Grammar Correction

英文文法校正

110034545 張芳綺

CONTENTS

目錄

PART
1

研究動機與目的

研究動機與目的

研究動機

在進行英文寫作時常常會犯一些文法上的小錯誤，若無細心檢查或對文法不熟悉可能會繼續使用錯誤的語句而不自知

研究目的

訓練一個能夠進行文法除錯的模型，輸入錯誤的英文句子後能夠輸出校正過後文法正確的英文句子，節省人工檢查的時間與成本

5W1H

Who 進行英文寫作者或文法校正人員

When 當文法有錯誤時

What 進行英文文法校正除錯

Why 自動偵測錯誤並校正，節省人工成本

Where 要進行文法除錯的任何地點

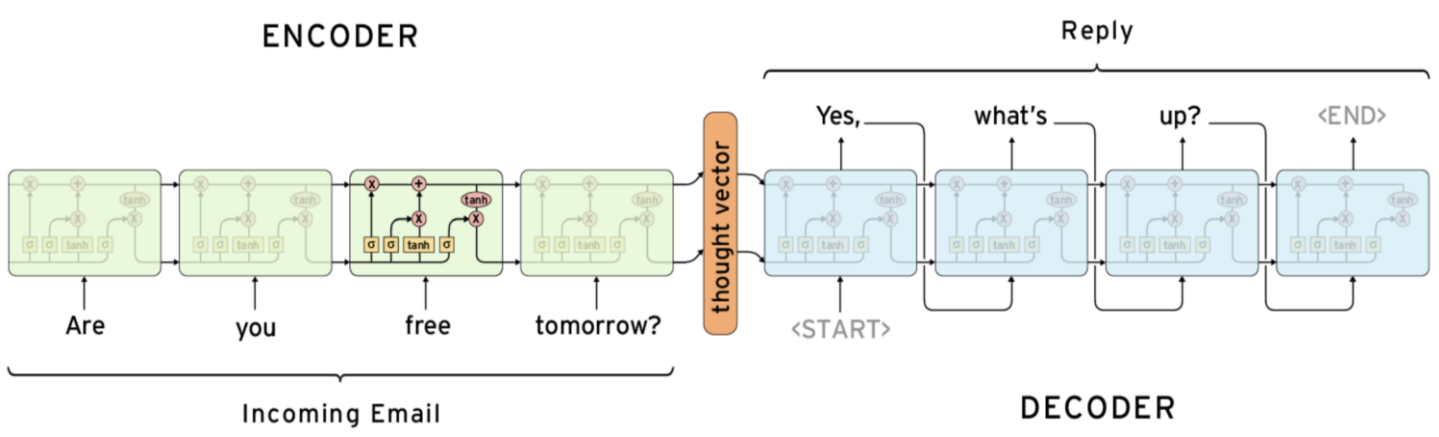
How 深度學習(Seq2Seq、LSTM)

PART
2

研究方法

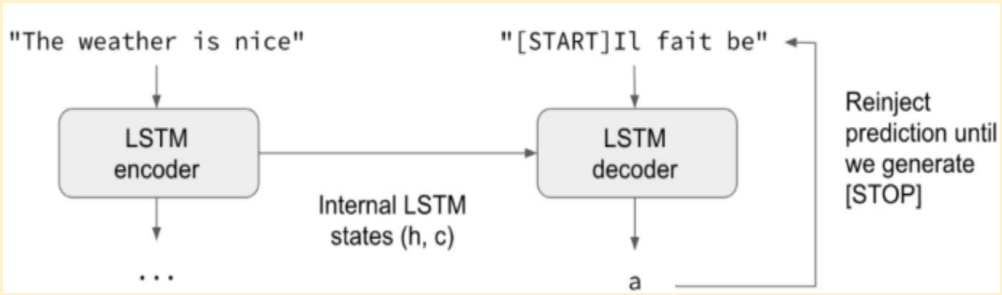
研究方法

Seq2Seq 架構



- Seq2seq架構也被稱作encoder-decoder framework。Encoder把輸入的文字轉換成機器理解的context vector，Decoder把context vector轉換成我們能理解的文字。

LSTM



- 此架構串連兩個LSTM隱藏層，一個隱藏層擔任編碼器(encoder)的角色，以LSTM處理，但不管輸出，只保留記憶狀態(State)，讓另一個隱藏層使用，另一個隱藏層額外再考慮前文，兩者綜合起來，預測下一個翻譯的字，這個機制為解碼器(decoder)。

PART
3

Case Study

資料集描述

```
Number of samples: 153182
Number of unique input tokens: 41936
Number of unique output tokens: 41713
Max sequence length for inputs: 282
Max sequence length for outputs: 285
```

- 共有153182組句子
- 前一句為錯誤文法，後一句為正確文法
- 錯誤文法的句子中共有41936個相異單字
- 正確文法的句子中共有41713個相異單字

```
In my opinion the TV channels should show the violent movies and TV shows in night , not in the afternoon .     In my opinion the TV channels should show the violent movies and TV shows at night , not in the afternoon .
There I began to study Family Therapy , discipline that I enjoy too much , because I 've seen I can help people about their troubles ..     There I began to study Family Therapy , discipline that I enjoy too much , because I 've seen I can help people with their troubles ..
It seems  me , that in this song there are romantic notes  .    It seems to me , that in this song there are romantic notes .
So , I want to say that I admire your right disposition and sacrifice in solve the problems related a your phobia , as for example going out the company . So , I want to say that I admire your right disposition and sacrifice to solve the problems related to your phobia , as for example going out the company .
I  graduated  Chung university .        I  graduated from Chung university .
I like listening  music and travelling on the weekends .        I like listening to music and travelling on the weekends .
Let me tell you why you should apply  this job .        Let me tell you why you should apply for this job .
In Munchen I went been in Beer Square . In Munchen I went to in Beer Square .
Next I go to the gym and I practice exercises for a little while , because I need to go for the college .       Next I go to the gym and I practice exercises for a little while , because I need to go to the college .
I am now at Beijing , it 's amazing here ; new modern buildings with a lot of Chinese history . I am now in Beijing , it 's amazing here ; new modern buildings with a lot of Chinese history .
Here in Brazil it is very common we have problems with family business .        Here in Brazil it is very common to have problems in family business .
I go for every game and cheer for the home team .       I go to every game and cheer for the home team .
I love painting romantic landscapes , reading books , listening  all types of music and playing video games .   I love painting romantic landscapes , reading books , listening to all types of music and playing video games .
Shirt  $ 30.00  Shirt for $ 30.00
```

實作流程步驟

1. 資料預處理

- 文字轉數字
- padding
- 處理dirty data

2. Word to vector

- 用word2vec實作embedding_layer
- 數字轉向量

3. Encode

- 將embedded data丟入encoder (LSTM1)
- output 一個context vector

4. Decode

- 將context vector丟入decoder (LSTM2)
- decode 出target sentence

資料預處理

- ◆將文本資料讀進來後把錯誤句子(input)和正確句子(target)分開存取
- ◆把錯誤句子和正確句子中所有出現過的單子用dictionary分開存取
- ◆target句子部分要加上<Start>、<End>的標籤
- ◆用np.zeros 的方式將句子做padding，保證每個句子長度一樣

```
with open(data_path, 'r', encoding='utf-8') as f:
    lines = f.read().split('\n')
for line in lines[: min(num_samples, len(lines) - 1)]:
    input_text, target_text = line.split('\t')
    # We use "tab" as the "start sequence" character
    # for the targets, and "\n" as "end sequence" character.
    target_text = '\t' + target_text + '\n'
    input_texts.append(input_text)
    target_texts.append(target_text)
    # for char in str(input_text).split():
    for char in mysplit(input_text):

        if char not in input_characters:
            input_characters.add(char)
    # for char in str(target_text).split():
    for char in mysplit(target_text):

        if char not in target_characters:
            target_characters.add(char)
```

```
input_token_index = dict(
    [(char, i) for i, char in enumerate(input_characters)])
target_token_index = dict(
    [(char, i) for i, char in enumerate(target_characters)])
```

```
encoder_input_data = np.zeros(
    (len(input_texts), max_encoder_seq_length),
    dtype='int32')
decoder_input_data = np.zeros(
    (len(input_texts), max_decoder_seq_length),
    dtype='int32')
decoder_target_data = np.zeros(
    (len(input_texts), max_decoder_seq_length,1),
    dtype='int32')
```

資料預處理

- ◆將句子單字根據dictionary 來轉換成數字的形式，其中 0 代表沒有意義
- ◆Dirty data 的清理：因為原本的資料集中屬於正確文法的句子裡也有一些錯誤，所以手動進行糾正，把discuss、explain、mention、describe後的about刪除

```
for i, (input_text, target_text) in enumerate(zip(input_texts, target_texts)):
    input_text=mysplit(input_text)
    target_text=mysplit(target_text)
    for t, char in enumerate(input_text):
        encoder_input_data[i, t] = input_token_index[char]
    for t, char in enumerate(target_text):
        # decoder_target_data is ahead of decoder_input_data by one timestep
        index=target_token_index[char]
        decoder_input_data[i, t] = index
        if t > 0:
            # decoder_target_data will be ahead by one timestep
            # and will not include the start character.
            decoder_target_data[i, t - 1,0] = index
# Define an input sequence and process it.
```

```
def  data_clean(input_data):
    clean=[]
    for  data  in  (input_data):
        clean_data=re.sub(r'discuss  about',r'discuss',data)
        clean_data=re.sub(r'explain  about',r'explain',clean_data)
        clean_data=re.sub(r'mention  about',r'mention',clean_data)
        clean_data=re.sub(r'describe  about',r'describe',clean_data)
        clean.append(clean_data)
    return  clean
```

Word to vector

- 利用pretrained好的word2vec模型來實作embedding_layer
- 將已經轉為數字的句子單字當作input丟入embedding layer 轉換成向量

```
EMBEDDING_DIM=300
## input
embedding_matrix = np.zeros((num_encoder_tokens, EMBEDDING_DIM))
for word, i in input_token_index.items():#word_index.items():
    if i >= MAX_NB_WORDS:
        continue
    embedding_vector = embeddings_index.get(word)
    if embedding_vector is not None:
        # input data中的詞不再word2vec裡向量取0，有的話取word2vec的向量
        embedding_matrix[i] = embedding_vector

## target
embedding_matrix1 = np.zeros((num_decoder_tokens, EMBEDDING_DIM))
for word, i in target_token_index.items():#word_index.items():
    if i >= MAX_NB_WORDS:
        continue
    embedding_vector = embeddings_index.get(word)
    if embedding_vector is not None:
        # target data中的詞不再word2vec裡向量取0，有的話取word2vec的向量
        embedding_matrix1[i] = embedding_vector
# 將訓練好的词向量加载如embedding layer
#  trainable = False，代表词向量不更新
embedding_layer = Embedding(num_encoder_tokens,
                            EMBEDDING_DIM,
                            weights=[embedding_matrix],
                            trainable=False)
embedding_layer1 = Embedding(num_decoder_tokens,
                            EMBEDDING_DIM,
                            weights=[embedding_matrix1],
                            trainable=False)
```

```
encoder_inputs =  Input(shape=(None, ), dtype='int32',)
encoder_embedding = embedding_layer(encoder_inputs)
```

Encode and Decode

- Encode：將embedded data當作input丟入encoder (LSTM1)後output 一個 context vector
- Decode：將context vector丟入decoder(LSTM2) decode出target sentence

```
encoder = (LSTM(latent_dim, return_state=True))
encoder_outputs, state_h, state_c = encoder(encoder_embedding)
# We discard `encoder_outputs` and only keep the states.
encoder_states = [state_h, state_c]
```

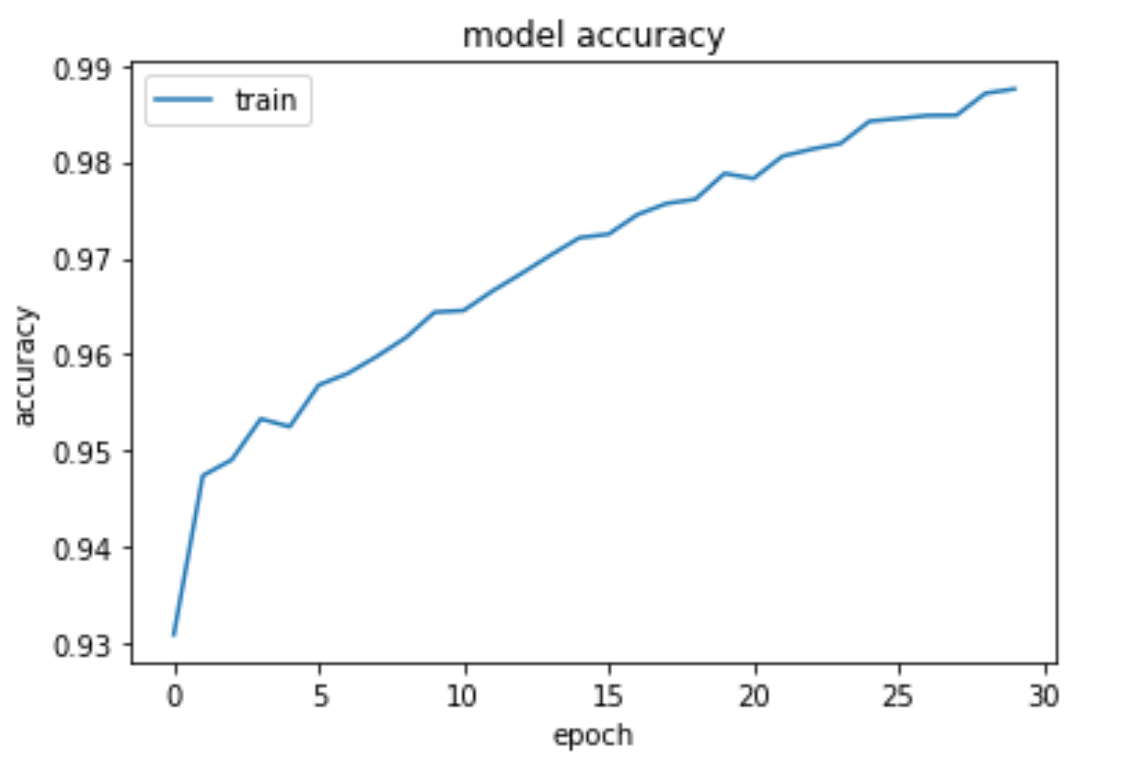
```
# Set up the decoder, using `encoder_states` as initial state.
decoder_inputs =  Input(shape=(None, ), dtype='int32',)
# We set up our decoder to return full output sequences,
# and to return internal states as well. We don't use the
# return states in the training model, but we will use them in inference.
decoder_embedding = embedding_layer1(decoder_inputs)
decoder_lstm = (LSTM(latent_dim, return_sequences=True, return_state=True))
decoder_outputs, _, _ = decoder_lstm(decoder_embedding,
                                     initial_state=encoder_states)
decoder_dense = (Dense(num_decoder_tokens, activation='softmax'))

# decoder_dense = TimeDistributed(Dense(num_words, activation='softmax'))
decoder_outputs = decoder_dense(decoder_outputs)
```

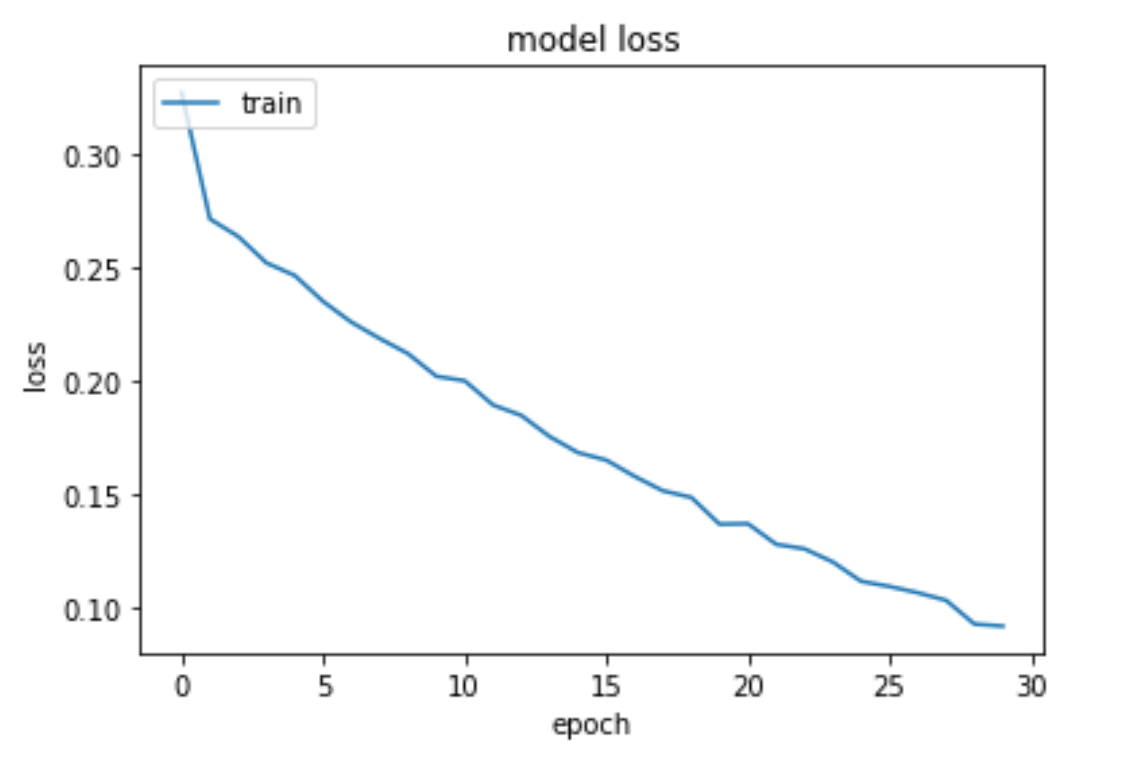
模型架構

- 兩層Input layer，兩層Embedding layer，兩層LSTM，一層Dense
- Encoder：input_1、embedding_1、lstm_1
- Decoder：input_2、embedding_2、lstm_2、dense_1

```
Model: "model_2"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)           [(None, None)]       0           []

 input_2 (InputLayer)           [(None, None)]       0           []

 embedding_1 (Embedding)        (None, None, 300)    12580800    ['input_1[0][0]']

 embedding_2 (Embedding)        (None, None, 300)    12513900    ['input_2[0][0]']

 lstm_1 (LSTM)                  [(None, 300),        721200      ['embedding_1[0][0]']
                                 (None, 300),
                                 (None, 300)]

 lstm_2 (LSTM)                  [(None, None, 300),  721200      ['embedding_2[0][0]',
                                 (None, 300),                     'lstm_1[0][1]',
                                 (None, 300)]                     'lstm_1[0][2]']

 dense_1 (Dense)                (None, None, 41713)  12555613    ['lstm_2[0][0]']

==================================================================================================
Total params: 39,092,713
Trainable params: 13,998,013
Non-trainable params: 25,094,700
```

模型訓練成果 —30個epoch





模型訓練成果 —短句

Input sentence: I really like wear jeans and t - shirts .
Decoded sentence: I really like to wear jeans and T - shirts .

Input sentence: Many people like play golf .
Decoded sentence: Many people like to play volleyball .

Input sentence: I 'm working on my office .
Decoded sentence: I ' m working in my office .

Input sentence: Also you should n't yell in the street .
Decoded sentence: Also you should n ' t yell at the street .

Input sentence: I also like play a guitar .
Decoded sentence: I also like to play a guitar .

Input sentence: Let me tell you why you should apply this job .
Decoded sentence: Let me tell you why you should apply for this job .

Input sentence: I am sending out invitations on 30 of my friends .
Decoded sentence: I am sending out invitations to 30 of my friends .

Input sentence: Welcome my home .
Decoded sentence: Welcome to my home .

Input sentence: Sometimes I talk phone and send emails .
Decoded sentence: Sometimes I talk on the phone and send emails .

Input sentence: I am good shape .
Decoded sentence: I am in good shape .

Input sentence: I worked with him 5 years in Metis company .
Decoded sentence: I worked with him for 5 years at a company .

Input sentence: It will be in Rio de Janeiro and I will arrive there about 6 pm .
Decoded sentence: It will be on July , , I will go there and and I will go there .

模型訓練成果 —長句

Input sentence: I think our future is positive and colourful ; every day we have new chances working on the quality of our thoughts and words .
Decoded sentence: I think the future is very good and and have a better work in the market and the quality of the people will be more careful with you .

Input sentence: Before the age of laptops we used note books to take notes of everything , but now we can use the laptops also on the meetings .
Decoded sentence: Before the online catalog they have to spend time on the internet , I will send it to the people to the people in the world and s more .

Input sentence: In my opinion the TV channels should show the violent movies and TV shows in night , not in the afternoon .
Decoded sentence: In my opinion the TV channels should show the violence and sex and The movie is to go to the movies .

PART
4

結論與改善方向

結論與改善方向

- 從成果可以看出目前的模型在短句的表現成果較佳, 長句的處理沒辦法準確的更改錯誤, 也可能直接改變原本的語意。
- 在訓練時限制句子長度或再訓練多一點epoch可能會訓練得更好。
- 可能可以增加attention layer幫助模型專注在應該專注的地方。
- 可能可以把LSTM換成Transformer來提升效果,或是參考google的bert模型架構。

參考資料

- https://gau820827.medium.com/%E6%95%99%E9%9B%BB%E8%85%A6%E5%AF%AB%E4%BD%9C-ai%E7%90%83%E8%A9%95-seq2seq%E6%A8%A1%E5%9E%8B%E6%87%89%E7%94%A8%E7%AD%86%E8%A8%98-pytorch-python3-31e853573dd0

- https://ithelp.ithome.com.tw/articles/10194403?fbclid=IwAR1PegT2qp7KBMaIusY6YANhEsmbmHFS9PFGl0CBkHI7k38-JRBbKnAPOFc

- https://ithelp.ithome.com.tw/articles/10223055

- https://dataxujing.github.io/seq2seqlearn/chapter1/

THANK YOU