2022

Data Mining Final Project

組員: 卓冠廷、李照棋、王伊婷

CONTENTS

目 錄

PART 01

QA BERT

Question **A**nswering with a fine-tuned BERT

Question Answering with a fine-tuned BERT

Let machine answer machine

01 將前處理資料加工

將前處理後的資料中的 q 和 r 結合成文章，並根據 agree 和 disagree 設計問題

```
Text:
Q:http news.telegraph.co.uk news mai ... nabort15.xml r:i think that i don't have quite enough tobasco in my bloody mary .

Question:
What does q disagree with r?
```

02 將文章和問題丟入模型

預訓練模型：
'bert-large-uncased-whole-word-masking-finetuned-squad’
分詞器(tokenizer) :
'bert-large-uncased-whole-word-masking-finetuned-squad'

```
model = BertForQuestionAnswering.from_pretrained('bert-large-uncased-whole-word-masking-finetuned-squad')
tokenizer = BertTokenizer.from_pretrained('bert-large-uncased-whole-word-masking-finetuned-squad')
```

03 獲得並解析結果

過濾無法辨識的答案，並重新解讀

04 結果

準確率(根據 Agree 和 disagree 設計問題) :
61.12%

可能的問題：

1. 問的問題太單一不太符合文章的敘述。
2. 因為輸出是文章中的起點和終點，對非連續的單詞或短句答案不利，最終會只預測出一個單詞或連續不相關的段落。

PART 02

QA Bert (Improvement)

Question **A**nswering with a fine-tuned BERT
Improvement

QA Bert (Improvement)

Problem we want to solve

01 q’ , r’ 段落不清楚

因為所獲得的答案為文章中的段落，所以 r’ 可能會參雜 q 的語句或單詞，q’ 同理。

用後處理過濾 q , r 交雜的答案

將 r’ 參雜 q 的語句或單詞利用文章座標做過濾，q’ 同理。

02 問題與文章相關性低

問題只以 agree , disagree 的角度發問，和文章內文無相關，造成模型容易隨機辨識並沒有真的判斷。

利用2次發問設計問題

第一次先問文章中的關鍵字，再以文章座標分類 q , r 分別的關鍵字，最後以這些關鍵字組成問題提問。

QA Bert (Improvement)

Pipeline

01 將前處理資料加工

將前處理後的資料中的 q 和 r 結合成文章，並根據 agree 和 disagree 設計問題，如：What does q agree/disagree about ?
What does r agree/disagree about ?

02 將文章和問題丟入模型

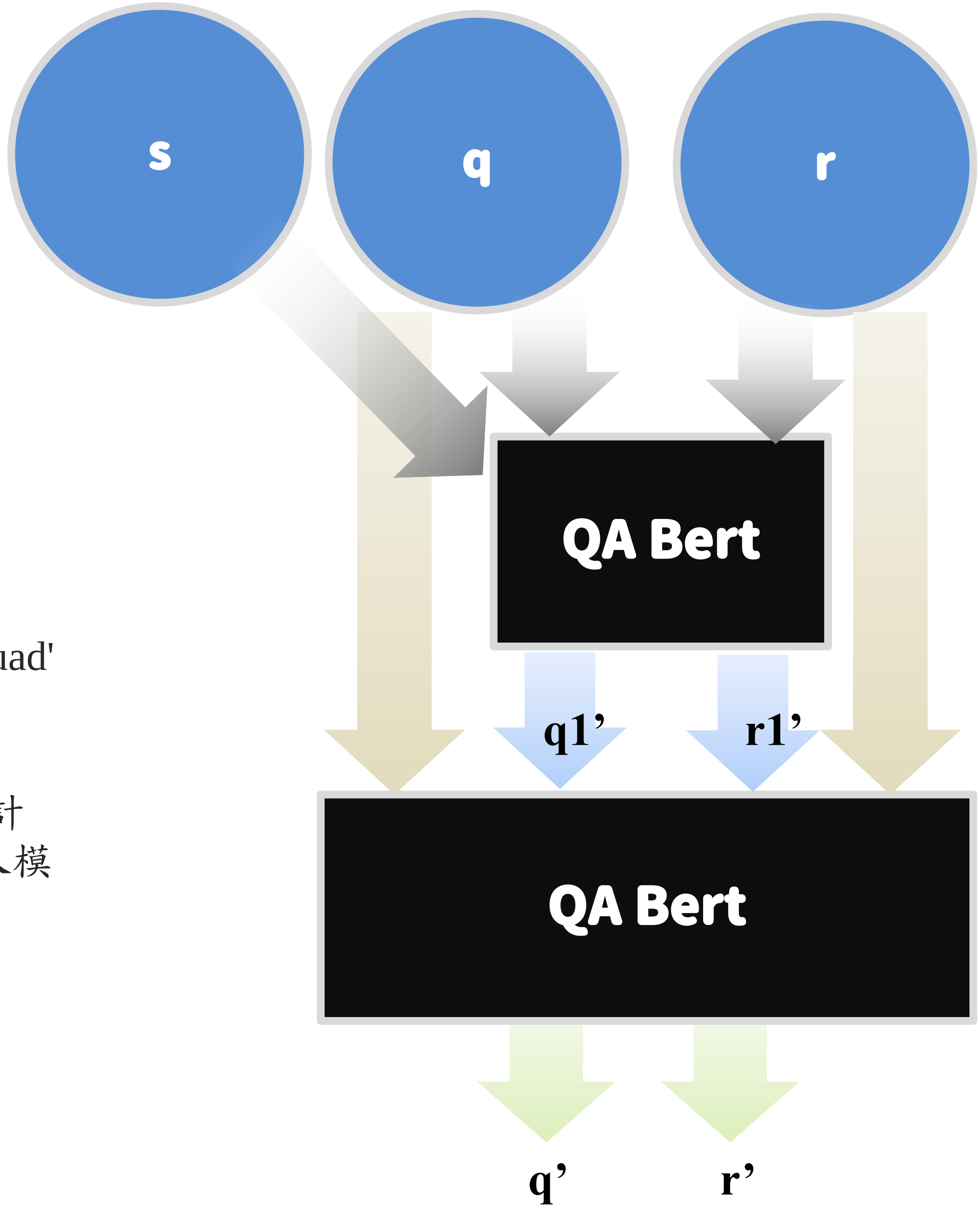
預訓練模型：'bert-large-uncased-whole-word-masking-finetuned-squad’
分詞器(tokenizer) : 'bert-large-uncased-whole-word-masking-finetuned-squad'

03 結果後處理並再度丟入模型

以文章座標分類 q , r 分別的關鍵句，從關鍵句若關鍵句大於 1 句則統計最常出現的名詞(以 nltk 判斷)，最後以這個關鍵名詞組成問題重新丟入模型提問。
問題格式為：Why does q/r agree/disagree about Noun1?

04 結果

Continue …



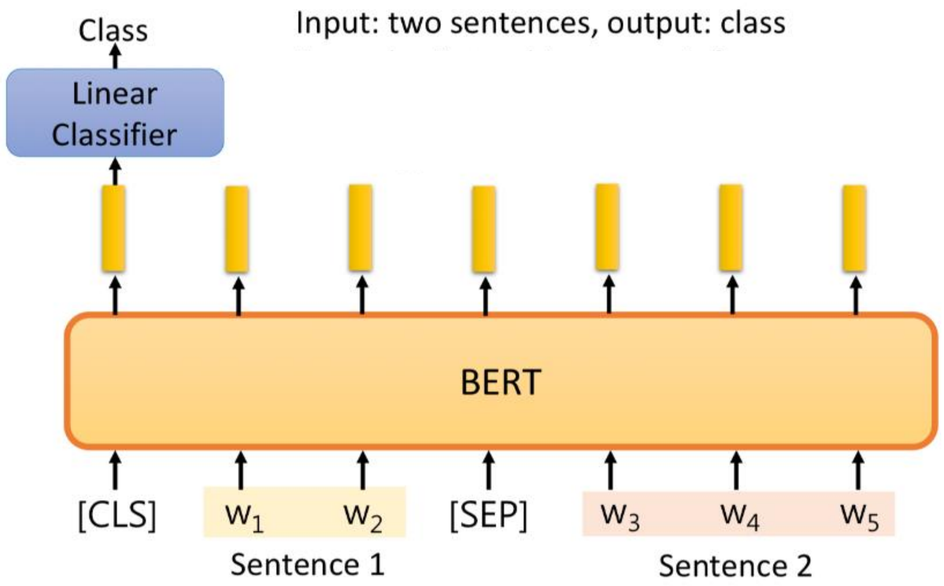
PART 03

Combined Model

BERT Classifier + Transformer

BERT Classifier

Bidirectional **E**ncoder **R**epresentations
from **T**ransformers and **L**inear **Cl**assifier



模型

透過將q和r丟入模型中訓練(預訓練權重選'bert-base-cased')，最終可獲得80%的準確率，但這是針對分辨s的部分，而此競賽需要的是獲得q'和r'

問題

得Bert的sequence outputs(X)與linear classifier(Y=AX)的q和r分別的權重值(A)，然而，即使從sequence outputs取出q'與r'，仍不知該如何將s作為input

解決辦法

選用Transformer做純粹的sequence extraction，並將提取結果丟上AI CUP競賽中，結果意外的好

Transformer

Sequence extraction from q and r

01 模型與預訓練權重

預訓練權重： tf-small
模型 : TFAutoModelForSeq2SeqLM

Model size variants

Model	Parameters	# layers	d_{model}	d_{ff}	d_{kv}	# heads
Small	60M	6	512	2048	64	8
Base	220M	12	768	3072	64	12
Large	770M	24	1024	4096	64	16
3B	3B	24	1024	16384	128	32
11B	11B	24	1024	65536	128	128

T5 model size variants. Source: T5 paper.

02 將 (q q’) 與 (r r’) 丟入訓練

隨機抽樣：
文章短(1~9句)的預測結果很好
文章長(10句以上)的預測結果很差
AI CUP準確率(有無前處理):
71.81% / 72.42%

```
84 token length: 4
actual: I really think it's funny .
predict: I really think it's funny
F1: 0.999999995 , Precision: 1.0 , Recall: 1.0
```

```
11 token length: 14
actual: find that appalling
predict: First , there is no us on your part regarding this . I am talking to you . Others here that argue your same positions have been much less beligerent
F1: 0.15789473184210542 , Precision: 0.15789473684210525 , Recall: 0.15789473684210525
```

03 將訓練集中文章長短文章分開訓練

隨機抽樣：
文章長(10句以上)的預測結果提升
AI CUP準確率(有無前處理) :
72.54% / 73.85%

```
41 token length: 5
actual: The coelacanth , according to fossil records , and according to evolutionists , allegedly went extinct millions and millions of years ago .
predict: The coelacanth , according to fossil records , and according to evolutionists , allegedly went extinct millions and millions of years ago
F1: 0.999999995 , Precision: 1.0 , Recall: 1.0
```

04 增加訓練時間

AI CUP準確率(有無前處理) :
75.7% / 76.12%

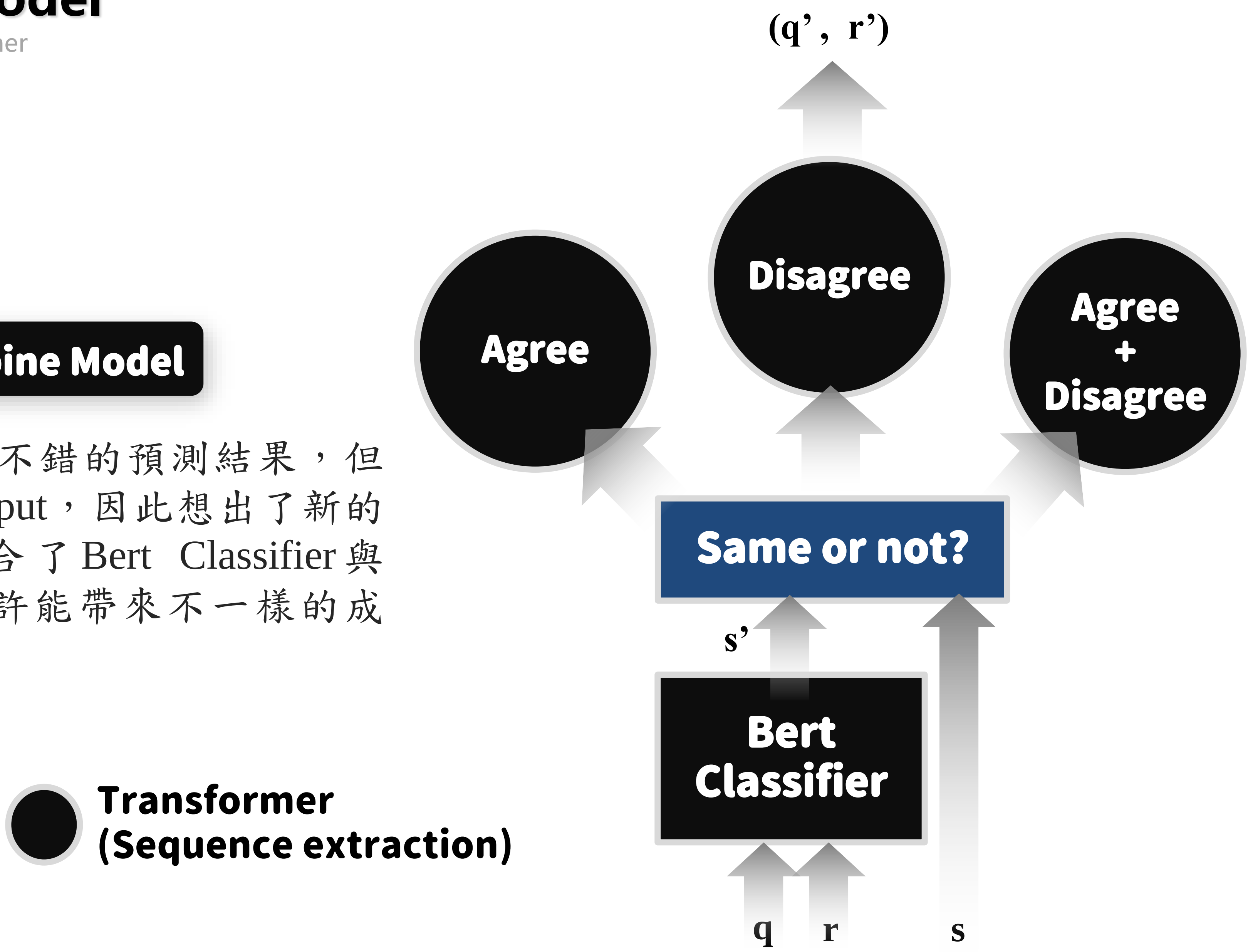
```
10 token length: 17
actual: I personly would not condone an abortion , however wouldn't condem person who wanted one
predict: I personly would not condone an abortion , however wouldn't condem person who wanted one its there choice .
F1: 0.999999995 , Precision: 1.0 , Recall: 1.0
```

Combined Model

BERT Classifier + Transformer

Combine Model

Transformer帶來不錯的預測結果，但並未加入s當作input，因此想出了新的模型架構，結合了Bert Classifier與Transformer，期許能帶來不一樣的成果展現



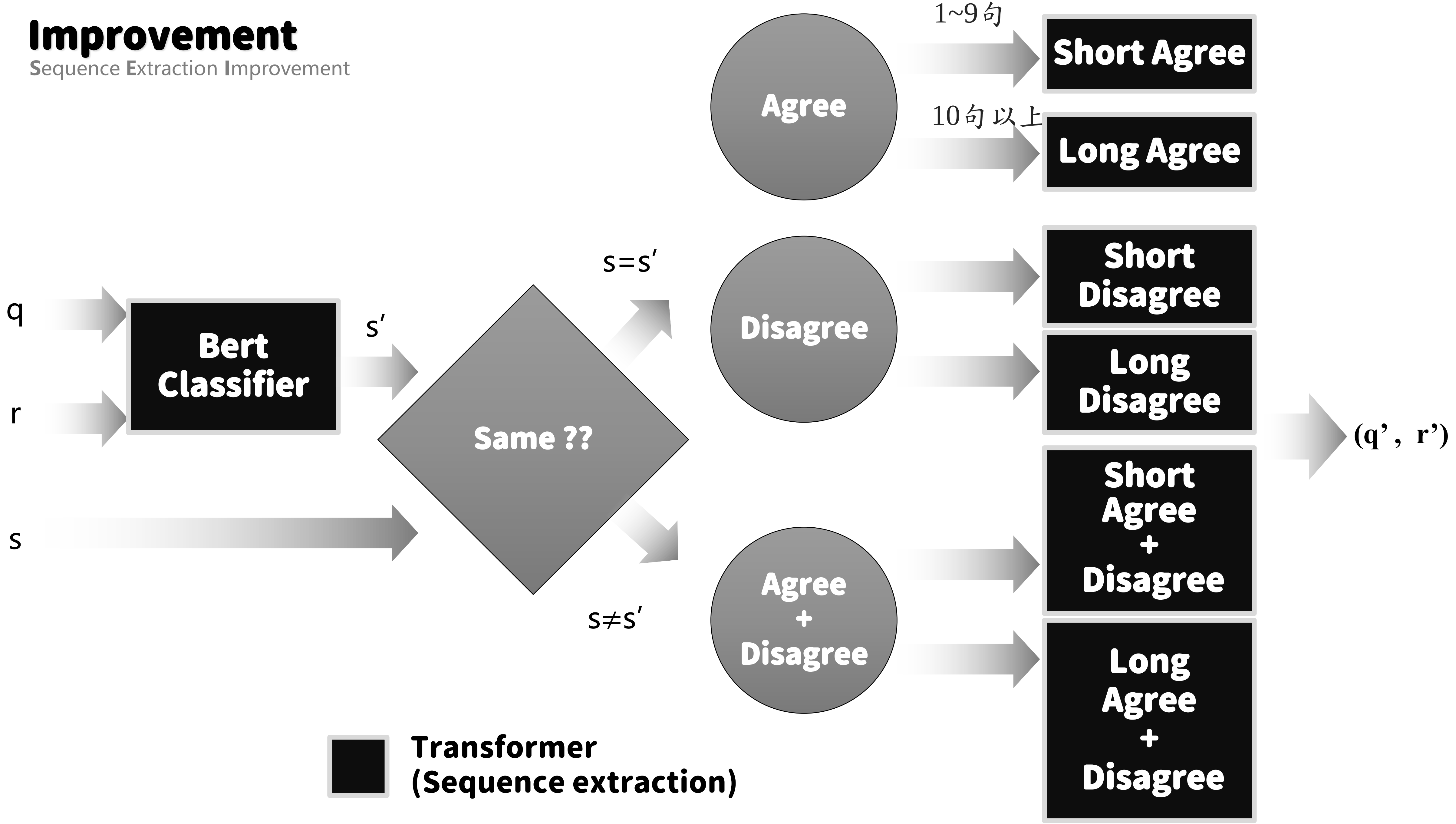
PART 04

Implement

Implement Combined Model

Improvement

Sequence Extraction Improvement



BERT Classifier

Bidirectional **E**ncoder **R**epresentations from **T**ransformers
and **L**inear **C**lassifier

01 模型與預訓練權重

預訓練權重： "bert-base-cased"
模型：BertForSequenceClassification

```
name            module
----------------------
bert:embeddings
bert:encoder
bert:pooler
dropout         Dropout(p=0.1, inplace=False)
classifier      Linear(in_features=768, out_features=2, bias=True)
```

02 將 (q , r , s) 丟入訓練

模型準確度：80.32%
將q跟r的句子一起做Word embedding
經Encoder和Pooler後得到Sequence output
進到linear classifier 做分類

```
訓練: 613 / 617
訓練: 614 / 617
訓練: 615 / 617
訓練: 616 / 617
訓練: 617 / 617
classification acc: 0.803245647249191
predictions: tensor([0, 0, 0,  ..., 0, 0, 1], device='cuda:0')
```

03 預測的s’ ,做(s , s’) 比較

總共分三個Case:
0: s = Agree & s’= Agree
1: s = Disagree & s’= Disagree
2: s ≠ s’
遇同一ID不同Case的結果採投票表決
若同票則選Case3

3367	DISAGREE	DISAGREE	1
3368	DISAGREE	DISAGREE	1
3368	DISAGREE	DISAGREE	1
3368	DISAGREE	DISAGREE	1
3368	DISAGREE	DISAGREE	1
3368	AGREE	DISAGREE	2
3369	AGREE	AGREE	0
3369	AGREE	AGREE	0
3369	AGREE	AGREE	0
3369	AGREE	AGREE	0
3369	DISAGREE	AGREE	2
3370	DISAGREE	DISAGREE	1
3370	DISAGREE	DISAGREE	1

Transformer

Text Extraction for q and r

01 s = Agree & s’ = Agree (Long)

隨機抽樣取平均

```
Average Result
F1: 0.6853123930796285 , Precision: 0.6853123980796283 , Recall: 0.6853123980796283
```

```
57 token length: 16
actual: You guys know me . Always happy to correct anyone .
predict: you guys know me. Always happy to correct anyone. As electrolyte pointed out , the first step in the scientific method is the observation . scientist observes something that raises question . Then comes the hypothesis , that is an attempt to explain the observation
F1: 0.35294117147058834 , Precision: 0.35294117647058826 , Recall: 0.35294117647058826
```

02 s = Agree & s’ = Agree (Short)

隨機抽樣取平均

```
Average Result
F1: 0.9218450466639736 , Precision: 0.9218450516639736 , Recall: 0.9218450516639736
```

```
181 token length: 3
actual: If guns make you happy and you do not break and criminal codes . Then you need no reason nor do you have to justify yourself to
predict: If guns make you happy and you do not break and criminal codes . Then you need no reason nor do you have to justify yourself to anyone
F1: 0.999999995 , Precision: 1.0 , Recall: 1.0
```

Transformer

Text **E**xtraction for q and r

03 s = Disagree & s’ = Disagree (Long)

隨機抽樣取平均

```
Average Result
F1: 0.676324341081451 , Precision: 0.6763243460814508 , Recall: 0.6763243460814508
```

```
152 token length: 33
actual: ToE is constructed on fraud premise .
predict: This post contains perfectly simple explanation of why ToE is sham along with challenge to any Darwinist to prove me wrong .
F1: 0.16216215716216234 , Precision: 0.16216216216216217 , Recall: 0.16216216216216217
```

04 s = Disagree & s’ = Disagree (Short)

隨機抽樣取平均

```
Average Result
F1: 0.920355534522206 , Precision: 0.9203555395222062 , Recall: 0.9203555395222062
```

```
37 token length: 4
actual: Then you freely admit that you lied when you said this , and I quote People like Arch are setting it as opposed to science and in that position it
will be doomed to fail .
predict: Then you freely admit that you lied when you said this , and I quote People like Arch are setting it as opposed to science
F1: 0.999999995 , Precision: 1.0 , Recall: 1.0
```

Transformer

Text Extraction for q and r

05 $s \neq s'$ (Long)

隨機抽樣取平均

```
Average Result
F1: 0.8422083532944202 , Precision: 0.8422083582944202 , Recall: 0.8422083582944202
```

```
204 token length: 10
actual: No I won't n't walk into that situation .
predict: No I won't n't walk into that situation .
F1: 0.999999995 , Precision: 1.0 , Recall: 1.0
```

06 $s \neq s'$ (Short)

隨機抽樣取平均

```
Average Result
F1: 0.8215786688107447 , Precision: 0.8215786738107447 , Recall: 0.8215786738107447
```

```
8 token length: 3
actual: It can go both ways . We all doubt . It is what you do with it that matters .
predict: It can go both ways .
F1: 0.999999995 , Precision: 1.0 , Recall: 1.0
```

Thanks for your listening