UET at WNUT-2020 Task 2: A Study of Combining Transfer Learning Methods for Text Classification with RoBERTa

Huy Quang Dao
University of Engineering and Technology
Vietnam National University, Ha Noi
huydao98.uet@gmail.com

Tam Minh Nguyen
Sun Asterisk Inc.
nguyen.minh.tamb
@sun-asterisk.com

Abstract

This paper reports our approach and the results of our experiments for W-NUT task 2: Identification of Informative COVID-19 English Tweets. In this paper, we test out the effectiveness of transfer learning method with state of the art language models as RoBERTa on this text classification task. Moreover, we examine the benefit of applying additional fine-tuning and training techniques including fine-tuning discrimination, gradual unfreezing as well as our custom head for the classifier. Our best model results in a high F1-score of 89.89 on the task's test dataset and that of 90.96 on the public validation set without ensembling multiple models and additional data.

1 Introduction

Identification of Informative COVID-19 English Tweets (Nguyen et al., 2020) is the task of "providing users the information related to the virus". It is meaningful in the sense that with an increasing amount of tweets about the virus, many among them are uninformative and even harmful to the viewers. Manually identifying uninformative tweets is costly. Therefore, a system which can perform the task automatically would be tremendously helpful.

With the rise of deep learning, particularly transfer learning for solving text classification problem, we would like to propose an approach that leads to high performance (represents by a high F1-score of 89.89) on the task's test dataset. This approach uses pre-training method with state-of-the-art pre-trained language model RoBERTa (Liu et al., 2019), combines with many existing fine-tuning techniques including one-cycle-policy learning rate(Smith, 2018),fine-tuning discrimination, gradual unfreezing (Howard and Ruder, 2018), label smoothing (Pereyra et al., 2017), and our custom-head model.

Other than using pre-training with a state-of-the-art pre-trained language model, there is no clear winning factor for our success as all fine-tuning techniques need to incorporate to form our best model. Our main contributions are:

• We perform numerous experiments to support our hypothesis. Which is fine-tuning state-of-the-art pre-trained language models such as RoBERTa is more beneficial to the Identification of Informative COVID-19 English Tweets task than several training-from-scratch models.

• We combine many fine-tuning techniques to form our best model and experiments with the effect of each technique by gradually stacking them onto our base model then observe their effects.

2 Related work

2.1 Language model pre-training

Universal feature representation function, through pre-training language model on large amount of unlabeled data, namely ELMo (Peters et al., 2018), GPT (Radford, 2018), BERT (Devlin et al., 2018) and XLNet (Yang et al., 2019) has brought tremendous gains in performance for many NLP tasks. The rise of those models has been most beneficial to transfer learning for downstream tasks such as text classification, Question Answering, Text Summarization. Different from learning fixed feature vectors of words without regard to its context such as Word2Vec (Mikolov et al., 2013) or Glove (Pennington et al., 2014), those above self-training method learn context-dependent word representation, results in high quality features learning for text (Jawahar et al., 2019).

2.2 Tranfering training techniques

ULMfit (Howard and Ruder, 2018) introduced a novel fine-tuning method which is task-adaptive pre-training which boosts upmost NLP downstream

Proceedings of the 2020 EMNLP Workshop W-NUT: The Sixth Workshop on Noisy User-generated Text, pages 475–479
Online, Nov 19, 2020. ©2020 Association for Computational Linguistics

task's performance. The authors fine-tune the pre-trained language model in order to adapt the weights to a new task distribution. Moreover, the authors experiment with a combination of several training techniques, that they suggest, to bring a gain in performance for many common transfer learning settings, such as Learning Rate Discrimination, Gradual Unfreezing, and Slanted Triangular Learning Rate. Our work has been mostly inspired by this paper.

Don't stop pre-training (Gururangan et al., 2020) sheds light on the effectiveness of domain-adaptive pre-training and task-adaptive pretraining in 4 different domains with 8 downstream tasks, 2 tasks each domain, including tasks with limited and redundant labeled data. The paper points out that task – adaptive pre-training results in better performance compared to only fine-tuning weights on downstream tasks. Unfortunately, we did not experiment with this technique for WNUT-2020 Task 2: Identification of informative COVID-19 English Tweets competition.

2.3 RoBERTa

RoBERTa (Liu et al., 2019) was built based on BERT's language masking strategy. However, RoBERTa model modifies several key hyperparameters in BERT including removing BERT's next-sentence pretraining objective and training with much larger mini-batches as well as learning rates. It outperforms BERT on a variety of NLP tasks and archives comparable performance with the SOTA model XLNET (Yang et al., 2019). Likewise BERT, RoBERTa has two different settings, RoBERTa Base which uses 12 layers of Transformer Encoder and 24 Transformers Encoder Layers with RoBERTa Large. We experiment with both RoBERTa Base and RoBERTa Large as out base model and show a comparison in performance between them.

3 System description

3.1 Pretraining and backbone mode:

Fine-tuning the downstream task's model in Natural Language Processing using pre-trained language models, such as BERT, has been experimentally shown to be effective, both in terms of convergence time and performance (Gururangan et al., 2020). However, the choice of the pre-trained model affects the result of the downstream task. Since RoBERTa Base and RoBERTa Large have achieved significant gain in performance in many common fine-tuning settings, we experiment using both models as the backbone for our task's base model.

Our base model is simply designed to test out the effectiveness of our choice of backbone, custom head, and training techniques. We use both RoBERTa Base and RoBERTa Large (Liu et al., 2019) as our backbone in all model settings to compare the effectiveness of each backbone.

Our base model includes a backbone extracting all information of an input sequence into the [CLS] token's features of the last backbone's layer. Those features are then linearly projected onto 2D, followed by a Softmax activation to predict the probability of the label being "Informative" or "Uninformative" given the input. Notice that we use cyclical learning rate and label smoothing to all model's settings in our experiment including the base model's settings.

More complex model settings use a more complex custom head and are gradually added advanced training techniques.

In order to emphasize the significant contribution of the pre-training method using the SOTA language model, we also train used-to-be state of the art models in text classification including Hierarchical Attention Networks (Yang et al., 2016) combined with Weight-Dropped GRU (Merity et al., 2017), Bidirectional Long Short-Term Memory Networks with Two Dimensional Convolutional Neural Network (Zhou et al., 2016) and Non-Static Convolutional Neural Network (Kim, 2014). As expected, the results of training those models from scratch have been around 8.0 lower F1-score than our base model. The detailed result is demonstrated in the Experiment Results section.

3.2 Custom head design

The higher layer in RoBERTa model captures higher-level features and semantic meaning (Jawahar et al., 2019) . We would like to incorporate more types of features by using not only [CLS] token's features of the last backbone's layers but those of 4 last backbone layers. We concatenate all those features, subsequently linearly project them on lower-dimensional space. We refer to the output of this process as features of branch one. Note that with each linear layer except the last one, we always stack one Batch Normalization (Ioffe and Szegedy, 2015) layer following ReLU activation.

We suspect that it is burdening to use only [CLS] token's features as the only source of information for the prediction of the model. To resolve this, we also utilize features of the rest of the tokens of the last backbone layers in our custom head. We put the rest tokens' features through a Bidirectional LSTM (Hochreiter and Schmidhuber, 1997), 1D Convolutional Neural Network, and Max Pooling Over Time to extract and summarize useful information. The output of this process is referred to as feature branch two. We then aggregate 2 branches' features through concatenation. Finally, we apply a linear classifier on top of those combined features. The choice of 4 layers are the only heuristic, not thoroughly scientific, this choice can be further examined in our future works.

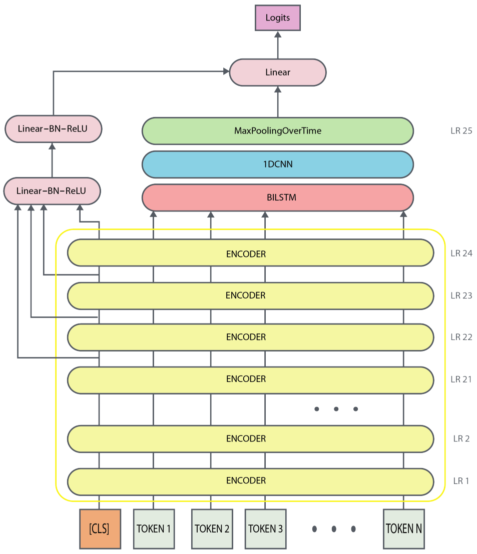

Figure 1: Architecture of the best proposed model.

Figure 1 describes our best architecture which has RoBERTa Large as the backbone, and our custom head on top of it. We refer to section 3 as the source of more details.

3.3 Transfer learning techniques

Learning rate discrimination: Intermediate layers in BERT model learn a "rich hierarchy of linguistic information, starting with surface features at the bottom, syntactic features in the middle followed by semantic features at the top" (Jawahar et al., 2019). We expect the same behavior for its variants which is RoBERTa (Liu et al., 2019). Therefore, when adapting to a new domain or a new task, it would be more appropriate to strongly adapt the layer that captures the semantic meaning and slowly, slightly adapts the weights of lower layers which contain most general knowledge (Yosinski et al., 2014). We achieve that by setting the learning rate differently for each layer, the lower layer has a small learning rate while higher ones update their weights at a higher rate.

Cyclical learning rate: The cyclical learning rate has been empirically shown to improve neural network performance. The learning rate should neither be set to be too large nor too small but first to get warm up by a gradual increase, followed by a gradual decrease. By doing that, learning would less likely to over-fit (due to small learning rate) or diverse (due to large learning rate). For more specifications, we refer to the experiment section.

Gradual unfreezing: ULMfit (Howard and Ruder, 2018) pointed out the risk of catastrophic forgetting appears as the result of fine-tuning all layers of the backbone model simultaneously. To avoid that, the paper suggests to apply multi-phase training, each unfreezes one layer, from top to bottom gradually. In the experiments section, we show that incorporating these techniques results in better overall models' performance

4 Experimental Results

4.1 Data preparation

Data description: The original dataset contains 7000 samples for training and 1000 ones for validating, and an additional public validation set. It is an almost balanced dataset with 3303 samples being labeled INFORMATIVE and 3697 are UNINFORMATIVE tweets. The average length of the sample in each class is 40 tokens.

Data preprocessing: We first exclude all emoji in the dataset. Our justification is that: we observe the same fraction of sequences containing emoji in both INFORMATIVE and UNINFORMATIVE data, indicating that information from emoji is unlikely to be useful for this classification task.

For the tweet domain, there are typical elements such as hashtags, URL-links, mentions that needed to be handled. However, the given dataset has already replaced URL-links and mentions with special tokens @USER and @HTTPURL respectively. We choose to discard all other non-English languages in the dataset and keep hashtags elements.

4.2 Model's hyper-parameters settings and system configuration

In order to test out the effect of all model's settings in our experiments, we prefix hyper-parameters

Model	**Precision**	**Recall**	**F1**	**Accuracy**
BiLSTM + 2DCNN (*)	77.89	83.31	80.51	80.90
Non Static CNN (*)	81.63	80.38	81.00	82.18
HAN + WD-GRU (*)	82.65	82.08	82.36	83.40
RoBERTa Base	87.60	92.97	90.21	90.46
RoBERTa Large	**88.40**	**92.84**	**90.57**	**90.86**

Table 1: Comparing pretraining using RoberTa and training from scratch with previous SOTA models on text classification. (*) denotes for our implementation.

Model	**Precision**	**Recall**	**F1**	**Accuracy**
RoBERTa Base	87.60	92.97	90.21	90.46
RoBERTa Base + DLr	87.84	93.09	90.39	90.64
RoBERTa Base + DLr + Head	88.66	92.33	90.46	90.80
RoBERTa Base + DLr + Head + GU	88.40	93.22	90.75	91.02
RoBERTa Large	88.40	92.84	90.57	90.86
RoBERTa Large + DLr	87.84	93.74	90.69	90.90
RoBERTa Large + DLr + Head	**88.75**	92.84	90.75	91.06
RoBERTa Large + DLr + Head + GU	88.15	**93.96**	**90.96**	**91.14**

Table 2: Comparing models using RoBERTa Base and RoBERTa Large as backbone. DLr denotes Discrimination Learning Rate, Head denotes Custom Head and GU denotes Gradual Unfreezing.

across all settings except for the number of minibatch sizes and the number of training epochs. Due to the limits of GPU memory, each model uses RoBERTa Base has a minibatch of size 50 whereas those of model using RoBERTa Large are 10. We train with the number of epochs which equals the number of encoder layers plus 1 and 10 epochs for the model using and not-using Gradual unfreezing, the final result is averaged results of 10 times training with different random seeds. In a different setting, we set LSTM's hidden size is the same as RoBERTa hidden size, the kernel size of 1D CNN is 3. The maximum learning rate using in the one-cycle policy is 1e-5. Our choice of optimizer is AdamW (Loshchilov and Hutter, 2017). All out models were trained using only one GPU Tesla T4 with 16GB memory.

4.3 Evaluation metrics

We report our results with Accuracy, F1-score, Recall, and Precision metrics on the validation set. However, most of our analysis focuses on F1-score since it is the harmonic mean of Precision and Recall and it is a better representative of performance than Accuracy when the data is not perfectly balanced.

4.4 Results and analysis

Table 1 compared the performance of several architectures training from scratch and pre-training method using RoBERTa in our base settings. These traditional architecture are Hierarchical Attention Networks (Yang et al., 2016) combined with Weight-Dropped GRU (Merity et al., 2017) (HAN + WD-GRU), Bidirectional Long Short-Term Memory Networks with Two Dimensional Convolutional Neural Network (BiLSTM + 2DCNN) (Zhou et al., 2016) and Non Static Convolutional Neural Network (Kim, 2014) (Non-static CNN) . As far as we expect, the pre-training method significantly out-performs training from scratch with traditional architecture results in a gain of around 8.0 F1-score.

Table 2 compared the performance of 4 models using RoBERTa Base and 4 models using RoBERTa Large as the backbone. Each backbone is tested out with a base setting (the only backbone with a linear classifier, details are described in section 3), then that base setting is gradually added with fine-tuning techniques (DLr is discriminative learning rate; GU is gradual Unfreezing) and our custom head.

There are two standing out observations in table 2. Firstly, RoBERTa Large outperforms RoBERTa Base in all model settings. Which meets our expectations.

Secondly, gradually adding up training technique and custom head leads to a gradual increase in F1-score. This emphasizes the positive effect of all techniques and custom heads on this task's performance. Overall, our winning model results in an 89.89 F1-score.

5 Conclusion

In this paper, we experiment with the effectiveness of transfer learning using state-of-the-art language pre-trained model RoBERTa with the incorporation of several fine-tuning and training techniques for the informative tweet identifying the task. Our best model results in high F1-scores in both public and test dataset.

For future work, we would like to explore the effectiveness of different text augmentation strategies and task-adaptive pre-training instead of only fine-tuning the classification task.

References

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

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

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

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy.

Yoon Kim. 2014. Convolutional neural networks for sentence 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.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2017. Regularizing and optimizing lstm language models.

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

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.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations.

A. Radford. 2018. Improving language understanding by generative pre-training.

Leslie N. Smith. 2018. A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California. Association for Computational Linguistics.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3320–3328. Curran Associates, Inc.

Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. 2016. Text classification improved by integrating bidirectional lstm with two-dimensional max pooling.