BERT Goes Brrr: A Venture Towards the Lesser Error in Classifying Medical Self-Reporters on Twitter

Alham Fikri Aji[*•] **Haryo Akbarianto Wibowo**[*]
Made Nindyatama Nityasya[*] **Radityo Eko Prasojo**[*⋆] **Tirana Noor Fatyanosa**[*◦]
[*] Kata.ai Research Team, Jakarta, Indonesia
[•] School of Informatics, University of Edinburgh
[⋆] Faculty of Computer Science, Universitas Indonesia
[◦] Graduate School of Science and Technology, Kumamoto University
{aji,haryo,made,ridho,tirana.fatyanosa}@kata.ai
fatyanosa@dbms.cs.kumamoto-u.ac.jp

Abstract

This paper describes Kata.ai's submission for the Social Media Mining for Health (SMM4H) 2021 shared task. We participated in three tasks: classifying adverse drug effect, COVID-19 self-report, and COVID-19 symptoms. Our system is based on BERT model pre-trained on the domain-specific text. In addition, we perform data cleaning and augmentation, as well as hyperparameter optimization and model ensemble to further boost the BERT performance. We achieved the first rank in both classifying adverse drug effects and COVID-19 self-report tasks.

1 Introduction

Over the years, social media has been used as a massive data source to monitor health-related issues (Weissenbacher et al., 2018, 2019; Klein et al., 2020), such as flu trends (Achrekar et al., 2011; Paul and Dredze, 2012), adverse drug effects (Cocos et al., 2017; Pierce et al., 2017), or viral disease outbreak such as the COVID-19 (Sarker et al., 2020; Klein et al., 2021). In general, leveraging massive self-reported data is considered useful for supplementing the otherwise long and costly process of clinical trials in obtaining a more comprehensive picture of the issue in hand.

Nevertheless, analyzing text data from social media is challenging due to its noisy nature, which stems from the prevalence of linguistic errors and typos. In this work, we leverage BERT (Devlin et al., 2018) to handle noisy text through domain-specific pre-training, data cleaning, augmentation, hyperparameter optimization, and model ensemble. With this training pipeline, we achieved the best performance in Social Media Mining for Health (SMM4H) 2021 shared task (Magge et al., 2021) for classifying Adverse Drug Effect and COVID-19 self-report from Twitter text.

Text	Label
How is it that Vyvanse gives me dry mouth, but I still produce this much saliva in my sleep?	ADE
I need Temazepam and alprazolam.... Is there any doctor can prescribe for me?? :/	NoADE

(a) Task 1a : Classification of adverse drug effect (ADE) mentions in English tweets

Text	Label
This girl in my class really had the coronavirus, I'm booking an appointment with my doctor for a check up	1
Read someone on facebook say she hopes the coronavirus doesn't come with the goods she ordered online. Either way, you're quarantined from my facebook, you racist bitch!	0

(b) Task 5 : Classification of tweets self-reporting potential cases of COVID-19

Text	Label
Maybe they've been asked too early. I had a total loss of smell and taste in week 3. In week 1 I only had phantom smells and that's when you test positive.	Self
My brother came home from Paris with a sore throat and a fever and I know he gave me coronavirus. I KNOW IT.	Nonpersonal
Months after Covid-19 infection, patients report breathing difficulty and fatigue https://t.co/H3wcVLxL6y	Lit-News

(c) Task 6 : Classification of COVID-19 tweets containing symptoms

Table 1: Task data examples.

2 BERT Goes Brrr

We participated in 3 classification tasks: Task 1a to classify the adverse drug effect (ADE), Task 5 to classify COVID-19 potential case, and Task 6 to classify COVID-19 symptoms. The distribution of

Proceedings of the Sixth Social Media Mining for Health Workshop 2021, pages 58–64
June 10, 2021. ©2021 Association for Computational Linguistics

Task 1			Task 5			Task 6		
Label	Train	Valid	Label	Train	Valid	Label	Train	Valid
ADE	1231	65	0	5439	594	Lit-News_mentions	4277	247
NoADE	16113	848	1	1026	122	Nonpersonal_reports	3442	180
						Self_reports	1348	73
All	17344	913	All	6465	717	All	9067	500

Table 2: Distribution of datasets.

the datasets are given in Table 2. All tasks' text data are taken from Twitter, with some examples shown in Table 1. More detailed information about the dataset can be found in (Klein et al., 2021; Magge et al., 2021).

We used BERT for all three tasks, implemented with the Huggingface toolkit (Wolf et al., 2020). For each task, we started off by fine-tuning the off-the-shelf BERT-base (Devlin et al., 2018), which resulted in a fairly good performance (Table 3). Then, we improved by using domain-specific BERT instead, then by performing data cleaning, data augmentation, hyperparameter optimization, and finally model ensembling. Table 3 shows the F1-Score improvement by incorporating each of those techniques. Detailed experiments for each technique are in Section 3. Note that some techniques are not used in certain tasks, specified by the dash symbol on the table.

Among the 3 tasks, we achieved the best score for Task 1a and Task 5. Our standing for Task 6 by the time of this paper submission is currently unknown. Still, our performance on Task 6 is above the median, as seen in Table 4. We note that our Task 1a performance on the test set drops significantly compared to its performance on the valid set, indicating overfitting on the valid set. Unfortunately, further analysis on the test set was not feasible since the labels are not provided.

3 Improving BERT

In this section, we dissect each technique we introduce to our submission model.

3.1 Baseline Model

BERT (Devlin et al., 2018) is a pretrained language model based on the Transformer (Vaswani et al., 2017). It is, alongside its many variants, the current state-of-the-art for many NLP applications. It also dominates the previous year's SMM4H shared task and comes out as the winning system (Klein et al., 2020; Weissenbacher et al., 2019).

There are many BERT pre-trained models. To have a good starting point, we explored several pre-trained models. First, we compared general BERT models such as DistilBERT, ALBERT, BERT-base,[1] and BERT-large.[2] Then, knowing that our datasets are tweets that potentially contain medical terms, we explored some domain-specific models: Bio-ClinicalBERT[3] which is trained on biomedical and clinical text (Alsentzer et al., 2019), BERTweet[4] which is trained on English tweets (Nguyen et al., 2020), and BERTweet-Covid19[5] which is built by continuing the pre-trained BERTweet using English tweets related to COVID-19 (Nguyen et al., 2020). We found that BERTweet-Covid19 gives the best result even in the non-COVID-19 related data like Task 1's ADE (see Table 5).

We also considered another COVID-19 tweets pretrained model, that is COVID-Twitter-BERT (CT-BERT)[6] (Müller et al., 2020). It is based on the BERT-large model, while the BERTweet-Covid19 is a BERT-base model. We found that fine-tuning on this model using the recommended hyperparameters is relatively unstable compared to the BERTweet-Covid19 model, though it does outperform it occasionally. As such, we used this model in the later steps, that is, only with hyperparameter optimization and ensembling.

3.2 Data Cleaning

We focused on eliminating tokens that are potential sources of bias. We found that masking Twitter handles, URLs, emails, phone numbers, and money yields the best results. In our experiments, masking all numerical tokens produces worse results.

Furthermore, we also performed a routine HTML tag cleanup, as well as hashtag expan-

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

[2] https://huggingface.co/bert-large-uncased

[3] https://huggingface.co/emilyalsentzer/Bio_ClinicalBERT

[4] https://huggingface.co/vinai/bertweet-base

[5] https://huggingface.co/vinai/bertweet-covid19-base-uncased

[6] https://huggingface.co/digitalepidemiologylab/covid-twitter-bert

Method	Valid F1-Score		
	Task 1a	Task 5	Task 6
BERT-base model	70.87	73.03	98.27
+ Domain-specific BERT	79.14	74.60	98.55
+ Data Cleaning	82.93	77.64	98.87
+ Data Augmentation	–	80.31	–
+ Hyperparameter Optimization	84.30	82.20	–
+ Model Ensembling (submitted system)	87.80	86.27	98.90

Table 3: Our system performance on valid set.

Task	Our Performance			Median Performance			Standing
	F1-Score	Precision	Recall	F1-Score	Precision	Recall	
Task 1a	0.54	0.603	0.489	0.44	0.505	0.409	1st place
Task 5	0.79	0.781	0.789	0.74	0.739	0.743	1st place
Task 6	0.94	0.944	0.944	0.93	0.932	0.932	-

Table 4: Our submitted system performance on test set, compared with the median performance.

Method	Task 1a	Task 5	Task 6
BERT-base	70.87	73.03	98.27
BERT-large	77.78	73.60	98.39
Bio-ClinicalBERT	68.97	68.57	97.33
BERTweet	75.76	71.09	**98.55**
BERTweet-Covid19	**79.14**	**74.60**	**98.55**

Table 5: Baseline on valid set. Task 1a: F1-Score for the ADE class. Task 5: F1-Score for the "potential case" class. Task 6: Macro F1-Score for all classes.

sion (e.g., "#SaveTheEarth" becomes "save the earth"). To this end, we leveraged Ekphrasis (Baziotis et al., 2017) tokenization and masking pipeline. Finally, we performed emoji codification (e.g. into `:thumbsup:`, `:red_heart:`, etc.) using the python emoji package.[7] The emoji codes are treated as special tokens, following the configuration of our chosen base models (Nguyen et al., 2020; Müller et al., 2020).

Below are some data cleaning attempts that did not improve our final model performance.

1. We handpicked some relevant Twitter handles to keep unmasked (such as @WHO). We also tried to pick top-*n* most frequent handles to stay unmasked. Both did not yield better results.
2. We crawled the URLs to get their titles. Using a keyword-based extraction, we determine whether the title is relevant to COVID-19, and if so, we append the title to the end of the tweet. This did not improve the performance of our models.
3. We tried to fix grammatical and typography informality (such as the use of contraction) using Ekphrasis's toolkit, which is based on Norvig's spell checker algorithm. This does not provide better results, not even when using BERT-base or BERT-large.

3.3 Data Augmentation

The provided training data is imbalanced: the number of positive class data is significantly less (Table 2). Therefore, we tried 2 approaches to deal with this issue, namely data oversampling and class weighting. In data oversampling, we duplicate the minority class training data. On the other hand, class weighting simply increases the gradient weight of the minority class.

Additional training data, including the synthetic one, has been shown to improve the model performance (Wei and Zou, 2019; Ma, 2019). Hence, we also explored augmentation data by paraphrasing the training. We create paraphrases by using round-trip translation (Mallinson et al., 2017): our English dataset is translated into another pivot language, then translated back into English. We've tried different pivot languages as well as different translation engines. Based on our manual judgement, using Google Translate and German as the pivot provides the best paraphrase.

Method	Data size	F1
BERTweet-Covid19 (Bc19)	6.5k	74.60
Bc19 + Class-weight	6.5k	75.20
Bc19 + Oversampling	10.5k	76.74
Bc19 + Paraphrase Aug.	12.9k	76.45
Bc19 + Class-weight + Paraphrase Aug.	12.9k	76.49
Bc19 + Oversampling + Paraphrase Aug.	21.1k	**77.65**

Table 6: Task 5 result on data augmentation

[7] https://pypi.org/project/emoji/

Experimental results on data augmentation and data balancing can be seen in Table 6. Our result shows that oversampling is better than class-weighting for dealing with imbalanced training data. Orthogonally, data augmentation can also improve performance. The combination of both data oversampling and data augmentation can increase performance even higher. However, it should be noted that the size of the training data has also increased significantly.

Note that our baseline in this experiment is BERTweet-Covid19 without data cleaning. On uncleaned raw input, we achieved F1-Score of 77.65, as shown in Table 6. However, applying oversampling + paraphrase augmentation on cleaned data can further improve the F1-Score to 80.31.

Interestingly, Task 1a does not benefit from data augmentation or data balancing. Furthermore, adding extra training data from past years' training set does not help as well. Therefore, we only apply data augmentation for Task 5.

3.4 Hyperparameter Optimization

Nowadays, it is common knowledge that optimizing hyperparameter can improve the performance of machine learning algorithms (Kaur et al., 2020; Yang and Shami, 2020; Fatyanosa and Aritsugi, 2020). Current research on the transformer (Murray et al., 2019; Zhang and Duh, 2020) also moving towards hyperparameter optimization (HPO) as the transformer models are susceptible on the chosen hyperparameters (Murray et al., 2019).

The purpose of this section is to determine the best hyperparameter combination of the baseline model for Task 1a and Task 5. We did not optimize the model for Task 6 as the results were already good.

HPO is a time-consuming task. Therefore, performing manual HPO would be inefficient, and it is advisable to utilize automatic optimization. There are several well-known automatic HPO approaches. In this paper, we only use bayesian HPO, specifically, the Tree-structured Parzen Estimator (TPE).

TPE selects the next possible combination of hyperparameters by building probabilistic models. To simplify the search process of the best hyperparameter combination, we employ the Hyperopt (Bergstra et al., 2013) package. As stated in Section 3.1, we also explored a stable and better hyperparameter configuration for Covid-Twitter-BERT.

Table 7 shows all the optimized hyperparameters and their ranges and values. The range for BS was selected following the capabilities of our GPU. We tried two optimizers: AdamW (Loshchilov and Hutter, 2017) and AdaBelief (Zhuang et al., 2020). The ranges for LR, EPS, and WD were selected based on recommendation from (Zhuang et al., 2020).

Hyper-parameter	Definition	Range/Value
BS	Batch size	Min: 8, Max: 32
LR	Learning rate	Min: 1e-6, Max: 1e-4
OP	Optimizer	['AdamW', 'AdaBelief']
EPS	Epsilon	Min: 1e-16, Max: 1e-8
WD	Weight Decay	Min: 0, Max: 1e-2

Table 7: Hyperparameter Range

We set the same random seed to 1 for our baseline. In HPO experiments, we tried to open the possibility of a better model by randomizing the seeds. This assumption is based on several studies suggesting that random seeds influence machine learning algorithms (Madhyastha and Jain, 2019; Risch and Krestel, 2020). It is important to note that the random seeds were not tuned; instead, they were generated randomly in each iteration of the TPE.

As predicted, HPO indeed increase the F1-Score for Task 1a and Task 5 when training the baseline model. After HPO, the results for Task 1a increased by 1.65% and Task 5 increased by 2.35% as shown in Table 3.

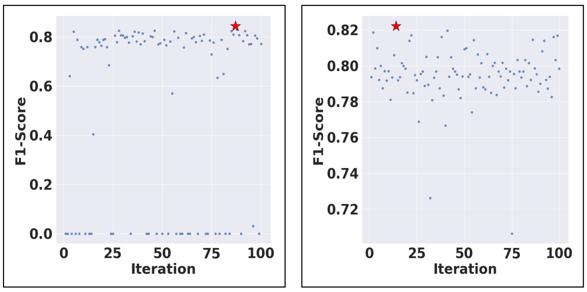
The HPO implementations for the two tasks were executed in the same search space and the same total number of iterations (100 iterations). The visual comparison of the results is illustrated in Figure 1. It shows that the optimal solution for Task 1a is obtained after 87 iterations. Meanwhile, Task 5 only needs 14 iterations. Although the faster discovery of the best combination is preferable in terms of execution time, this scenario can also mean that the algorithm is stuck in local optima.

In terms of execution time, an average of 21 min and 41 min were needed to finish an iteration for Task 1a and Task 5, respectively. Note that the execution time may vary depending on the model and the GPU. The HPO was implemented on NVIDIA Tesla V100 GPU.

Owing to the validation data optimization, it is predictable that HPO bias towards the validation set. Consequently, the model shown strong overfitting, specifically for Task 1a, where the results obtained are very far from the baseline results. The next step to combat the overfitting is to employ ensemble methods.

Task	Hyperparameter					Model	Best F1-Score
	BS	LR	OP	EPS	WD		
Task 1a	11	1.55E-05	AdamW	2.08E-09	0.002085	vinai/bertweet-covid19-base-cased	84.30
Task 5	19	5.21E-05	AdaBelief	5.10E-09	0.000421	digitalepidemiologylab/covid-twitter-bert	82.20

Table 8: HPO results



(a) Task 1a

(b) Task 5

Figure 1: Visual comparison of HPO

3.5 Model Ensemble

Motivated by some past successful results (Chen et al., 2019; Casola and Lavelli, 2020), we ensembled some trained models on Task 1a and Task 5, which are picked from the best performing HPO models. In the implementation of the ensemble technique, to predict the label of an instance, we summed all of the chosen models' probability score and took the highest score as the label.

Typically, a model ensemble considers all of the chosen models. However, our experiments showed that this configuration does not produce the best results for Task 1a (Table 9). We then proceeded to perform an exhaustive search for every possible combinations (that is, the power set) of the chosen models.

Method	Task 1a		Task 5	
Best HPO result	84.30	(1 model)	82.20	(1 model)
All Ensemble	82.71	(15 models)	83.40	(10 models)
Best Ensemble	**87.81**	(5 models)	**86.28**	(5 models)
Top-5 Ensemble	83.58	(5 models)	82.03	(5 models)

Table 9: Result of the model ensemble. **All Ensemble** ensembles all handpicked models. **Best Ensemble** ensemble the subset model of all handpicked models. **Top-5 Ensemble** is the top five best model ensemble result. The "**n** models" represents number of models used to produce the result.

As shown in Table 9, we found the best ensemble involves a subset of five models for both Task 1a and Task 5. There is also a significant gap between the performance of the best subset ensemble with the full model ensemble for both tasks. Regarding Task 1a's "All Ensemble" worse performance, we hypothesize that there might be some "noisy" models among the chosen ones. While our exhaustive search may alleviate this problem, it takes a lot of time that also increases exponentially with respect to the number of chosen models. We leave optimizing this process as future work.

Interestingly, simply choosing the best-performing models does not produce the best ensembled model. As shown in Table 9, model ensemble of the top-5 best F1 ("Top-5 Ensemble") performs worse than the "Best Ensemble". In fact, top-5 ensemble performed worse than a single non-ensembled model from the best HPO result.

4 Conclusion

We describe our team submission for Social Media Mining for Health Applications shared task 2021. Our system achieved the best performance for classifying Adverse Effect mentions and self-reporting potential cases of COVID-19 in English tweets.

Our system is based on BERT model. We observe improvement over the off-the-shelf BERT-base from using domain-specific BERT, rigorous data cleaning, data augmentation, hyperparameter optimization, and model ensembling. Among those techniques, we find that domain-specific BERT, data cleaning, and model ensembling improve the performance on all tasks, whereas data augmentation and hyperparameter optimization are more situational.

Overall, we obtain 17% and 13% improvement on Task 1a and Task 5 respectively (Table 3). On Task 6, we only obtain 0.6% improvement. This is because we did not perform data augmentation and hyperparameter optimization on this dataset, and because the base model already returns a high score of 98.27. We argue that these training pipelines can be used to improve the performance of general text classification tasks.

5 Acknowledgements

We thank Masayoshi Aritsugi (Kumamoto University) for comments on the manuscript.

References

Harshavardhan Achrekar, Avinash Gandhe, Ross Lazarus, Ssu-Hsin Yu, and Benyuan Liu. 2011. Predicting flu trends using twitter data. In *2011 IEEE conference on computer communications workshops (INFOCOM WKSHPS)*, pages 702–707. IEEE.

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. DataStories at SemEval-2017 task 4: Deep LSTM with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754, Vancouver, Canada. Association for Computational Linguistics.

J. Bergstra, D. Yamins, and D. D. Cox. 2013. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, page I–115–I–123. JMLR.org.

Silvia Casola and Alberto Lavelli. 2020. Fbk@ smm4h2020: Roberta for detecting medications on twitter. In *Proceedings of the Fifth Social Media Mining for Health Applications Workshop & Shared Task*, pages 101–103.

Shuai Chen, Yuanhang Huang, Xiaowei Huang, Haoming Qin, Jun Yan, and Buzhou Tang. 2019. Hitsz-icrc: a report for smm4h shared task 2019-automatic classification and extraction of adverse effect mentions in tweets. In *Proceedings of the fourth social media mining for health applications (# SMM4H) workshop & shared task*, pages 47–51.

Anne Cocos, Alexander G Fiks, and Aaron J Masino. 2017. Deep learning for pharmacovigilance: recurrent neural network architectures for labeling adverse drug reactions in twitter posts. *Journal of the American Medical Informatics Association*, 24(4):813–821.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Tirana Noor Fatyanosa and Masayoshi Aritsugi. 2020. Effects of the Number of Hyperparameters on the Performance of GA-CNN. In *2020 IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT)*, pages 144–153. IEEE.

Sukhpal Kaur, Himanshu Aggarwal, and Rinkle Rani. 2020. Hyper-parameter optimization of deep learning model for prediction of Parkinson's disease. *Machine Vision and Applications*, 31(5):32.

Ari Klein, Ilseyar Alimova, Ivan Flores, Arjun Magge, Zulfat Miftahutdinov, Anne-Lyse Minard, Karen O'connor, Abeed Sarker, Elena Tutubalina, Davy Weissenbacher, et al. 2020. Overview of the fifth social media mining for health applications (# smm4h) shared tasks at coling 2020. In *Proceedings of the Fifth Social Media Mining for Health Applications Workshop & Shared Task*, pages 27–36.

Ari Z Klein, Arjun Magge, Karen O'Connor, Jesus Ivan Flores Amaro, Davy Weissenbacher, and Graciela Gonzalez Hernandez. 2021. Toward using twitter for tracking covid-19: A natural language processing pipeline and exploratory data set. *Journal of medical Internet research*, 23(1):e25314.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Edward Ma. 2019. Nlp augmentation. https://github.com/makcedward/nlpaug.

Pranava Madhyastha and Rishabh Jain. 2019. On model stability as a function of random seed. *arXiv*, pages 929–939.

Arjun Magge, Ari Klein, Ivan Flores, Ilseyar Alimova, Mohammed Ali Al-garadi, Antonio Miranda-Escalada, Zulfat Miftahutdinov, Eulàlia Farré-Maduell, Salvador Lima López, Juan M Banda, Karen O'Connor, Abeed Sarker, Elena Tutubalina, Martin Krallinger, Davy Weissenbacher, and Graciela Gonzalez-Hernandez. 2021. Overview of the sixth social media mining for health applications (# smm4h) shared tasks at naacl 2021. In *Proceedings of the Sixth Social Media Mining for Health Applications Workshop & Shared Task*.

Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2017. Paraphrasing revisited with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 881–893.

Martin Müller, Marcel Salathé, and Per E Kummervold. 2020. Covid-twitter-bert: A natural language processing model to analyse covid-19 content on twitter. *arXiv preprint arXiv:2005.07503*.

Kenton Murray, Jeffery Kinnison, Toan Q. Nguyen, Walter Scheirer, and David Chiang. 2019. Auto-sizing the transformer network: Improving speed, efficiency, and performance for low-resource machine translation. *arXiv*, (Wngt):231–240.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English Tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.

Michael J Paul and Mark Dredze. 2012. A model for mining public health topics from twitter. *Health*, 11(16-16):1.

Carrie E Pierce, Khaled Bouri, Carol Pamer, Scott Proestel, Harold W Rodriguez, Hoa Van Le, Clark C Freifeld, John S Brownstein, Mark Walderhaug, I Ralph Edwards, et al. 2017. Evaluation of facebook and twitter monitoring to detect safety signals for medical products: an analysis of recent fda safety alerts. *Drug safety*, 40(4):317–331.

Julian Risch and Ralf Krestel. 2020. Bagging {BERT} Models for Robust Aggression Identification. *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, (May):55–61.

Abeed Sarker, Sahithi Lakamana, Whitney Hogg-Bremer, Angel Xie, Mohammed Ali Al-Garadi, and Yuan-Chi Yang. 2020. Self-reported covid-19 symptoms on twitter: an analysis and a research resource. *Journal of the American Medical Informatics Association*, 27(8):1310–1315.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Davy Weissenbacher, Abeed Sarker, Arjun Magge, Ashlynn Daughton, Karen O'Connor, Michael Paul, and Graciela Gonzalez. 2019. Overview of the fourth social media mining for health (smm4h) shared tasks at acl 2019. In *Proceedings of the fourth social media mining for health applications (# SMM4H) workshop & shared task*, pages 21–30.

Davy Weissenbacher, Abeed Sarker, Michael Paul, and Graciela Gonzalez. 2018. Overview of the third social media mining for health (smm4h) shared tasks at emnlp 2018. In *Proceedings of the 2018 EMNLP workshop SMM4H: the 3rd social media mining for health applications workshop & shared task*, pages 13–16.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Li Yang and Abdallah Shami. 2020. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415:295–316.

Xuan Zhang and Kevin Duh. 2020. Reproducible and Efficient Benchmarks for Hyperparameter Optimization of Neural Machine Translation Systems. *Transactions of the Association for Computational Linguistics*, 8:393–408.

Juntang Zhuang, Tommy Tang, Yifan Ding, Sekhar Tatikonda, Nicha Dvornek, Xenophon Papademetris, and James Duncan. 2020. Adabelief optimizer: Adapting stepsizes by the belief in observed gradients. *Conference on Neural Information Processing Systems*.