國立清華大學

智慧化企業整合

Project3
真假新聞辨識

指導教授:邱銘傳 教授

110034562 黃鉦程

111/01/07

第一章、前言

1.1 背景介紹

科技網路蓬勃發展，使得現在成為了資訊爆炸的時代，很多的問題都能夠透過網路搜索，獲取最新的資訊，但也衍生了假新聞的問題，訊息可能是不實資訊故意誤導大眾，帶來政治、經濟、心理成就感和利益的新聞或宣傳，包括通過傳統新聞媒體（印刷和廣播）或線上社群媒體傳播的故意錯誤資訊或惡作劇，社交媒體上傳播錯誤訊息可能導致暴力、自殺等問題，成為一種負面循環。虛假新聞破壞了媒體的正當報導，使記者更難以報導重大新聞報導。BuzzFeed 的一項分析發現，關於 2016 年美國總統選舉的前 20 個虛假新聞報導在 Facebook 上的點擊率超過了 19 個主要媒體的前 20 則選舉報導。匿名代管的虛假新聞網站也缺乏已知的出版商也受到批評，因為它們難以起訴假新聞的來源。

1.2 5W1H

新聞每天都會看到，但是有些資訊卻是假的，抑或是有時候政治的操弄，使得訊息是假的，令人難以分辨。我們透過 5W1H 來了解此次的研究目的及研究方法。

What?	假新聞時有所聞
Where?	網路上
When?	閱讀新聞的時候
Who?	幫助社會大眾辨別
Why?	減少資訊誤導、暴力、自殺行為發生
How?	運用 LSTM 辨別新聞真假

第二章、文獻回顧

2.1 自然語言處理（Natural Language Processing , NLP）

自然語言處理是人工智慧和語言學領域的分支學科，此領域探討如何處理及 運用自然語言，包括多方面和步驟，基本有認知、理解、生成等部分。自然語言 的認知與理解，是透過複雜的數學模型及演算法來讓電腦把輸入的語言變成有意 義的符號和關係，然後根據目的再處理；而自然語言生成系統則是把電腦數據轉 化為自然語言。

早期的 NLP 技術主要基於統計的概念去訓練模型，讓演算法閱讀大量類似字典的文章段落，再讓演算法計算單字、句子出現的機率，然而此種方式無法使系統很好地辨識複雜的文法，同時，這樣子的模型所產生的字句更是生硬且結構錯亂。但隨著深度學習與演算法模型的突破，新的訓練方式已能更好的處理以上所提的問題。

2.2 遞歸神經網路（Recurrent Neural Network , RNN）

遞歸神經網路就是進行預測（或者回歸）的時候，不僅要考慮到當前時刻的輸入，還要考慮上一個時刻的輸入，預測的結果不僅與當前狀態有關，還與上一個時刻的狀態有關。遞歸神經網路常應用在處理時間、空間序列上有強關聯的訊息，因此常被應用在 NLP（Natural Language Processing，自然語言處理）領域上。由於我們在閱讀文章時，是根據上下文來理解文章意義，RNN 的概念在於將狀態在自身網絡中循環傳遞，因此可以接受更廣泛的時間序列結構輸入，允許訊息持續存在。

2.3 長短期記憶模型（Long Short-Term Memory , LSTM）

因應 RNN 模型短期記憶的先天限制，LSTM 模型於是設計了一些方式解決。也就是除了 RNN 的記憶路徑之外，新增了遺忘、選擇和忽視的路徑讓神經網絡能更適合記憶長期訊息。長短期記憶模型是一種特殊的 RNN 模型，它不同於 RNN 只有單一的神經網路層（tanh），而是有四個層，以特別的方式進行溝通，LSTM 模型包括輸入門(input gate)、輸出門(output gate)、忘記門(forget gate)和專門的記憶存儲單元(memory cell)，適合於處理和預測時間序列中間隔和延遲非常長的重要事件。

第三章、資料前處理

2.1 資料來源

本研究由 Kaggle 網站上的資料集中，取得真新聞與假新聞的資料集作為模型的資料來源。原始資料集中，真新聞總共有 21417 筆資料，而假新聞中總共有 23481 筆資料。

2.2 資料欄位

此次資料及總共有四個資訊，分別有標題、內文、主題及日期。標題為發布新聞時的重點摘要，內文是新聞所要講述的內容，主題為此篇新聞分類，日期則是新聞的發布日期。

1. Title：標題。
2. text：內文。
3. Subject：主題。
4. Date：日期。

	title	text	subject	date
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017

2.3 資料前處理

(1)導入函式庫與資料集

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import nltk
import re
from wordcloud import WordCloud
nltk.download('stopwords')
nltk.download('punkt')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
True
```

```
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Embedding, LSTM, Conv1D, MaxPool1D
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, accuracy_score
```

```
fake = pd.read_csv('/content/drive/MyDrive/2021Fall/5.智慧化企業整合/7.Project3/Fake.csv')
```

```
real = pd.read_csv('/content/drive/MyDrive/2021Fall/5.智慧化企業整合/7.Project3/True.csv')
```

(2)出版社去除、空值移除

真新聞有出版社的資訊，而假新聞沒有此一資訊，將文字與出版社去除。

```
real.head()
```

	title	text
0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...
2	Senior U.S. Republican senator: 'Let Mr. Muell...	WASHINGTON (Reuters) - The special counsel inv...

```
real.head()
```

	title	text
0	As U.S. budget fight looms, Republicans flip t...	The head of a conservative Republican faction...
1	U.S. military to accept transgender recruits o...	Transgender people will be allowed for the fi...
2	Senior U.S. Republican senator: 'Let Mr. Muell...	The special counsel investigation of links be...

將新聞標題與內文有嚴重缺失的移除。

```
[index for index,text in enumerate(real.text.values) if str(text).strip() == '']
```

```
[8970]
```

```
real.iloc[8970]
```

```
title      Graphic: Supreme Court roundup
text
subject                       politicsNews
date                       June 16, 2016
Name: 8970, dtype: object
```

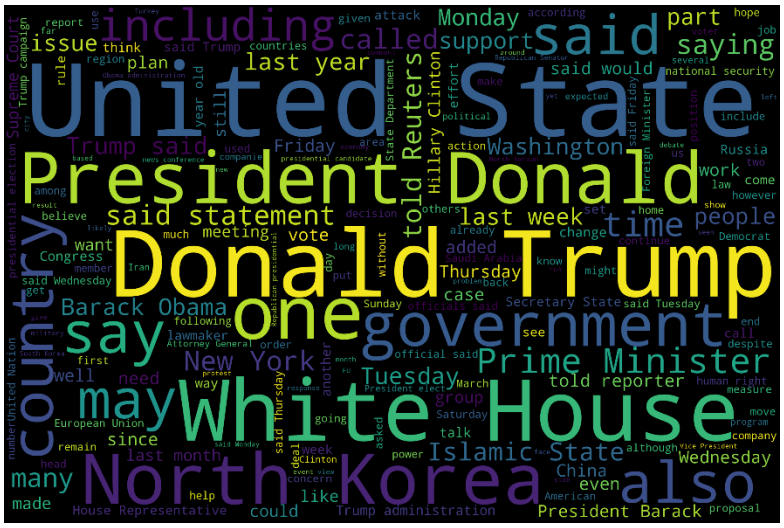
```
real = real.drop(8970, axis=0)
```

(3)字詞分析視覺化

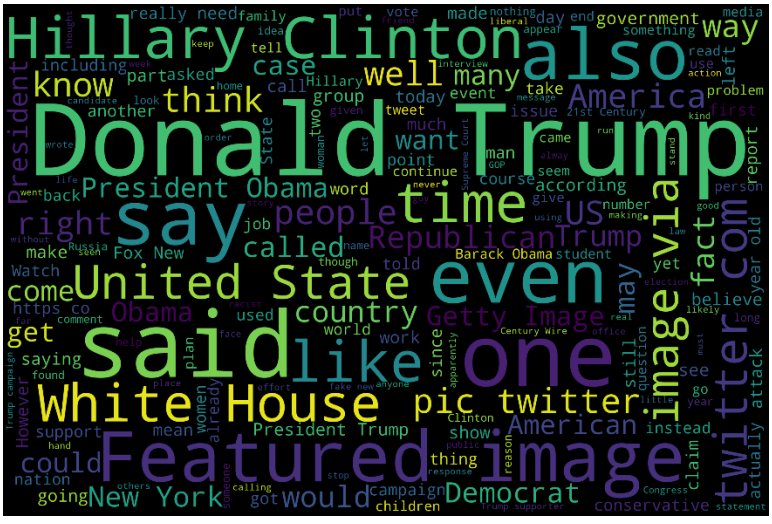
利用函式 WorldCloud()，文字雲了解資料集關鍵字，使用 nltk 的 stopwords 來去除停用詞。

```
text = ''
for news in fake.text.values:
        text += f" {news}"
wordcloud = WordCloud(
        width = 3000,
        height = 2000,
        background_color = 'black',
        stopwords = set(nltk.corpus.stopwords.words("english"))).generate(text)
fig = plt.figure(
        figsize = (40, 30),
        facecolor = 'k',
        edgecolor = 'k')
plt.imshow(wordcloud, interpolation = 'bilinear')
plt.axis('off')
plt.tight_layout(pad=0)
plt.show()
```

真新聞文字雲



假新聞文字雲



(4)資料標籤化、分詞、移除停用詞

將真新聞的資料都標成 1，假新聞的資料標成 0。並將標題與內文的文字合併成一個欄位。再將其他不需要的欄位移除。最後將文字做分詞並且移除所有的停用詞。

```
real["class"] = 1
fake["class"] = 0
```

```
real["text"] = real["title"] + " " + real["text"]
fake["text"] = fake["title"] + " " + fake["text"]
```

```
real = real.drop(["subject", "date","title", "publisher"], axis=1)
fake = fake.drop(["subject", "date", "title"], axis=1)
```

```
data = real.append(fake, ignore_index=True)
del real, fake
```

```
y = data["class"].values
X = []
stop_words = set(nltk.corpus.stopwords.words("english"))
tokenizer = nltk.tokenize.RegexpTokenizer(r'\w+')
for par in data["text"].values:
        tmp = []
        sentences = nltk.sent_tokenize(par)
        for sent in sentences:
                sent = sent.lower()
                tokens = tokenizer.tokenize(sent)
                filtered_words = [w.strip() for w in tokens if w not in stop_words and len(w) > 1]
                tmp.extend(filtered_words)
        X.append(tmp)
```

(5)向量化

One-hot encoding 是資訊密度低、維度高的向量，word embedding 改成資訊密度高、維度低的向量來代表一個詞。 Word2vec 是一群用來產生詞向量的相關模型，可以訓練出自己的 word vectors，也可以用 pre-trained word vectors 查看 similar words。

```
#Dimension of vectors we are generating
EMBEDDING_DIM = 100
#Creating Word Vectors by Word2Vec Method (takes time...)
w2v_model = gensim.models.Word2Vec(sentences=X, size=EMBEDDING_DIM, window=5, min_count=1)
```

```
len(w2v_model.wv.vocab)
```

```
122248
```

參數	
Size = 100	詞向量的維度大小，維度太小會無法有效表達詞與詞的關係。
Window = 5	CBOW 下決定 Word2Vec 一次取多少詞來預測中間詞。
Min_count = 1	出現次數大於等於 min_count 的詞，才會納入 Word2Vec 的詞典中。

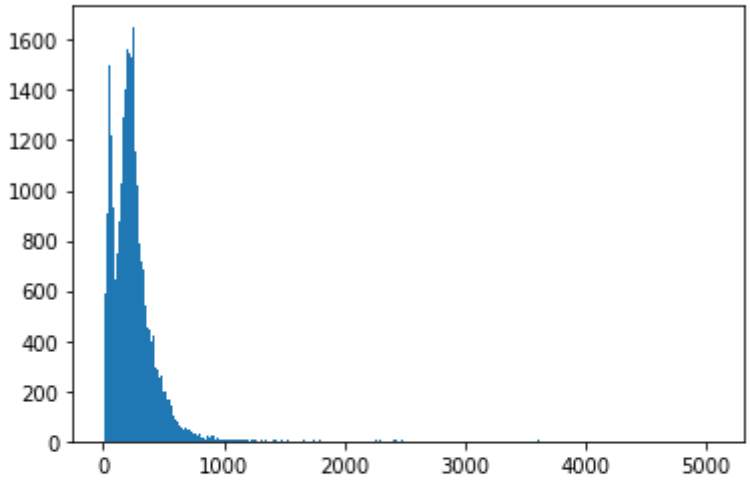
(6) 排序(sequence)與填充(padding)

利用 texts_to_sequences()函式，將每個句子用數字序列 表示。利用 pad_sequences()函式，使每個序列擁有相同的長度。將每個序列都刪減或補值成 700 個字。

```
tokenizer = Tokenizer()
tokenizer.fit_on_texts(X)

X = tokenizer.texts_to_sequences(X)
```

```
plt.hist([len(x) for x in X], bins=500)
plt.show()
```




```
nos = np.array([len(x) for x in X])
len(nos[nos < 700])
```

```
43982
```

```
maxlen = 700
#Making all news of size maxlen defined above
X = pad_sequences(X, maxlen=maxlen)
```

第三章、模型架構

3.1 LSTM 模型架構

此 LSTM 模型架構為一嵌入層（Embedding）、一輸出層（Dense）以及一層 keras 中的所定義的 LSTM 層。模型的可訓練參數總共有 117377 個。

```
#Defining Neural Network
model = Sequential()
#Non-trainable embeddidng layer
model.add(Embedding(vocab_size, output_dim=EMBEDDING_DIM, weights=[embedding_vectors], input_length=maxlen, trainable=False))
#LSTM
model.add(LSTM(units=128))
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['acc'])
```

```
model.summary()
```

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 700, 100)	12224900
lstm (LSTM)	(None, 128)	117248
dense (Dense)	(None, 1)	129

```
Total params: 12,342,277
Trainable params: 117,377
Non-trainable params: 12,224,900
```

第四章、模型分析

透過調整詞向量的維度，可以發現維度越大精度也不一定會比較準確。

```
model.fit(X_train, y_train, validation_split=0.3, epochs=6)
```

```
Epoch 1/6
737/737 [==============================] - 601s 812ms/step - loss: 0.1448 - acc: 0.9497 - val_loss: 0.0914 - val_acc: 0.9706
Epoch 2/6
737/737 [==============================] - 597s 810ms/step - loss: 0.0882 - acc: 0.9699 - val_loss: 0.0730 - val_acc: 0.9762
Epoch 3/6
737/737 [==============================] - 598s 811ms/step - loss: 0.0590 - acc: 0.9801 - val_loss: 0.0833 - val_acc: 0.9660
Epoch 4/6
737/737 [==============================] - 596s 808ms/step - loss: 0.0645 - acc: 0.9788 - val_loss: 0.0823 - val_acc: 0.9672
Epoch 5/6
737/737 [==============================] - 599s 813ms/step - loss: 0.0308 - acc: 0.9896 - val_loss: 0.0327 - val_acc: 0.9882
Epoch 6/6
737/737 [==============================] - 599s 812ms/step - loss: 0.0196 - acc: 0.9934 - val_loss: 0.0295 - val_acc: 0.9912
<keras.callbacks.History at 0x7fda34fcab10>
```

```
y_pred = (model.predict(X_test) >= 0.5).astype("int")
```

```
accuracy_score(y_test, y_pred)
```

```
0.9908240534521158
```

```
model.fit(X_train,  y_train,  validation_split=0.3,  epochs=6)
```

```
Epoch 1/6
737/737 [==============================] - 918s 1s/step - loss: 0.1268 - acc: 0.9554 - val_loss: 0.0847 - val_acc: 0.9722
Epoch 2/6
737/737 [==============================] - 898s 1s/step - loss: 0.0608 - acc: 0.9794 - val_loss: 0.0398 - val_acc: 0.9872
Epoch 3/6
737/737 [==============================] - 906s 1s/step - loss: 0.0283 - acc: 0.9908 - val_loss: 0.0663 - val_acc: 0.9756
Epoch 4/6
737/737 [==============================] - 886s 1s/step - loss: 0.0226 - acc: 0.9926 - val_loss: 0.0234 - val_acc: 0.9915
Epoch 5/6
737/737 [==============================] - 944s 1s/step - loss: 0.0131 - acc: 0.9961 - val_loss: 0.0214 - val_acc: 0.9932
Epoch 6/6
737/737 [==============================] - 892s 1s/step - loss: 0.0155 - acc: 0.9948 - val_loss: 0.0385 - val_acc: 0.9874
<keras.callbacks.History at 0x7ff37d128590>
```

```
y_pred  =  (model.predict(X_test)  >=  0.5).astype("int")
```

```
accuracy_score(y_test,  y_pred)
```

```
0.9878841870824053
```

參數	
Embedding	(122249,100,700)
LSTM	(128)
Dense	(1,sigmoid)
Loss_function	binary_crossentropy
optimizer	adam
epoch	6
Loss	0.0295
accuracy	0.99082

參數	
Embedding	(122249,200,700)
LSTM	(128)
Dense	(1,sigmoid)
Loss_function	binary_crossentropy
optimizer	adam
epoch	6
Loss	0.0295
accuracy	0.98788

第五章、結果與討論

5.1 結論

透過此模型的應用，我們可以有效的區分新聞是真是假，讓人民能夠不被誤導。Word embedding 維度越高效率精度會越低。

5.2 未來展望

蒐集國內中文新聞，並透過自然語言處理，也能夠訓練出模型。