SunBear at WNUT-2020 Task 2: Improving RoBERTa-Based Noisy Text Classification with Knowledge of the Data domain

Linh Bao Doan
Sun Asterisk Inc.
doan.bao.linh
@sun-asterisk.com

Viet-Anh Nguyen
Sun Asterisk Inc.
nguyen.viet.anh
@sun-asterisk.com

Quang Pham Huu
Sun Asterisk Inc.
pham.huu.quang
@sun-asterisk.com

Abstract

This paper proposes an improved custom model for WNUT task 2: Identification of Informative COVID-19 English Tweet. We improve experiment with the effectiveness of fine-tuning methodologies for state-of-the-art language model RoBERTa (Liu et al., 2019). We make a preliminary instantiation of this formal model for the text classification approaches. With appropriate training techniques, our model is able to achieve 0.9218 F1-score on public validation set and the ensemble version settles at top 9 F1-score (0.9005) and top 2 Recall (0.9301) on private test set.

1 Introduction

Since the outbreak of COVID-19 pandemic, frequently updated information becomes a huge problem of concern. Social media platforms consequently become real-time sources for news about flare-up data. In any case, the flare-up has been spreading quickly, we observe a monstrous amount of information on social networks, for example around 4 million COVID-19 English Tweets every day on Twitter, in which most of these Tweets are uninformative. Therefore, it is crucial to collect the informative ones (for example Corona Virus Tweets identified with new cases or dubious cases) for downstream applications. In any case, manual ways to deal with recognizing useful Tweets require critical human endeavors, and hence are expensive.

Based on the dataset provided in WNUT-2020 Task 2: Identification of informative COVID-19 English Tweets (Nguyen et al., 2020), we propose a fine-tuning strategy to adopt the universal language model RoBERTa as an backbone model for text classification purposes. We also conduct several experiments in varied fine-tuning architectures on the pre-trained RoBERTa. Our best model results in a high F1-score of 0.9005 on the task's private test dataset and that of 0.9218 on the public validation set with Multilayer Perceptron Head.

2 Related work

One of the most important parts in text classification problems is input representation. Traditional methods construct context-independent embeddings for words. *Mikolov et al.* (Mikolov et al., 2013) introduce an open-source *Word2Vec*, which consists of two models: Continuous Bag of Words (CBOW) and Skip-gram model. The models were trained on 1.6 billion words to learn linguistic contexts of words. While Word2Vec is a self-supervised algorithm, GloVe (Pennington et al., 2014) is trained unsupervised to form word embeddings. GloVe factorizes co-occurrence matrix of words, resulting in dense word vectors. However, both GloVe and Word2Vec fail representing rare or out-of-vocabulary words. FastText (Mikolov et al., 2018) mitigates this problem by decomposing words as a sum of character n-grams. This handles unseen words very well because these character n-grams may still occur in other words. In contrast to context-independent embeddings, modern language models encode word semantics within contexts. Word vectors obtained from these methods achieve better results on downstream tasks because a word in different contexts expresses different meanings. Bidirectional Encoder Representations from Transformers (Devlin et al., 2018), or BERT for short, outperforms the previous best result with GLUE score of 80.4%, which is 7.6% improvement. There are two variants of BERT: base and large; the large model is a stack of 24 Transformers' encoders for a total of 340M parameters while the base one has only 12 encoders. GPT-2 (Radford et al., 2019) by OpenAI is a gigantic model with 1.5 billion parameters and 48 layers, setting new state-of-the-art results on 7 out of 8 datasets. Face-

Proceedings of the 2020 EMNLP Workshop W-NUT: The Sixth Workshop on Noisy User-generated Text, pages 485–490
Online, Nov 19, 2020. ©2020 Association for Computational Linguistics

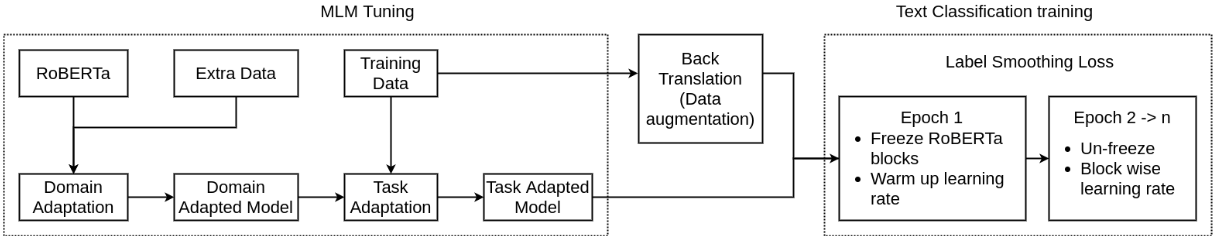

Figure 1: Our overall pipeline for hierarchical MLM tuning and main task training.

book Research team improves training procedures for BERT, introducing RoBERTa (Liu et al., 2019). The improvements include extended training time on a ten-times bigger dataset, increased batch size, using byte-level encoding with larger vocabulary, excluding next sentence predicting task, and dynamic masking pattern modifying.

3 Proposed method

Figure 1 illustrates our process. For MLM tuning we propose hierarchical tuning process that consists of two steps: Domain adaptation using extra COVID data and Task adaptation using the given training data. After MLM Tuning, we utilize different training techniques for text classification such as back translation, warm-up learning rate, layer freezing and layer-wise learning rates. This section provides details of this pipeline.

3.1 RoBERTa network for Text Classification Task

Taking advantage of RoBERTa as a backbone, we propose a customized network with appreciably modifications. Figure 2 illustrates our proposed architecture. The "base" version of RoBERTa is used. It has 12 Transformer blocks, each block outputs a 768-D vector for each token. Since the output of different Transformer blocks represent different semantic levels for the inputs, in our experiments we combine outputs of those Transformer blocks by concatenation. This combination is fed to a classification head. We propose two types of the head:

- **MLP Head:** A simple feed forward network with one hidden layer. This head takes the last token embedding as its input.

- **BiLSTM Head:** A recurrent neural network with one Bidirectional LSTM layer. This network takes embeddings of all tokens.

The hyperparameters are shown in Section 4.

3.2 Fine-tuning Masked Language Model (MLM)

3.2.1 Direct tuning on task data

RoBERTa apparently is an excellent language model since it was trained on a huge dataset in a broad domain. However, the general domain is also a drawback when it comes to downstream tasks with completely different domains such as classifying users' tweets on Twitter. Therefore, in order to produce high-quality outputs from the model, there is a need of fine-tuning MLM task on the task dataset for RoBERTa. This adapts the universal language model into our narrow domain, giving it prior knowledge for later classification training.

Choosing learning rate is the key factor for the convergence. If learning rate is too small, the model may converge too slow causing harder to fit to new data distribution. On the other hand, large learning rate can lead to the problem of useful feature forgetting. Hence, we employ warm-up learning rate scheduler (Howard and Ruder, 2018) to help the model converge faster while preserving its good initialization.

3.2.2 Hierarchical tuning with extra data

We assume fine-tuning only on the dataset might cause overfitting on the chosen dataset only. Hence, we propose a hierarchical fine-tuning strategy for RoBERTa: the first phase we train with custom domain COVID Tweets dataset for ***domain adaptation***, then the second phase is a fine-tuning process with WNUT Task 2 dataset for ***task adaptation***. Our custom COVID Tweets dataset is gathered from Twitter platform, including unlabeled 1 million posts in general COVID domain, which has the hashtag of **#Covid**, **#Covid19**, and **#Coronavirus**. We expect this model to generalize better on different distributed dataset in the same field of COVID Tweets.

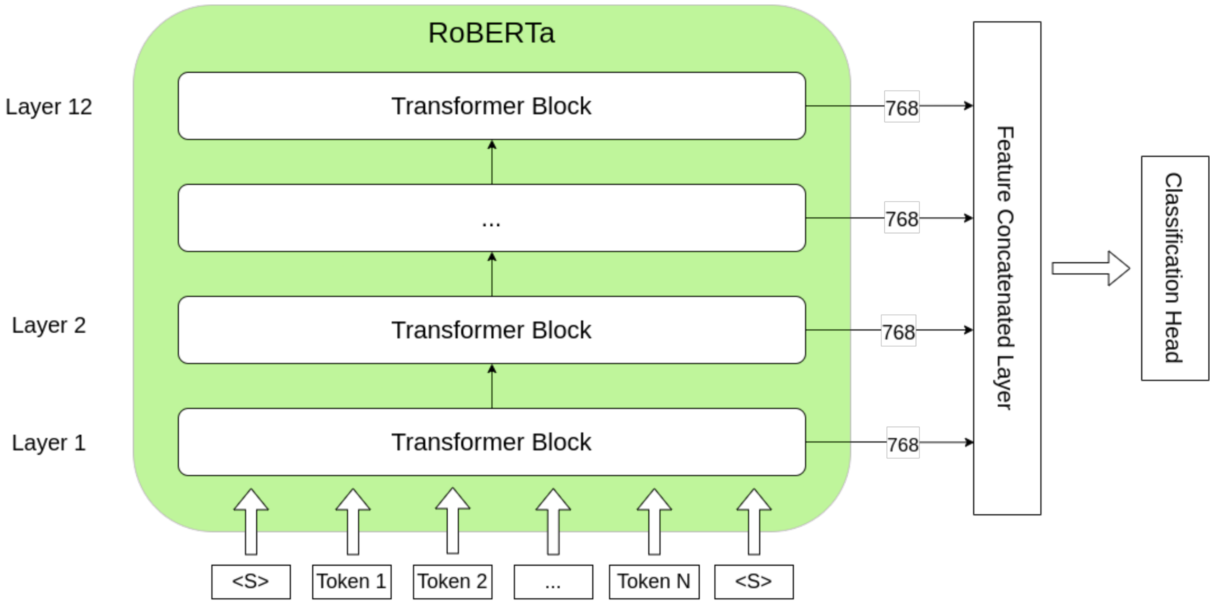

Figure 2: The architecture of the proposed model. The input is tokenized into a sequence of BPE tokens. RoBERTa, the "base" version, takes this sequence and propagates it through 12 Transformer layers. By concatenating outputs from these 12 layers, we form a long sentence representation for the follow-up classification head, which is a simple Multi-layer Perceptron/Long Short-Term Memory network.

3.3 Text classification training

3.3.1 Back Translation

Recently research (Xie et al., 2019; Edunov et al., 2018) have shown that back-translating monolingual data can be used as a potential form of data augmentation in Text Classification. The idea behind back translation is to translate a sentence from the original language (English) to another selected language and then translate back to the original language. This utilizes the power of current well-developed translation engines. In our experiment, 25% of the data samples is back-translated into Vietnamese, the same amount goes for Italian and French, and the rest 25% is kept unchanged. This assures the languages contribute equally to the overall dataset. Totally, the dataset size is increased by 75%.

3.3.2 Model freezing with layer-wise learning rates

Layer freezing helps preserving useful knowledge that a pre-trained neural network has learned. Since RoBERTa has been trained on a huge dataset, we would not want the model to derive too far from its pre-train weights. The training procedure is divided into 2 steps:

- **Step 1:** We freeze RoBERTa to train the classification head for the first epoch. Warm-up learning rate (Section 3.2.1) is also applied. Because RoBERTa's weights are already well trained, this step helps escape from narrow local optimum.

- **Step 2:** RoBERTa is unfrozen, a whole network is trained. In RoBERTa, upper layers produce embeddings with more context-specific than lower layers. This motivates us to further apply layer-wise learning rate: set a small learning rate for the shallowest layer, increase the learning rate as the layer goes deeper.

3.3.3 Label Smoothing

When training a huge neural network on a relatively small dataset, overconfidence is a problem leading to bad behaviours of the model. This phenomenon occurs when the model gives predictions with confidence higher than its accuracy. While there have been a lot of studies for overfitting reduction, overconfidence problem attracts less attention from researchers. In this study, we employ label smoothing (Szegedy et al., 2015) to prevent model from being too certain about its predictions. Instead of assigning "hard" one-hot encoded ground truth, label smoothing adds a small perturbation into the label by a smoothing parameter α.

Model	Precision	Recall	F1	Accuracy
RoBERTa + MLP Head	0.9407	0.8740	0.9061	0.9080
RoBERTa + BiLSTM Head	0.9322	0.8853	0.9082	0.9110
Direct tuning + MLP Head + Label smoothing	**0.9492**	0.8960	**0.9218**	**0.9240**
Direct tuning + BiLSTM Head + Label smoothing	0.9364	**0.8983**	0.9170	0.9200
Direct tuning + MLP Head + Back translation + Label smoothing	0.9343	0.8909	0.9121	0.9150
Hierarchical tuning + MLP Head + Back translation + Label smoothing	0.9449	0.8745	0.9084	0.9100

Table 1: Comparison of different tuning and training techniques on the public validation set.

$$y'_k = y_k(1-\alpha) + \alpha/K$$

, where y_k is output probabilities of K classes.

Moreover, label smoothing also helps stabilize the training process. When using cross-entropy loss, one-hot encoded labels cause numerical instabilities if the prediction is close to one-hot form. In that case, the loss will become $1 \log 0 = -\infty$. By setting $\alpha \neq 0$, this problem can be solved.

4 Experiments and Results

4.1 Experiment setup

Our set-up is proceeded as following instruction. We trained our networks with PyTorch framework on GPU GeForce GTX 2080Ti with batch size 32 for 20 epochs. We used AdamW (Loshchilov and Hutter, 2017) for the optimization and a learning rate of $3e-5$, decayed 0.01 except for LayerNorm layers. Label smoothing hyperparameter α was empirically experimented with multiple values of 0, 0.1, 0.15, 0.2 and the last value possessed promising results. The numbers of hidden units of MLP Head and BiLSTM Head to 768 and 256 respectively.

4.2 Evaluation metrics

Evaluation metrics for assessing are Accuracy, F1-score, Recall and Precision metrics on public validation set. Accuracy can be used when the class distribution is similar while F1-score is a better choice of metric when there are imbalanced classes.

$$precision = \frac{TP}{TP+FP}$$

$$recall = \frac{TP}{TP+FN}$$

$$F_1 = \frac{2}{precision^{-1} + recall^{-1}}$$

, where TP: True Positive, FP: False Positive, FN: False Negative

4.3 Results

Table 1 compares the performance of multiple trial architectures training with pre-trained method using RoBERTa in our base settings. The original RoBERTa with MLP Head shows the better result than LSTM head, but the difference is not really noticeable (0.9082 vs. 0.9061). When applying direct tuning MLM and label smoothing, the gap has been widened, specifically, 0.9218 for MLP Head and 0.9170 for LSTM Head.

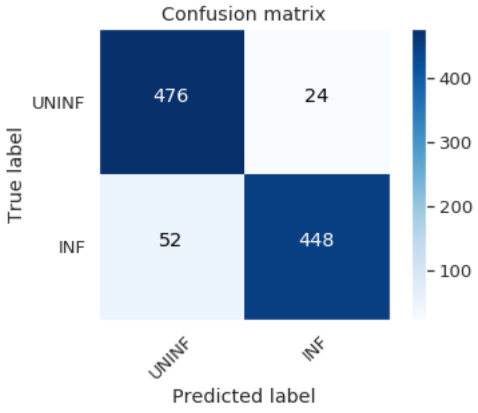

Figure 3: Confusion matrix on the public validation set

In the table, hierarchical tuning and back translation method did not yield better results than direct tuning without back translation one. Nevertheless, we expect this method can generalize well on many different distributed datasets and thus, we ensembled the two versions with voting and submitted to the private test benchmark. We ended up at top 9 on the leaderboard with 0.9005 F1-score and

Table 2: Some failures of our system.

Text	Model prediction	Truth label
Some people metaphorically shake their walking sticks at the TV like Grandpa Simpson &amp; rage ”flu has already killed thousands in USA”, ”but guns have already killed over 6,000 in USA this year” - all true. But Coronavirus is In Addition to those deaths. HTTPURL	INFORMATIVE	UNINFORMATIVE
2/26 PCR test 2/27 Negative result 2/28 X-ray shows n.p. Discharge. Stay near Haneda airport 2/29 Akita airport Return 3/6 Follow up: Visit B in Akita. Fever &amp; cough - B consults with designated outpatient service(C) And PCR + covid19 HTTPURL	INFORMATIVE	UNINFORMATIVE
Amazon and Facebook ask Seattle employees to work from home after coronavirus cases HTTPURL	UNINFORMATIVE	INFORMATIVE
Third of Sacramento coronavirus cases linked to church events - Los Angeles Times. Pathetic! HTTPURL	UNINFORMATIVE	INFORMATIVE

0.9301 Recall, in which our Recall score reached the second place.

4.4 Error Analysis

A question that as important as designing a subtle method is “what make the model fail”. By answering this question we can gain an insight into our model performance and further improve it. The method used for analyzing is the bottom row in Table 1. Firstly, we plot confusion matrix (Figure 3), observe that both True Positive and True Negative are evenly distributed with a small proportion of False Negative and False Positive, indicating our model did not bias towards any classes. Secondly, we randomly sample some failures the model made (Table 2). It seems like sentences containing more numbers are usually (mis)classified as INFORMATIVE while the ones containing less numbers are classified as UNINFORMATIVE. This can be explained that INFORMATIVE tweets provide information about recovered, suspected, confirmed and death cases. Therefore, numbers appearance is inevitable.

5 Conclusion

In this paper, we have explored and proposed our pipeline to solve the Identification of Informative COVID-19 English Tweet task by using a pretrained universal language model. By conducting a lot of experiments, we have demonstrated that the use of RoBERTa and our fine-tuning strategy is highly effective in text classification tasks. With our proposed methods, we have achieved prominent results on the WNUT Task 2.

For future work, we will design more complex classification head architectures to improve model’s performance as well as solving problems indicated in Section 4.4. Furthermore, we would like to employ our model and pipeline in different languages such as Vietnamese to see how they adapt to new languages.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification.

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.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

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*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

A. Radford, Jeffrey Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2015. Rethinking the inception architecture for computer vision.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V. Le. 2019. Unsupervised data augmentation for consistency training.