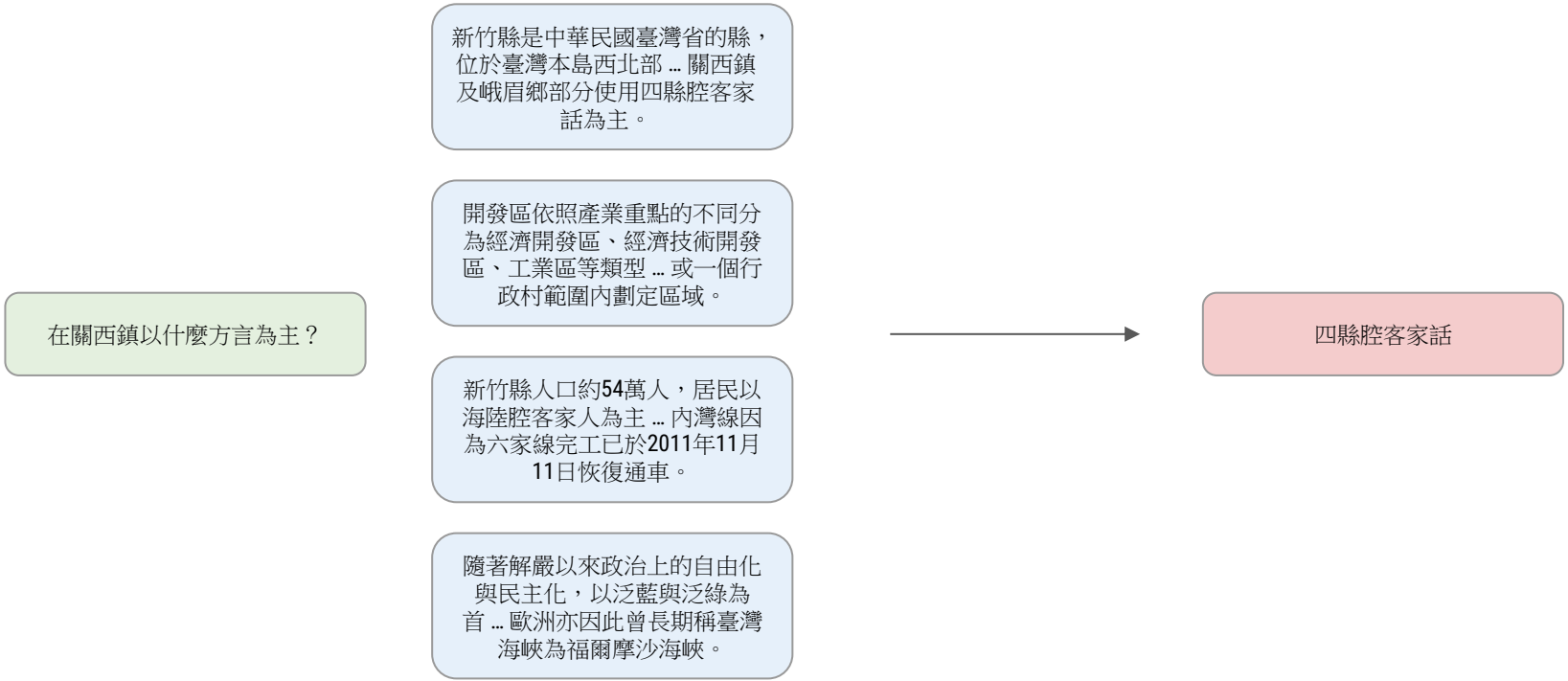
NTU ADL 2024 Fall
HW1 Report

NTNU EE 蔡杰宇 41375007H

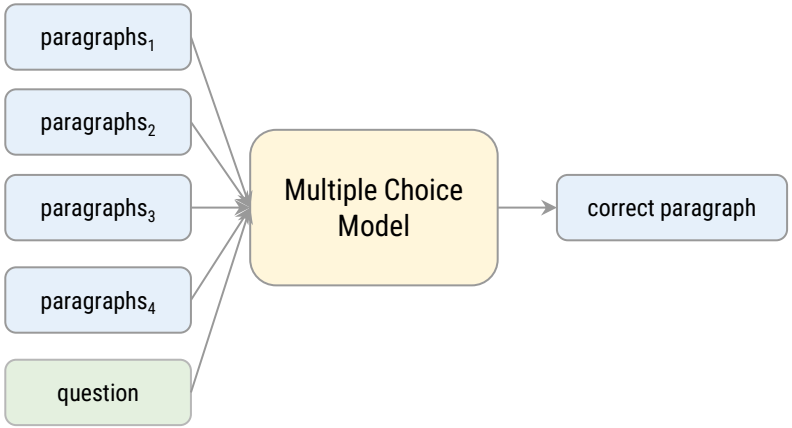
Task Description - Chinese Extractive Question Answering (QA)



Task Description - Chinese Extractive Question Answering (QA)

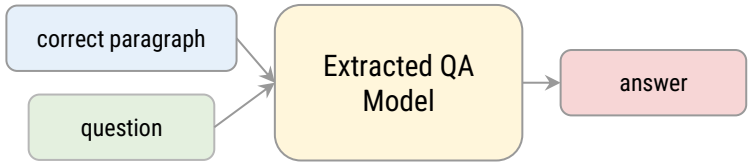
Paragraph Selection:

Determine which paragraph is relevant.



Span selection:

Determine the start and end position of the answer span.



The answer is always a **span in the correct paragraph.*

Report - Q1: Data processing (2%)

```python
from transformers import AutoTokenizer

model_name = "bert-base-chinese"

tokenizer = AutoTokenizer.from_pretrained(model_name)

context = "在關西鎮以什麼方言為主？"
answer = "四縣腔客家話"

input = tokenizer(text = context, text_pair = answer,
                  padding = "max_length", truncation=True)
print(input)
output = tokenizer.decode(input['input_ids'])
print(output)
```

0.2s Python

- Tokenizer (1%):
  - Tokenizer processing separates English words into subwords, which are meaningful unit. The advantage is that the model can use the meaning by these tokens to predict unseen words.
  - Chinese word are separates to each individual words, like

```
{'input_ids': [101, 1762, 7302, 6205, 7120, 809, 784, 7938, 3175, 6241, 4158, 712, 8043, 102, 1724, 5238,
[CLS] 在 關 西 鎮 以 什 麼 方 言 為 主 ？ [SEP] 四 縣 腔 客 家 話 [SEP] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [P
```

  - There are some special token like
    - [CLS] : for the beginning of the sequence.
    - [SEP] : using it to separate different sentences
    - [PAD] : padding token, in order to pad sequence to same length
    - [UNK] : if input have unseen token, it will be represented this token

Report - Q1: Data processing (2%)

```
context = "在關西鎮以什麼方言為主？"
answer = "四縣腔客家話"
input = tokenizer(text = context, text_pair = answer,
                  truncation=True,
                    return_overflowing_tokens=True, return_offsets_mapping=True)

print(input['offset_mapping'])
```

✓ 0.0s Python

```
[[(0, 0), (0, 1), (1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 7), (7, 8), (8, 9), (9, 10), (10, 11), (11,
```

- Answer Span (1%):
  - (a). In question answering task, when doing tokenization, I use the “return_offset_mapping” option to record the start and end position in sequence of each token. By doing this, I can convert the answer span to corresponding position.
  - (b). Since we need to predict start, end positions of the span, so we must to calculate the logits probabilities to identify the best answer. Using above function to find the best answer.

$$i^*, j^* = \text{argmax}_{i,j} \mathrm{P}(s_i)\mathrm{P}(e_j) = \arg\max_{i,j} e^{l(s_i)}\, e^{l(e_j)}$$

$$= \arg\max_{i,j} \log\left(e^{l(s_i)} e^{l(e_j)}\right) = \arg\max_{i,j} \log\left(e^{l(s_i)}\right) + \log\left(e^{l(e_j)}\right)$$

$$= \arg\max_{i,j} l\,(s_i) + l(e_j)$$

Report - Q2: Modeling with BERTs and their variants (4%)

In Multiple choice task

	Bert-base-Chinese	Chinese-lert-base
Performance (acc in validation)	0.96078	0.96975
Loss function	Cross-Entropy Loss	Cross-Entropy Loss
Optimizer	AdamW	AdamW
Learning rate	5e-5	5e-5
Batch size	8(4 * 2 gradient accumulation step)	8(4 * 2 gradient accumulation step)
Max seq length	512	512
Epoch	5	5

Report - Q2: Modeling with BERTs and their variants (4%)

In question answering task, because my best model is Chinese-lert-base in mc task, I use Chinese-lert-base model to infer test dataset.

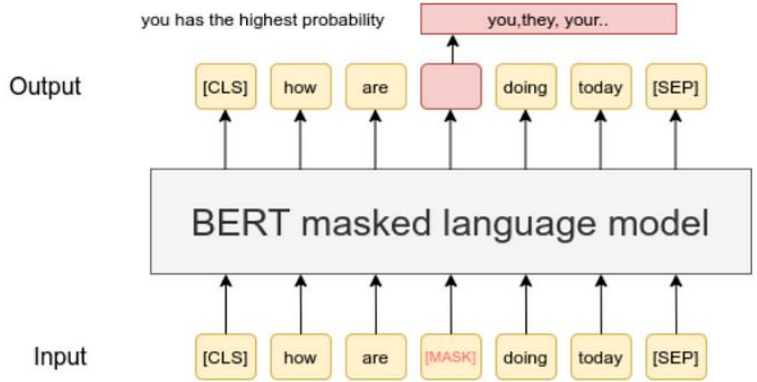
	Chinese-lert-base	Chinese-lert-large
Performance (EM in validation / Private score / Public score)	86.075 / 0.80265 / 0.79298	86.274 / 0.81222 / 0.80116
Loss function	Cross-Entropy Loss	Cross-Entropy Loss
Optimizer	AdamW	AdamW
Learning rate	8e-5	2e-5
Batch size	32 (8 * 4 gradient accumulation step)	32 (8 * 4 gradient accumulation step)
Max seq length	512	512
Epoch	25	25

Report - Q2: Modeling with BERTs and their variants (4%)

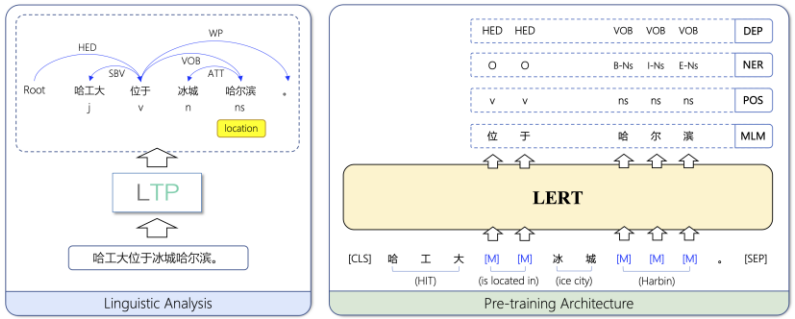
The difference between Bert and Lert

- Bert is a model in NLP that uses MLM (Masked Language Model) and NSP(Next Sentence Prediction) tasks to train. In contrast, Lert not only focuses surface-level textual but also integrates linguistic features like DEP (Dependency Parsing), NER (Named Entity Recognition), POS (Part-of-Speech) task to train model, which provide the model richer language representations.

- In addition, Lert use LIP (Linguistic Information-Enhenced Pre-training) mechanism, which captures and utilizes linguistic information more effectively.



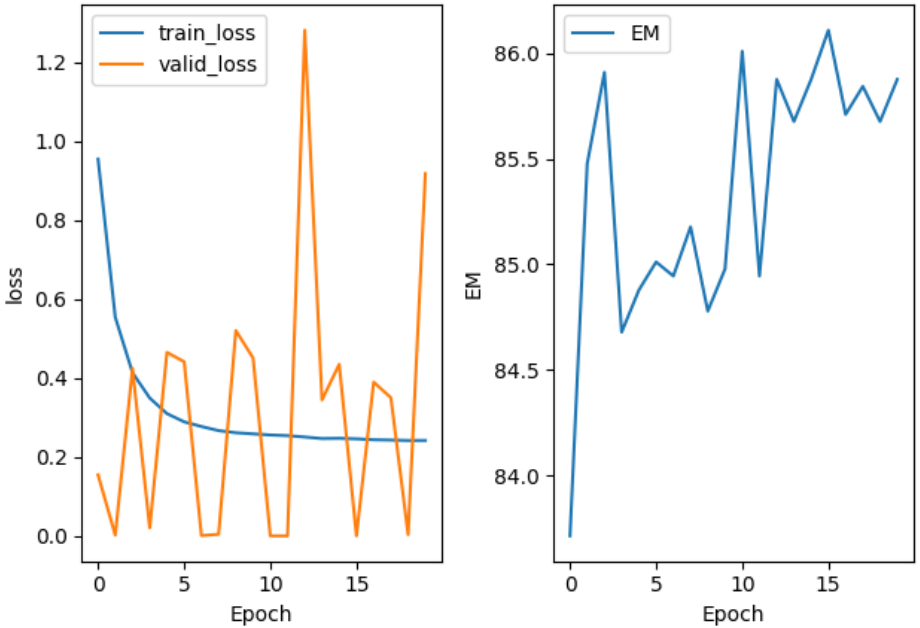
Bert



Lert

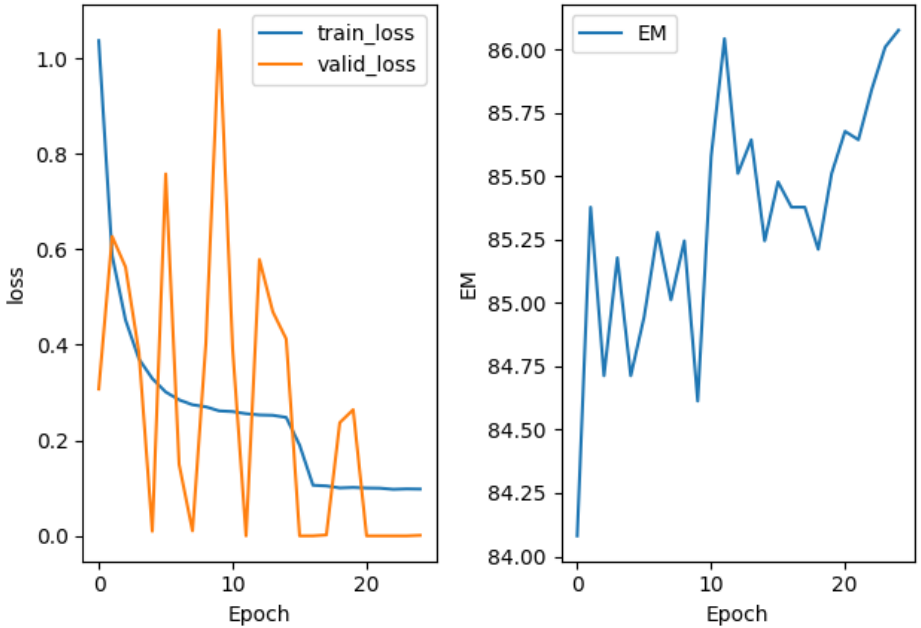
Report - Q3: Curves (1%)

chinese-lert-base_{qa2}



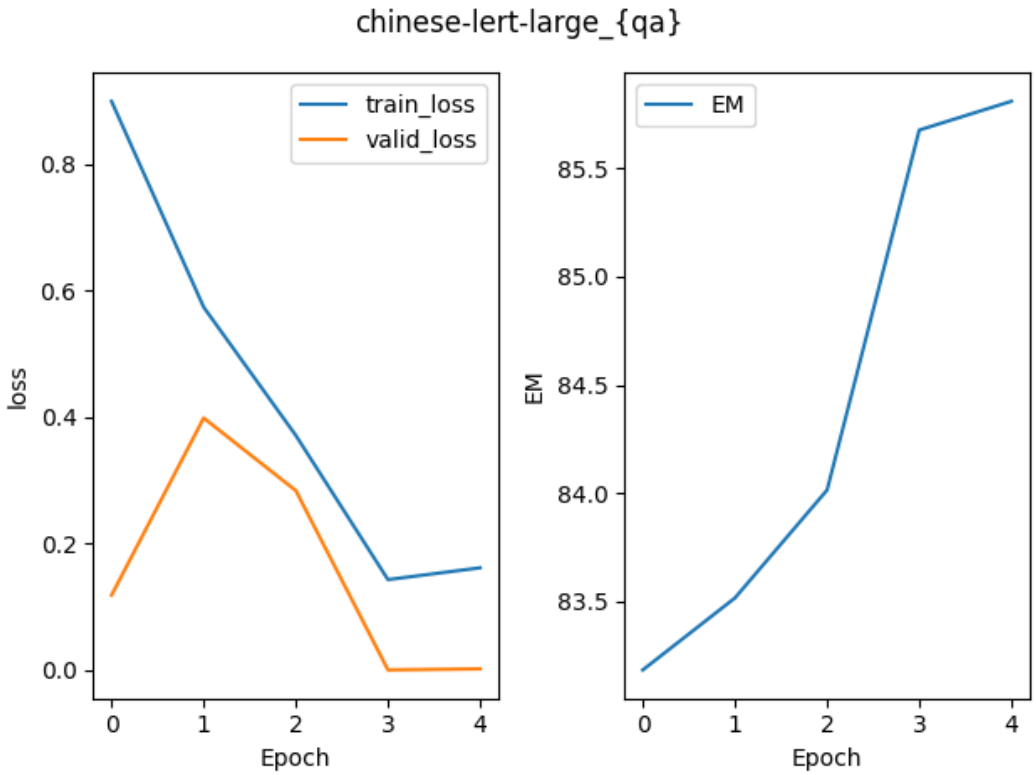
Epoch = 20

chinese-lert-base_{qa3}



Epoch = 25

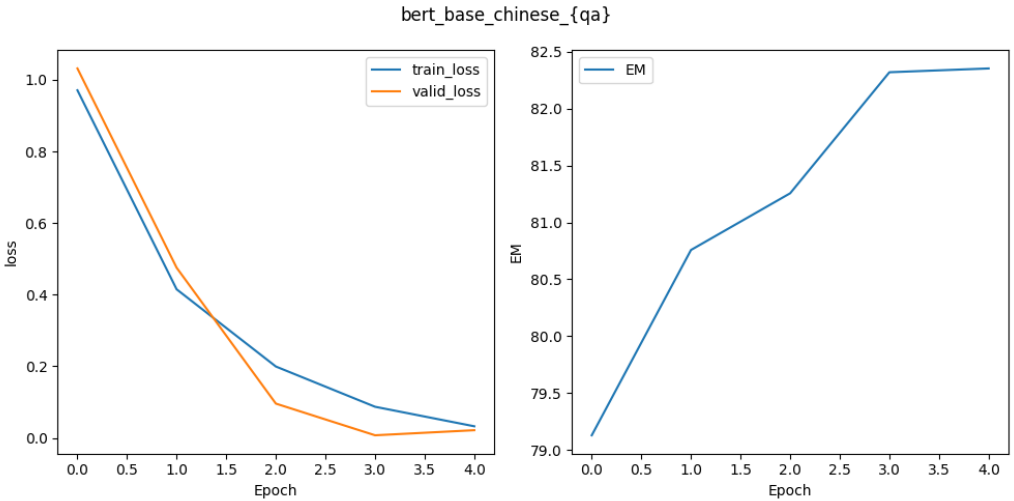
Report - Q3: Curves (1%)



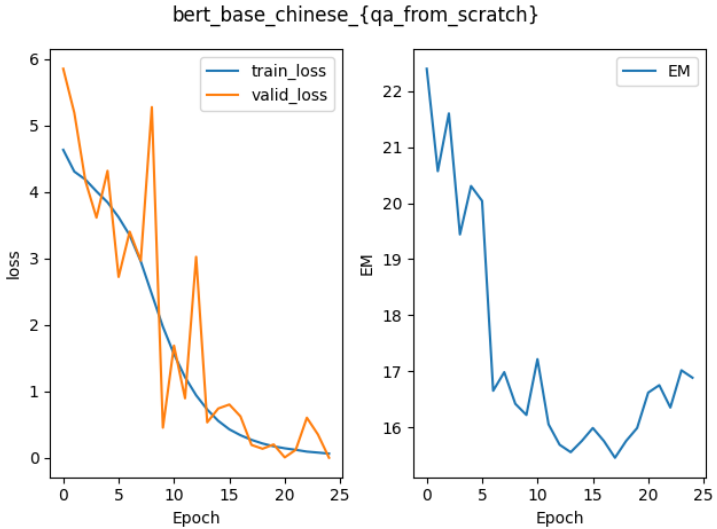
Epoch = 5

Report - Q3: Curves (1%)

Pretrained model vs train from scratch (bert-base-Chinese)



Epoch = 5



Epoch = 25

“Lottery Ticket Hypothesis”, above plot illustrate initial weight from pretrain model is important and its performance is better than random model weight. (EM value of from scratch model even drops XD)

Report - Q4: Pre-trained vs Not Pre-trained (2%)

Pretrained model vs train from scratch for question answering (bert-base-Chinese)

	Pretrained model	Train from scratch
Performance (“Best” EM in validation)	82.352	22.399
Loss function	Cross-Entropy Loss	Cross-Entropy Loss
Optimizer	AdamW	AdamW
Learning rate	8e-5	8e-5
Batch size	32 (8 * 4 gradient accumulation step)	32 (8 * 4 gradient accumulation step)
Max seq length	512	512
Epoch	5	25