ADL Homework 2 Report

Name：Liu Chia Ming **Student Number：**R10H41007 **Date：**November 11 2022

A. Data Processing：

1) Tokenizer

The tokenizer for bert-base-chinese (and almost every chinese BERT model) is character-based. That is, a chinese character is considered as a “token”. However, some non-chinese character are still tokenized at a vocabulary-level, (e.x. some numbers, english words). Additionally, there are some special tokens added to the tokenized sentence to serve special purpose. [PAD] is for padding, [UNK] is for unknown token, [MASK] is for the masked-language-modeling (MLM) pre-training objective, [CLS] is added to the beginning of the sentence for classification purpose, and [SEP] is for partitioning sentences if required. The following example shows the real tokenization result performed by the bert-base-chinese tokenizer：

- Original sentence: [你有幾顆 apples]
- Tokenized sentence: [[CLS], 你, 有, 幾, 顆, apple, ##s, [SEP]]

2) Answer Span

2-1)

After bert-tokenization, the original positions of the tokens can be return by offset mapping. The token whose original positions correspond to / contain the start position is the start token, and the token whose original positions correspond to / contain the end position is the end token.

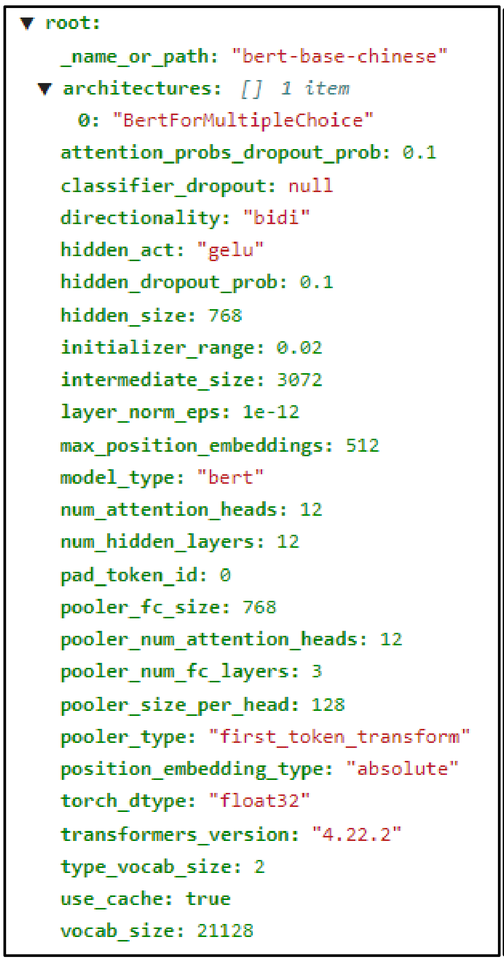
2-2)

For each pair (i.e. start position and end position), taking the probability product and remove all incompatible pair (i.e. start position > end position or subsentence longer than sentence), then we can find the pair which have maximum probability, serve it as our final prediction.

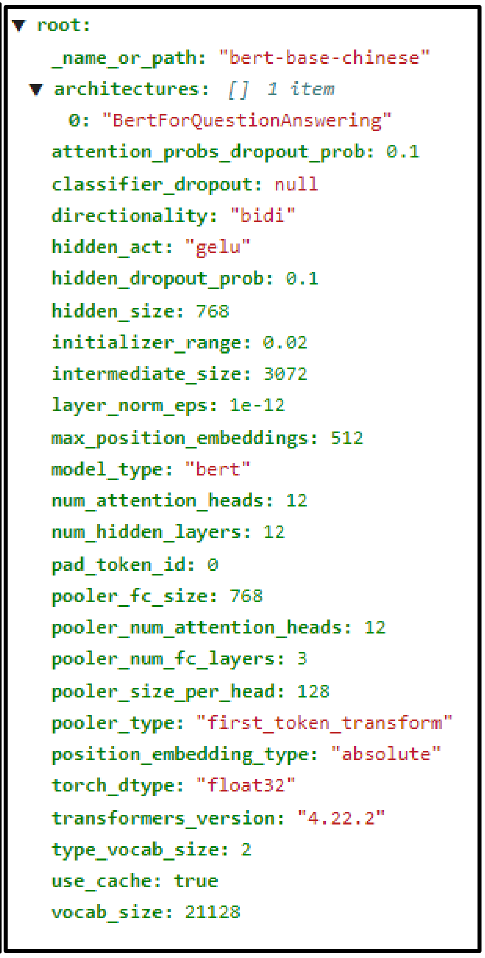
B. Model with BERTs and their Variants：

1) Bert Model：

1-1)



```
▼ root:
    _name_or_path: "bert-base-chinese"
  ▼ architectures: [] 1 item
      0: "BertForMultipleChoice"
    attention_probs_dropout_prob: 0.1
    classifier_dropout: null
    directionality: "bidi"
    hidden_act: "gelu"
    hidden_dropout_prob: 0.1
    hidden_size: 768
    initializer_range: 0.02
    intermediate_size: 3072
    layer_norm_eps: 1e-12
    max_position_embeddings: 512
    model_type: "bert"
    num_attention_heads: 12
    num_hidden_layers: 12
    pad_token_id: 0
    pooler_fc_size: 768
    pooler_num_attention_heads: 12
    pooler_num_fc_layers: 3
    pooler_size_per_head: 128
    pooler_type: "first_token_transform"
    position_embedding_type: "absolute"
    torch_dtype: "float32"
    transformers_version: "4.22.2"
    type_vocab_size: 2
    use_cache: true
    vocab_size: 21128
```



```
▼ root:
    _name_or_path: "bert-base-chinese"
  ▼ architectures: [] 1 item
      0: "BertForQuestionAnswering"
    attention_probs_dropout_prob: 0.1
    classifier_dropout: null
    directionality: "bidi"
    hidden_act: "gelu"
    hidden_dropout_prob: 0.1
    hidden_size: 768
    initializer_range: 0.02
    intermediate_size: 3072
    layer_norm_eps: 1e-12
    max_position_embeddings: 512
    model_type: "bert"
    num_attention_heads: 12
    num_hidden_layers: 12
    pad_token_id: 0
    pooler_fc_size: 768
    pooler_num_attention_heads: 12
    pooler_num_fc_layers: 3
    pooler_size_per_head: 128
    pooler_type: "first_token_transform"
    position_embedding_type: "absolute"
    torch_dtype: "float32"
    transformers_version: "4.22.2"
    type_vocab_size: 2
    use_cache: true
    vocab_size: 21128
```

1-2) Performance：**EM**

- Kaggle Public：0.72875
- Kaggle Private：0.7579

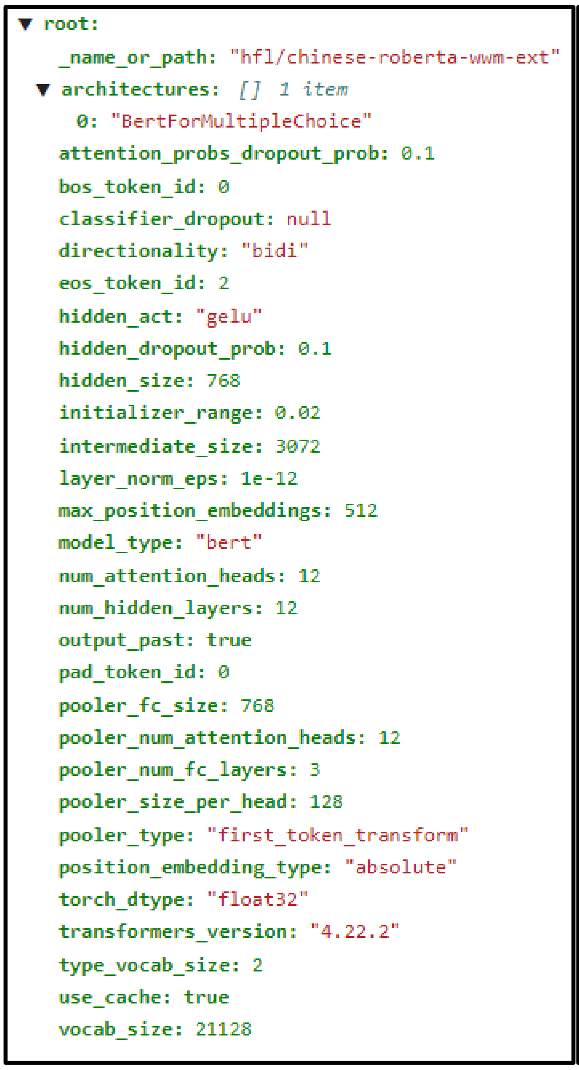
1-3) Loss Function： Cross Entropy

1-4) Training setting are both same between context selection and span selection

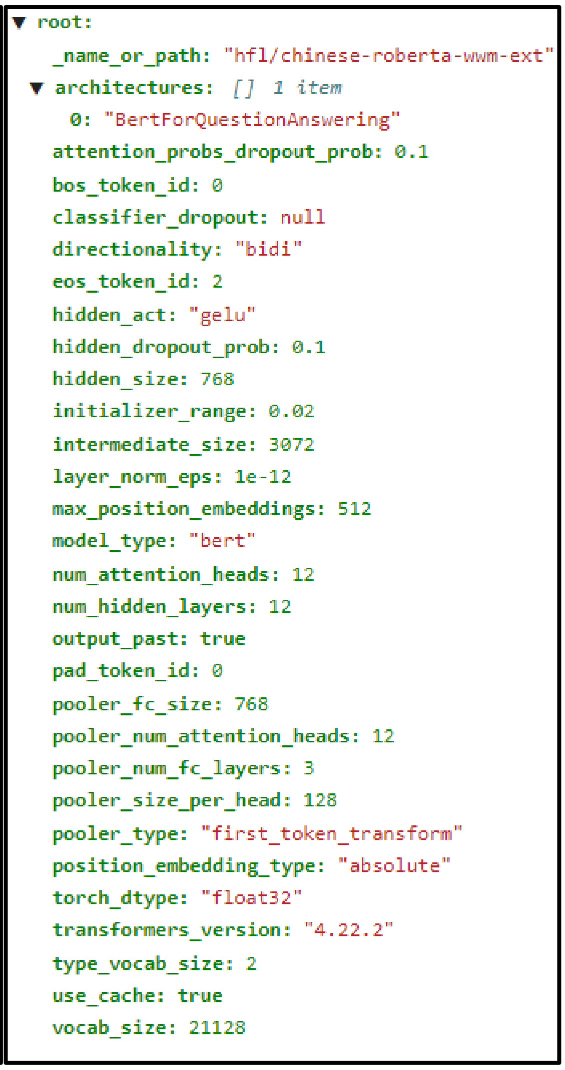
- Optimization Algorithm：AdamW with lr=3e-5
- lr scheduler: linear scheduler with warmup, warmup_ratio = 0.1
- batch size: 2
- gradient accumulation step: 2

2) hf1/Roberta-wwm-ext

2-1)



```
▼ root:
    _name_or_path: "hfl/chinese-roberta-wwm-ext"
  ▼ architectures: [] 1 item
      0: "BertForMultipleChoice"
    attention_probs_dropout_prob: 0.1
    bos_token_id: 0
    classifier_dropout: null
    directionality: "bidi"
    eos_token_id: 2
    hidden_act: "gelu"
    hidden_dropout_prob: 0.1
    hidden_size: 768
    initializer_range: 0.02
    intermediate_size: 3072
    layer_norm_eps: 1e-12
    max_position_embeddings: 512
    model_type: "bert"
    num_attention_heads: 12
    num_hidden_layers: 12
    output_past: true
    pad_token_id: 0
    pooler_fc_size: 768
    pooler_num_attention_heads: 12
    pooler_num_fc_layers: 3
    pooler_size_per_head: 128
    pooler_type: "first_token_transform"
    position_embedding_type: "absolute"
    torch_dtype: "float32"
    transformers_version: "4.22.2"
    type_vocab_size: 2
    use_cache: true
    vocab_size: 21128
```



```
▼ root:
    _name_or_path: "hfl/chinese-roberta-wwm-ext"
  ▼ architectures: [] 1 item
      0: "BertForQuestionAnswering"
    attention_probs_dropout_prob: 0.1
    bos_token_id: 0
    classifier_dropout: null
    directionality: "bidi"
    eos_token_id: 2
    hidden_act: "gelu"
    hidden_dropout_prob: 0.1
    hidden_size: 768
    initializer_range: 0.02
    intermediate_size: 3072
    layer_norm_eps: 1e-12
    max_position_embeddings: 512
    model_type: "bert"
    num_attention_heads: 12
    num_hidden_layers: 12
    output_past: true
    pad_token_id: 0
    pooler_fc_size: 768
    pooler_num_attention_heads: 12
    pooler_num_fc_layers: 3
    pooler_size_per_head: 128
    pooler_type: "first_token_transform"
    position_embedding_type: "absolute"
    torch_dtype: "float32"
    transformers_version: "4.22.2"
    type_vocab_size: 2
    use_cache: true
    vocab_size: 21128
```

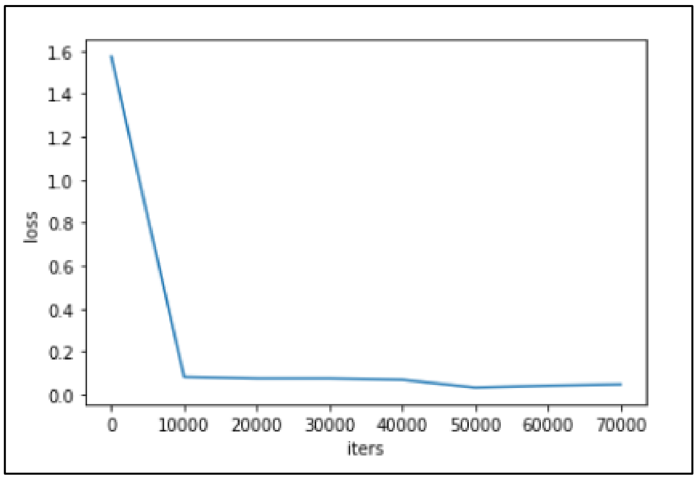
2-2) Performance：**EM**

- Kaggle Public：0.7839
- Kaggle Private：0.79945

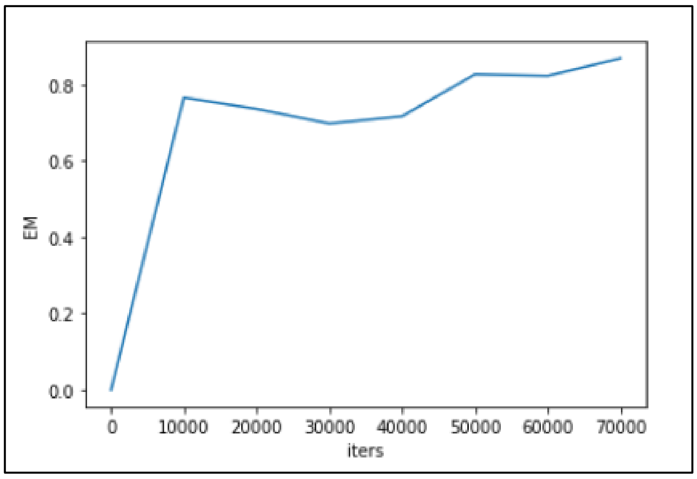
2-3) The difference between pretrained model for bert-base-chinese and chinses-roberta-wwm-ext have two part. First, the BERT model leverages two pre-training tasks, masked-language modeling (MLM) and next sentence prediction (NSP), while RoBERTa only uses MLM to pre-train the model.
Second, during pre-training, BERT randomly masks WordPiece tokens without constraints, while chinese-roberta-wwm-ext uses whole word masking (WWM) to only mask Chinese phrases segmented by segmentation tool to maintain the integrity of the language.

C. Curve：

1) Loss

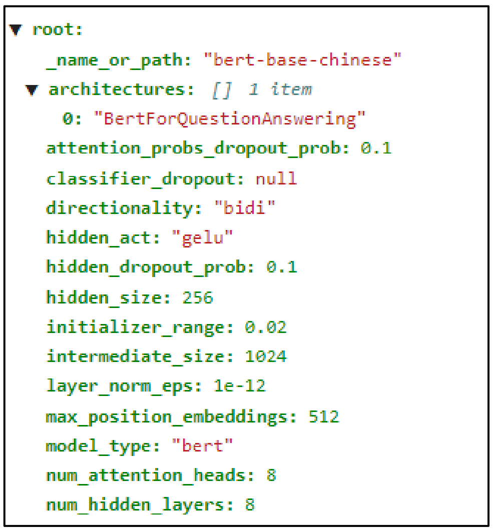


2) EM



D. Pretrained vs Not Pretrained ：

1) I directly use Q2 bert-base-chinse for QA model and <u>change to not load pretrained weight</u>, and reduce number of hidden size, layer and Attention head, configs are showed below.



```
▼ root:
    _name_or_path: "bert-base-chinese"
  ▼ architectures: [] 1 item
      0: "BertForQuestionAnswering"
    attention_probs_dropout_prob: 0.1
    classifier_dropout: null
    directionality: "bidi"
    hidden_act: "gelu"
    hidden_dropout_prob: 0.1
    hidden_size: 256
    initializer_range: 0.02
    intermediate_size: 1024
    layer_norm_eps: 1e-12
    max_position_embeddings: 512
    model_type: "bert"
    num_attention_heads: 8
    num_hidden_layers: 8
```

- Valid Set **EM** Performance： (**non-Pretrained**, Bert-Pretrained) = (**0.06983**, 0.7713)
- Comparsion：We can find that pre-trained model outperforms not pre-trained model very

large. I think that the reason is the framework of transformer are so huge, if we don’t have enough data to train, it will easily go to overfitting, so for hw2 condition, it better to use pre-trained model.