DATAMAFIA at WNUT-2020 Task 2: A Study of Pre-trained Language Models along with Regularization Techniques for Downstream Tasks

Ayan Sengupta
Noida, India
ayan.sengupta007@gmail.com

Abstract

This document describes the system description developed by team *datamafia* at WNUT-2020 Task 2: Identification of informative COVID-19 English Tweets. This paper contains a thorough study of pre-trained language models on downstream binary classification task over noisy user generated Twitter data. The solution submitted to final test leaderboard is a fine tuned RoBERTa model which achieves F1 score of 90.8% and 89.4% on the dev and test data respectively. In the later part, we explore several techniques for injecting regularization explicitly into language models to generalize predictions over noisy data. Our experiments show that adding regularizations to RoBERTa pre-trained model can be very robust to data and annotation noises and can improve overall performance by more than 1.2%.

1 Introduction

The recent outbreak of Coronavirus disease (COVID-19) has turned the world topsy-turvy with more than 25M+ infected people so far and 800K+ deaths across the globe[1]. Government officials, researchers, health workers and fear trapped common people are largely relying on online information to monitor, tackle and overcome the situation. Social media platforms, particularly - Twitter and Facebook, have become an easily accessible source of information related to the current affairs. Very recently, few researchers (Drias and Drias, 2020; Samuel et al., 2020) have conducted large scale analysis on Twitter data in the context of COVID-19. However, as mentioned by Nguyen et al. 2020b, a huge majority of the information shared on Twitter are not informative and can pose an additional burden to those who are relying on social media to monitor the pandemic. For example - a tweet like *"Half of Uruguay's COVID-19 cases can be traced to a single fashion designer"* can be speculative and possibly does not contain any insightful information. On the other hand, a tweet like *"Currently 32000+ deaths and their talking spreading it far and wide...BBC News - Coronavirus: Trump unveils plan to reopen states in phases"* can be very useful to a larger population.

To overcome this situation, shared task 2 of WNUT 2020 by Nguyen et al. 2020b allows to automatically identify whether a Tweet is informative in the context of COVID-19 or not. The task dataset contains 10K tweets (written mostly in English) and the associated label - INFORMATIVE and, UNINFORMATIVE labelled by human annotators.

In this task, we use a fine-tuned pre-trained RoBERTa$_{\text{base}}$ model (Liu et al., 2019) to learn the contextual representation of texts. We discover further that the last 4 layers of RoBERTa contain semantically rich hidden representation and are diverse, which, when used together can lead to better performance. In our final submitted model, as described in section 2.1, we use the concatenated hidden states of all the tokens from the last 4 layers of RoBERTa$_{\text{base}}$. Upon further investigation, we realize that the overparameterized large transformer models can be prone to overfitting when fine-tuned on noisy data and ambiguous annotations. In the later part of our study (section 2.3), we explore various different techniques to inject regularization externally to pre-trained language models to improve generalization capabilities. Although, ensembling diverse set of classifiers (Opitz and Maclin, 1999) is known to be an effective technique for improving generalization, in real-life applications, large ensemble systems aren't much effective for drawing inferences on low-resource devices. Further, interpreting model predictions are also difficult for complex ensemble systems. Due to these operational challenges, we refrain ourselves from using ensem-

[1] https://covid19.who.int/

bles and rather focus on single-model systems in our work. We have open-sourced our system and the experiments at Github[2].

2 System Description

In this shared task we use the original train, dev and test datasets provided in the challenge[3]. Dev dataset is used for only validation and evaluating our models. We omit the description of the datasets in this paper due to page constraint, and it can be found in the task description by Nguyen et al. 2020b.

2.1 System Model

After the introduction of self-attention based transformer architecture (Vaswani et al., 2017), several large auto-regressive and auto-encoder based language models (Devlin et al., 2018; Liu et al., 2019; Yang et al., 2019) and their variants have been developed and have showed great results on various NLP downstream tasks including - text classification, Named entity recognition (NER), Natural Language Inference (NLI) etc. Very recently, Nguyen et al. 2020a has developed pre-trained model particularly for English tweets. We use the RoBERTa$_{\text{base}}$ (Liu et al., 2019) as our base language model to learn the hidden representation from the text data. Clark et al. 2019; Kovaleva et al. 2019; Hao et al. 2019 showed that different attention heads from different layers of BERT learn different features from text data. Keeping this in mind, we evaluate all the 12 layers of RoBERTa$_{\text{base}}$ and figured out that the last 4 layers learn diverse set of hidden representations and can influence the final output the most. Although, original BERT and RoBERTa uses only [CLS] token embedding for classification task, our experiment shows that using all the token hidden states can lead to better generalization. Architecture of our submitted model ($Model_{system}$) is shown in Figure 1.

2.2 Other Baselines

Apart from our original submission, we explore other language models and their variants in great details. Each of these models are used to learn the overall representation of the text which is followed by a logistic dense layer to calculate the probability of a text being INFORMATIVE.

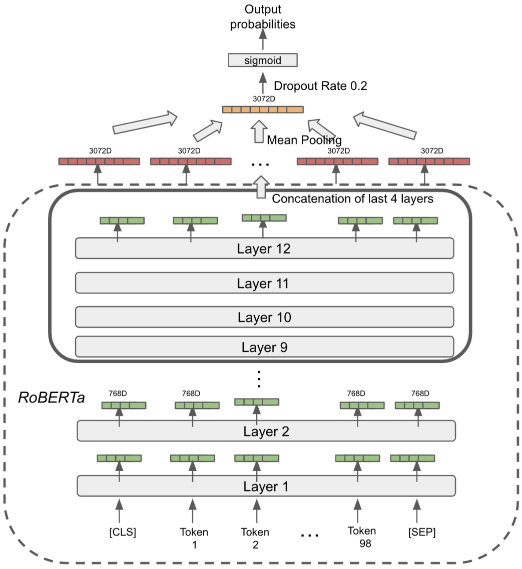


Figure 1: Model architecture by team *datamafia*

- ○ [CLS] representation of language models - BERT$_{\text{base}}$ (Devlin et al., 2018), RoBERTa$_{\text{base}}$, ALBERT$_{\text{base}}$(Lan et al., 2020), BERTweet (Nguyen et al., 2020a)
- ○ Mean/Max of all token hidden states from last/all (concatenated) layers from RoBERTa$_{\text{base}}$
- ○ RoBERTa$_{\text{base}}$ + CNN - We use architecture similar to the one explored by Ma 2019b. We use Wavenet (van den Oord et al., 2016) instead of ordinary convolution layer.

In all these models, we use a dropout of 0.2 before applying the final logistic activation.

2.3 Techniques for Injecting Regularizations into Language Models

Although being an highly over-parameterized models, BERT and its variants are robust to overfitting while fine-tuning (Hao et al., 2019), empirical results from Lee et al. 2020 show that the instability when it is fine-tuned on small and noisy data. Unlike BookCorpus (Zhu et al., 2015) or English Wikipedia data, as used by most of the language models for pretraining, Twitter data is very noisy, unstructured and lacks many linguistic characteristics. To tackle the noisy nature of the dataset, we explore various strategies for regularizing base language model to make it robust to text noises.

- ○ **Transformer Hidden Dropout** - Dropout (Srivastava et al., 2014) is an effective technique to reduce overfitting. Original BERT

[2] https://github.com/victor7246/WNUT-2020-Task-2

[3] https://competitions.codalab.org/competitions/25845

and RoBERTa language models use hidden dropout rate of 0.1 in the FFN layers. We experiment with various dropout rates $dropout(p) \in [0, .3]$.

- **Regularization** - As explored by Schwarz et al. 2018, Kirkpatrick et al. 2017, we add an additional L2 regularization penalty term to final loss. We use λ as regularization coefficient to control the effect of penalty term on the overall loss.
- **Mixout** - Mixout is a technique recently proposed by Lee et al. 2020, and shows strong performance improvement when used with BERT on downstream finetuning tasks. We use the parameter $mixout(\mathbf{w}_{pre})$ to tune the effect of mixout in our model.
- **Multi-Sample Dropout** - Inoue 2019 proposed multi-sample dropout to accelerate training as well as, better generalization. Multi-Sample dropout uses an average of multiple dropouts for a single sample.
- **Text Augmentation** - We inject artificial noise to training data by randomly masking a certain % of all tokens and replacing them with contextually similar word predicted by BERT. For text augmentation we use nlpaug package (Ma, 2019a). For augmentation we use parameter $aug_p \in [0, .3]$ to denote the proportion of the tokens to be masked for each text.

To our best knowledge, next to the work by Lee et al. 2020, our work is the first large-scale empirical study to show the effectiveness of different regularization techniques on pre-trained language models over noisy text data.

2.4 Hyperparamater Settings

In this work, for training, validation and testing, we use the raw data only, without using any further pre-processing. All the pre-trained language models are kept with default configurations. For the base language models we use Huggingface's transformer library (Wolf et al., 2019). We use the default BytePairEncoding (BPE) for each of the language models to tokenize raw texts with max sequence length of 100. Shorter texts are padded with [*PAD*] token id. For all the models, we use Adam optimizer (Kingma and Ba, 2014) with an initial learning rate of $2e-5$, $\beta_1 = 0.9, \beta_2 = 0.999$ and weight decay rate of 0.01. We run each of the experiments for max 15 epochs with an early-stopping criteria based on validation F1 score with a patience of 5. We use a batch size of 32 for both training as well as, validation. Models are checkpointed at each epoch where validation F1 increases from the previous best. We conduct all our experiments on 1 Tesla T4 GPU. All the conducted experiments are logged with Wandb[4,5] (Biewald, 2020).

Dataset	F1	Precision	Recall
Test	89.40	88.57	90.25
Dev	90.84	86.56	95.55

Table 1: Performance of $Model_{system}$ on test & dev datasets

3 Results

We evaluate the performances of all the models using F1, Precision and Recall scores.

3.1 System and Baseline Results

Table 1 shows system model's performance on the test and dev dataset. In Table 2, we have demonstrated the performance of all the baseline methods on dev dataset. RoBERTa shows the most stable performance among all the language models. Even with just [*CLS*] token representation, RoBERTa works pretty well.

We can also observe that using more than one layer of RoBERTa usually works better than using only the last layer.

3.2 Performance of Different Reg. Methods

Table 3 shows the performance of regularization techniques described in section 2.3 on the dev data. We observe that RoBERTa language model with any sort of regularization works better than the one without any regularization added. Figure 2 shows the effect of each regularization method on RoBERTa model on the dev data. Among all the methods, multi-sample dropout and using augmented data show most stability w.r.t all the evaluation metrics. Individual dropout layers in $Model_{multi}$ act differently on each samples and show high variability among each other with, avg.

[4] https://app.wandb.ai/victor7246/wnut-task2

[5] https://app.wandb.ai/victor7246/wnut-task2-regularization

Model Identifier	Model Description	F1	Precision	Recall
$Model_{system}$	***Our submitted model***	90.84	86.56	**95.55**
$Model_{BERT}$	BERT$_{\text{base}}$ with [CLS]	88.20	**89.35**	87.08
$Model_{RoBERTa}$	RoBERTa$_{\text{base}}$ with [CLS]	**90.87**	87.16	94.91
$Model_{ALBERT}$	ALBERT$_{\text{base}}$ with [CLS]	89.21	88.20	90.25
$Model_{BERTweet}$	BERTweet with [CLS]	89.96	88.84	91.10
$Model_{mean12}$	RoBERTa$_{\text{base}}$ mean of all tokens (layer 12)	89.61	86.27	93.22
$Model_{max12}$	RoBERTa$_{\text{base}}$ max of all tokens (layer 12)	89.53	84.56	95.12
$Model_{meanall}$	RoBERTa$_{\text{base}}$ mean of all tokens (all layers concat)	90.76	87.13	94.70
$Model_{maxall}$	RoBERTa$_{\text{base}}$ max of all tokens (all layers concat)	90.65	88.02	93.43
$Model_{CNN}$	RoBERTa$_{\text{base}}$ + CNN	90.57	87.70	94.49

Table 2: Performance of all baseline models on dev. Best scores are highlighted in **bold**.

Model Identifier	Model Description	F1	Precision	Recall
$Model_{noreg}$	No Regularization	90.10	87.75	92.58
$Model_{dropout}$	$dropout(p) = 0.1$	91.19	89.25	93.22
$Model_{l2}$	$\lambda = 0.02$	91.08	**92.37**	89.38
$Model_{multi}$	7-Sample Dropout with $dropout(p) = 0.1$	91.22	87.09	**95.76**
$Model_{aug}$	$aug_p = .1$ with $dropout(p) = 0.1$	**92.04**	88.78	95.55
$Model_{mixout}$	$mixout(\mathbf{w}_{pre)} = 0.6$	90.40	87.20	93.86

Table 3: Performance of different regularization techniques with RoBERTa$_{\text{base}}$ model on dev data. We use [CLS] token representation for the classification.

correlation being -0.004. This diversity works like an ensemble and help the model classifying ambiguous examples correctly. On the other hand, by randomly replacing word tokens in the augmented texts, language models learn the overall context better without depending too much on any particular phrase.

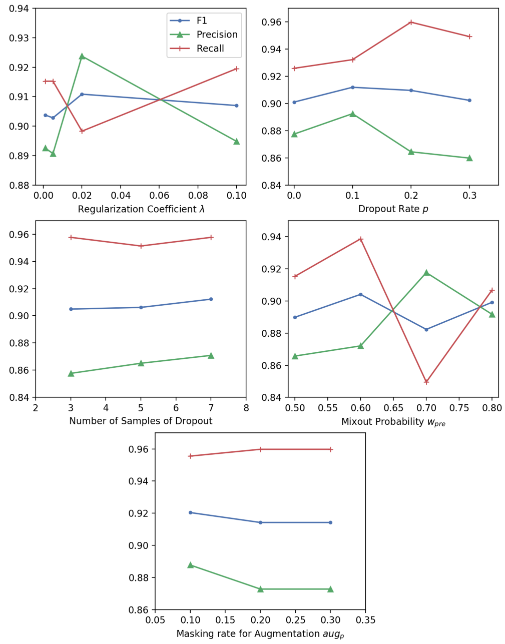


Figure 2: Effect of Regularization Parameters on Classification Performance

4 Result Analysis

In this section, we inspect the language models and explain their predictive capabilities. During our exploratory data analysis (EDA), we plot top words present in the corpus, conditioned on the INFORMATIVE and UNINFORMATIVE classes and figure out that "case", "covid", "death", "virus" etc. remains top words for both the classes of documents.

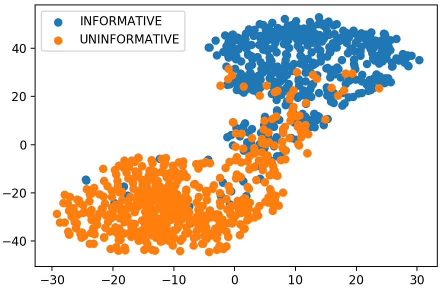
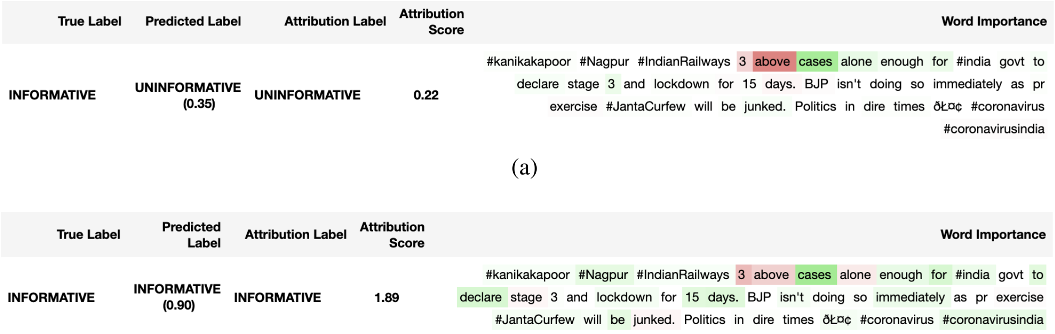


Figure 3: t-SNE plot of embeddings extracted by $Model_{system}$

We observe that any language model in just 2-3 epochs of fine-tuning can achieve a F1 score of more than 89%, however, due to the inherent noise of the tweets, around 10% of the examples are ambiguous and difficult to be classified correctly for almost all the standalone models. Figure 3 shows the two different clusters of representations extracted



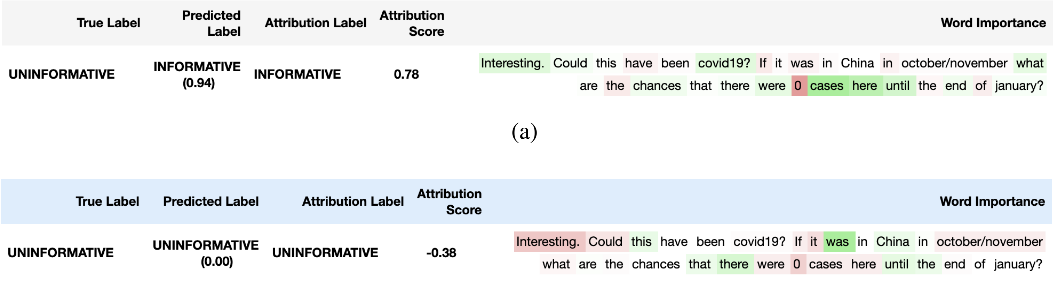
True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
INFORMATIVE	UNINFORMATIVE (0.35)	UNINFORMATIVE	0.22	#kanikakapoor #Nagpur #IndianRailways 3 above cases alone enough for #india govt to declare stage 3 and lockdown for 15 days. BJP isn't doing so immediately as pr exercise #JantaCurfew will be junked. Politics in dire times ðŁ¤¢ #coronavirus #coronavirusindia

(a)

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
INFORMATIVE	INFORMATIVE (0.90)	INFORMATIVE	1.89	#kanikakapoor #Nagpur #IndianRailways 3 above cases alone enough for #india govt to declare stage 3 and lockdown for 15 days. BJP isn't doing so immediately as pr exercise #JantaCurfew will be junked. Politics in dire times ðŁ¤¢ #coronavirus #coronavirusindia

(b)

Figure 4: Explanations for an INFORMATIVE tweet predicted by $Model_{system}$ (a) and $Model_{aug}$ (b). Tokens with high(low) attribute scores are highlighted in green(red).



True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
UNINFORMATIVE	INFORMATIVE (0.94)	INFORMATIVE	0.78	Interesting. Could this have been covid19? If it was in China in october/november what are the chances that there were 0 cases here until the end of january?

(a)

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
UNINFORMATIVE	UNINFORMATIVE (0.00)	UNINFORMATIVE	-0.38	Interesting. Could this have been covid19? If it was in China in october/november what are the chances that there were 0 cases here until the end of january?

(b)

Figure 5: Explanations for an UNINFORMATIVE tweet predicted by $Model_{system}$ (a) and $Model_{aug}$ (b)

by RoBERTa [*CLS*] token (embedded onto a lower dimensional space) on the dev dataset, with several misclassified ambiguous examples. From Table 2 we can understand that all the models have an inductive bias towards the positive class, which lead to relatively poor precision but high recall. There are 89 examples in the validation set which are wrongly classified by $Model_{system}$. However, 47 of them are correctly predicted by either of the regularized models (models described in Table 3), and 70% of those examples are UNINFORMATIVE. We closely inspect the predictions using model interpretation tool Captum (Kokhlikyan et al., 2019), which uses a gradient based attribution method in explaining the predictions. A token with high positive attribution score are assigned more importance by the model and correlates positively with the overall prediction. Similarly, a word with high negative attribution score affects the final outcome adversely. In Figure 4, we explain predictions by $Model_{system}$ and the $Model_{aug}$ on an INFORMATIVE tweet, and found that the system model fails to capture the overall semantics correctly. On the other hand, $Model_{aug}$ looks at contextually more important words like - "declare", "lockdown", "immediately" etc. and predicts the tweet to be INFORMATIVE. Similar observations are found in Figure 5, where, $Model_{system}$ assigns more importance towards frequently occurring words like - "cases" and predicts wrongly. On the contrary, $Model_{aug}$ understands the subtle sarcastic tone of the sentence by looking at the phrases "interesting", "was in China" and classifies correctly with high confidence.

5 Conclusion

In this paper, we present a large-scale empirical study of language models with explicit regularizations. We conclude that using hidden states from multiple layers from a language model helps in understanding the context better and using an additional regularization, we can improve the stability and generalization capabilities of large pre-trained models. In future, we wish to use the insights captured by this work in building a custom and robust language model particularly for noisy user generated texts that are found in social media. We strongly believe that our study will help the research community in using the language model on real-life applications more effectively.

References

Lukas Biewald. 2020. Experiment tracking with weights and biases. Software available from wandb.com.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does bert look at? an analysis of bert's attention. In *BlackBoxNLP@ACL*.

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.

Habiba H. Drias and Yassine Drias. 2020. Mining twitter data on covid-19 for sentiment analysis and frequent patterns discovery. *medRxiv*.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2019. Visualizing and understanding the effectiveness of BERT. 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 4143–4152, Hong Kong, China. Association for Computational Linguistics.

Hiroshi Inoue. 2019. Multi-sample dropout for accelerated training and better generalization. *CoRR*, abs / 1905.09788.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Jonathan Reynolds, Alexander Melnikov, Natalia Lunova, and Orion Reblitz-Richardson. 2019. Pytorch captum. `https://github.com/pytorch/captum`.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of BERT. 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 4365–4374, Hong Kong, China. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Cheolhyoung Lee, Kyunghyun Cho, and Wanmo Kang. 2020. Mixout: Effective regularization to finetune large-scale pretrained language models. In *International Conference on Learning Representations*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei you, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs / 1907.11692.

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

Guoqin Ma. 2019b. Tweets classification with bert in the field of disaster management. In *StudentReport@Stanford.edu*.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020a. Bertweet: A pre-trained language model for english tweets.

Dat Quoc Nguyen, Thanh Vu, Afshin Rahimi, Mai Hoang Dao, Linh The Nguyen, and Long Doan. 2020b. WNUT-2020 Task 2: Identification of Informative COVID-19 English Tweets. In *Proceedings of the 6th Workshop on Noisy User-generated Text*.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio.

D. Opitz and R. Maclin. 1999. Popular ensemble methods: An empirical study. *jair*.

Jim Samuel, G. G. Md. Nawaz Ali, Md. Mokhlesur Rahman, Ek Esawi, and Yana Samuel. 2020. Covid-19 public sentiment insights and machine learning for tweets classification. *mdpi*.

Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. 2018. Progress & compress: A scalable framework for continual learning. In *Proceedings of Machine Learning Research*, volume 80 of *Proceedings of Machine Learning Research*, pages 4528–4537, Stockholmsmässan, Stockholm Sweden. PMLR.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. Rush. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dÁlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5753–5763. Curran Associates, Inc.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.