Practical Deep Learning System Performance

Weiyao Xie, Ruoyu Li

April 2^{ed} 2021

1 Abstract

In these years, NLP models have evolved so fast that most state of the art models can surpass human behaviours on GLUE benchmarks easily. SuperGlue has been a very powerful successor of Glue Benchmarks which contains a range of new tasks that are more challenging than those in GLUE benchmark. The research team created SuperGLUE for two reasons. One is to better record how NLP models are progressing towards general purpose Natural Language Understanding. Second is by increasing the difficulty of the tasks, it provides researchers with more improvement room. Our final project explores the effectiveness of fine tuning BERT using different SuperGlue tasks. The model we use here is the BERT base model. We add a classification head on top of the fine tuned BERT model. Then we measure the fine tuned model performance on a downstream sentiment analysis task we picked.

2 SuperGLUE

BoolQ

1.1 SuperGLUE task and Downstream Task

The downstream task we picked is a sentiment analysis task. The dataset contains paragraph of reviews written by IMDB users on the internet. Each data


```
{
  "question": "is france the same timezone as the uk",
  "passage": "At the Liberation of France in the summer of 1944, Metropolitan France kept GMT+2 as i
  "answer": false,
  "title": "Time in France",
}
```


Figure 1: BoolQ example

Label	Target	Context-1	Context-2
F	bed	There's a lot of trash on the bed of the river	I keep a glass of water next to my bed when I sleep

Figure 2: WiC example

entry contains a paragraph of review, and a binary label indicating whether this review is positive or negative. This task is not trivial because it requires very detailed semantics to distinguish the sentiment contained in the movie reviews. Also some paragraphs are really long which makes it more difficult to extract the necessary semantics. Once we have the downstream task, we can focus on how to choose our fine tuning task. We picked out three tasks from SuperGLUE for transfer learning: BOOLQ, WiC, and COPA. Before we go into the reasons behind our choices, I want to first give some introductions on these tasks.

BoolQ is a question answering dataset for yes/no questions containing 15942 examples. These questions are naturally occurring which means they are generated in unprompted and unconstrained settings, and can contain diverse context and topics. Figure 1 is an example of a BoolQ data sample.

WiC

A system's task on the WiC dataset is to identify the intended meaning of words. WiC is framed as a binary classification task. Each instance in WiC has a target word w, either a verb or a noun, for which two contexts are provided. Each of these contexts triggers a specific meaning of w. The task is to identify if the occurrences of w in the two contexts correspond to the same meaning or not.

COPA

The Choice Of Plausible Alternatives (COPA) evaluation provides researchers with a tool for assessing progress in open-domain commonsense causal reasoning. COPA consists of 1000 questions, split equally into development and test sets of 500 questions each. Each question is composed of a premise and two alternatives, where the task is to select the alternative that more plausibly has a causal relation with the premise.

Premise: The man broke his toe. What was the CAUSE of this?
Alternative 1: He got a hole in his sock.
Alternative 2: He dropped a hammer on his foot.

Figure 3: COPA example

There are two reasons why we choose these three tasks. Firstly, they all use accuracy as the primary evaluation metrics, which is same as our downstream task. This makes it easy to evaluate how much knowledge learned from transfer learning. Secondly, since BoolQ contains context/toppics from different settings, we believe there will be a big overlap between BoolQ and IMDB dataset.

However, for the other two tasks, the contexts are very different from IMDB dataset, we think it will be interesting to contrast this difference when evaluating the final model.

3 Transfer Learning

We initialize three same BERT base models for these three transfer learning tasks. Then my goal is to train the models to the same level. By that I mean I want the models to learn the approximately the same amount knowledge from the tasks. To do so, I set a threshold of 70% accuracy, and trained each model to that accuracy. This is also the goal of my hyperparameters searching. The only problem is that COPA can never achieve 70% accuracy during fine tuning, probably due to the small amount of data and complex setting. The following table shows the hyperparameters I used to achieve the threshold accuracy.

SuperGLUE	Max sequence length	Batch Size	Learning rate	Epoch
COPA	128	32	2e-5	3
WiC	128	32	2e-5	6
BoolQ	128	32	2e-5	3

Table 1: Parameters to reach the threshold of 70% accuracy

4 Model Implementation

4.1 IMDB Data Preprocessing

IMDB data is a very raw {text, label} data found on `https://ai.stanford.edu/~amaas/data/sentiment/aclImdb_v1.tar.gz`. We first used tensorflow function "text_dataset_from_directory" to fetch the data from internet and store them in a pandas DataFrame. Then we did a trian_test_split on the dataset and transfer the data into tensors using the tokenizer trained in Bert and set the maximum length to 510.

Later, we defined our own pytorch Dataset - CommentDataset to transfer the input into the expected form of our Bert Model and load them into a DataLoader.

4.2 Transfer Learning

We used three fine-tuned Bert and the base Bert to do transfer learning on IMDB. Thus, we need to add a classification head to all the Bert models in order for them to learn to use the features extracted by Bert models. We simple use two fully-connected layers as follows:

```
self.linear_layer = nn.Linear(768, 32)
self.dropout = nn.Dropout(0.5)
self.non_linearity = nn.ReLU()
self.clf = nn.Linear(32, 1)
```

5 Model Evaluations

5.1 Expectations

5.1.1 BoolQ

BoolQ, as we know, is a question answering dataset for yes/no questions containing 15942 examples. Apparently, the pattern and relationship between input and output is very similar to IMDB. We expect the model performs very well on IMDB dataset.

5.1.2 COPA

COPA, as we know, is a high level Q&A tasks with open-domain commonsense casual reasoning. This kind of Q&A is too high level. The model fine-tuned with this data tends to extract more information that might benefit when dealing with IMDB dataset. From human perspective, we expect the model performs relatively well on IMDB dataset.

5.1.3 WIC

WIC is a task to identify if the occurrences of w in the two contexts correspond to the same meaning or not. This task requires model to focus on the context around certain words and such effect of context will decrease as the distance to the word increase. It would ignore or purposely decrease the effect of certain part of the sentence, which is unexpected for IMDB task. From human perspective, we expect the model performs badly on IMDB dataset.

5.2 Results

We can see from Figure.4 that as we expected the model trained on COPA actually outperforms its original dataset. And BoolQ, beyond our expectation outperforms a lot and actually has a very high evaluation accuracy on IMDB dataset. The reason might be that Boolq and COPA extract higher level of information needed by IMDB, which leads to a very good performance on IMDB. However, those information is not enough for Boolq and COPA.

However, for WIC, even though it has almost 70% accuracy on WIC dataset, it has only 50% accuracy on IMDB dataset. Fine-tuning with WIC has negative effect on the model. The best fine-tuning task we get is BoolQ. First, we don't need to fine-tune too long to get a high accuracy on BoolQ data. And the improvement on IMDB is surprisingly great!

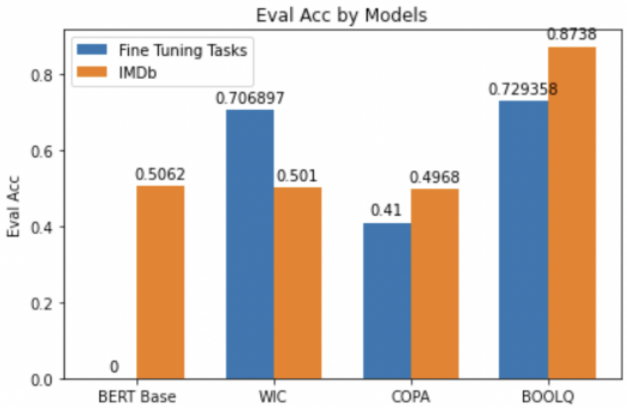

Figure 4: Evaluation Results

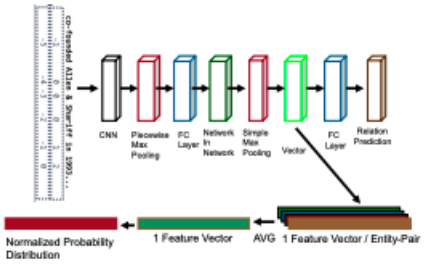

Figure 5: COPA example

6 Model Divergence

From the above results, we can see BoolQ has much better transfer learning performance on BERT than the other two tasks. We wonder what has caused this significant difference and want to make sense of it. The P2L paper we studied during class states that we are able to predict the effectiveness of different tasks on transfer learning by computing the divergence between their representation. In the original paper, the team use a vector to represent the entire image dataset.

From the above results, we can see BoolQ has much better transfer learning performance on BERT than the other two tasks. We wonder what has caused this significant difference and want to make sense of it. The P2L paper we studied during class states that we are able to predict the effectiveness of different tasks on transfer learning by computing the divergence between their representation. In the original paper, the team use a vector to represent the entire image dataset. The vector is obtained by getting the image features from the second last hidden layer from VGG-16 and then taking average of all images. The

overall conclusion of the paper is that the more similar the source dataset is to the target dataset, the more effective the source dataset is at transfer learning. Based on this work, we decided to apply the similar techniques on the SuperGLUE tasks. Since we are dealing with text data, we need to do some modification on the original method. We tried out two experiments. First, we use a sentence to represent an entire dataset. We extract the hidden states from the last hidden layer of BERT. The size is $24 * 768$ because we are padding the length of sentence to 24 and 768 is the embedding size we use in this BERT base model.

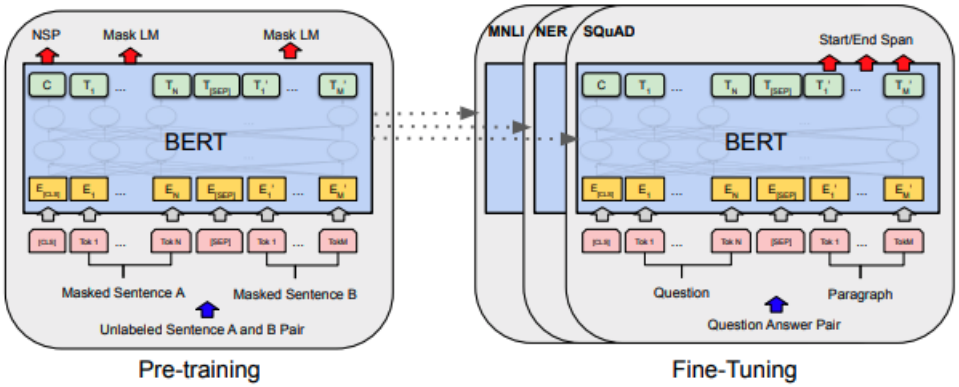

Figure 6: Extracting hidden states from BERT

With the vector representing each sentence in the dataset, we than take the average of all sentences in the dataset to get one vector with size $24 * 768$ that can be used to represent the entire dataset.
The second experiment is that we use one word to represent the entire dataset. Instead of geting the sentence representation which is composed by word embeddings, we now only use the $[CLS]$ token from BERT hidden state. $[CLS]$ is commonly used as an embedding in classification problem. Then we take the average of $[CLS]$ across all data samples and eventually get a vector of length 768. This is the word that we want to use to represent the dataset.
With the dataset representation vector, we can compute the divergence of dataset using these vectors. For easier interpretation and easier implementation, we are using L1 distance as our divergence metrics. The result is shown in the table below. , WiC = , BoolQ =

Task	Sentence Level Distance	Word Level Distance
COPA	23.698	4.26

WiC	31.357	3.40
BoolQ	56.765	8.64

Table 2: Divergence between SuperGLUE tasks and IMDB dataset

The resuls are very surprising because we expect the best performing BoolQ to have the closest distance to the IMDB dataset.

7 Conclusion

Obviously, there are a lot of work we can do to further improve this experiment. The main problem we have in the current experiment is the padding and truncating problem. Since we want to foucs on BoolQ due to the best perfoemance, we are adjusting all the other dataset. BoolQ has very short questions, so we have to truncate a lot texts in the IMDB dataset so that it can have the same size vector to represent the dataset. However, by applying the huge truncation on each data sample, we are potentially discarding a lot of important information. This may damage our experiment results and can be a possible explaination of why our results are so surprising. Another improvement we can make is that we can expand the model configuration. Instead of using 768 as our embedding size, we can choose a larger embedding size because in P2L paper, the vector size used to represent the datasize is larger than 4000, but we are using 768 here.

Overall, our experiments still show that if the correct tasks are choosen, SuperGLUE benchmarks can be a very good source to finetune the model.