Learning From How Human Correct

SUBMISSION DATE / POSTED DATE

27-01-2021 / 22-03-2021

CITATION

Guo, Tong (2021): Learning From How Human Correct. TechRxiv. Preprint.
https://doi.org/10.36227/techrxiv.13647974.v2

DOI

10.36227/techrxiv.13647974.v2

Learning From How Human Correct

Tong Guo

779222056@qq.com

Abstract. In industry NLP application, our manually labeled data has a certain number of noisy data. We present a simple method to find the noisy data and relabel them manually, meanwhile we collect the correction information. Then we present novel method to incorporate the human correction information into deep learning model. Human know how to correct noisy data. So the correction information can be inject into deep learning model. We do the experiment on our own text classification dataset, which is manually labeled, because we relabel the noisy data in our dataset for our industry application. The experiment result shows that our method improve the classification accuracy from 91.7% to 92.5%. The 91.7% baseline is based on BERT training on the corrected dataset, which is hard to surpass.

Keywords: Deep Learning · Text Classification

1 Introduction

In recent years, deep learning [2] and BERT-based [1] model have shown significant improvement on almost all the NLP tasks. However, past methods did not inject human correction information into the deep learning model. Human interact with the environment and learn from the feedback from environment to correct the their own error or mistake. Our method try to solve the problem that let deep learning model imitate how human correct.

In order to solve the problem we present a learning framework. The framework mainly works for our industry application dataset, because the problem starts from our industry application. In order to solve the text classification problem in our industry application, we first label a dataset. Then we can find the noisy data in the dataset and relabel them. The relabeling step collects the human correction information.

To the best of our knowledge, this is the first study exploring the improvement of injecting human correction information into deep model for natural language understanding. Our key contribution are 4 folds:

1. Based on our dataset, we first present the simple step to find the noisy data and relabel the noisy data.

2. We present the method to inject the human correct information into BERT for text classification.

3. The experiment result shows our framework gets gain of 0.8% accuracy against the strong baseline.

4. Our learning framework can apply to a broad set of deep learning industry applications whose dataset is manually labeled first.

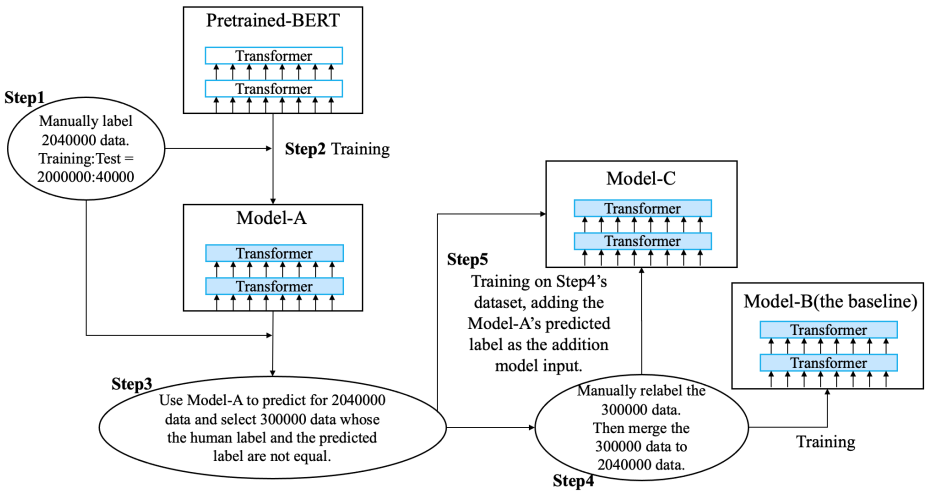

Fig. 1. Our learning framework

2 Relate work

BERT [1] is constructed by the multi-layer transformer encoder [10], which produces contextual token representations that have been pre-trained from unlabeled text and fine-tuned for the supervised downstream tasks. BERT achieved state-of-the-art results on many sentence-level tasks from the GLUE benchmark [3]. There are two steps in BERT's framework: pre-training and fine-tuning. During pre-training, the model is trained on unlabeled data by using masked language model task and next sentence prediction task. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks.

Our method is different to semi-supervised learning. Semi-supervised learning solve the problem that making best use of a large amount of unlabeled data. These works include UDA [6], Mixmatch [7], Fixmatch [8], Remixmatch [9]. These works do not have the human correction information. Our work has a clear goal that is to learn how human correct their mistake.

3 Our method

In this section, we describe our method in detail. Our learning framework is shown in Fig 1. The framework includes 5 steps:

Step 1, in order to solve the industry text classification problem. We label 2,040,000 data and split them into 2,000,000 training data and 40,000 test data. The 2,040,000 data are sampled from our application database whose data size is 500 million.

Step 2, we train / fine-tune the BERT model on the 2,000,000 training data. We named the result model of this step Model-A.

Step 3, we use Model-A to predict for all the 2,040,000 data. Then we find 300,000 data whose predicted label and human label are not equal. We consider it is the noisy data. In detail, there are 294,120 noisy data in the training dataset and 5,880 noisy data in the test dataset.

Step 4, we manually relabel 300,000 data and merge back to the 2,040,000 data. Then we get the merged 2,040,000 data. During the relabeling, the last label by human and the model-A's predicted label are listed as the references for people. But labeling people also make their own decision.

Step 5, we add the Model-A's predicted one-hot label as the addition input for training / fine-tuning a new model. The detail encoding method for the predicted label is described in the next section. We named the result model of this step Model-C. The Model-A's predicted one-hot labels represent the before-corrected information. The training ground truth for Model-C contains the 294,120 corrected human label, which represent the corrected information. So Model-C is learning how to correct and learning the text classification task in the same time.

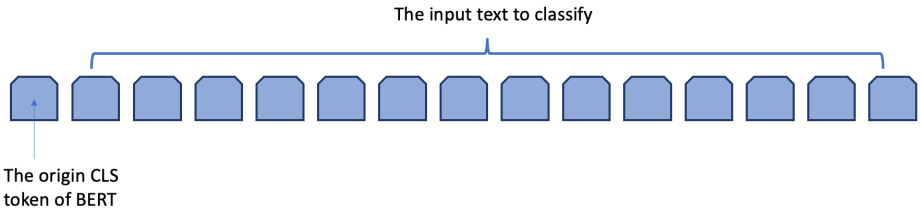

Fig. 2. The encoding detail for Model-A, which is corresponding to Fig 1.

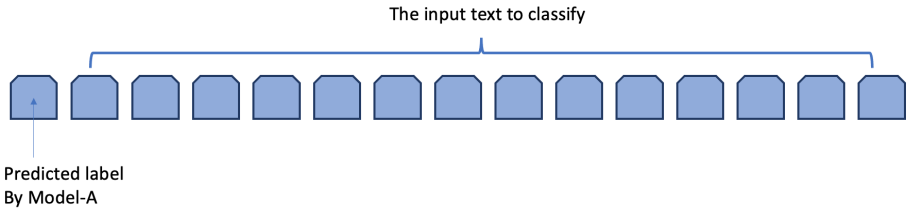

Fig. 3. The encoding detail for Model-C, which is corresponding to Fig 1.

4 The Model

We use BERT as our model. The training steps in our method belongs to the fine-tuning step in BERT. We follow the BERT convention to encode the input text. The encoding detail for Model-A is shown in Fig 2. The encoding detail for Model-C is shown in Fig 3.

5 Experiments

In this section we describe detail of experiment parameters and show the experiment result. The detail result is shown in Table 2. The data size in our experiment is shown in Table 1.

In fine-tuning, we use Adam [4] with learning rate of 1e-5 and use a dropout [5] probability of 0.1 on all layers. We use BERT-Base (12 layer, 768 hidden size) as our pre-trained model.

Table 1. The data size in our experiment.

Data Size	Description
500 million	All the data in our application database.
2,040,000	The data we label in step 1 of Fig 1.
300,000	All the noisy data we select from the 2,040,000 data to relabel.
294,120	The noisy data in the training dataset.
40,000	The test data we split from the 2,040,000 data as test dataset.
5,880	The noisy data in the testing dataset.

Table 2. The experiment result. The test dataset is the 40,000 data. The accuracy reported by human is 5000 data that sampled from the 500 million data. Model-A, Model-B and Model-C are corresponding to Fig 1.

Model In Fig 1	Test Dataset Accuracy	Human Evaluate Accuracy
Model-A	83.3%	88.0%
Model-B	91.7%	97.2%
Model-C	92.5%	97.7%

6 Analysis

In step 4 of Fig 1, the manually relabeling can correct the noisy 300,000 data. Because the selected 300,000 data is unfitting to the Model-A, the relabeling's 'worst' result is the origin last human's label.

The step 5 in Fig 1 is the core contribution of our work. In step 5, the predicted label of Model-A contains the before-human-corrected information. The ground truth for Model-C contains the after-human-corrected information. So the model is learning the human correction.

We could use the before-corrected human label (i.e., the ground truth in step 1 of Fig 1) as the input for Model-C. But this way can not apply to real industry inference. Because we can not get the before-corrected human label as the input of Model-C in real industry application. In real industry application, we use the Model-A to predict one-hot label as input for Model-C.

Human evaluation accuracy is higher than the test dataset accuracy, because we randomly sampled 5000 data from the 500 million data. The sampled 5000 data represents the great majority of the 500 million data.

7 Conclusion

Human interact with environment and learn from the feedback from environment to correct human's own error. Base on the human's correction idea, we design a learning framework to inject the information of human's correction into the deep model. The experiment result shows our idea works. Our idea can apply to a broad set of deep learning industry applications.

References

1. Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.
2. Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Advances in neural information processing systems, 2012, 25: 1097-1105.
3. Wang A, Singh A, Michael J, et al. GLUE: A multi-task benchmark and analysis platform for natural language understanding[J]. arXiv preprint arXiv:1804.07461, 2018.
4. Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.
5. Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. The journal of machine learning research, 2014, 15(1): 1929-1958.
6. Xie Q, Dai Z, Hovy E, et al. Unsupervised data augmentation for consistency training[J]. arXiv preprint arXiv:1904.12848, 2019.
7. Berthelot D, Carlini N, Goodfellow I, et al. Mixmatch: A holistic approach to semi-supervised learning[J]. arXiv preprint arXiv:1905.02249, 2019.
8. Sohn K, Berthelot D, Li C L, et al. Fixmatch: Simplifying semi-supervised learning with consistency and confidence[J]. arXiv preprint arXiv:2001.07685, 2020.
9. Berthelot D, Carlini N, Cubuk E D, et al. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring[J]. arXiv preprint arXiv:1911.09785, 2019.
10. Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in neural information processing systems. 2017: 5998-6008.