Recent Advances in Natural Language Processing (RANLP) (2019)
10. Hinton, G.E., Sabour, S., Frosst, N.: Matrix capsules with EM routing. In: Proceedings of ICLR 2018 (2018)
11. Huang, G., Liu, Z., Weinberger, K.Q.: Densely Connected Convolutional Networks. Proceedings of IEEE CVPR 2017 (2017)
12. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. Proceedings of CoRR 2015 (2015)
13. Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L.E., Brown, D.E.: Text Classification Algorithms: A Survey. Information **10**(4) (2019)
14. Kumar, R., Ojha, A.K., Malmasi, S., Zampieri, M.: Benchmarking Aggression Identification in Social Media. In: Proceedings of TRAC (2018)
15. Malmasi, S., Zampieri, M.: Detecting Hate Speech in Social Media. In: Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP). pp. 467–472 (2017)
16. Malmasi, S., Zampieri, M.: Challenges in Discriminating Profanity from Hate Speech. Journal of Experimental & Theoretical Artificial Intelligence **30**, 1 – 16 (2018)
17. Mathew, B., Illendula, A., Saha, P., Sarkar, S., Goyal, P., Mukherjee, A.: Temporal effects of unmoderated hate speech in gab. arXiv preprint arXiv:1909.10966 (2019)
18. Modha, S., Mandl, T., Majumder, P., Patel, D.: Overview of the HASOC track at FIRE 2019: Hate Speech and Offensive Content Identification in Indo-European Languages. In: Proceedings of the 11th annual meeting of the Forum for Information Retrieval Evaluation (2019)
19. Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., Chang, Y.: Abusive Language Detection in Online User Content. In: Proceedings of WWW (2016)
20. Plum, A., Ranasinghe, T., Calleja, P., Orasan, C., Mitkov, R.: RGCL-WLV at SemEval-2019 Task 12: Toponym Detection. In: Proceedings of SemEval-2019 (2019)
21. Ravaut, M., Gorti, S.: Gradient descent revisited via an adaptive online learning rate. arXiv preprint arXiv:1801.09136 (2018)
22. Risch, J., Krestel, R.: Delete or not Delete? Semi-Automatic Comment Moderation for the Newsroom. In: Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018). pp. 166–176 (2018)
23. Scherer, D., Müller, A.C., Behnke, S.: Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition. In: Proceedings of ICANN 2010 (2010)
24. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Signal Processing **45**, 2673–2681 (1997)
25. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research **15**, 1929–1958 (2014)
26. Sun, C., Qiu, X., Xu, Y., Huang, X.: How to fine-tune bert for text classification? ArXiv **abs/1905.05583** (2019)

27. Tompson, J., Goroshin, R., Jain, A., LeCun, Y., Bregler, C.: Efficient object localization using Convolutional Networks. Proceedings of IEEE CVPR 2015 (2015)
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention Is All You Need. In: Proceedings of NIPS (2017)
29. Wan, L., Zeiler, M.D., Zhang, S., LeCun, Y., Fergus, R.: Regularization of Neural Networks using DropConnect. In: ICML (2013)
30. Wiegand, M., Siegel, M., Ruppenhofer, J.: Overview of the GermEval 2018 Shared Task on the Identification of Offensive Language. In: Proceedings of GermEval (2018)
31. Wu, Y., Yang, F., Liu, Y., Zha, X., Yuan, S.: A Comparison of 1-D and 2-D Deep Convolutional Neural Networks in ECG Classification. In: Proceedings of IEEE Engineering in Medicine and Biology Society (2018)
32. Yao, M., Chelmis, C., Zois, D.S.: Cyberbullying detection on instagram with optimal online feature selection. In: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). pp. 401–408. IEEE (2018)
33. Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., Kumar, R.: 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: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 1415–1420 (2019)
34. Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., Kumar, R.: SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). In: Proceedings of The 13th International Workshop on Semantic Evaluation (SemEval) (2019)