Bagging BERT Models for Robust Aggression Identification

Julian Risch and Ralf Krestel
Hasso Plattner Institute, University of Potsdam
Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam, Germany
julian.risch@hpi.de, ralf.krestel@hpi.de

Abstract

Modern transformer-based models with hundreds of millions of parameters, such as BERT, achieve impressive results at text classification tasks. This also holds for aggression identification and offensive language detection, where deep learning approaches consistently outperform less complex models, such as decision trees. While the complex models fit training data well (low bias), they also come with an unwanted high variance. Especially when fine-tuning them on small datasets, the classification performance varies significantly for slightly different training data. To overcome the high variance and provide more robust predictions, we propose an ensemble of multiple fine-tuned BERT models based on bootstrap aggregating (bagging). In this paper, we describe such an ensemble system and present our submission to the shared tasks on aggression identification 2020 (team name: Julian). Our submission is the best-performing system for five out of six subtasks. For example, we achieve a weighted F1-score of 80.3% for task A on the test dataset of English social media posts. In our experiments, we compare different model configurations and vary the number of models used in the ensemble. We find that the F1-score drastically increases when ensembling up to 15 models, but the returns diminish for more models.

Keywords: neural networks, offensive language, aggression, hate speech, ensemble learning, transformer model, BERT

1. Robust Aggression Identification

Aggression in social media posts, such as tweets or Facebook posts, has become omnipresent. Ignoring it is inappropriate because it can inflict real damage in real-world life (Hsueh et al., 2015; Rösner et al., 2016). Text classification approaches can detect such malicious behavior, and more fine-grained classifications can identify subclasses of aggression, for example, different severity levels or target groups (Zampieri et al., 2019a). These classifiers alone cannot solve the problem of online aggression because they do not reach its root cause — the attackers behind aggressive posts. However, they still play an essential role in combating aggression by supporting content moderators, who remove these posts from online platforms or criminal prosecutors, who hold attackers accountable.

The current trend for research on natural language processing with deep neural networks is to develop more and more complex models. The complexity is expressed in the number of parameters, which is in the hundreds of millions for transformer-based language models, such as Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018). More precisely, *large* BERT models span 24 layers and 340 million parameters, and even *base* BERT models span 12 layers and 110 million parameters. Typically, these models are pre-trained on large corpora, for example, on collections of web pages with billions of tokens. For down-stream tasks, e.g., text classification, they are fine-tuned on smaller datasets. While the pre-training is unsupervised, the fine-tuning for down-stream tasks is typically supervised learning.

The fine-tuning fits the model well to the labeled training data, and the model's bias is typically low. It does not suffer from underfitting. The strong classification performance reported on training, validation, and test datasets proves this. In fact, overfitting can be more of an issue, especially for smaller datasets. The number of parameters is much larger than the typical number of samples in hand-labeled datasets. Standard regularization techniques, such as dropout and limiting the number of training steps with early stopping, can be used to cope with overfitting problems.

However, the model's variance is high. Even slight variations in the input data or a slight change of the random seed, which affects, for example, the randomly initialized weights of the final prediction layer (prediction head) result in large changes in classification performance. In our initial experiments, we find that the performance varies in a range of up to five percentage points in F1-score.

Contributions. We address the issue of high variance of fine-tuned BERT models on small datasets with an ensembling approach. To this end, we propose to combine the predictions of multiple BERT models that are trained with bootstrap aggregating on slightly differing training datasets and with varying weight initialization in the final prediction layer. Our experiments show that an ensemble achieves a two percentage points higher F1-score than single models. Further, we optimize the number of ensembled models and find that the performance increases for up to 15 models and stays the same for larger ensembles.

Outline. The rest of this paper is structured as follows: In Section 2, we give an overview of related shared tasks and transformer-based neural networks. We then briefly introduce the dataset and point out the imbalanced class distribution in Section 3. Further, the training procedure for the BERT models and the ensembling technique is described in the same section. Our experiments in Section 4 evaluate the F1-score on the validation and test datasets, and we describe the model configurations that achieved the best results. An additional experiment studies how the number of ensembled models affects the classification performance. In Section 5, we discuss the results and analyze misclassi-

fications based on confusion matrices before we conclude with directions for future work in Section 6.

2. Related Work

The last three years came with a variety of shared tasks in the broad field of aggression identification. We give an overview of these tasks in the following. Afterward, we summarize related work on transformer neural networks and ensembles for aggression identification since we combine both techniques in our approach.

Shared Tasks. The by far largest shared task concerning the number of participants and the dataset size is the Kaggle challenge on toxic comment classification.[1] The dataset comprises English user comments from Wikipedia discussion pages. Thanks to a large number of shared tasks in conference workshops, labeled datasets cover a diverse set of languages besides English. For example, there is Spanish (Fersini et al., 2018), Italian (Bosco et al., 2018), Hindi (Kumar et al., 2018a; Bhattacharya et al., 2020), Bangla (Bhattacharya et al., 2020), German (Wiegand et al., 2018; Struß et al., 2019), and Arabic, Danish, Greek, and Turkish (Zampieri et al., 2020).

The shared tasks differ not only in language but also in the precise task and respective class labels. For example, HatEval deals with hate speech against immigrants and women (Basile et al., 2019), HaSpeeDe with hate speech detection in general (Bosco et al., 2018), IberEval has a task on automatic misogyny identification (Fersini et al., 2018), OffensEval covers offensive language (Zampieri et al., 2019b), and TRAC focuses on aggression (Kumar et al., 2018a; Bhattacharya et al., 2020). To the best of our knowledge, there is no common definition for the task of identifying aggressive or otherwise offensive social media posts. Instead, the different shared tasks use varying terminology: hate speech, toxic comments, offensive language, abusive language, aggression, and misogyny identification. Waseem et al. (2017) provide an overview of abusive language detection subtasks.

Transformer Models. Our approach builds on Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018). BERT is a task-agnostic language representation model, which consists of multiple layers of bidirectional transformers by Vaswani et al. (2017). After being pre-trained on a large corpus, it can not only be fine-tuned for text classification but also for many other tasks, such as named entity recognition, question answering, and text summarization. The training objective of the model uses a masking technique. Given a sentence, 15% of the input tokens are masked, and the task is to predict these tokens. This technique overcomes the limitation of unidirectional processing and is also superior to language models that combine right-to-left and left-to-right processing (Peters et al., 2018). Our implementation uses the Python-based framework for adapting representation models (FARM) by *deepset*.[2]

BERT has been used in other shared tasks on hate speech or offensive language detection (Mozafari et al., 2019; Nikolov and Radivchev, 2019). We first published the idea of ensembling multiple BERT models in the context of a shared task on offensive language detection for German tweets (Risch et al., 2019). However, our experiments in this previous publication only show that ensembles of five or ten BERT models outperform a single model. It does not answer what the optimal number of models in such an ensemble is.

Our submission to the last edition of the aggression identification shared task in 2018 uses another ensembling technique: stacking (Risch and Krestel, 2018). The predictions of bidirectional recurrent neural networks and logistic regression classifiers are weighted for each social media post individually. Depending on features extracted from the post, such as its text length or the number of out-of-vocabulary words, one or the other classifier's predictions are emphasized. Thereby, we account for the fact that individual classifiers are specialized to make predictions for longer or shorter posts, for example. The difference to the bootstrap aggregating approach is that the goal was not to reduce variance but to combine classifiers that were trained on different features (word embeddings, character n-grams). On the English dataset, the best single model achieves an F1-score of 58% and the ensemble 61% for English. The results on the Hindi dataset are similar (best single model: 61% and ensemble: 63%).

3. Bootstrap Aggregating BERT Models

This section presents our approach for the shared task. It begins with a brief description of the task dataset and further describes the classification model, the training procedure, and the ensembling strategy. The Python code for our submission is publicly available online.[3]

3.1. Dataset

The shared task[4] is based on three datasets: an English, a Hindi, and a Bangla dataset of about 6000 social media posts each (Kumar et al., 2020). It comprises two independent tasks. The first task, task A: aggression identification, is a 3-way classification into non-aggressive (NAG), covertly aggressive (CAG), and overtly aggressive (OAG) posts. Covertly aggressive posts include indirect attacks that use, e.g., satire or rhetorical questions, while overtly aggressive posts contain lexical features that are considered aggressive (Kumar et al., 2018b). Table 1 gives an overview of the dataset sizes for this task. The second task, task B: misogynistic aggression identification, is a binary classification task with two labels: gendered (GEN) and non-gendered (NGEN). Gendered aggression is defined as attacks based on gender (roles), and includes homophobic and transgender attacks (Kumar et al., 2018b). Table 2 gives an overview of the dataset sizes for this task. Figure 1 and Figure 2 list one English-language example post per class label.

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

[2] https://github.com/deepset-ai/FARM

[3] https://hpi.de/naumann/projects/repeatability/text-mining.html

[4] https://sites.google.com/view/trac2/shared-task

Table 1: Training, validation, and test dataset sizes for task A per language.

	Training			Validation			Test			Total		
	NAG	CAG	OAG	NAG	CAG	OAG	NAG	CAG	OAG	NAG	CAG	OAG
English	3375	453	435	836	117	113	690	224	286	4901	794	834
Hindi	2245	910	829	578	211	208	325	191	684	3148	1312	1721
Bangla	2078	898	850	522	218	217	712	225	251	3312	1341	1318

Table 2: Training, validation, and test dataset sizes for task B per language.

	Training		Validation		Test		Total	
	NGEN	GEN	NGEN	GEN	NGEN	GEN	NGEN	GEN
English	3954	309	993	73	1025	175	5972	557
Hindi	3323	661	845	152	633	567	4801	1380
Bangla	3114	712	766	191	986	202	4866	1105

text: Great video😘😘😘
tokens: Great, video, `[UNK]`, `[UNK]`, `[UNK]`
label: non-aggressive (NAG)

RSS agenda is to demolished opposite options
tokens: RS, ##S, agenda, is, to, demolished, opposite, options
label: covertly aggressive (CAG)

You are soo fucked up that you can't understand someone else's perspective...
tokens: You, are, so, ##o, fucked, up, that, you, can, ', t, understand, someone, else, ', s, perspective, ., ., .
label: overtly aggressive (OAG)

Figure 1: Training samples for task A (aggression identification).

text: I think feminists are lesbians,OAG,GEN
tokens: I, think, feminist, ##s, are, lesbian, ##s
label: gendered (GEN)

text: kill all those womens who file faje rape and dowry cases,CAG,NGEN
tokens: kill, all, those, women, ##s, who, file, f, ##aj, ##e, rape, and, do, ##wry, cases
label: non-gendered (NGEN)

Figure 2: Training samples for task B (misogynistic aggression identification).

3.2. Classification Model

The tokenizer for BERT uses word pieces so that the model learns an embedding for each token. The vocabulary consists of 30,000 tokens. Custom tokens can be added to extend this vocabulary, but then there is no pre-trained representation for the added tokens. A larger dataset than the one provided for this task is needed to make proper use of custom tokens.

We refrain from any complex data pre-processing and use only three small steps. First, all characters are converted to lowercase. Second, we insert whitespaces before and after every emoji so that they can be tokenized as separate tokens. Third, we limit the sequence length to 200 tokens. The sequence length defines how many tokens are cut off from overly long sequences. Only a few posts are affected by this choice. With a maximum sequence length of 200 tokens, 0.9% of all training samples are affected. A maximum sequence length of 220 or 230 tokens reduces this number to 0.5%.

The tokenizer is the same as used for pre-training the BERT model. For this reason, emojis and non-Latin characters are unknown tokens, which are replaced with a common `[UNK]` symbol. Without inserting whitespace around emojis, the example post "Great video😘😘😘" would be tokenized as "Great, `[UNK]`". With our pre-processing, it is tokenized as "Great, video, `[UNK]`, `[UNK]`, `[UNK]`". On the word embedding level, we use a dropout of 10%, which means that every tenth word is randomly removed from the input to regularize the model.

We use the BERT *base* model, which has 768 hidden units.[5] Therefore, the final prediction layer is a dense layer with softmax activation that maps the 768-dimensional vectors to three outputs for the multi-class classification and to two outputs for the binary classification.

3.3. Training Procedure

We train each model for up to ten training epochs and halt the training if no learning progress is made for two subsequent evaluation periods. This early stopping mechanism monitors the weighted F1-score on a 10% validation set. An evaluation on this set runs every 40 batches. With a batch size of 48, there are approximately two evaluations per epoch.

Each training process starts with a different random seed. Thereby, not only does the random initialization of the weights of the final prediction layer vary among the models, but also the random data split for the early stopping is chosen differently. As the loss function, we use cross-

[5] `https://huggingface.co/bert-base-uncased`

entropy loss weighted by the class distribution. The learning rate is set to $5 \cdot 10^{-5}$ but uses a warmup phase as it is standard for fine-tuning BERT models. We use a linear learning rate warmup for the first 30% of the training up to the rate of $5 \cdot 10^{-5}$. Afterward, the rate linearly decays until the end of the training (ten epochs max). Deviations from this general configuration for different runs of our approach are described in Section 4.

3.4. Ensembling Strategy

The motivation for our ensembling approach is the instability of the classification performance across different fine-tuning runs of the same model. For example, Devlin et al. report[6] that the accuracy on small datasets, such as the Microsoft Research Paraphrase Corpus (MRPC) with 3,600 samples varies between 84% and 88%. This variance occurs when fine-tuning even the exact same pre-trained model. The recommended approach is to restart the fine-tuning multiple times. When fine-tuning BERT models on the shared task dataset, we are confronted with the same varying classification performance. Slight changes to the training data and model hyperparameters, such as the random seed, cause the fine-tuned models to achieve very different results on the hold-out test dataset. These models only differ in the model weights in the final dense layer (the prediction head) when the training starts. In summary, the BERT models that are fine-tuned on the small shared task dataset are unstable and have a high variance.

Our ensembling strategy is a variance reduction technique: bootstrap aggregation (bagging). We train up to 25 BERT models of the same kind on slightly different subsets of the data. A soft majority voting combines the predictions of these models:

$$\hat{y} = \underset{j}{\operatorname{argmax}} \sum_{i=1}^{n} p_{i,j}$$

where $p_{i,j}$ is the probability for class label j predicted by the i-th classifier (out of n classifiers). It sums up the probability mass assigned per class label and chooses the label with the highest probability as the ensemble's prediction. In other words, it chooses the class label that is most likely predicted. In contrast to that, a hard majority voting would choose the label that is most often predicted.

4. Evaluation

We evaluate our approach for both shared tasks on the test dataset and report the best model configurations. Two additional experiments study how the ensembling affects classification performance. The first experiment shows how many models should be ensembled to achieve the best performance. The second experiment is an ablation study to find out whether the random data splits or the random weight initialization cause the ensemble's superior performance compared to single models.

4.1. Shared Task Performance

The shared task uses the weighted F1-score for the evaluation. As a consequence, the score for the majority class is more important than for the other classes. Table 3 lists the performance that our approach achieved on the test dataset. In five out of six tasks, our approach outperforms all other shared task participants (15 teams). The only exception is the English-language version of task B. We believe the inferior results of our model for this task are caused by using a case-sensitive BERT model. For all other tasks, we used case-agnostic BERT models, which outperform the case-sensitive ones.

The largest gap to the second-best submission is at the English-language version of task A. Our approach achieves a 4.4 percentage points better F1-Score than the second-best approach.

Table 4 lists the model configurations that achieved the best results on the test dataset. Note that the number of submissions for the test dataset was limited to three per task and language. Therefore, we can evaluate only a small set of different configurations. This limitation is also the reason why we can only assume that a case-agnostic BERT model would achieve a higher F1-score for the English version of task B than the case-sensitive model that we used for our submission. We did not submit the predictions of such a case-sensitive model due to the limited number of allowed submissions.

4.2. Optimizing the Number of BERT Models

With the following experiment, we study how many models should be included in the ensemble to achieve the highest weighted F1-score at the shared task. To this end, we fine-tune 100 BERT models that only differ in the initial random seed. All these models have the same architecture and the same hyperparameters, such as batch size or learning rate. However, the varying seed determines the randomly initialized weights for the final dense layer of the model (the prediction head), the order in which the training samples are processed, their distribution among the training batches, and finally, the 90% training and 10% percent validation split.

For each number from 1 to 50, which we call ensemble size, we select subsets of the 50 fine-tuned models of that size. For example, to build an ensemble of 50 models out of 100 trained models, there are $\binom{100}{50} \approx 10^{29}$ possible combinations. As we cannot evaluate that many combinations, we randomly sample 1 000 combinations per ensemble size. The ensemble's predictions are generated with soft majority voting. Each ensemble is then evaluated on the exact same hold-out test dataset.

The top line in Figure 3 (random dataset split, random weight initialization) shows the weighted F1-scores that are achieved on average across the 1 000 combinations per ensemble size. The score increases for ensembles of up to 10 to 15 models, after which the advantage of adding even more models diminishes. The performance of a single model is, on average, about four percentage points worse than the best ensemble. We could not use the official test dataset for our experiment. Therefore, we use the official validation dataset for the evaluation and 90% of the official training dataset for training. 10% of the training dataset are used for the early stopping mechanism. The model seems to underfit because this mechanism halts the training too

[6] https://github.com/google-research/bert/blob/master/README.md

Table 3: Weighted F1-score (in percent) on the test dataset. Our approach outperforms the best submission by other teams in five out of six subtasks.

	English		Hindi		Bangla	
	Task A	Task B	Task A	Task B	Task A	Task B
Our Submission	**80.29**	85.14	**81.28**	**87.81**	**82.19**	**93.85**
Best Other Submission	75.92	**87.16**	79.44	86.89	80.83	92.97

Table 4: Configurations of our best-performing submissions on the test dataset.

	English		Hindi		Bangla	
	Task A	Task B	Task A	Task B	Task A	Task B
Language of models	English	English	multilingual	multilingual	multilingual	multilingual
Number of models	20	25	15	15	15	25
Letter casing	uncased	cased	uncased	uncased	uncased	uncased
Sequence length	220	220	200	200	200	230
Cross entropy loss	weighted	weighted	non-weighted	weighted	weighted	weighted
Hold-out data	10%	10%	20%	10%	20%	10%
Patience	2	2	1	2	1	2

early on the smaller dataset.

This experiment — in particular the fine-tuning of 100 BERT models and combining and evaluating the predictions of thousands of subsets of these models — is computationally expensive. It took approximately seven hours on two Nvidia GeForce GTX 1080 Ti GPUs with 11GB memory to complete the experiment. Training time and inference time increase linearly with the ensemble size.

4.3. Ablation Study

This experiment studies whether training on slightly different subsets of data or differently initializing weights in the final prediction layer (prediction head) causes the ensemble's strong performance. Our hypothesis is that the reason is the weight initialization. To test this hypothesis, we compare four different variations of our approach. Figure 3 shows the weighted F1-scores for all four variations per ensemble size.

First, we vary not only the random seeds for the weight initialization but also the training and validation split. As a consequence, the training data of the models differ slightly. Second, we vary the random seeds for the weight initialization while using the exact same training and validation split. For this variation, all models are trained on the exact same training data. Third, we use the same weight initialization for all models but vary the random splits of training and validation data. Fourth, we keep both the weight initialization and data splits fixed across all models. In the fourth variation, all trained models are identical, and thus, ensembling does not improve the performance. The test set is the exact same in all four variations.

The plot in Figure 3 confirms our hypothesis. The strong performance of our ensembles is mainly caused by using varying weight initializations for the individual models. The varying training and validation dataset splits have a smaller effect.

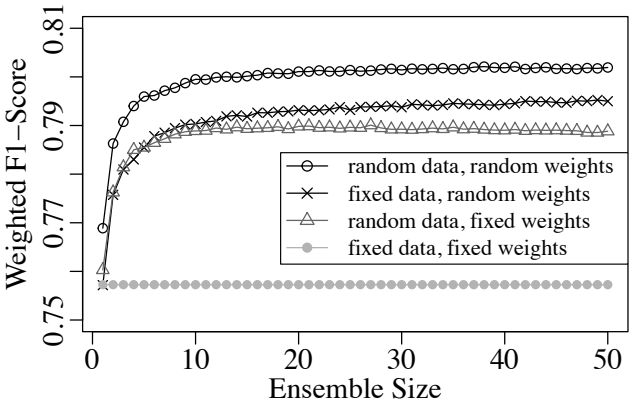

Figure 3: The increased performance of an ensemble of BERT models is mainly due to random weight initialization rather than random splits of training and validation data.

5. Discussion

Figure 4, Figure 5, and Figure 6 show normalized confusion matrices for task A on the test datasets. For task A on the English test dataset, the most frequent (with regard to relative numbers) misclassification is predicting *CAG* instead of *OAG* (28% of all posts labeled as *OAG*). On the Hindi dataset, *NAG* is more frequently misclassified as *CAG* (23% of all posts labeled as *NAG*). On the Bangla dataset, *CAG* is most often misclassified as *NAG* (31% of all posts labeled as *CAG*). For all three languages, *NAG* and *CAG* are often mixed up, and the same holds for *CAG* and *OAG*. This result is not to our surprise as *NAG* is more similar to *CAG* than to *OAG* and *OAG* is more similar to *CAG* than to *NAG*. A non-aggressive post is easier to distinguish from an overtly aggressive post than from a covertly aggressive one.

A weakness of our approach is the vocabulary of the BERT models. First, the meaning of emojis is ignored, and they are tokenized as unknown symbols, although they frequently occur in the dataset. For example, 😂 is the most frequent emoji in the English training dataset (488 occur-

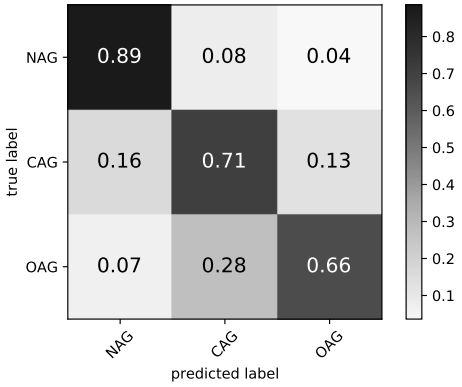

Figure 4: Confusion matrix for task A on the English test dataset.

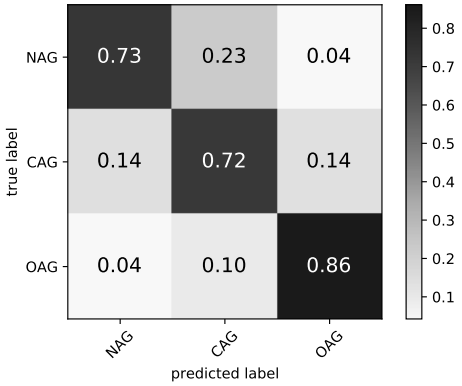

Figure 5: Confusion matrix for task A on the Hindi test dataset.

rences) followed by 👍 (239 occurrences). We assume that the model's performance could be improved by replacing each emoji with its text representation from the Unicode standard, such as *face with tears of joy* or *thumbs up*.

Moreover, the Hindi and Bangla datasets contain non-Latin characters. The pre-trained multilingual BERT that we use for our submission discards all these characters. However, there is another BERT model that overcomes this issue. It is called *multilingual cased* and is trained on non-normalized text (no lower casing, accent stripping, or Unicode normalization). This model is tailored to datasets with non-Latin characters, and we assume it would perform better than our current approach for the Hindi and Bangla datasets.

Last but not least, note that the class distribution of the Hindi test dataset for both tasks is much different compared to the training and validation datasets. Presumably, the reason for that is that the test dataset was sampled from a different social media platform than the training and validation datasets. More details can be found in the dataset description paper (Bhattacharya et al., 2020).

6. Conclusions and Future Work

When fine-tuning complex neural networks, such as BERT, one issue on small datasets is the instability of the classification performance. From one random weight initialization to the next or with slight changes to the training data, the performance can vary significantly, and training needs to

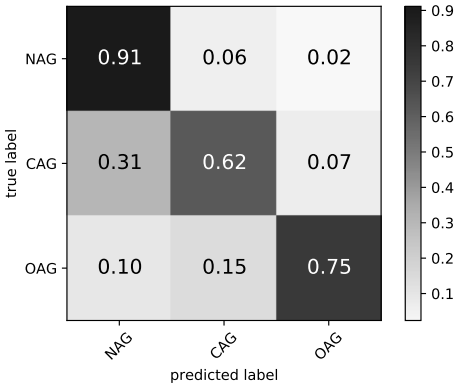

Figure 6: Confusion matrix for task A on the Bangla test dataset.

be restarted many times to select a well-performing model. To overcome the issue of instability, we use bootstrap aggregating (bagging) as a variance reduction technique and combine the predictions of multiple BERT models in an ensemble. Our approach outperforms all other participating teams at five out of six tasks. In our experiments, we further show that the classification performance of an ensemble increases for up to 15 BERT models. Adding more models does not improve the ensemble. The ensembling approach outperforms a single BERT model by approximately two percentage points on average. One direction for future work is to evaluate ensembles of BERT and its successors, such as generalized autoregressive pre-training for language understanding (XLnet) (Yang et al., 2019).

7. Bibliographical References

Basile, V., Bosco, C., Fersini, E., Nozza, D., Patti, V., Pardo, F. M. R., Rosso, P., and Sanguinetti, M. (2019). Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval@NAACL)*, pages 54–63. Association for Computational Linguistics.

Bhattacharya, S., Singh, S., Kumar, R., Bansal, A., Bhagat, A., Dawer, Y., Lahiri, B., and Ojha, A. K. (2020). Developing a multilingual annotated corpus of misogyny and aggression. *arXiv preprint arXiv:2003.07428*.

Bosco, C., Felice, D., Poletto, F., Sanguinetti, M., and Maurizio, T. (2018). Overview of the EVALITS 2018 hate speech detection task. In *Proceedings of the Evaluation Campaign of Natural Language Processing and Speech Tools for Italian (EVALITA)*, volume 2263, pages 1–9. CEUR.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Fersini, E., Rosso, P., and Anzovino, M. (2018). Overview of the task on automatic misogyny identification at IberEval 2018. In *Proceedings of the Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval@SEPLN)*, pages 214–228. CEUR.

Hsueh, M., Yogeeswaran, K., and Malinen, S. (2015). "leave your comment below": Can biased online comments influence our own prejudicial attitudes and behaviors? *Human Communication Research*, 41(4):557–576.

Kumar, R., Ojha, A. K., Malmasi, S., and Zampieri, M. (2018a). Benchmarking aggression identification in social media. In *Proceedings of the Workshop on Trolling, Aggression and Cyberbullying (TRAC@COLING)*, pages 1–11. Association for Computational Linguistics.

Kumar, R., Reganti, A. N., Bhatia, A., and Maheshwari, T. (2018b). Aggression-annotated Corpus of Hindi-English Code-mixed Data. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*. European Language Resources Association.

Kumar, R., Ojha, A. K., Malmasi, S., and Zampieri, M. (2020). Evaluating aggression and misogyny identification in social media. In *Proceedings of the Workshop on Trolling, Aggression and Cyberbullying (TRAC@LREC)*. European Language Resources Association.

Mozafari, M., Farahbakhsh, R., and Crespi, N. (2019). A bert-based transfer learning approach for hate speech detection in online social media. In *Proceedings of the International Conference on Complex Networks and Their Applications (COMPLEX NETWORKS)*, pages 928–940. Springer.

Nikolov, A. and Radivchev, V. (2019). Nikolov-radivchev at semeval-2019 task 6: Offensive tweet classification with bert and ensembles. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval@NAACL)*, pages 691–695. Association for Computational Linguistics.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 2227–2237. Association for Computational Linguistics.

Risch, J. and Krestel, R. (2018). Aggression identification using deep learning and data augmentation. In *Proceedings of the Workshop on Trolling, Aggression and Cyberbullying (TRAC@COLING)*, pages 150–158. Association for Computational Linguistics.

Risch, J., Stoll, A., Ziegele, M., and Krestel, R. (2019). hpidedis at germeval 2019: Offensive language identification using a german bert model. In *Proceedings of the Conference on Natural Language Processing (KONVENS)*, pages 403–408. German Society for Computational Linguistics & Language Technology.

Rösner, L., Winter, S., and Krämer, N. C. (2016). Dangerous minds? effects of uncivil online comments on aggressive cognitions, emotions, and behavior. *Computers in Human Behavior*, 58:461–470.

Struß, J. M., Siegel, M., Ruppenhofer, J., Wiegand, M., and Klenner, M. (2019). Overview of germeval task 2, 2019 shared task on the identification of offensive language. In *Proceedings of the Conference on Natural Language Processing (KONVENS)*, pages 352–363. German Society for Computational Linguistics & Language Technology.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008. Curran Associates, Inc.

Waseem, Z., Davidson, T., Warmsley, D., and Weber, I. (2017). Understanding abuse: A typology of abusive language detection subtasks. In *Proceedings of the Workshop on Abusive Language Online (ALW@ACL)*, pages 78–84. Association for Computational Linguistics.

Wiegand, M., Siegel, M., and Ruppenhofer, J. (2018). Overview of the germeval 2018 shared task on the identification of offensive language. In *Proceedings of the Conference on Natural Language Processing (KONVENS)*, pages 1–10. Austrian Academy of Sciences.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In H. Wallach, et al., editors, *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5753–5763. Curran Associates, Inc.

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., and Kumar, R. (2019a). Predicting the type and target of offensive posts in social media. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 1415–1420. Association for Computational Linguistics.

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., and Kumar, R. (2019b). Semeval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In *Proceedings of the International Workshop on Semantic Evaluation (SemEval@NAACL)*, pages 75–86. Association for Computational Linguistics.

Zampieri, M., Nakov, P., Rosenthal, S., Atanasova, P., Karadzhov, G., Mubarak, H., Derczynski, L., Pitenis, Z., and Çöltekin, c. (2020). SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). In *Proceedings of the International Workshop on Semantic Evaluation (SemEval@COLING)*. Association for Computational Linguistics.