EdinburghNLP at WNUT-2020 Task 2: Leveraging Transformers with Generalized Augmentation for Identifying Informativeness in COVID-19 Tweets

Nickil Maveli
ILCC, School of Informatics
University of Edinburgh
n.maveli@sms.ed.ac.uk

Abstract

Twitter has become an important communication channel in times of emergency. The ubiquitousness of smartphones enables people to announce an emergency they're observing in real-time. Because of this, more agencies are interested in programatically monitoring Twitter (disaster relief organizations and news agencies) and therefore recognizing the informativeness of a tweet can help filter noise from large volumes of data. In this paper, we present our submission for *WNUT-2020 Task 2: Identification of informative COVID-19 English Tweets*. Our most successful model is an ensemble of transformers including *RoBERTa*, *XLNet*, and *BERTweet* trained in a Semi-Supervised Learning (SSL) setting. The proposed system achieves a F1 score of **0.9011** on the test set (ranking **7**th on the leaderboard), and shows significant gains in performance compared to a baseline system using fasttext embeddings.

1 Introduction

In late December 2019, the outbreak of a novel coronavirus causing COVID-19 was reported[1]. Due to the rapid spread of the virus, the World Health Organization declared a state of emergency. Social media platforms such as Twitter provides a powerful lens for identifying people's behavior, decision-making, and information sources before, during, and after wide-scope events, such as natural disasters (Becker et al., 2010). Identifying relevant information in tweets is challenging due to the low signal-to-noise ratio.

The basic goal of WNUT-2020 Task 2 (Nguyen et al., 2020) is to automatically identify whether a COVID-19 English Tweet is Informative or not. Such Informative Tweets provide information about recovered, suspected, confirmed and death cases as well as location or travel history of the cases. About 4M COVID-19 English Tweets are daily being posted on twitter, the majority of which being not informative. In many instances, it's not always clear whether a person's words are actually announcing a disaster response. Consider an example of an UNINFORMATIVE tweet from the dataset as shown in Table 1:

Text	Label
1) Some thoughts on China's 1Q macro numbers. China's economy was the first to suffer the consequences of fighting the novel coronavirus and is the first on the road to recovery. After an initial cover-up and more than 3,000 deaths, China appears to have brought COVID-19	0

Table 1: A hard to classify tweet.

However, this observation is hard to decipher examining only the vocabulary used; the tweet contains a variety of top frequent informative words ("*coronavirus*", "*covid-19*", "*deaths*"). This example hints that in order to reach meaningful results, we may have to examine contextual linguistic features, model the annotator's bias, introduce adversarial examples, etc (Geva et al., 2019; Goodfellow et al., 2015).

In this paper, we build an ensemble of Transformer (Vaswani et al., 2017) models to leverage its strength in capturing contextual information. The data used to train these models is an augmented version carefully prepared to alleviate confirmation bias and thereby improve generalization. The final inference result is the majority voting of the class from all constituent models through optimal thresholding as a post-processing step. Our best model (ensemble) achieves F1 scores of **0.9248**

[1] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7159299/

Proceedings of the 2020 EMNLP Workshop W-NUT: The Sixth Workshop on Noisy User-generated Text, pages 455–461

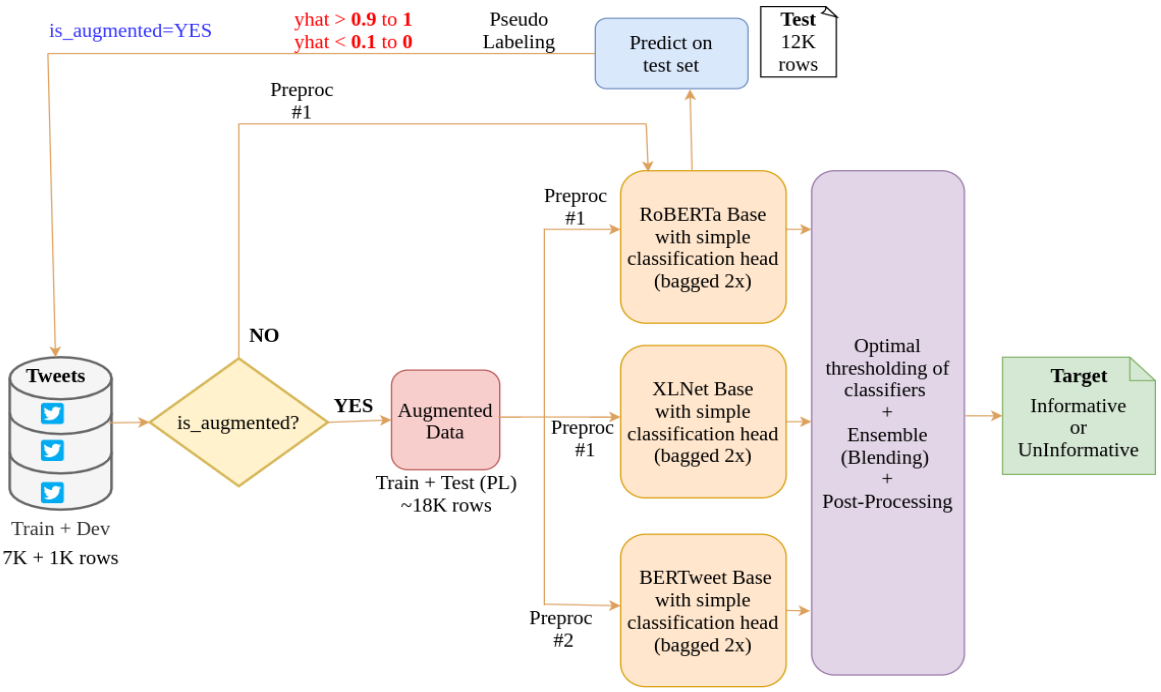

Figure 1: Our proposed model architecture. A RoBERTa model does the classification on the 12K test-set, while being trained using 7K train-set. Later, 11K most confident predictions are appended to the train-set. The new concatenated data is fed to the ensemble models leading to better generalization and improved model performance.

and **0.9011** on the dev and test set respectively.

2 Related Work

Recently, research has started to investigate the use of deep learning in the area of disaster response. For example, Caragea et al. (2016) used CNN to identify informative messages in data from flooding disasters and reported significant improvements in performance over SVM and fully connected ANN. Nguyen et al. (2017) used CNNs on situational awareness crisis data and noted improvements over traditional algorithms. Lazreg et al. (2016) used LSTM network to learn a model from crisis tweets and used this model to generate snippets of information summarizing the tweets. Wang and Lillis (2019) classified actionable tweets using ELMo contextual word embeddings, whereas Ma (2019) used a monolingual BERT-based model for disaster-related tweet classification.

Text classification generally consists of two processes: an encoder that converts texts to numerical representations and a classifier that estimates hidden relations between the representations and class labels. The text representations are generated using N-gram statistics (Wang and Manning, 2012), word embeddings (Joulin et al., 2017; Wang et al., 2018). More recently, powerful pre-trained models for text representations, e.g. BERT (Devlin et al., 2018), have shown state-of-the-art performance on text classification tasks using only the simple classifier of a fully connected layer.

3 System Description

We formulated this task as a binary text classification problem with INFORMATIVE and UNINFORMATIVE as the class names. As shown in Figure 1, the framework of our Informativeness classification model consists of three modules: *Transformer* and *BERTweet* ensemble learning, generalized augmentation via pseudo-labeling, optimal thresholding via post-processing to adjust distribution of class labels in target.

3.1 Data Preprocessing

The preprocessing pipeline consists of the following two strategies.

- **Preproc #1:** Texts are lowercased. Non-ascii letters, urls, @RT:[NAME], @[NAME] are removed. Break apart common single tokens; Eg: *RoBERTa* makes a single token for "...", so convert all single `[...]` tokens into three `[.][.][.]` tokens. Similarly, split "!!!". All *Transformer* models use this preprocessing strategy.

- **Preproc #2:** Texts are normalized using `TweetTokenizer`[2]. Some of the normalization steps are - Expand text contractions("*can't*" to "*cannot*", "*M*" to "*million*", etc), text normalization ("*p . m .*" to "*p.m.*", etc). All *BERTweet* models use this preprocessing strategy.

3.2 Model

We trained 6 models: 2 each of *RoBERTa-base*, *XLNet-base-cased*, and *BERTweet-base* respectively on a 5-fold setup to find the optimal epoch and it's performance was evaluated on the validation set after every epoch. Later, it was trained on the complete dataset.

3.2.1 RoBERTa

The meaning of words can vary subtly from one context to another, and *RoBERTa* generates contextualized word representations to capture the context-sensitive semantics of words (Liu et al., 2019). The use of word representations from *RoBERTa* has resulted in state-of-the-art performance in a variety of language understanding tasks. Given a sentence s consisting of n words $\{w_1, \ldots, w_n\}$, *RoBERTa* model generates their contextualized representations $\{\mathbf{v}^c_{s,w_1}, \ldots, \mathbf{v}^c_{s,w_n}\}$.

3.2.2 XLNet

XLNet is an auto-regressive language model which outputs the joint probability of a sequence of tokens based on the transformer architecture with recurrence (Yang et al., 2019). It's training objective calculates the probability of a word token conditioned on all permutations of word tokens in a sentence, as opposed to just those to the left or just those to the right of the target token.

3.2.3 BERTweet

It is the first public large-scale language model pre-trained for English Tweets that is trained using a 80GB corpus of 850M English Tweets (Dat Quoc Nguyen and Nguyen, 2020). It uses the same architecture as *BERT-base*, which is trained with a masked language modeling objective (Devlin et al., 2018). *BERTweet-base* model claims to do better than *RoBERTa-base* and outperforms previous SOTA models on three downstream Tweet NLP tasks of POS tagging, NER and text classification.

[2] https://github.com/VinAIResearch/BERTweet/blob/master/TweetNormalizer.py

Parameter	Version 1	Version 2
Max Sequence Length	128	192
Epochs	4	4
Batch Size	16	16
Learning Rate	2e-5	3e-5
Optimizer	Adam	AdamW (0.01)
FGM	no	yes

Table 2: Training Hyperparameters. To perform bagging, Version 1 and Version 2 were used.

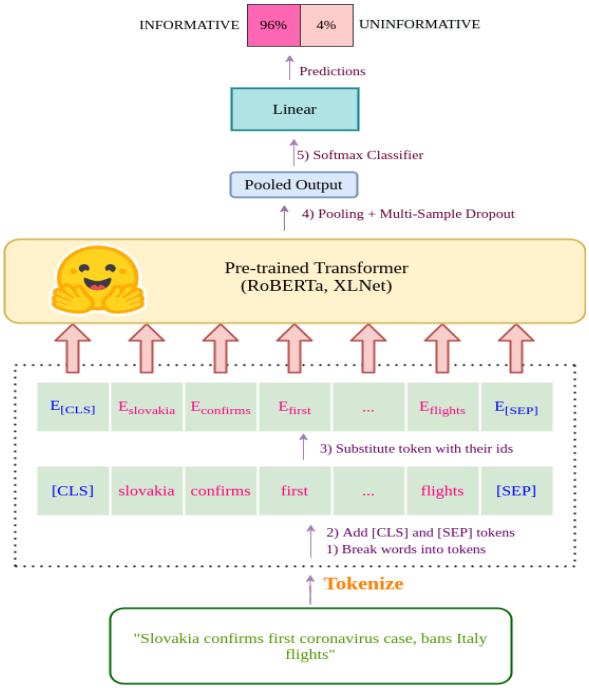

Figure 2: Pre-trained Transformer model architecture for informativeness classification.

3.2.4 Loss

Training Loss, Binary Cross Entropy Loss is defined as follows:

$$BCE = \begin{cases} -log(f(s_1)) & if \quad t_1 = 1 \\ -log(1 - f(s_1)) & if \quad t_1 = 0 \end{cases}$$

where $f()$ is the $sigmoid$ function and s_1 and t_1 are the score and the ground truth label for the class C_1, which is also the class C_i in C.

4 Experimentation

We use only the dataset provided by the organizers to perform our experiments. Overall, there are a total of 10K Tweets split in the ratio of 70/10/20 into train/dev/test set respectively. However, for the final evaluation, 12K unlabeled noisy Tweets were provided, out of which 2K test Tweets were the actual ones the models were evaluated upon.

MODEL	WITHOUT AUGMENTATION			WITH AUGMENTATION		
	PRECISION	RECALL	F1	PRECISION	RECALL	F1
ROBERTA$_{BASE_1}$	0.8652	0.9386	0.9004	0.9619	0.8833	0.9209
ROBERTA$_{BASE_2}$	0.8760	0.9280	0.9012	**0.9640**	0.8818	0.9211
XLNET$_{BASE_1}$	0.8583	0.9364	0.8956	0.9619	0.8798	0.9190
XLNET$_{BASE_2}$	0.8580	0.9343	0.8945	0.9619	0.8731	0.9153
BERTWEET$_{BASE_1}$	0.8630	0.9343	0.8973	0.9534	0.8858	0.9184
BERTWEET$_{BASE_2}$	0.8483	**0.9597**	0.9006	0.9449	0.8974	0.9206
ENSEMBLE	0.8790	0.9386	0.9078	0.9513	0.8998	**0.9248**

Table 3: Results on Dev Data.

Model	P	R	F1
Baseline FASTTEXT	0.7730	0.7288	0.7503
RoBERTa-XLNet-BERTweet-Ensemble	0.8768	0.9269	0.9011

Table 4: Results on Test set.

4.1 Setup

We installed the Python `transformers` library developed by huggingface (Wolf et al., 2019). Pretrained *RoBERTa-base* and *XLNet-base-cased* models with a single linear layer which is simply a feed-forward network that acts as a classification head were used. Figure 2 shows a high-level overview of the architecture.

To speed up training, sequence bucketing by removing unnecessary padding was employed (Khomenko et al., 2017). To improve the robustness of neural networks, and improving resistance to adversarial attacks, Fast Gradient Method (FGM) was used (Miyato et al., 2017) at the end of *Transformer* models. Multi-Sample Dropout (Inoue, 2019) was used when using dropout before the last layer with $p = 0.5$, seemed to converge loss faster. Output of each dropout layer was then passed to a shared weight *fc* layer. Next, we took the average of the outputs from *fc* layer as the final output. Table 2 lists the chosen parameters while model training.

For the *BERTweet-base* model, tweets were normalized and tokenized[3] with a *CNN-Dropout* layer for the inference. Through a bunch of hyperparameters experimented from a finite sample space, we set the *batch_size* = 16, *epochs* = 5, *max_seq_len* = 128, *learning_rate* = $3e-6$, along with Learning Rate Schedulers (Loshchilov and Hutter, 2017).

[3] https://www.kaggle.com/christofhenkel/setup-tokenizer

4.2 Augmentation

Data is carefully augmented with the help of pseudo labeling which is the process of adding confident predicted test data to the training data. Inorder to make the Cross Validation (CV) less over-optimistic, we exclude the pseudo labels from validation folds. In other words, get the labels, run the kfold on only the original data points with real labels, and add the labels to train exactly at training time. That way the CV isn't biased by easy and artificially noiseless targets.

$$\hat{y_{new}} = \begin{cases} 1 & if \quad \hat{y_r} \geq 0.9 \\ 0 & if \quad \hat{y_r} < 0.1 \end{cases}$$

where $\hat{y_r}$ is the meta-prediction on the 12K test-set using *RoBERTa-base* and $\hat{y_{new}}$ is the new label associated with it. These are then concatenated back to the train set, making an augmented data of 18915 rows to develop the final model. In other words, 11915 out of 12K rows in the test-set were identified as confident predictions after pseduo labeling. The thresholds were decided based on several optimization ranges so as to maximize the F-score on holdout dev set.

4.3 Post-Processing

The idea here is to make the distribution of labels in dev/test set to match corresponding distribution of labels in train set so as to maintain the class ratio. Hence, probabilities from all the 6 models were added and a majority voting cutoff value of 4 was found out by fine-tuning that maximized the F-score on holdout dev set.

$$p = \sum_{i=1}^{6} \vec{p_i}$$

$$p_{out} = \begin{cases} 1 & if \quad p \geq 4 \\ 0 & if \quad p < 4 \end{cases}$$

where $\vec{p_i}$ is the probability vector calculated by the 6 models $i \in \{1, \ldots, 6\}$. p is the ensemble output, whereas p_{out} is the final prediction.

5 Results and Error Analysis

Ablation analysis was performed to compare the performance of our model variants. We can evaluate the effect of contextual features by comparing our model with and without augmentation. Table 3 summarizes the performance on Dev Data.

Without augmentation, we notice a situation of high recall, low precision. Our classifier thinks a lot of tweets belong to INFORMATIVE class. This likely leads to a higher number of false positive measurements, and a lower overall accuracy. For the *BERTweet*$_{BASE\_2}$ model that gives the highest recall, 81 false positive and 19 false negative cases were identified. Whereas with augmentation, a situation of low recall, high precision was observed. This makes sense as the model has access to more positive training samples and is able to make better decisions. Our classifier is very picky, and does not think many tweets are INFORMATIVE. For the *RoBERTa*$_{BASE\_2}$ model that gives the highest precision, 61 false positive and 17 false negative cases were identified. Ideally, in the real-world scenario, the high recall case would be more favourable as we want the model to label everything that could potentially be an INFORMATIVE Tweet, because a human personnel will most likely then interpret these results.

Understandably, the fine-tuned *RoBERTa* model outperformed every other experimented models. Bagging the models also lead to lower variance and robust predictions. Table 4 shows the final results wherein our model improves the organizer's baseline by 20%. The effect of augmentation in the final ensemble was drastic as the F-score increased by about 1.87%. Moreover, the idea of summing the probabilities of single models while ensembling worked better in comparison to choosing the most common label after finding different cutoff points that maximized F-score of individual models.

The confusion matrix of our best model is as shown in Figure 3. We look through the examples where our model made misclassification, and summarize the patterns of these error examples along with their attention visualization (Vig, 2019).

- Inaccurate interpretation of contexts. In the sentence, "*Writing 101: don't put 2 numbers side by side. The punctuation is easy to miss.*

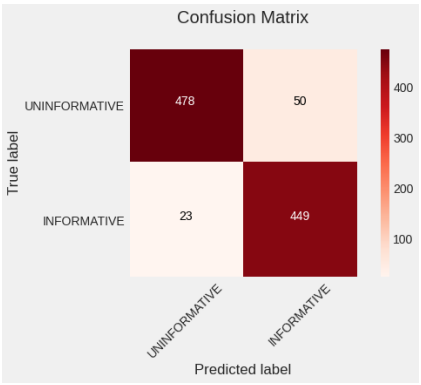

Figure 3: Confusion Matrix.

I first read this as being 51,385 people have died in Ontario from Covid.", much of the attention weights are focused on the latter part. Our model may not capture this shift correctly given the long-distance dependency, which results in a false positive prediction. See Figure 4 (Appendix A) for attention visualization.

- Misinformation due to ambiguity and subjectivity. In the sentence, "*I just remember this news recently China keeping two sets of coronavirus pandemic numbers? "Leaked" infection numbers over 154,000; deaths approach 25,000*", it could be well evident that some events may not really happen as the source of the news lacked credibility. This could have prompted inter-annotator disagreement. See Figure 5 (Appendix A) for attention visualization.

6 Conclusion

We adopted an ensemble approach to reduce the variance of predictions and improve the model performance. The empirical results showed the effectiveness of our model. We also performed an error analysis to gain insights into the model behavior. In future, we would like to combine user-related tweet features (followers, friends, favorite counts, etc) and tweet-related meta features (retweets, creation_date, sentiment, etc) along with contextual representation. Moreover, extending to multilingual tweets (Chowdhury et al., 2020) is a potential future direction to pursue.

Acknowledgements

We would like to thank the anonymous reviewers for their time and comments which have helped make this paper and its contribution better.

References

Hila Becker, Mor Naaman, and Luis Gravano. 2010. Learning similarity metrics for event identification in social media. In *WSDM*, pages 291–300. ACM.

Cornelia Caragea, A. Silvescu, and A. Tapia. 2016. Identifying informative messages in disaster events using convolutional neural networks. In *ICIS 2016*.

Jishnu Ray Chowdhury, Cornelia Caragea, and Doina Caragea. 2020. Cross-lingual disaster-related multi-label tweet classification with manifold mixup. In *ACL (student)*, pages 292–298. Association for Computational Linguistics.

Thanh Vu Dat Quoc Nguyen and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English Tweets. *arXiv preprint*, arXiv:2005.10200.

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.

Mor Geva, Yoav Goldberg, and Jonathan Berant. 2019. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. In *EMNLP/IJCNLP (1)*, pages 1161–1166. Association for Computational Linguistics.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *ICLR (Poster)*.

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

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *EACL (2)*, pages 427–431. Association for Computational Linguistics.

Viacheslav Khomenko, Oleg Shyshkov, Olga Radyvonenko, and Kostiantyn Bokhan. 2017. Accelerating recurrent neural network training using sequence bucketing and multi-gpu data parallelization. *CoRR*, abs/1708.05604.

Mehdi Ben Lazreg, Morten Goodwin, and Ole-Christoffer Granmo. 2016. Information abstraction from crises related tweets using recurrent neural network. In *AIAI*, volume 475 of *IFIP Advances in Information and Communication Technology*, pages 441–452. Springer.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, 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.

Ilya Loshchilov and Frank Hutter. 2017. SGDR: stochastic gradient descent with warm restarts. In *ICLR (Poster)*. OpenReview.net.

Guoqin Ma. 2019. Tweets classification with bert in the field of disaster management.

Takeru Miyato, Andrew M. Dai, and Ian J. Goodfellow. 2017. Adversarial training methods for semi-supervised text classification. In *ICLR (Poster)*. OpenReview.net.

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

Dat Tien Nguyen, Kamla Al-Mannai, Shafiq R. Joty, Hassan Sajjad, Muhammad Imran, and Prasenjit Mitra. 2017. Robust classification of crisis-related data on social networks using convolutional neural networks. In *ICWSM*, pages 632–635. AAAI Press.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. In *ACL (3)*, pages 37–42. Association for Computational Linguistics.

Congcong Wang and David Lillis. 2019. Classification for crisis-related tweets leveraging word embeddings and data augmentation. In *TREC*, volume 1250 of *NIST Special Publication*. National Institute of Standards and Technology (NIST).

Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. 2018. Joint embedding of words and labels for text classification. In *ACL (1)*, pages 2321–2331. Association for Computational Linguistics.

Sida I. Wang and Christopher D. Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *ACL (2)*, pages 90–94. The Association for Computer Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*, pages 5754–5764.

A Appendix

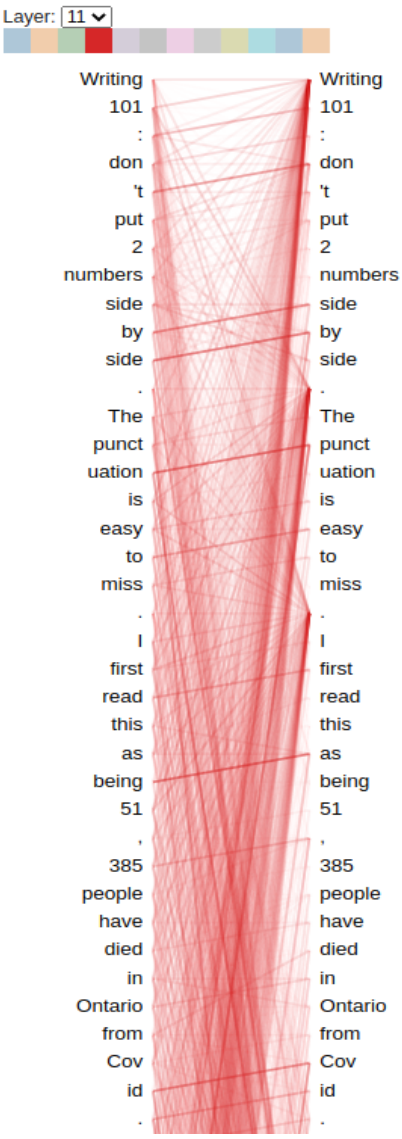

Figure 4: Attention-head view for the last layer of *RoBERTa-base* showing attention to other words predictive of word. In this pattern, attention seems to be directed to other words that are predictive of the source word, excluding the source word itself. In the example below, most of the attention from "id" is directed to "Cov", whereas most of the attention from "Cov" is not focused on "id".

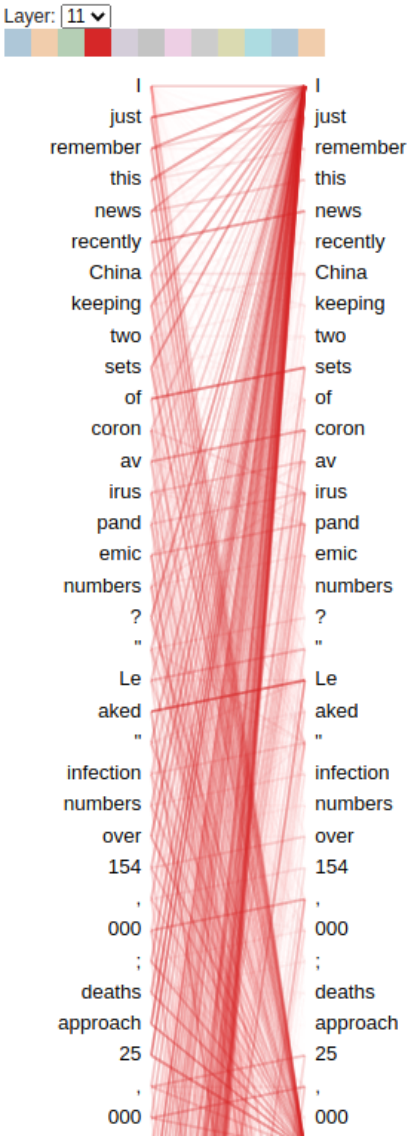

Figure 5: Attention-head view for the last layer of *RoBERTa-base* showing attention to either the previous or the next token in the sentence. For instance, most of the attention for "China" is directed to the previous word "I". Considering a different example, most of the attention for "coron" is directed to the next word "irus" skipping "av" in between.