Q1:

Tokenizer:

I use BertTpkenizerFast to convert each word that can be recognized by Bert-based model into a unique id, and special tokens [CLS],[SEP], and [UNK] are representations of classification tokens, separation tokens, and unknown words respectively.

Example for context selection:

[CLS] tokenized question... [SEP] tokenized context 1... [SEP] [PAD]...
[CLS] tokenized question... [SEP] tokenized context 2... [SEP] [PAD]...
[CLS] tokenized question... [SEP] tokenized context 3... [SEP] [PAD]...
[CLS] tokenized question... [SEP] tokenized context 4... [SEP] [PAD]...

Example for QA :

[CLS] tokenized question... [SEP] tokenized context split 1... [SEP]
[CLS] tokenized question... [SEP] tokenized context split2... [SEP]
.
.
.
[CLS] tokenized question... [SEP] tokenized context split n... [SEP] 0,0,0(padding)...

Answer Span:

a. I use offset mapping(return by the tokenizer) to convert the start/end position, for each tuple(start/end of a word) in offset mapping if the first element of the tuple matches the answer start position in the context, then the corresponding token is the start token, and if the second element of the tuple matches the answer end position in the context, then the corresponding token is the end token.

b. 1. if the answer is not in the context, then return (0,0)

2. if the answer is longer than 30, it will not be considered

3. if in the start position and the end position are in the same truncated split and the start position is bigger than the end position, the answer will not be considered

After finding the highest start/end position pair, not in the above situations, I use offset mapping to decode the tokenized start/end position.

Model:

1. My model: bert-base-chinese
   - Hidden size: 768
   - Hidden Layers: 12
   - Attention Heads: 12
   - Intermediate Size: 3072
   - max_len: 512
   - doc_stride: 128
2. Performance
   Context Selection:0.9054
   QA:0.798

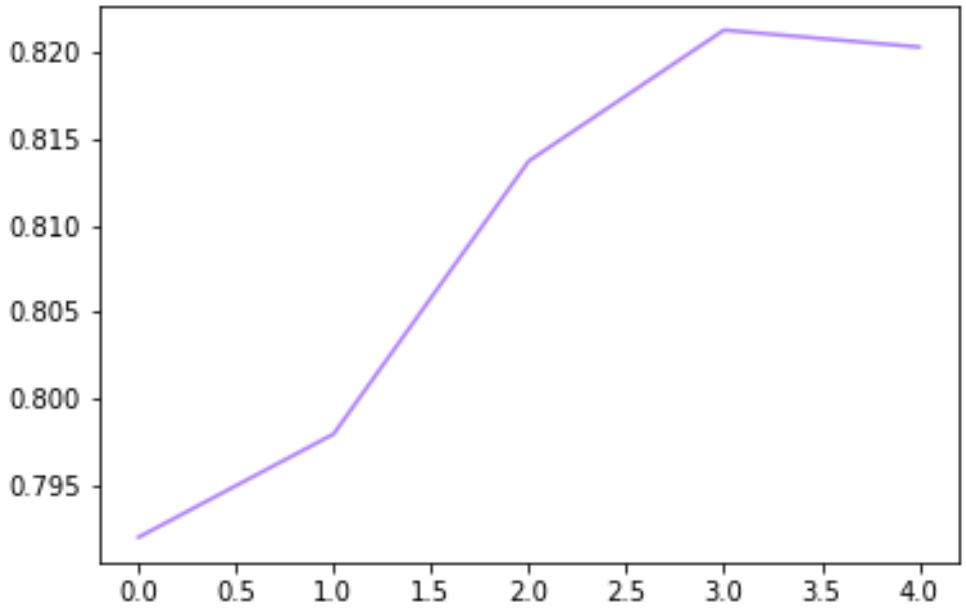
3. Loss Function: CrossEntropy Loss
4. Optimization: • Algorithm: AdamW with lr = 2e-5 and weight decay = 1e-5
5. Scheduler: get_cosine_schedule_with_warmup

1. My model: hfl/chinese-macbert-base
   - Hidden size: 768
   - Hidden Layers: 12
   - Attention Heads: 12
   - Intermediate Size: 3072
   - max_len: 512
   - doc_stride: 128
3. Performance
   Context Selection:0.9608
   QA:0.819

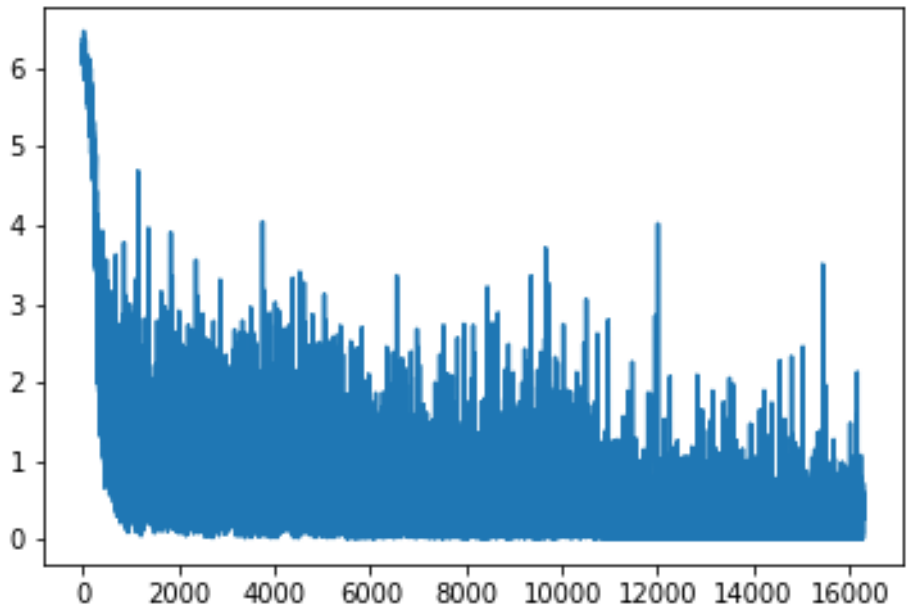
3. Loss Function: CrossEntropy Loss
4. Optimization: • Algorithm: AdamW with lr = 2e-5 and weight decay = 1e-5
5. Scheduler: get_cosine_schedule_with_warmup
6. difference with bert: Instead of using [Mask] token, they replace the original word with a similar word.

Curve

Acc(epoch)



Loss(step)



Not pretrained

1. Model:
   - Hidden size:192
   - Hidden Layers: 3
   - Attention Heads: 3
   - Intermediate Size: 3072
   - max_len: 512
   - doc_stride: 128
2. Performance 0.1784(20 epoch, not easy to train…..)
3. Loss Function: CrossEntropy Loss
4. Optimization: • Algorithm: AdamW with lr = 2e-5 and weight decay = 1e-5
5. Scheduler: get_cosine_schedule_with_warmup