Taipei QA BOT

Using BERT

P.HUANG

https://github.com/p208p2002/taipei-QA-BERT

資料集

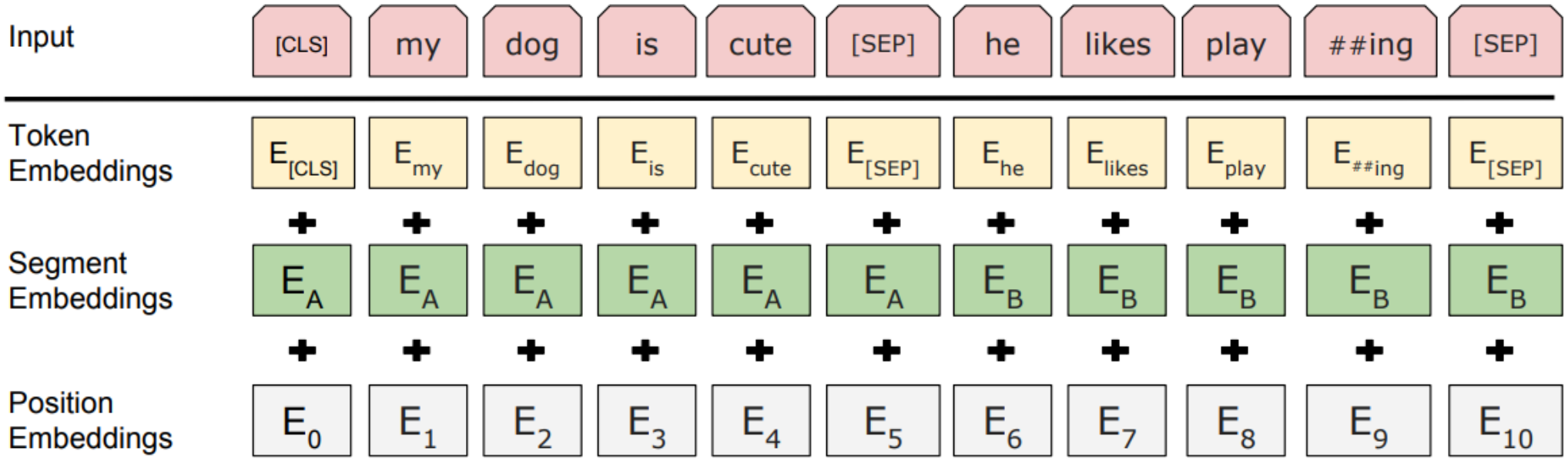
https://raw.githubusercontent.com/p208p2002/taipei-QA-BERT/master/Taipei_QA_new.txt

台北QA問答

- 資料筆數:7986
- 資料總類別:149

臺北市政府文化局 2018臺北藝術節FAQ
臺北市政府文化局 臺北市信義區Neo19大樓後方人行道後場使用作業通告(107/06/03-05)
臺北市政府文化局 2018台北電影節FAQ
臺北市政府文化局 臺灣新文化運動紀念館位置與聯絡方式
臺北市政府文化局 臺灣新文化運動紀念館開館及參觀時間?
臺北市政府文化局 2018臺北兒童藝術節FAQ
臺北市政府文化局 藝文補助之申請時間及計畫執行時間
臺北市政府文化局 新芳春茶行地點、開放時間、聯絡方式
臺北市政府文化局 如何前往松山文創園區
臺北市政府文化局 被指定古蹟之建築物要符合何種條件，才能辦理容積移轉？
臺北市政府文化局 所有權人接獲古蹟公告後，如不服指定程序該如何處理？
臺北市政府文化局 臺北市政府古蹟歷史建築紀念建築聚落建築群考古遺址史蹟及文化景觀審議會如何組成？
臺北市政府文化局 如何申請「古蹟」指定、「歷史建築」登錄？「古蹟」指定、「歷史建築」登錄的程序為何？
臺北市政府文化局 錢穆故居營業時間？是否須收取門票？聯絡電話？交通資訊？
臺北市政府文化局 林語堂故居營業時間？是否須收取門票？聯絡電話？交通資訊？
臺北市政府文化局 臺北二二八紀念館開館及參觀時間?交通方式?
臺北市政府文化局 所有權人的房舍被指定為古蹟後，可以獲得什麼權益？
臺北市政府文化局 施工中，如發現疑似古蹟之建築物該如何處理？
臺北市政府文化局 我的家鄰近古蹟，我的權益會不會受損？
臺北市政府文化局 如何解除「古蹟」或「歷史建築」之身分？
臺北市政府文化局 古蹟週邊的開發行為有什麼限制？
臺北市政府文化局 什麼人有資格修復古蹟？
臺北市政府文化局 古蹟遭受重大災害之破壞，政府會如何處理
臺北市政府文化局 士林官邸正館參觀Q&A
臺北市政府文化局 紫藤廬營業時間？是否須收取門票？聯絡電話？
臺北市政府文化局 台北故事館營業時間？是否須收取門票？聯絡電話？
臺北市政府文化局 臺北服飾文化館(西園二九)營業時間？是否須收取門票？聯絡電話？交通資訊？
臺北市政府文化局 李國鼎故居開放時間？有無須收取門票？

BERT Input Review



Using BertTokenizer

```
from transformers import BertTokenizer

tokenizer = BertTokenizer(vocab_file='bert-base-chinese-vocab.txt')
```

Token Embedding

tokenizer.tokenize

STR_INPUT = “what’re you doing”

tokenizer.tokenize(STR_INPUT)

=> what are you do ##ing

tokenizer.convert_tokens_to_ids

WORD_PIECE_STRING = “what are you do ##ing”

tokenizer.convert_tokens_to_ids(WORD_PIECE_STRING)

=> [190,256,79,13,65]

id從vacab.txt查詢而來

tokenizer.build_inputs_with_special_tokens

tokenizer.build_inputs_with_special_tokens(SENTENCE_A, SENTENCE_ B)

=>[CLS] SENTENCE_A[SEP] SENTENCE_B[SEP]

tokenizer.build_inputs_with_special_tokens(A)

=>[CLS] SENTENCE_A[SEP]

* tokenizer.build_inputs_with_special_tokens() 輸入的實際上是**WordPiece ids**，輸出也會是**WordPiece ids**

Segment Embedding and Position Embedding

Segment Embedding and Position Embedding

假設我們的輸入

[CLS] what are you do ##ing[SEP]

轉成id後

- =>[100,500,465,46,321,8744,102]

Segment Embedding

- [0,0,0,0,0,0,0]

Postion Embedding

- [1,1,1,1,1,1,1] (假設你需要補齊batch長度，剩餘補0)

資料前處理

答案字典 – id 與 label互轉

```python
class AnsDic(object):
    def __init__(self, answers):
        self.answers = answers #全部答案(含重複)
        self.answers_norepeat = sorted(list(set(answers))) # 不重複
        self.answers_types = len(self.answers_norepeat) # 總共多少類
        self.ans_list = [] # 用於查找id或是text的list
        self._make_dic() # 製作字典

    def _make_dic(self):
        for index_a,a in enumerate(self.answers_norepeat):
            if a != None:
                self.ans_list.append((index_a,a))

    def to_id(self,text):
        for ans_id,ans_text in self.ans_list:
            if text == ans_text:
                return ans_id

    def to_text(self,id):
        for ans_id,ans_text in self.ans_list:
            if id == ans_id:
                return ans_text

    @property
    def types(self):
        return self.answers_types

    @property
    def data(self):
        return self.answers

    def __len__(self):
        return len(self.answers)
```

將資料轉換輸入格式

等一下準備一起丟進去
Dataloader做batch切割

Input WordPiece ids

Input Position ids

Input Segment ids

對應投影片3
的三個輸入
Embedding

```
data_features = {'input_ids':input_ids,
                 'input_masks':input_masks,
                 'input_segment_ids':input_segment_ids,
                 'answer_lables':answer_lables,
                 'question_dic':question_dic,
                 'answer_dic':ans_dic}

output = open('trained_model/data_features.pkl', 'wb')
pickle.dump(data_features,output)
return data_features
```

問題與答案字典Class
存起來供predict
找查id與text

def convert_data_to_feature()

輸入格式轉Dataset

```python
def makeDataset(input_ids, input_masks, input_segment_ids, answer_lables):
    all_input_ids = torch.tensor([input_id for input_id in input_ids], dtype=torch.long)
    all_input_masks = torch.tensor([input_mask for input_mask in input_masks], dtype=torch.long)
    all_input_segment_ids = torch.tensor([input_segment_id for input_segment_id in input_segment_ids], dtype=torch.long)
    all_answer_lables = torch.tensor([answer_lable for answer_lable in answer_lables], dtype=torch.long)

    full_dataset = TensorDataset(all_input_ids, all_input_masks, all_input_segment_ids, all_answer_lables)

    # 切分訓練與測試資料集
    train_size = int(0.8 * len(full_dataset))
    test_size = len(full_dataset) - train_size
    train_dataset, test_dataset = torch.utils.data.random_split(full_dataset, [train_size, test_size])

    return train_dataset,test_dataset
```

Dataset餵給Dataloader

```
train_dataset, test_dataset = makeDataset(input_ids = input_ids, input_masks = input_masks, input_segment_ids = input_segment_ids, ans
train_dataloader = DataLoader(train_dataset,batch_size=16,shuffle=True)
test_dataloader = DataLoader(test_dataset,batch_size=16,shuffle=True)
```

Model fine-tune

加載BERT預訓練模型與設定

```
from transformers import BertConfig, BertForSequenceClassification, BertTokenizer, AdamW
```

```
bert_config, bert_class, bert_tokenizer = (BertConfig, BertForSequenceClassification, BertTokenizer)
```

```
config = bert_config.from_pretrained('bert-base-chinese',num_labels = 149)
model = bert_class.from_pretrained('bert-base-chinese', from_tf=bool('.ckpt' in 'bert-base-chinese'), config=config)
```

覆寫設定

其他設定與預設值可以參考Transformers下的class BertConfig 與class PretrainedConfig

Fine-tune

```
model.zero_grad()
for epoch in range(30):
    running_loss_val = 0.0
    running_acc = 0.0
    for batch_index, batch_dict in enumerate(train_dataloader):
        model.train()
        batch_dict = tuple(t.to(device) for t in batch_dict)
        outputs = model(
            batch_dict[0],
            # attention_mask=batch_dict[1],
            labels = batch_dict[3]
            )
        loss,logits = outputs[:2]
        loss.sum().backward()
        optimizer.step()
        # scheduler.step()  # Update learning rate schedule
        model.zero_grad()

        # compute the loss
        loss_t = loss.item()
        running_loss_val += (loss_t - running_loss_val) / (batch_index + 1)

        # compute the accuracy
        acc_t = compute_accuracy(logits, batch_dict[3])
        running_acc += (acc_t - running_acc) / (batch_index + 1)

        # log
        print("epoch:%2d batch:%4d train_loss:%2.4f train_acc:%3.4f"%(epoch+1, batch_index+1, running_loss_val, running_acc))
                    0.0: float
    running_loss_val = 0.0
    running_acc = 0.0
    for batch_index, batch_dict in enumerate(test_dataloader):
        model.eval()
        batch_dict = tuple(t.to(device) for t in batch_dict)
        outputs = model(
            batch_dict[0],
```

順序對應
投影片 14

IntelliCode Python support requires you to u
Python Language Server (preview).
來源: Visual Studio IntelliCode (延... Enable it and

第 42 行，第 5 欄　空格: 4　UTF-

結果

```
epoch:23 batch: 100 test_loss:0.9882 test_acc:77.7981
```

Predict

結果預測與轉回答案標籤

```python
def toBertIds(q_input):
    return tokenizer.build_inputs_with_special_tokens(tokenizer.convert_tokens_to_ids(tokenizer.tokenize(q_input)))
```

```python
class AnsDic(object):
    def __init__(self, answers):
        self.answers = answers #全部答案(含重複)
        self.answers_norepeat = sorted(list(set(answers))) # 不重複
        self.answers_types = len(self.answers_norepeat) # 總共多少類
        self.ans_list = [] # 用於查找id或是text的list
        self._make_dic() # 製作字典

    def _make_dic(self):
        for index_a,a in enumerate(self.answers_norepeat):
            if a != None:
                self.ans_list.append((index_a,a))

    def to_id(self,text):
        for ans_id,ans_text in self.ans_list:
            if text == ans_text:
                return ans_id

    def to_text(self,id):
        for ans_id,ans_text in self.ans_list:
            if id == ans_id:
                return ans_text

    @property
    def types(self):
        return self.answers_types

    @property
    def data(self):
        return self.answers

    def __len__(self):
        return len(self.answers)
```

`model.eval()` —— 不要忘記他

```python
# predict
outputs = model(input_ids)
predicts = outputs[:2]
predicts = predicts[0]
max_val = torch.max(predicts)
label = (predicts == max_val).nonzero().numpy()[0][1]
ans_label = answer_dic.to_text(label)
```

預測結果

```
(base) torch12@4639e08a1246:~/project/taipei-qa-bert$ python predict.py
為何路邊停車格有編號的要收費，無編號的不用收費
臺北市停車管理工程處

債權人可否向稅捐稽徵處申請查調債務人之財產、所得資料
臺北市稅捐稽徵處稅務管理科

想做大腸癌篩檢，不知如何辦理
臺北市立聯合醫院
```

End

Using GPU

模型加載

device = torch.device('cuda’)

model = # new model class

model.to(device)

每一個Batch把資料從CPU RAM 推到 GPU RAM

batch = tuple(t.to(device) for t in batch)

僅適用於GPU = 1，若要同時使用多張GPU要再額外設定