ADL 2024 HW1 Report

陳彥廷，資管四，b10705013

Q1: Data processing

Q1-1 : Describe in detail about the tokenization algorithm you use. You need to explain what algorithm does in your own ways

The model I used is `macbert-base-chinese` , which is a Chinese version of BERT. The tokenization algorithm used in this model is `WordPiece` .

WordPiece is similar to BPE, which is a subword tokenization algorithm.

1. Spilt the input text into characters, and add a special prefix to identify the word.
   For example `word -> w、##o、##r、##d`
2. Depending on the previeous step, we can get a initial vocabulary.
3. Compute the score of each pair of characters, and merge the pair with the highest score.
   The formula is `freq_of_pair / (freq_of_first_element * freq_of_second_element)`
4. Merge the pair with the highest score, and update the vocabulary.
5. Repeat step 3 and 4 until the vocabulary size reaches the predefined size.

Q1-2 : How did you convert the answer span start/end position on characters to position on tokens after BERT tokenization?

When we tokenize the input text, we will record the `offset_mapping` of each token. The `offset_mapping` is a tuple of two integers, which represent the start and end position of the token in the original text.

Q1-3 : After your model predicts the probability of answer span start/end position, what rules did you apply to determine the final start/end position?

To find the answer span, we will use the score calculated by adding the start logits and end logits of each feature, and according to the hyperparameter `n_best_size` and `max_answer_length` , we will find the best answer span.

Q2: Modeling with BERTs and their variants

Q2-1 : Describe Model

- Model : `macbert-base-chinese`
- The performance of your model (on validation set):
  - Paragraph selection : 0.967
  - Span selection : 0.825
- The loss function you used : Cross Entropy
- The optimization algorithm, learning rate and batch size :

Hyper Parameter	Paragraph Selection	Span Selection
Loss Function	Cross Entropy	Cross Entropy
Optimization Algorithm	AdamW	AdamW
lr_scheduler_type	cosine	cosine
max_answer_length	X	40
n_best_size	X	40
Learning Rate	5e-5	5e-5

Hyper Parameter	Paragraph Selection	Span Selection
Max Seq Length	512	512
Batch Size per Device	32	8
Gradient Accumulation Steps	2	4
Batch Size	64	32
Epochs	1	3

MacBERT is an improved BERT with novel MLM as correction pre-training task, which mitigates the discrepancy of pre-training and fine-tuning.

Instead of masking with [MASK] token, which never appears in the fine-tuning stage, MacBERT use similar words for the masking purpose. A similar word is obtained by using Synonyms toolkit, which is based on word2vec similarity calculations.

MacBERT also uses a new technique called `N-gram Masking` , which masks a continuous sequence of words instead of a single word. When N-gram Masking is used, the model can learn the context of the masked sequence, which enhances the model's ability to understand the context.

Q2-2 : Try another type of pre-trained LMs and describe

- Model : `chinese-roberta-wwm-ext`
- The performance of your model (on validation set):
  - Paragraph selection : 0.963
  - Span selection : 0.824
- The difference between pre-trained LMs

Hyper Parameter	Paragraph Selection	Span Selection
Loss Function	Cross Entropy	Cross Entropy

Hyper Parameter	Paragraph Selection	Span Selection
Optimization Algorithm	AdamW	AdamW
lr_scheduler_type	cosine	cosine
max_answer_length	X	40
n_best_size	X	40
Learning Rate	5e-5	5e-5
Max Seq Length	512	512
Batch Size per Device	32	8
Gradient Accumulation Steps	2	4
Batch Size	64	32
Epochs	1	3

chinese-roberta-wwm-ext is a Chinese language model based on the RoBERTa architecture, optimized for handling Chinese natural language tasks. The key feature of this model is the Whole Word Masking (WWM) technique, when a part of a word is selected for masking, the entire word is masked. This helps the model better capture the full context of words rather than focusing on isolated characters.

RoBERTa is an improved version of BERT, which uses dynamic masking instead of static masking. Dynamic masking means that the masking pattern changes at each training epoch, which helps the model learn more about the context of the masked words.

The main difference between macbert-base-chinese and chinese-roberta-wwm-ext is the masking strategy. macbert-base-chinese uses similar words for masking, while chinese-roberta-wwm-ext uses whole-word masking. Besides, they have different pre-training tasks, macbert-base-chinese uses SOP (sentence order prediction) and MLM as correction, while chinese-roberta-wwm-ext uses dynamic masking and removes the NSP (next sentence prediction) task.

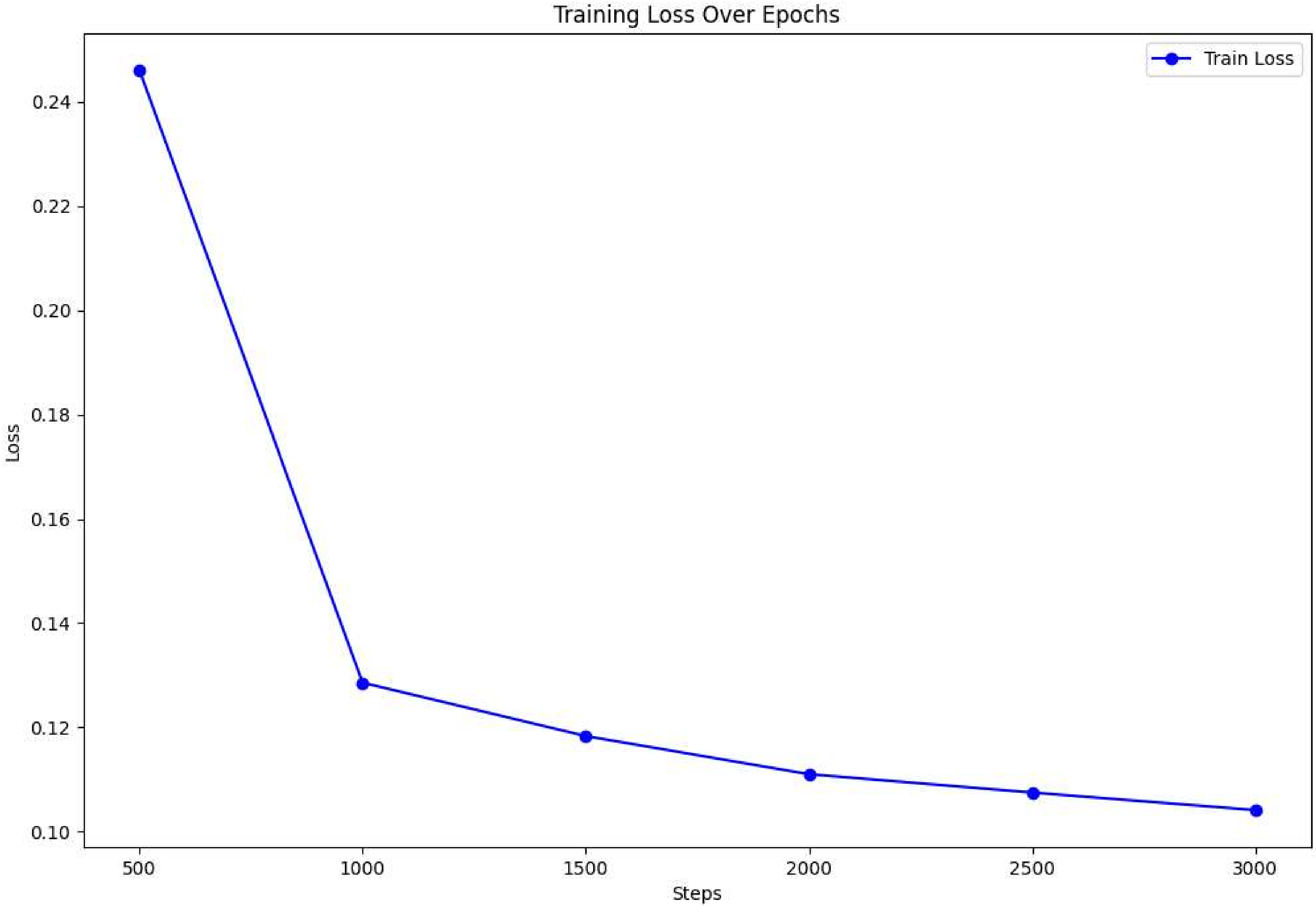
Q3: Curves

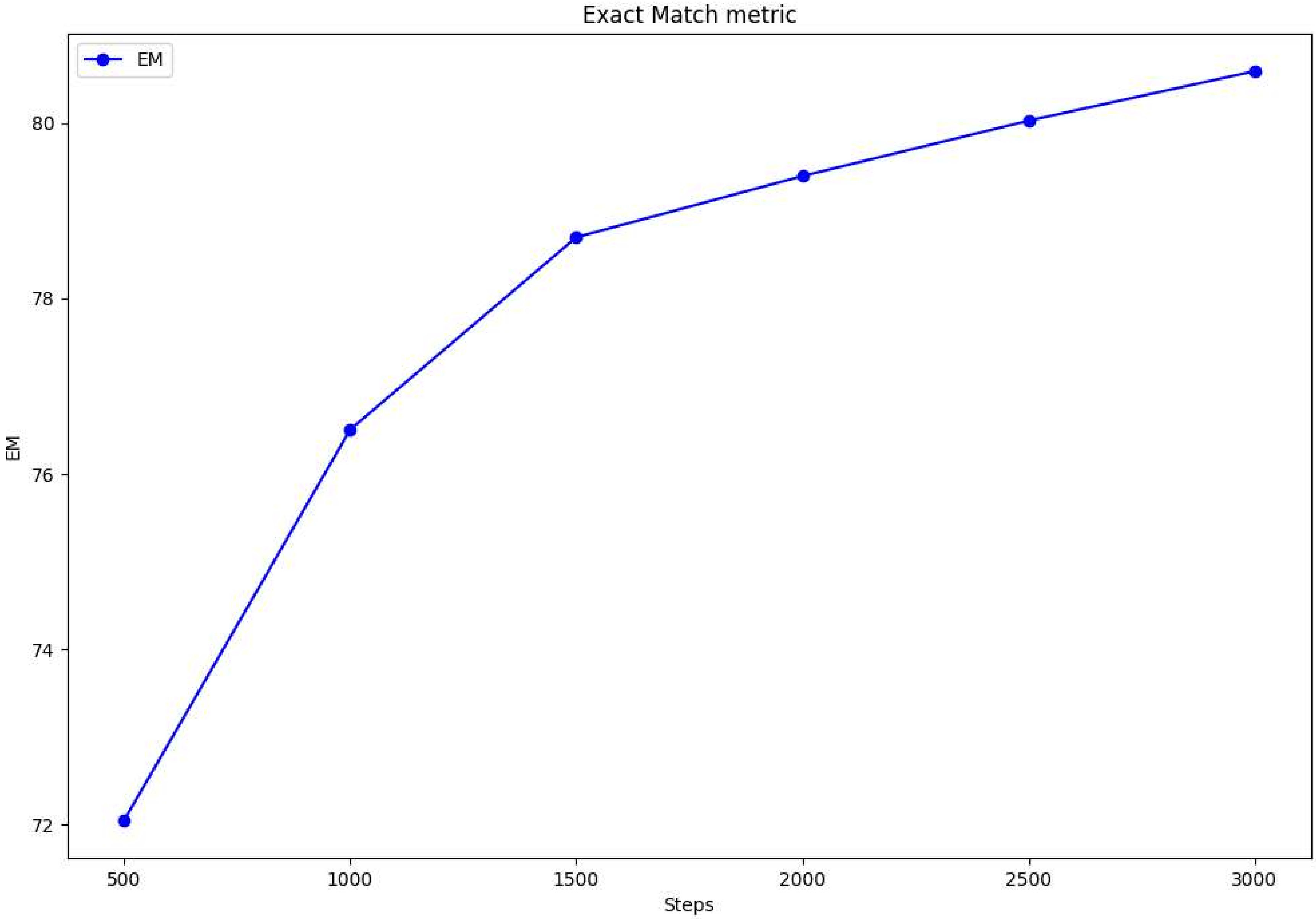
Q3-1 : Plot the learning curve of your span selection (extractive QA) model

- Learning curve of the loss value
- Learning curve of the Exact Match metric value

I plot the learning curve of the span selection model in training set.

Because of the VRAM limitation, I can only train the model for 1 epochs.


Training Loss Over Epochs
Train Loss
Loss
0.24
0.22
0.20
0.18
0.16
0.14
0.12
0.10
Steps
500
1000
1500
2000
2500
3000


Exact Match metric
EM
EM
80
78
76
74
72
500
1000
1500
2000
2500
3000
Steps

Q4: Pre-trained vs Not Pre-trained

Q4-1 : Train a transformer-based model from scratch

- The configuration of the model and how do you train this model
- The performance of this model v.s. BERT.
- The difference between the two models

I decide to train the model from scratch on the paragraph selection task.

Hyper Parameter	From Scratch	macbert
Loss Function	Cross Entropy	Cross Entropy
Optimization Algorithm	AdamW	AdamW
lr_scheduler_type	cosine	cosine
max_answer_length	X	40
n_best_size	X	40
Learning Rate	5e-5	5e-5
Max Seq Length	512	512
Batch Size per Device	16	8
Gradient Accumulation Steps	2	4
Batch Size	32	32
Epochs	1	3

Hyper Parameter	From Scratch	macbert
Performance	0.534	0.967

I use bert as the model, and WordPiece as the tokenizer.

We can see that the model trained from scratch has a lower performance compared to the Pre-trained model.

The main difference between the two models is the pre-training stage. The BERT model is pre-trained on a large corpus of text data, while the model trained from scratch is not pre-trained. Pre-training helps the model learn the language patterns and structures from the text data, which improves the model's performance on downstream tasks.

Q5: Bonus

Q5-1 : Instead of the paragraph selection + span selection pipeline approach, train an end-to-end transformer-based model and describe

- Model : `XLNet`
- The performance of your model : 0.399
- The loss function you used : Cross Entropy
- The optimization algorithm (e.g. Adam), learning rate and batch size

Hyper Parameter	End-to-End Selection
Loss Function	Cross Entropy
Optimization Algorithm	AdamW
lr_scheduler_type	cosine

Hyper Parameter	End-to-End Selection
max_answer_length	30
n_best_size	20
Learning Rate	5e-5
Max Seq Length	512
Batch Size per Device	32
Gradient Accumulation Steps	2
Batch Size	64
Epochs	1

XLNet is a transformer-based model that uses an autoregressive language model combined with a permutation language model to capture bidirectional context. Instead of predicting the next word in a fixed sequence, XLNet applies a permutation language model, which allows it to learn from all possible permutations of the word order in a sequence, making it capable of leveraging both left and right contexts for improved language understanding.

Reference

- ChatGPT
- Copilot
- WordPiece 標記化