Quora Insincere Questions Classification

Cheng Zeng

June 2020

1 Introduction

For the final project, we the select a complete for text classification. We select **Quora Insincere Questions Classification** [1]. The task is handle toxic and divisive content.. We use BERT [1] model provided by Google to train a classifier. To implement our model, we applied Tensorflow [2],Sklearn [3] and Pandas [4] to process data and modeling. Next we will introduce the description of the task and data. we Finally achieve the F1 score

1.1 Background

Insincere Sentence existential problem for any major website today is how to handle toxic and divisive content. Quora wants to tackle this problem head-on to keep their platform a place where users can feel safe sharing their knowledge with the world.

Quora is a platform that empowers people to learn from each other. On Quora, people can ask questions and connect with others who contribute unique insights and quality answers. A key challenge is to weed out insincere questions – those founded upon false premises, or that intend to make a statement rather than look for helpful answers.

In this competition, Kagglers will develop models that identify and flag insincere questions. To date, Quora has employed both machine learning and manual review to address this problem. With your help, they can develop more scalable methods to detect toxic and misleading content.

2 Methodology

The BERT model newly released by the Google AI team has shown amazing results in SQuAD1.1, the top-level test of machine reading comprehension: all two metrics surpass humans comprehensively, and it also achieved the best results in 11 different NLP tests. , Including pushing the GLUE benchmark to 80.4% (absolute improvement 7.6%), MultiNLI accuracy reaching 86.7% (absolute improvement rate 5.6%), etc. It is foreseeable that BERT will bring milestone changes to NLP, which is also the most important recent development in the field of NLP.

3 BERT

BERT uses the Transformer [3] Encoder model as the language model. BERT completely abandons RNN/CNN and other structures, and completely adopts the Attention mechanism to calculate the relationship between input-output. The Figure 1 show the detail of Transformer architecture. The model contain two sublayer.

1. Multi-Head Attention is the Self-attention that applied for input.
2. Feed Forward is used for convert for transformation of attention.

The BERT model is shown as the first one on the Figure 2. The difference between it and OpenAI GPT [2] is that Transformer Encoder is used, that is, the Attention calculation at each moment can get the input

[1] https://www.kaggle.com/c/quora-insincere-questions-classification
[2] https://tensorflow.google.cn/
[3] https://scikit-learn.org/stable/
[4] https://pandas.pydata.org/

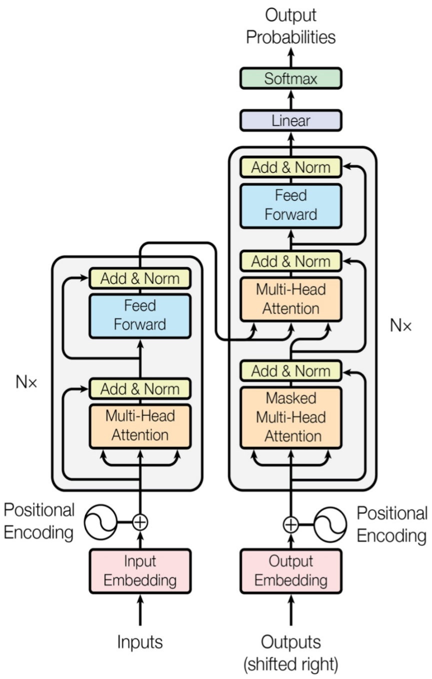

Figure 1: The Transformer - model architecture.

Figure 1: Transformer

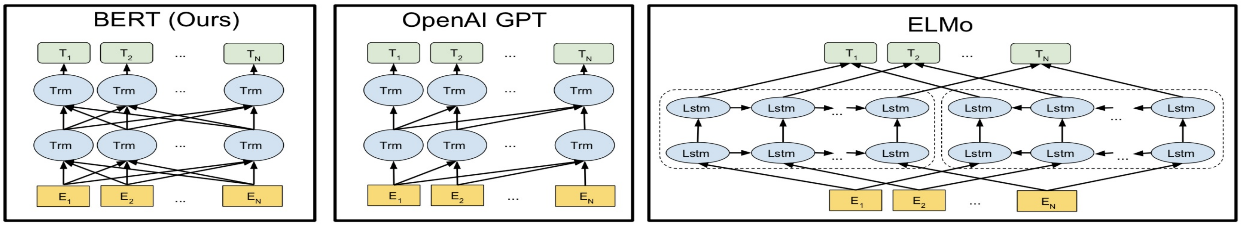

Figure 2: Three models

at all moments, and OpenAI GPT uses Transformer Decoder, each Attention calculation at this moment can only depend on the input at all moments before that moment, because OpenAI GPT uses a one-way language model.

Next, we introduce BERT's Pre-training tasks. Here, in order to facilitate token-level tasks such as sequence annotation, and at the same time benefit token-level tasks such as question and answer, two pre-training tasks are used.

3.1 Masked Language Model

The problem with the existing language model is that Bidirectional information is not used at the same time. Existing language models such as ELMo are known as bidirectional LM (BiLM), but in fact it is a splicing of two unidirectional RNN language models, as shown in Figure 3.

Because of the probability of a sentence can be calculated as follows:

$$p(S) = p(w_1, w_2, ..., w_m) = \prod_{i=1}^{m} p(w_i|w_1, w_2, ..., w_{i-1}) \tag{1}$$

The operation here is to randomize 15% of the tokens that will be masked in corpus, and then predict the

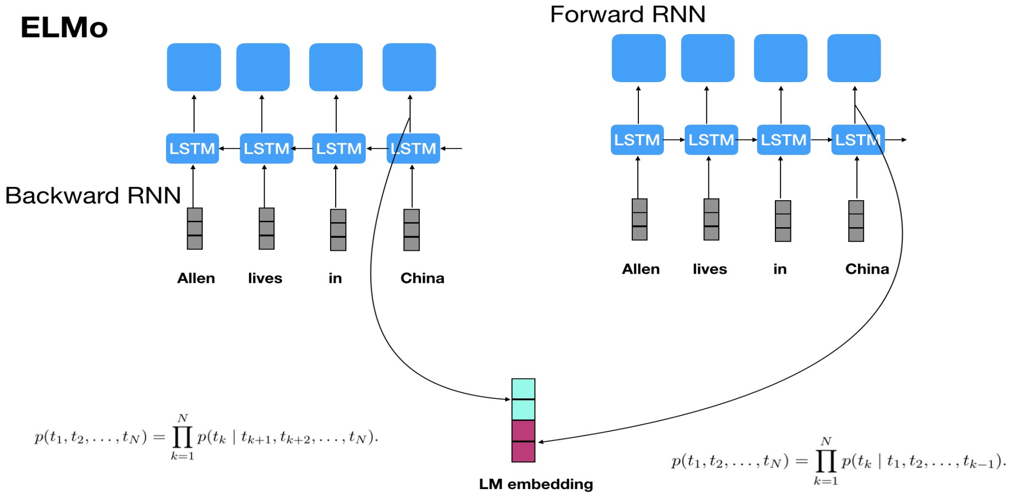
ELMo
Forward RNN
Backward RNN
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
Allen
lives
in
China
Allen
lives
in
China
$p(t_1, t_2, \ldots, t_N) = \prod_{k=1}^{N} p(t_k \mid t_{k+1}, t_{k+2}, \ldots, t_N).$
LM embedding
$p(t_1, t_2, \ldots, t_N) = \prod_{k=1}^{N} p(t_k \mid t_1, t_2, \ldots, t_{k-1}).$

Figure 3: EMLo model

Input = `[CLS] the man went to [MASK] store [SEP]`
`he bought a gallon [MASK] milk [SEP]`
Label = `IsNext`

Input = `[CLS] the man [MASK] to the store [SEP]`
`penguin [MASK] are flight ##less birds [SEP]`
Label = `NotNext`

Figure 4: EMLo model

masked token. Then the final hidden vectors output from the masked token position are fed to the softmax network to obtain the predicted result of the masked token.

There is a problem with this operation. There is no [MASK] token during fine-tuning, so there is a mismatch between pre-training and fine-tuning. In order to solve this problem, the following strategy is adopted:

- Token will be replace for 80%
- Selected tokens will be replaced by others for 10%.
- Tokens will not be replaced for 10%.

3.2 Next Sentence Prediction

Many NLP tasks that need to be solved depend on the relationship between sentences, such as question and answer tasks. This relational language model is not available, so the next sentence prediction is used as the second pre-training task. The training corpus for this task is two sentences to predict whether the second sentence is the next sentence of the first sentence, as shown in Figure 4

3.3 Input

After introducing two pre-training tasks, we introduce how the model constructs inputs. As shown in the figure below, the input includes the sum of three embedding, which are:

Parameter	batch size	epoch	maximum length	learning rate
value	128	2	32	0.0005

Table 1: Parameter setting

- Token embedding, which present the embedding of current words.
- Segment Embedding, which present the position of the current sentence.
- Position Embedding, which present the position of current word.

3.4 Model Training

- BERT-Base: L=12,H=768,A=12,Total parameters=110M
- BERT-Large: L = 24, H = 1024, A = 16, Total parameters = 340M

Where L stands for the number of layer in Transformers, H represents the dimension of Transformer and A represent for the number of Heads.

4 Training & Results

The data set has over 200,000 records. However the data is significantly imbalanced. The ratio of positive and negative is 9:1. When train our model, we set the parameters shown in Table 1. And we train split the training set and test set with the ratio 8:2. The result show we achieve 0.96 in accuracy and 0.65 in F1-score.

The result is not much perform well in this tasks. In the future we can apply many preprocessing method to improve the performance of our model.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[2] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.