ADL Homework 2 Report

Name: Chou Tun Hsiang **Student Number:** B07902050 **Date:** May 9 2021

1 Data Preprocessing

1.1 Tokenizer

The tokenization algorithm used by BERT, and also used in this homework, is called WordPiece, which first initializes the vocabulary to include every character present in the training data and progressively learns a given number of merge rules. In contrast to byte-pair encoding, WordPiece does not choose the most frequent symbol pair, but the one that maximizes the likelihood of the training data once added to the vocabulary.

1.2 Answer Span

(a) We can record the start and end character position of each token during tokenization, which allows us to find the answer span of tokens in the training stage.

(b) The top 20 start/end position with highest possibilities are chosen. Then the following rules are used to determine the final start and end positions of the answer.

1. The start/end position belongs to the context, not the question.
2. The start/end position is not a [cls] token.
3. The end position is not before the start position.
4. The length of the answer is shorter than 30 tokens.
5. The answer should be the combination of the start/end position with the highest possibilities.

2 Modeling with BERTs and their variants

2.1 BERT

(a) Model configuration:

- hidden size: 768
- number of hidden layers: 12
- number of attention heads: 12

- intermediate size: 3072
- max position embedding: 512
- activation: gelu

(b) Performance: EM = 0.771, F1 = 0.835 on public test set

(c) Loss function: cross entropy

(d) Training configuration:

- optimization algorithm: Adam with weight decay = 0.01
- learning rate: 2e-5
- batch size: 16
- epoch: 2

2.2 MacBERT

(a) Model configuration: the same with BERT

(b) Performance: EM = 0.815, F1 = 0.877 on public test set

(c) Comparison of BERT and MacBERT

There is no differences in the main neural architecture between BERT and MacBERT. The latter tries to improve BERT in the pre-training task. Instead of masking with [MASK] token, which never appears in the fine-tuning stage, similar words are used for the masking purpose. A similar word is obtained by using Synonyms toolkit (Wang and Hu, 2017), which is based on word2vec (Mikolov et al., 2013) similarity calculations.

3 Curves

The learning curve (check figure 1. below) illustrates the performance of MacBERT on the validation set, which makes up 10% of the original train set. One step is defined as one backward pass.

4 Pretrained vs Not-Pretrained

I trained DistilBert a from scratch for comparison with pretrained BERT model.

Model configuration:

- hidden size: 768

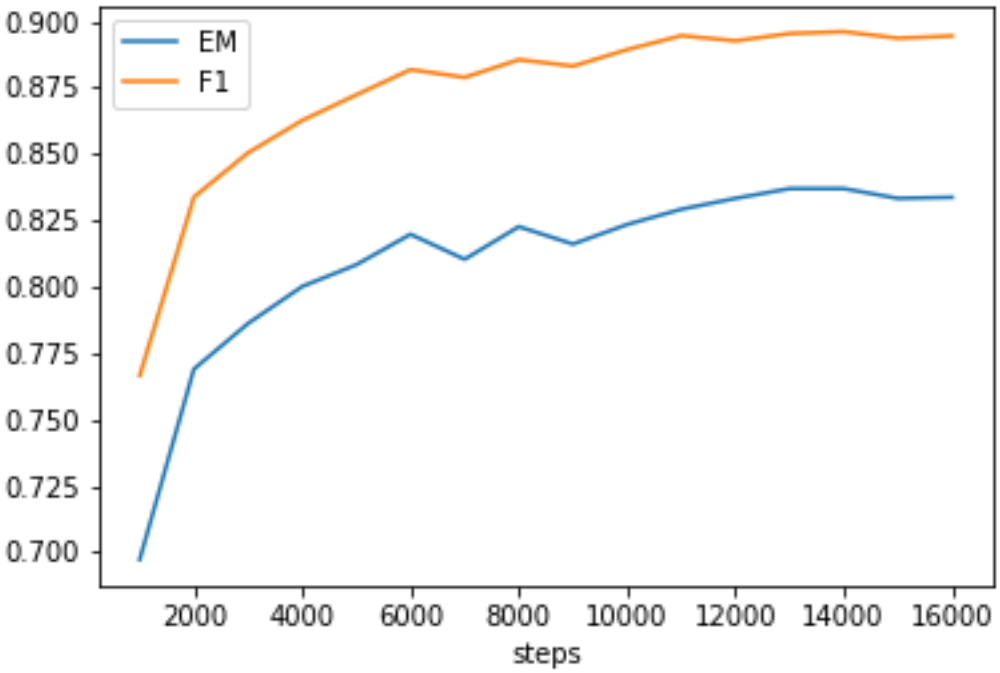

Figure 1: Learning curve of MacBert

- number of hidden layers: 6
- number of attention heads: 12
- intermediate size: 3072
- max position embedding: 512
- activation: gelu

Training configuration:

- loss function: cross entropy
- optimization algorithm: Adam with weight decay = 0.01
- learning rate: 2e-5
- batch size: 16
- epoch: 2

The unpretrained model is trained with the same configurations of BERT, and attains EM = 0.012 and F1 = 0.021 on public test set. Compared to a EM score of 0.771 and F1 score of 0.835 achieved by BERT, it reveals the importance of the pretrained weights.