
Machine Learning HW7

BERT - Question Answering

ML TAs

mlta-2023-spring@googlegroups.com

Links

Kaggle sample code:

<https://www.kaggle.com/code/yuxianglin032/ml2023-hw7-question-answering-ipynb>

Colab sample code:

<https://colab.research.google.com/drive/1m0fQjlfkK9vAovxPj9Nd3-hQuxeZB2w1?usp=sharing>

Kaggle competition:

<https://www.kaggle.com/t/e001cad568dc4d77b6a5e762172f44d6>

Gradescope:

<https://www.gradescope.com/courses/515619/assignments/2789862>

Google meet:

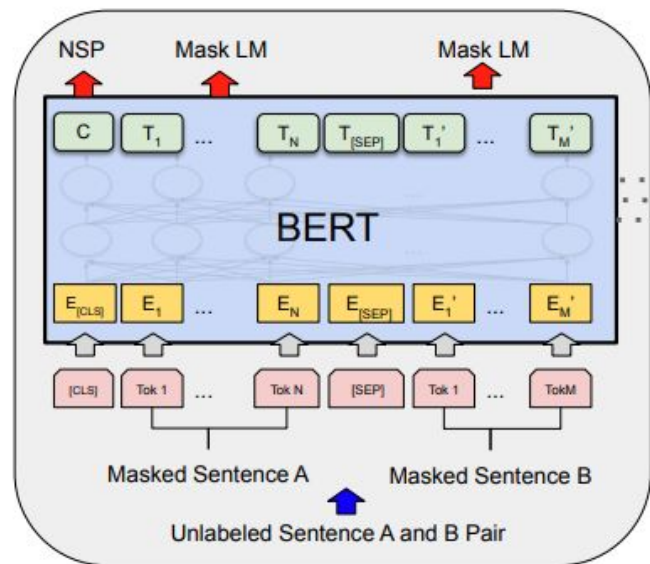
<https://meet.google.com/tti-oonf-zrx>

Outline

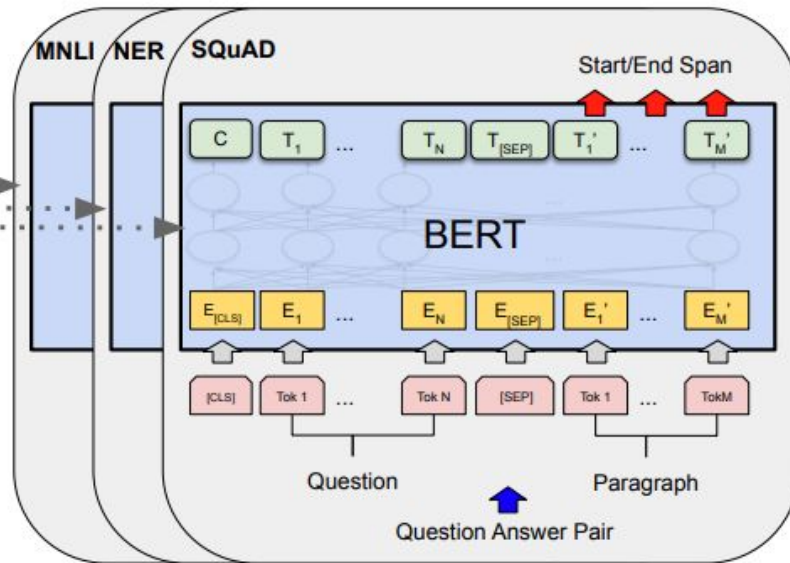
1. [Task Introduction](#)
2. [Tutorial](#)
3. [Hints](#)
4. [Grading](#)
5. [Report](#)
6. [Regulations](#)

1. Task Introduction

BERT



Pre-training



Fine-Tuning

<https://arxiv.org/abs/1810.04805>

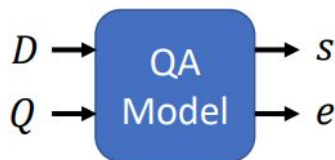
【機器學習2021】自督導式學習
(Self-supervised Learning) (二) –
BERT簡介

Task: Extractive Question Answering

- Extraction-based Question Answering (QA) (E.g. SQuAD)

Document: $D = \{d_1, d_2, \dots, d_N\}$

Query: $Q = \{q_1, q_2, \dots, q_N\}$



output: two integers (s, e)

Answer: $A = \{q_s, \dots, q_e\}$

In meteorology, precipitation is any product of the condensation of 17 spheric water vapor that falls under **gravity**. The main forms of precipitation include drizzle, rain, sleet, snow, **grau-pel** and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals **within a cloud**. Short, intense periods of rain 77 at 79 locations are called "showers".

What causes precipitation to fall?

gravity

$s = 17, e = 17$

What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?

grau-pel

Where do water droplets collide with ice crystals to form precipitation?

within a cloud

$s = 77, e = 79$

Traditional Chinese Reading Comprehension Dataset

DRCD: 台達閱讀理解資料集 Delta Reading Comprehension Dataset

ODSQA: Open-Domain Spoken Question Answering Dataset

- train: DRCD + DRCD-backtrans
 - 15329 paragraphs, 26918 questions
- dev: DRCD + DRCD-backtrans
 - 1255 paragraphs, 2863 questions
- test: DRCD + ODSQA
 - 1606 paragraphs, 3504 questions

Dataset: The format of the dataset

train & dev: with answer

```
{
  "questions": [
    {
      "id": 0,
      "paragraph_id": 11629,
      "question_text": "義大利因何原因而被稱為美麗的國度?",
      "answer_text": "擁有美麗的自然風光和為數眾多的人類文化遺產",
      "answer_start": 327,
      "answer_end": 347
    },
    {
      "id": 1,
      "paragraph_id": 10774,
      "question_text": "多少錢為2013年全球糖尿病所導致的耗費?",
      "answer_text": "五千四百八十億美元",
      "answer_start": 470,
      "answer_end": 478
    }
  ],
  "paragraphs": [
    "2010年引入的廣州快速交通運輸系統是世界第二大快速運輸系統。每日載客量可達100  

    廣州是京廣鐵路、廣深鐵路、廣茂鐵路、廣梅汕鐵路的終點站。2009年末，武漢客運  

    廣州自古已是華南地區著名的商埠，擁有2000多年的開放貿易歷史。1970年代末中國  

    廣州市內雨水充潤、土地肥沃，市區曾經有非常廣大的農業用地。兩千年前就已經有  

    古代廣州的手工業非常發達，船舶業、冶鑄和五金業、紡織業、食品加工、中成藥業、  

    中華人民共和國成立後工業國有化。五六十年代時，工業有所恢復，但文化大革命再  

    六朝時期的廣州對外貿易已相當興旺，外國海商「久停廣州，往來求利」。隋唐時期  

    從古至今，廣州基本上是嶺南地區的政治中心。秦末為南越國都城，漢朝征服南越國後
  ]
}
```

"paragraphs": [

"2010年引入的廣州快速交通運輸系統是世界第二大快速運輸系統。每日載客量可達100
廣州是京廣鐵路、廣深鐵路、廣茂鐵路、廣梅汕鐵路的終點站。2009年末，武漢客運
廣州自古已是華南地區著名的商埠，擁有2000多年的開放貿易歷史。1970年代末中國
廣州市內雨水充潤、土地肥沃，市區曾經有非常廣大的農業用地。兩千年前就已經有
古代廣州的手工業非常發達，船舶業、冶鑄和五金業、紡織業、食品加工、中成藥業、
中華人民共和國成立後工業國有化。五六十年代時，工業有所恢復，但文化大革命再
六朝時期的廣州對外貿易已相當興旺，外國海商「久停廣州，往來求利」。隋唐時期
從古至今，廣州基本上是嶺南地區的政治中心。秦末為南越國都城，漢朝征服南越國後

test: without answer, but the same format

```
{
  "questions": [
    {
      "id": 0,
      "paragraph_id": 765,
      "question_text": "喇叭裙是哪一個國家於文藝復興時男性常見的服飾?",
      "answer_text": null,
      "answer_start": null,
      "answer_end": null
    },
    {
      "id": 1,
      "paragraph_id": 373,
      "question_text": "首先在夜間戰鬥機上安裝雷達的是哪一個國家?",
      "answer_text": null,
      "answer_start": null,
      "answer_end": null
    }
  ],
  "paragraphs": [
    "在歐洲梵語的學術研究，由德國學者魯特漢斯雷頓開創。後來威廉瓊斯發現，歐語系也  

    善於在社交中口頭使用，並且在早期古典梵語文獻的發展中維持口頭傳統。在印度，書  

    梵語中，厚重、美觀的蘭札體流行於西藏和尼泊爾，並且隨著中國清朝統治階層對藏傳  

    馬祖列島是隸屬中華民國的羣島，位於臺灣海峽正北方，面臨閩江口，連江口，漢羅源  

    白犬列島，位於馬祖列島最南端的行政區，通稱炬光鄉，並封為東道，與西島。從南港  

    東犬燈塔創建於西元1872年，東犬燈塔為台灣第一批採用花崗石建造的洋式燈塔，樓高  

    島嶼之間距離比較近的，如南竿→北竿、東莒→西莒，早期靠的是一般的小遊艇，在面  

    位於馬祖列島，被中華民國博物館學會列入2004年9月版《台灣博物館名錄》的博物館
  ]
}
```

"paragraphs": [

"在歐洲梵語的學術研究，由德國學者魯特漢斯雷頓開創。後來威廉瓊斯發現，歐語系也
善於在社交中口頭使用，並且在早期古典梵語文獻的發展中維持口頭傳統。在印度，書
梵語中，厚重、美觀的蘭札體流行於西藏和尼泊爾，並且隨著中國清朝統治階層對藏傳
馬祖列島是隸屬中華民國的羣島，位於臺灣海峽正北方，面臨閩江口，連江口，漢羅源
白犬列島，位於馬祖列島最南端的行政區，通稱炬光鄉，並封為東道，與西島。從南港
東犬燈塔創建於西元1872年，東犬燈塔為台灣第一批採用花崗石建造的洋式燈塔，樓高
島嶼之間距離比較近的，如南竿→北竿、東莒→西莒，早期靠的是一般的小遊艇，在面
位於馬祖列島，被中華民國博物館學會列入2004年9月版《台灣博物館名錄》的博物館

2. Tutorial

Tokenization

'李宏毅教授2022機器學習'

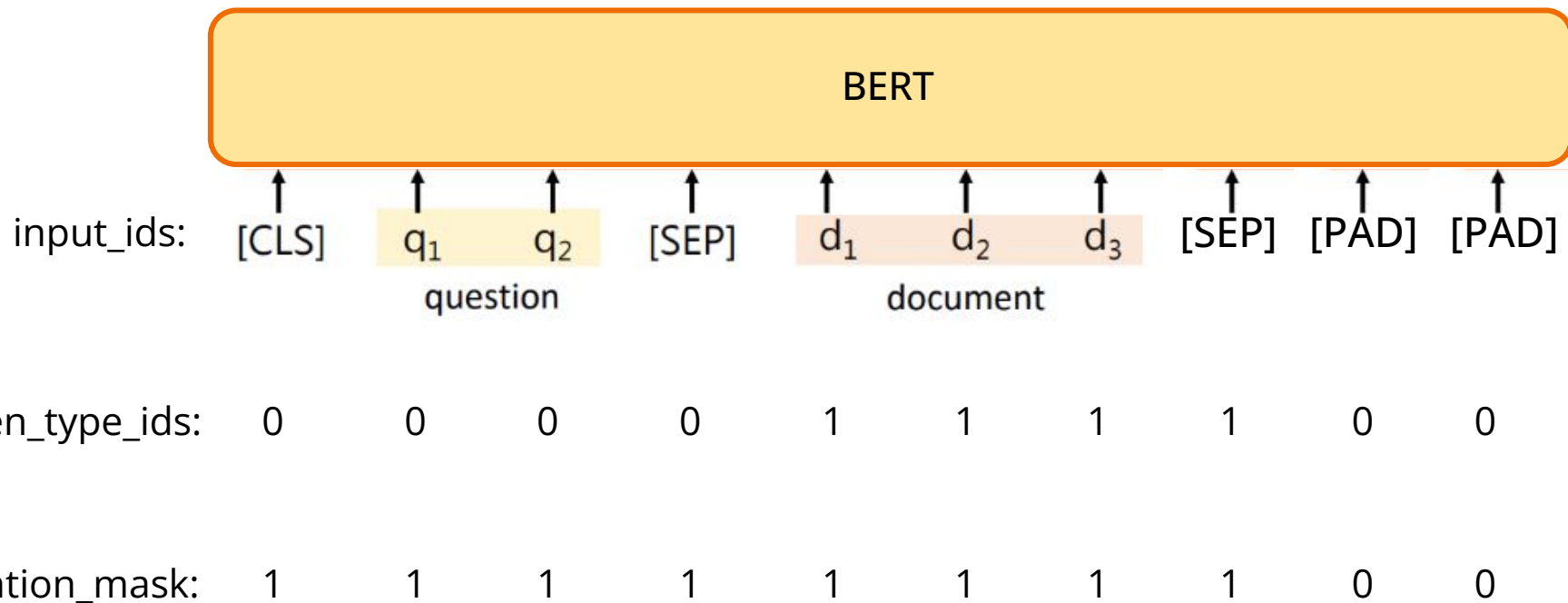
tokenize

['李', '宏', '毅', '教', '授', '2022', '機', '器', '學', '習']

tokens_to_ids

[3330, 2131, 3675, 3136, 2956, 10550, 3582, 1690, 2119, 5424]

Tokenization

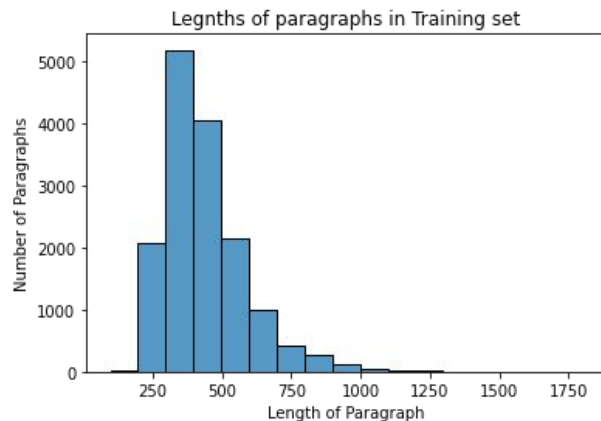


Why Long Paragraph is an Issue?

Total sequence length = question length + paragraph length + 3 (special tokens)

Maximum input sequence length of BERT is restricted to 512, why?

→ Self-Attention in transformer has $O(n^2)$ complexity



Therefore, we may not be able to process the whole paragraph.
What can we do?

Training

We know where the answer is in training!

Assumption: Info needed to answer the question can be found near the answer!

Simple solution: Just draw a window (as large as possible) around the answer!

e.g. window size = max_paragraph_len = 32

新加坡、馬來西亞的華文學術界在 1970 年代後開始統一使用簡體中文；然而 繁體字 在媒體中普遍存在著，例如華人商店的招牌、舊告示及許多非學術類中文書籍，香港和臺灣所出版的書籍也有在市場上流動 ...

Q: 新加坡的華文學術界在哪個年代後開始使用簡體中文？

A: 1970

Q: 馬來西亞的華人商店招牌主要使用什麼文字？

A: 繁體字

Testing

We do not know where the answer is in testing → split into windows!

e.g. window size = max_paragraph_len = 32

新加坡、馬來西亞的華文學術界在 1970 年代後開始統一使用簡體中文；然而 繁體字 在媒體中普遍存在著，例如華人商店的招牌、舊告示及許多非學術類中文書籍

Q: 新加坡的華文學術界在哪個年代後開始使用簡體中文？ A: 1970

Q: 馬來西亞的華人商店招牌主要使用什麼文字？ A: 繁體字

For each window, model predicts a start score and an end score → take the maximum to be answer!

	start score	start position	end score	end position	total score
window 1	0.5	23	0.4	26	0.9
window 2	0.3	35	0.7	37	1.0

Answer:
position 35 to 37

3. Hints

Hints for beating baselines

❖ Simple:

- Sample code

❖ Medium:

- Apply linear learning rate decay
- Change value of “doc_stride”

❖ Strong:

- Improve preprocessing
- Try other pretrained models

❖ Boss:

- Improve postprocessing
- Further improve the above hints

Estimated training time

	K80	T4	T4 (FP16)	P100	V100
Simple	40m	20m	8m	10m	7m
Medium	40m	20m	8m	10m	7m
Strong	2h	1h	25m	35m	20m
Boss	12.5h	6h	2.5h	4.5h	2h

❖ Training Tips (Optional):

- Automatic mixed precision (fp16)
- Gradient accumulation
- Ensemble

Linear Learning rate decay

Method 1: Adjust learning rate manually

- Decrement **optimizer.param_groups[0]["lr"]** by **learning_rate / total training step** per step

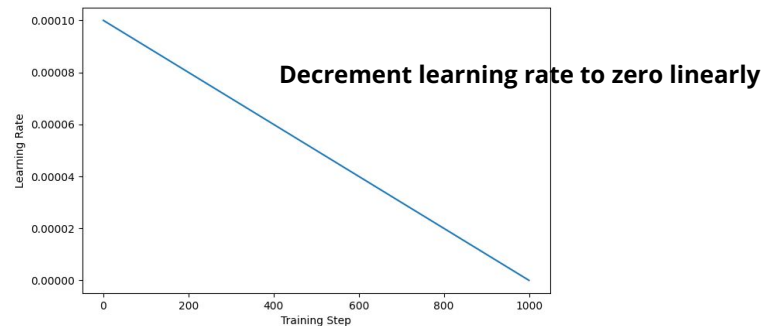
```
learning_rate = 1e-4
optimizer = AdamW(model.parameters(), lr=learning_rate)
total_step = 1000
for i in range(total_step):
    optimizer.param_groups[0]["lr"] -= learning_rate / total_step
```

This block is only an example!
You only need to add 1 or 2 lines

```
optimizer.param_groups[0]["lr"]
```

```
-1.5863074217500927e-18
```

Check whether learning rate after
training is very close to zero



Method 2: Adjust learning rate automatically by scheduler

- huggingface (Recommended) [Link](#)
- pytorch [Link](#)

```
scheduler = ...
for i in range(total_step):
    ...
    optimizer.step()
    scheduler.step()
    ... You only need to add 3 or 4 lines
```

You may also try other learning rate schedules (e.g. warmup)!

Doc stride Testing

```
##### TODO: Change value of doc_stride #####  
self.doc_stride = 150
```

We do not know where the answer is in testing ➡ split into windows!

e.g. window size = max_paragraph_len = 32

新加坡、馬來西亞的華文學術界在1970年代後開始統一使用簡體中文，然而繁體字在媒體中普遍存在著，例如華人商店的招牌、舊告示及許多非學術類中文書籍.....

Q: 新加坡的華文學術界在哪個年代後開始使用簡體中文？

A: 1970

Q: 馬來西亞的華人商店招牌主要使用什麼文字？

A: 繁體字

start position of
2nd window

start position of
1st window

doc_stride = distance between the start position of two consecutive windows

doc_stride is set to "max_paragraph_len" in sample code (i.e. no overlap)

What if answer is near the boundary of windows or across windows?

Hint: Overlapping windows

Preprocessing

Hint: How to prevent model from learning something it should not learn during training? (i.e. answers are not always near the middle of window)

```
##### TODO: Preprocessing #####
# Hint: How to prevent model from learning something it should not learn

if self.split == "train":
    # Convert answer's start/end positions in paragraph_text to start/end positions in tokenized_paragraph
    answer_start_token = tokenized_paragraph.char_to_token(question["answer_start"])
    answer_end_token = tokenized_paragraph.char_to_token(question["answer_end"])

    # A single window is obtained by slicing the portion of paragraph containing the answer
    mid = (answer_start_token + answer_end_token) // 2
    paragraph_start = max(0, min(mid - self.max_paragraph_len // 2, len(tokenized_paragraph) - self.max_paragraph_len))
    paragraph_end = paragraph_start + self.max_paragraph_len

    # Slice question/paragraph and add special tokens (101: CLS, 102: SEP)
    input_ids_question = [101] + tokenized_question.ids[:self.max_question_len] + [102]
    input_ids_paragraph = tokenized_paragraph.ids[paragraph_start : paragraph_end] + [102]
```

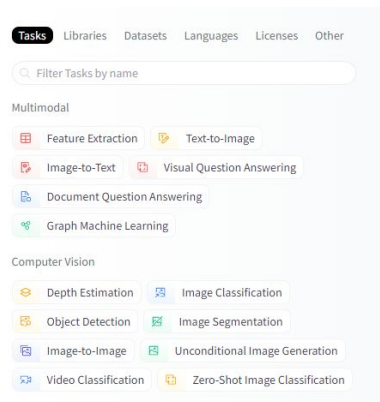
Other pretrained models

You can choose any model you like! [\[Link to pretrained models in huggingface\]](#)


Note 1: You are **NOT** allowed to use **pretrained models outside huggingface!**
(Violation = cheating = final grade x 0.9)

Note 2: Some models have  **Model card** describing details of the model


Note 3: Changing models may lead to error message, try it solve it yourself




Models 107

 Search Models


↑↓ Sort: Most Downloads

 bert-base-chinese


 Fill-Mask • Updated Dec 11, 2020 • 1,160k

 hfl/chinese-electra-180g-base-discriminator


Updated Mar 3 • 64k

 facebook/mbart-large-cc25


 Translation • Updated Mar 10 • 41k

 hfl/chinese-electra-180g-small-discriminator

Updated Mar 3 • 386k

 hfl/chinese-roberta-wwm-ext

 Fill-Mask • Updated Mar 3 • 45k

 hfl/chinese-bert-wwm

 Fill-Mask • Updated Mar 3 • 30k

Postprocessing

Hint: Open your prediction file to see what is wrong
(e.g. what if predicted end_index < predicted start_index?)

```
def evaluate(data, output):
    ##### TODO: Postprocessing #####
    # There is a bug and room for improvement in postprocessing
    # Hint: Open your prediction file to see what is wrong

    answer = ''
    max_prob = float('-inf')
    num_of_windows = data[0].shape[1]

    for k in range(num_of_windows):
        # Obtain answer by choosing the most probable start position / end position
        start_prob, start_index = torch.max(output.start_logits[k], dim=0)
        end_prob, end_index = torch.max(output.end_logits[k], dim=0)

        # Probability of answer is calculated as sum of start_prob and end_prob
        prob = start_prob + end_prob

        # Replace answer if calculated probability is larger than previous windows
        if prob > max_prob:
            max_prob = prob
            # Convert tokens to chars (e.g. [1920, 7032] --> "大 金")
            answer = tokenizer.decode(data[0][0][k][start_index : end_index + 1])

    # Remove spaces in answer (e.g. "大 金" --> "大金")
    return answer.replace(' ', '')
```

Training Tip: Automatic mixed precision

- PyTorch trains with 32-bit floating point (FP32) arithmetic by default
- Automatic Mixed Precision (AMP) enables automatic conversion of certain GPU operations from FP32 precision to half-precision (FP16)
- Offer about 1.5-3.0x speed up while maintaining accuracy

```
!pip install accelerate==0.16.0

# Change "fp16_training" to True to support automatic mixed
# precision training (fp16)
fp16_training = True
if fp16_training:
    accelerator = Accelerator(mixed_precision="fp16")
else:
    accelerator = Accelerator()
```

```
accelerator.backward(output, loss)
```

Reference:

[accelerate documentation](#)
[Intro to native pytorch automatic mixed precision](#)

Estimated training time

	T4	T4 (FP16)
Simple	20m	8m
Medium	20m	8m
Strong	1h	25m
Boss	6h	2.5h

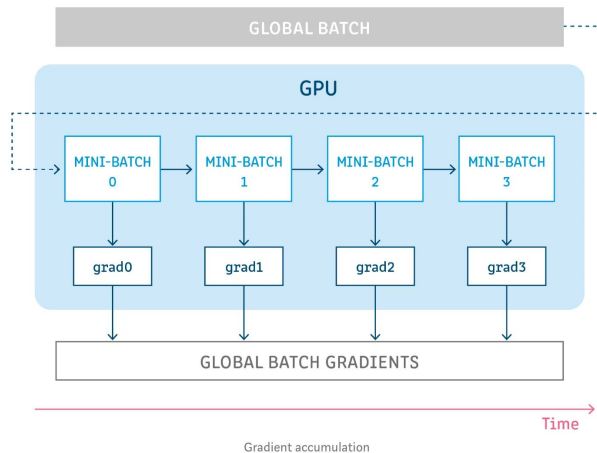
Warning: only work on some gpu
(e.g. T4, V100)

Training Tip: Gradient accumulation

Use it when gpu memory is not enough but you want to use larger batch size

- Split global batch into smaller mini-batches
- For each mini-batch: Accumulate gradient without updating model parameters
- Update model parameters

```
#### TODO: gradient_accumulation (optional)####  
# Note: train_batch_size * gradient_accumulation_steps = effective batch size  
# If CUDA out of memory, you can lower the train_batch_size and upper the gradient_accumulation_steps  
# Doc: https://huggingface.co/docs/accelerate/usage\_guides/gradient\_accumulation  
gradient_accumulation_steps = 16
```



Reference: [Gradient Accumulation in PyTorch](#)

4. Grading

Grading

Report	+4pts
Code Submission	+2pts
Simple Baseline (Public)	+0.5pt
Simple Baseline (Private)	+0.5pt
Medium Baseline (Public)	+0.5pt
Medium Baseline (Private)	+0.5pt
Strong Baseline (Public)	+0.5pt
Strong Baseline (Private)	+0.5pt
Boss Baseline (Public)	+0.5pt
Boss Baseline (Private)	+0.5pt

Kaggle (4pts)

Kaggle Link:

<https://www.kaggle.com/t/e001cad568dc4d77b6a5e762172f44d6>

Deadline: **5/12 (Fri.) 23:59**

Displayed name: <student ID>_<anything>

Testing Set: 3504 Questions (~50% public set, ~50% private set)

Evaluation Metric: Accuracy (Exact Match)

Submission Format: (csv)

Baselines	Public Score
Simple (0.5pt + 0.5pt)	0.46481
Medium (0.5pt + 0.5pt)	0.67139
Strong (0.5pt + 0.5pt)	0.75936
Boss (0.5pt + 0.5pt)	0.84108

```
ID,Answer
0,值勤制服
1,《諸羅縣志》
2,袁晁
3,柏林陸軍學院
4,長盤勝負
5,孫銘武
6,1874年
7,版圖向西擴張
```

Code Submission (2pts)

- **NTU COOL**

- **Deadline: 5/12 (Fri.) 23:59**
- Compress your code into **<student ID>_hw7.zip** (e.g. b10901118_hw7.zip)
- We can only see your last submission.
- **Do not submit your model or dataset.**
- If your code is not reasonable, your semester grade x 0.9.

Grading -- Bonus

- If your ranking in private set is top 3, you can choose to share a report to NTU COOL and get extra 0.5 pts.
- About the report
 - Your name and student_ID
 - Methods you used in code
 - Reference
 - in 200 words
 - Deadline is same as code submission
 - Please upload to NTU COOL's discussion of HW7

[Report Template](#)

5. Report

Report Questions: In-context learning (2 pts)

1. (2%) There are some difference between **fine-tuning** and **prompting**. Beside fine-tuning, in-context learning enable pre-trained model to give correct prediction on many downstream tasks with a few examples but without gradient descent. Please describe:
 - A. How encoder-only model (Bert-series) determines the answer in a extractive question answering task?
 - B. How decoder-only model (GPT-series) determines the answer in a extractive question answering task?

Report Questions: Prompts (2 pts)

2. (2%) The goal of this homework is to fine-tune a QA model. In this question, We wish you to try In-context learning on the same task and compare the difference between the prediction of in-context learning and fine-tuning.
- A. Try to Instruct the model with different prompts, and describe your observation on at least **3** pairs of prompts comparison.
 - B. Paste the screenshot of your prediction file in A. to Gradescope. (There are at least **6** screenshots)

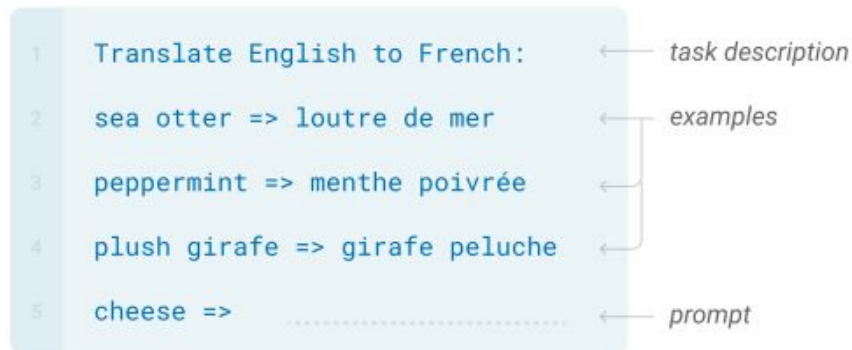
Notes:

- 1. The accuracy of prediction with few-shot learning doesn't matter when scoring**
- 2. The grading criteria are determined by Prof.Lee & TAs**

What is In-context Learning?

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.



Fine-tuning

The model is trained via repeated gradient updates using a large corpus of example tasks.



Report Questions: Instruct model with different examples

請從最後一篇的文章中找出最後一個問題的答案：

文章：〈文章1 內容〉

問題：〈問題1 敘述〉

答案：〈答案1〉

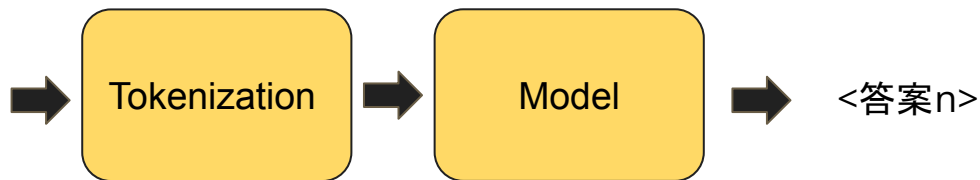
...

文章：〈文章n 內容〉

問題：〈問題n 敘述〉

答案：

Prompt



```
# K-shot learning
# Give model K examples to make it achieve better accuracy
# Note: (1) When K >= 4, CUDA_OUT_OF_MEMORY may occur.
#       (2) The maximum input length of XGLM is 2048
K = 2
```

Inference on

“hw7_in-context-learning_examples.json”,
which is the subset of “hw7_train.json”.

Report Questions: Bad Examples for Prompts Comparison

1. Compare between difference models with the similar prompts

I compared the performance of two different model:

"bloom-1b1-zh", "xglm-1.7B", and then I found...

2. Add some meaningless words to the prompts

I compared three prompt:

「根據文章, 找出問題的答案」,

「請根據文章, 找出問題的答案」,

「請根據文章, 找出以下問題的答案」,

and then I found...

Notes:

If comparisons as in example 2 **does improve** the performance, it may also be a good example. **The grading criteria are determined by Prof.Lee & TAs**

Report Questions: Good Examples for Prompts Comparison

1. Prompts with different languages

I compared two prompt:

“According to the passage, print the answer of the question”,

「根據文章, 找出問題的答案」, and then I found...

2. Prompts with different formats

I compared two prompt:

「根據文章, 找出問題的答案」,

「在P中, Q的答案是A」, and then I found...

6. Regulations

Regulations

- You should NOT plagiarize, if you use any other resource, you should cite it in the reference. (*)
- You should NOT modify your prediction files manually.
- Do NOT share codes or prediction files with any living creatures.
- Do NOT use any approaches to submit your results more than 5 times a day.
- **Do NOT search or use additional data.**
- **Do NOT use any pre-trained models outside huggingface.**
- Your **assignment will not be graded** and your **final grade x 0.9** if you violate any of the above rules.
- Prof. Lee & TAs preserve the rights to change the rules & grades.

(*) [Academic Ethics Guidelines for Researchers by the Ministry of Science and Technology](#)

Links

Kaggle sample code:

<https://www.kaggle.com/code/yuxianglin032/ml2023-hw7-question-answering-ipynb>

Colab sample code:

<https://colab.research.google.com/drive/1m0fQjlfkK9vAovxPj9Nd3-hQuxeZB2w1?usp=sharing>

Kaggle competition:

<https://www.kaggle.com/t/e001cad568dc4d77b6a5e762172f44d6>

Gradescope:

<https://www.gradescope.com/courses/515619/assignments/2789862>

Google meet:

<https://meet.google.com/tti-oonf-zrx>

If any questions, you can ask us via...

- NTU COOL (recommended)
 - https://cool.ntu.edu.tw/courses/24108/discussion_topics/188782
- Email
 - mlta-2023-spring@googlegroups.com
 - The title **should** begin with “[hw7]”
- TA hours
 - Each Friday During Class
 - Chinese TA hour: Monday 19:00 - 20:00
 - English TA hour: Monday 20:00 - 21:00
 - link: <https://meet.google.com/tti-oonf-zrx>