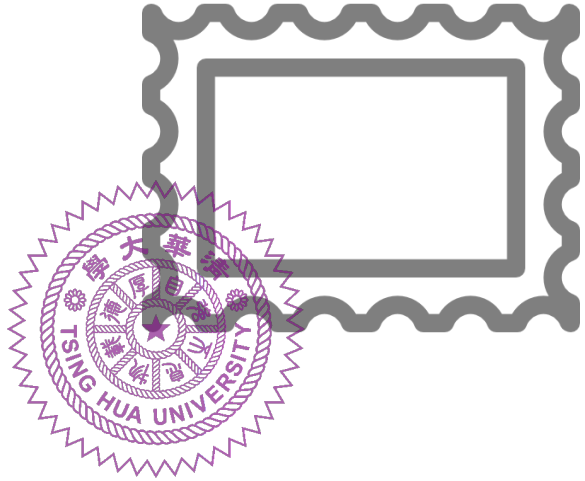
IIE FINAL PROJECT

垃圾郵件識別問題

109034542 林鈺婷

OUTLINE



01

背景介紹

研究動機&目的、 5W1H

01-背景介紹— 研究動機 & 目的

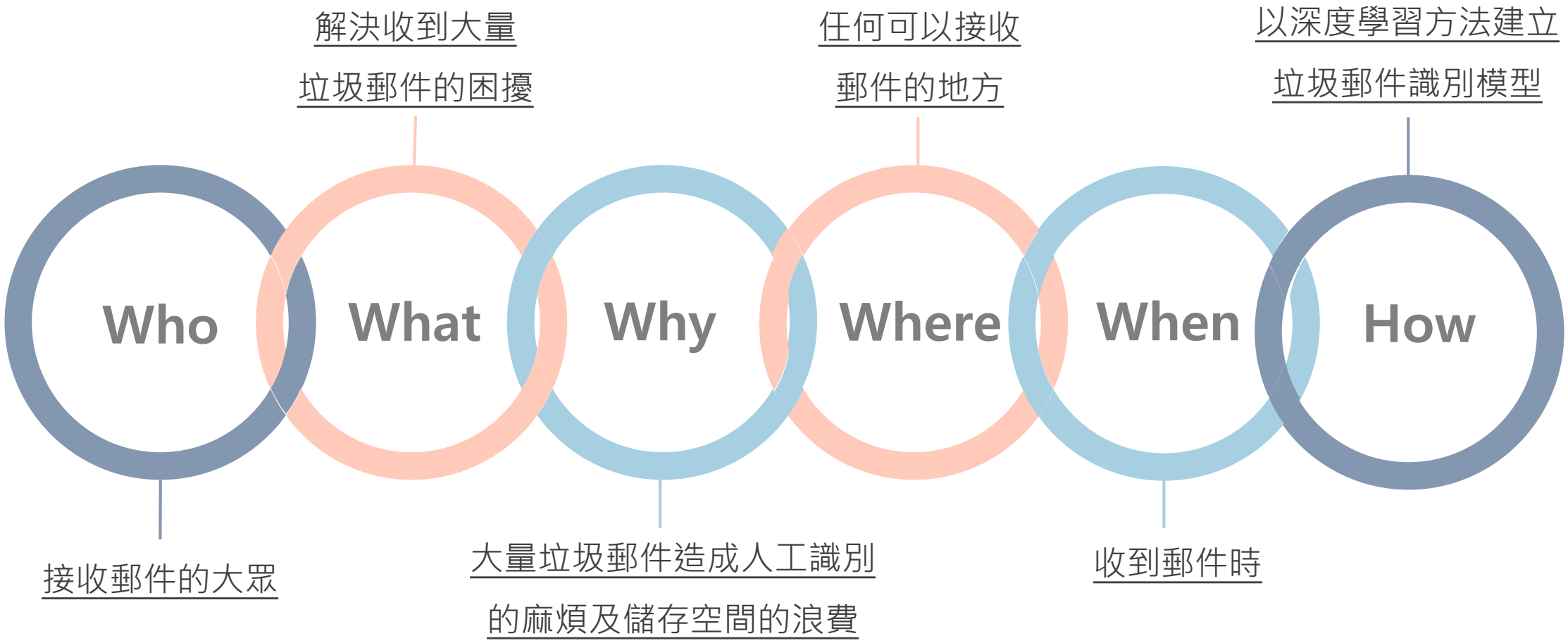
■ 研究動機

- ✓ 資訊快速發展，網路上各種資訊隨手可得，不僅帶來便利，也帶來煩惱
- ✓ 反垃圾郵件措施無法抵擋新的垃圾郵件模式
- ✓ 單靠人工識別的方式難以跟上垃圾郵件的成長速度

■ 目的

- ✓ 以人工智慧方法來區別垃圾郵件，減少人工識別所浪費的時間
- ✓ 期望透過人工智慧發現新的垃圾郵件模式

01-背景介紹— 5W1H


Who
What
Why
Where
When
How
解決收到大量
垃圾郵件的困擾
任何可以接收
郵件的地方
以深度學習方法建立
垃圾郵件識別模型
接收郵件的大眾
大量垃圾郵件造成人工識別
的麻煩及儲存空間的浪費
收到郵件時

02

資料前處理

資料輸入 & 空值檢查、字詞分析、Downsampling、
資料標籤化、Tokenization、Sequencing & Padding

02-資料前處理— 資料輸入 & 空值檢查

- 資料來源：UCI的郵件資料集
- 資料集大小：共計5572筆郵件資料
  - ✓ 有效郵件(ham)為4825筆
  - ✓ 垃圾郵件(spam)為747筆
- 欄位：label 與 message

```
1 #  Check  data  information
2 message.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 2 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   label    5572 non-null   object
 1   message  5572 non-null   object
dtypes: object(2)
memory usage: 87.2+ KB
```

	label	message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...
...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u...
5568	ham	Will ü b going to esplanade fr home?
5569	ham	Pity, * was in mood for that. So...any other s...
5570	ham	The guy did some bitching but I acted like i'd...
5571	ham	Rofl. Its true to its name

5572 rows × 2 columns

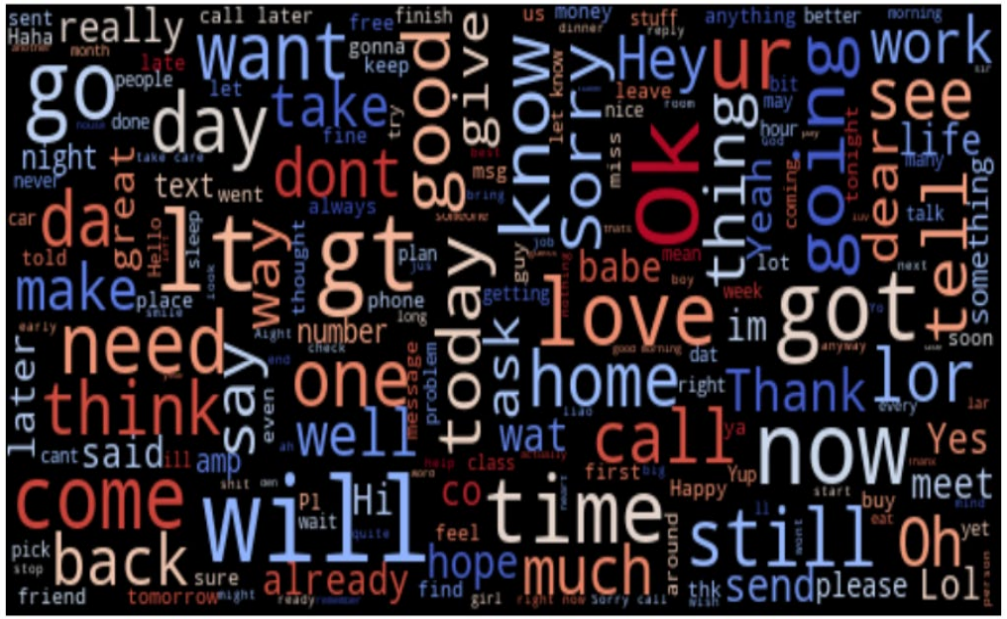
```
1 #  匯入資料集
2 url  =  'https://raw.githubusercontent.com/ShresthaSudip/SMS_Spam_Detection_DNN_LSTM_BiLSTM/master/SMSSpamCollection'
3 message  =  pd.read_csv(url,  sep  ='\t',names=["label",  "message"])
4 message
```

02-資料前處理— 字詞分析視覺化

- 利用 WorldCloud() 函式進行郵件字詞分析

✓ 垃圾郵件

✓ 有效郵件



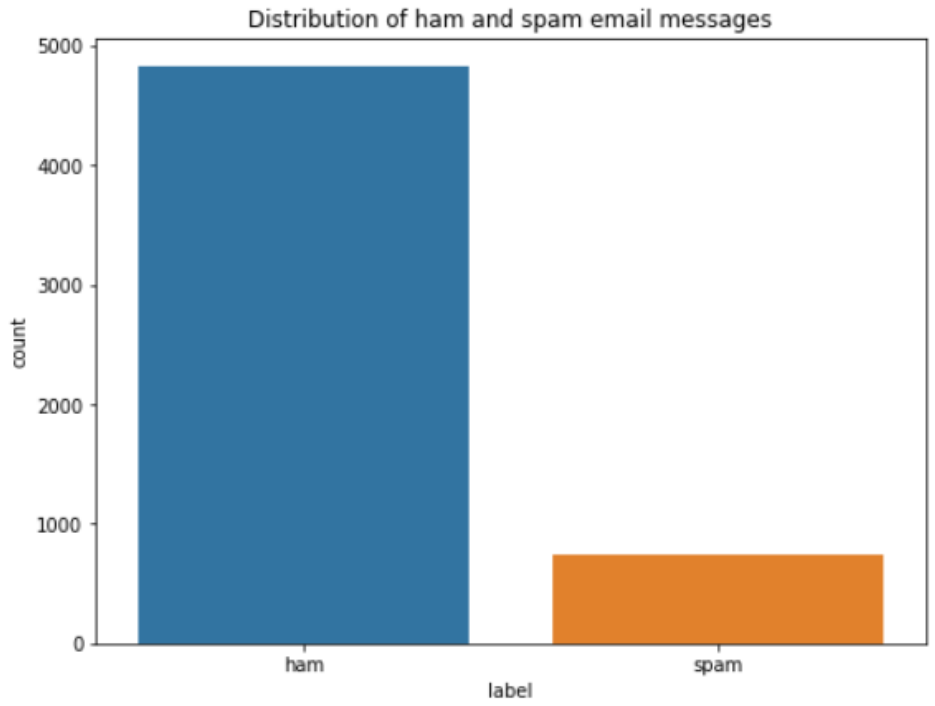

```
1 # wordcloud of spam messages
2 spam_msg_cloud = WordCloud(width =520, height =260,
3                            stopwords=STOPWORDS,max_font_size=50, background_color ="black", colormap='coolwarm').generate(spam_msg_text)
4 plt.figure(figsize=(16,10))
5 plt.imshow(spam_msg_cloud, interpolation='bilinear')
6 plt.axis('off') # turn off axis
7 plt.show()
```

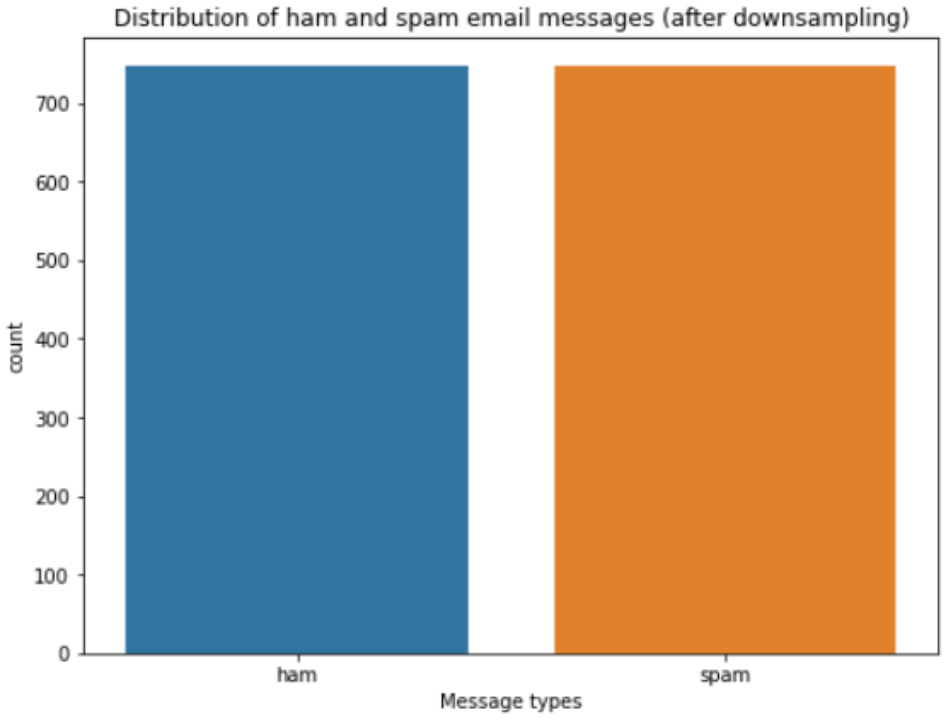
02-資料前處理— Downsampling

- 處理不平衡資料
  - ✓ 有效郵件為4825筆
  - ✓ 垃圾郵件為747筆

```
1 #Downsample the ham msg
2 ham_msg_df = ham_msg.sample(n = len(spam_msg), random_state = 44)
3 spam_msg_df = spam_msg
4 print(ham_msg_df.shape, spam_msg_df.shape)
```

```
(747, 2) (747, 2)
```





02-資料前處理— 資料標籤化 & 資料切分

■ 利用 map() 函式針對 label 欄位進行標籤化

✓ 有效郵件標記為0

✓ 垃圾郵件標記為1

```
#資料標籤化 (ham=0 & spam=1)
msg_df['msg_type']= msg_df['label'].map({'ham': 0, 'spam': 1})
msg_label = msg_df['msg_type'].values
msg_df
```

	label	message	msg_type
0	ham	Height of recycling: Read twice- People spend ...	0
1	ham	Yup song bro. No creative. Neva test quality. ...	0
2	ham	Feb <#> is "I LOVE U" day. Send dis to...	0
3	ham	Don't forget who owns you and who's private pr...	0
4	ham	Lol no. I just need to cash in my nitros. Hurr...	0
...	...	...	...
1489	spam	Want explicit SEX in 30 secs? Ring 02073162414...	1
1490	spam	ASKED 3MOBILE IF 0870 CHATLINES INCLU IN FREE ...	1
1491	spam	Had your contract mobile 11 Mnths? Latest Moto...	1
1492	spam	REMINDER FROM O2: To get 2.50 pounds free call...	1
1493	spam	This is the 2nd time we have tried 2 contact u...	1

1494 rows × 3 columns

■ 利用 train_test_split() 函式進行資料切分

✓ 訓練集 (Training data) 為80%

✓ 測試集 (Testing data) 為20%

```
# 將資料切分為 training and testing
train_msg, test_msg, train_labels, test_labels = train_test_split(msg_df['message'], msg_label, test_size=0.2, random_state=434)
```

02-資料前處理— 分詞(Tokenization)

■ 利用Tokenizer()函式，將訓練集資料內容切成單詞，並將每個單詞編號

```
# 定義 pre-processing-Tokenizer 的超參數
max_len = 50
trunc_type = "post"
padding_type = "post"
oov_tok = "<OOV>"
vocab_size = 500

tokenizer = Tokenizer(num_words = vocab_size, char_level=False, oov_token = oov_tok)
tokenizer.fit_on_texts(train_msg)

# Get the word_index
word_index = tokenizer.word_index
print(word_index)

# check how many words
tot_words = len(word_index)
print('There are %s unique tokens in training data. ' % tot_words)
```

```
{'<OOV>': 1, 'to': 2, 'you': 3, 'a': 4, 'i': 5, 'call': 6, 'the': 7, 'u': 8, 'your': 9, 'for': 10, 'is': 11, '2': 12, 'and': 13, 'now': 14, 'free': 15,
There are 4169 unique tokens in training data.
```

02-資料前處理— 排序(Sequencing) & 填充(Padding)

- 利用 texts_to_sequences() 函式，將訓練集與測試集的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

```
# Sequencing and padding on training and testing
training_sequences = tokenizer.texts_to_sequences(train_msg)
training_padded = pad_sequences (training_sequences, maxlen = max_len, padding = padding_type, truncating = trunc_type )
testing_sequences = tokenizer.texts_to_sequences(test_msg)
testing_padded = pad_sequences(testing_sequences, maxlen = max_len,
padding = padding_type, truncating = trunc_type)

# Before padding
print("Before padding:",(len(training_sequences[0]), len(training_sequences[1])))
# After padding
print("After padding:",(len(training_padded[0]), len(training_padded[1])))
print("training_padded[0]:",training_padded[0])
```

```
Before padding: (27, 24)
After padding: (50, 50)
training_padded[0]: [  1  47 186   9  34   1   3  24   1   2 274   2   7 152 275 135  34  10
  15   6   7  34 274  85  15  17   1   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
```

03

模型建構

Dense、LSTM 、Bi-LSTM Model

03-模型建構— Dense Model

- 利用EarlyStopping()函式，透過監控測試集的損失來判斷是否終止訓練，以避免過擬合情況發生。

```
# Dense model hyperparameters
max_len = 50
vocab_size = 500 # As defined earlier
embeding_dim = 16

# Dense model
model = Sequential()
model.add(Embedding(vocab_size, embeding_dim ,input_length=50))
model.add(GlobalAveragePooling1D())
model.add(Dense(24, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='mean_absolute_error',optimizer='adam' ,metrics=['accuracy'])

# fitting a dense spam detector model
early_stop = EarlyStopping(monitor='val_loss', patience=3)
history = model.fit(training_padded, train_labels, epochs=30,
                    validation_data=(testing_padded, test_labels),callbacks =[early_stop], verbose=2)
```

03-模型建構— Dense Model

■ 超參數調整過程

Embedding	(500,16,50)
dense	24,relu
dropout	0.2
dense	1,sigmoid
loss_function	binary_crossentropy
optimizer	adam
epochs	30
loss	0.1283
accuracy	0.9431

Embedding	(500,32,50)
dense	24,relu
dropout	0.2
dense	1,sigmoid
loss_function	binary_crossentropy
optimizer	adam
epochs	30
loss	0.1288
accuracy	0.9331

Embedding	(500,16,50)
dense	64,relu
dropout	0.2
dense	1,sigmoid
loss_function	binary_crossentropy
optimizer	adam
epochs	30
loss	0.1246
accuracy	0.9465

Embedding	(500,16,50)
dense	24,sigmoid
dropout	0.2
dense	1,sigmoid
loss_function	binary_crossentropy
optimizer	adam
epochs	30
loss	0.1420
accuracy	0.9431

Embedding	(500,16,50)
dense	24,relu
dropout	0.3
dense	1,sigmoid
loss_function	binary_crossentropy
optimizer	adam
epochs	30
loss	0.0631
accuracy	0.9465

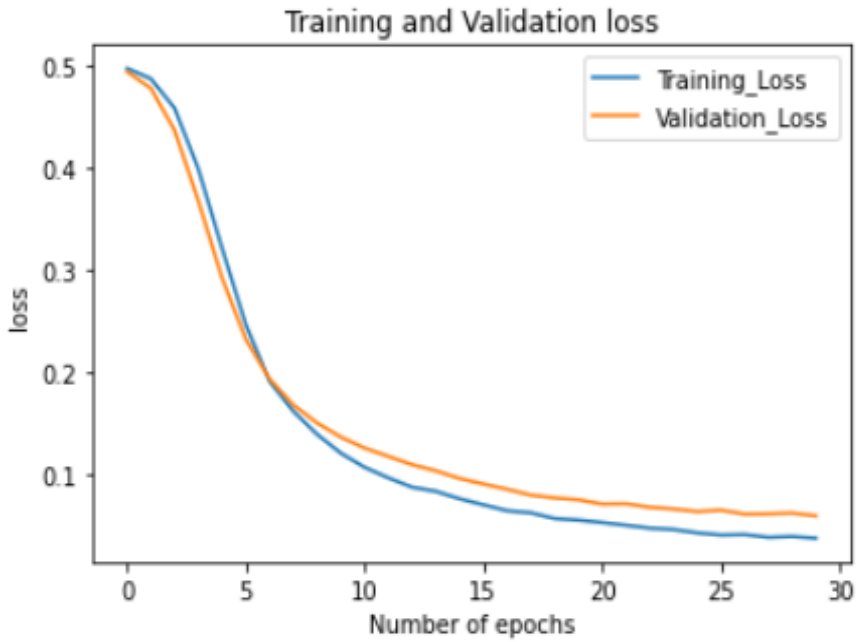
Embedding	(500,16,50)
dense	24,relu
dropout	0.2
dense	1,sigmoid
loss_function	mean_absolute_error
optimizer	adam
epochs	30
loss	0.0592
accuracy	0.9532

Embedding	(500,16,50)
dense	24,relu
dropout	0.2
dense	1,sigmoid
loss_function	binary_crossentropy
optimizer	sgd
epochs	30
loss	0.6512
accuracy	0.7659

Embedding	(500,16,50)
dense	24,relu
dropout	0.2
dense	1,softmax
loss_function	binary_crossentropy
optimizer	adam
epochs	30
loss	0.1126
accuracy	0.4716

03-模型建構— Dense Model

- 模型訓練集與測試集之損失&準確率



03-模型建構— LSTM Model

```
# #LSTM hyperparameters
vocab_size = 500 # As defined earlier
embeding_dim = 36

#LSTM model
model1 = Sequential()
model1.add(Embedding(vocab_size, embeding_dim, input_length=max_len))
model1.add(LSTM(32, dropout=0.2, return_sequences=True))
model1.add(LSTM(32, dropout=0.2, return_sequences=True))
model1.add(Dense(1, activation='sigmoid'))
model1.compile(loss = 'binary_crossentropy', optimizer = 'adam', metrics=['accuracy'])

# fitting a LSTM model
early_stop = EarlyStopping(monitor='val_loss', patience=2)
history = model1.fit(training_padded, train_labels, epochs=30,
                     validation_data=(testing_padded, test_labels),callbacks =[early_stop], verbose=2)
```

03-模型建構— LSTM Model

■ 超參數調整過程

Embedding	(500,16,50)
LSTM(n,dropout)	(20,0.2)
LSTM(n,dropout)	(20,0.2)
dense	1,sigmoid
loss_function	binary_crossentropy
optimizer	adam
epochs	30
loss	0.2359
accuracy	0.9243

Embedding	(500,32,50)
LSTM(n,dropout)	(20,0.2)
LSTM(n,dropout)	(20,0.2)
dense	1,sigmoid
loss_function	binary_crossentropy
optimizer	adam
epochs	30
loss	0.2259
accuracy	0.9321

Embedding	(500,36,50)
LSTM(n,dropout)	(20,0.2)
LSTM(n,dropout)	(20,0.2)
dense	1,sigmoid
loss_function	binary_crossentropy
optimizer	adam
epochs	30
loss	0.216
accuracy	0.9328

Embedding	(500,36,50)
LSTM(n,dropout)	(24,0.2)
LSTM(n,dropout)	(24,0.2)
dense	1,sigmoid
loss_function	binary_crossentropy
optimizer	adam
epochs	30
loss	0.2449
accuracy	0.9252

Embedding	(500,36,50)
LSTM(n,dropout)	(32,0.2)
LSTM(n,dropout)	(32,0.2)
dense	1,sigmoid
loss_function	binary_crossentropy
optimizer	adam
epochs	30
loss	0.1987
accuracy	0.9357

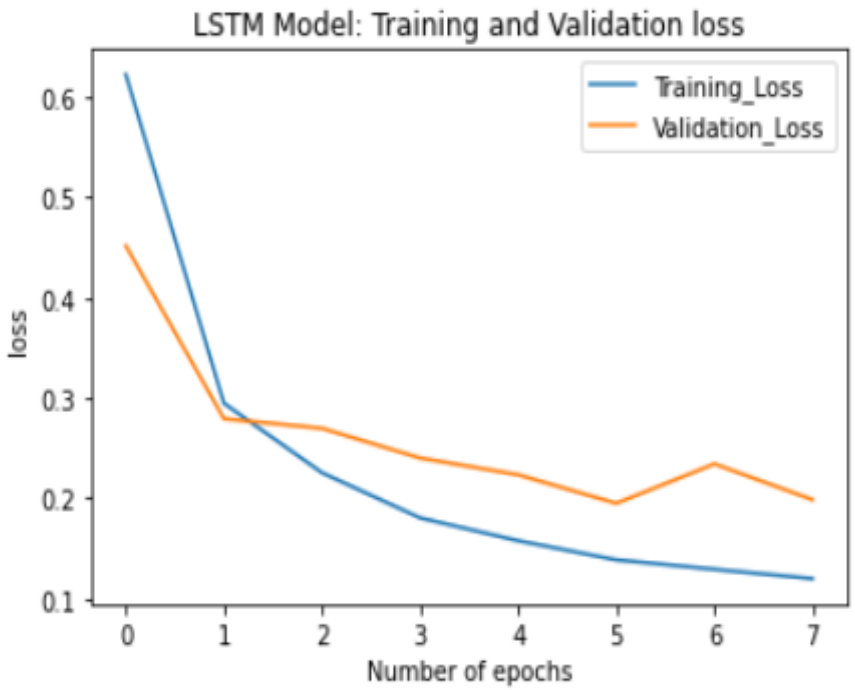
Embedding	(500,36,50)
LSTM(n,dropout)	(32,0.25)
LSTM(n,dropout)	(32,0.25)
dense	1,sigmoid
loss_function	binary_crossentropy
optimizer	adam
epochs	30
loss	0.2192
accuracy	0.9312

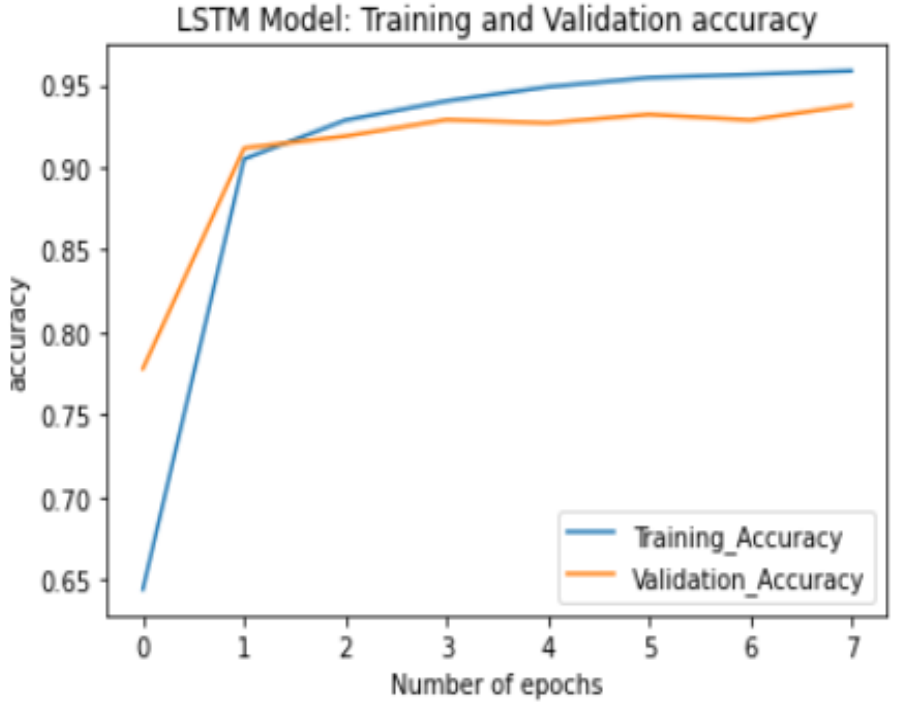
Embedding	(500,36,50)
LSTM(n,dropout)	(32,0.2)
LSTM(n,dropout)	(32,0.2)
dense	1,sigmoid
loss_function	mean_squared_error
optimizer	adam
epochs	30
loss	0.063
accuracy	0.9256

Embedding	(500,36,50)
LSTM(n,dropout)	(32,0.2)
LSTM(n,dropout)	(32,0.2)
dense	1,tanh
loss_function	binary_crossentropy
optimizer	adam
epochs	30
loss	0.221
accuracy	0.9097

03-模型建構— LSTM Model

■ 模型訓練集與測試集之損失&準確率





03-模型建構— Bidirectional-LSTM Model

```
# Bidirectional LSTM hyperparameters
vocab_size = 500 # As defined earlier
embeding_dim = 32

# Bidirectional LSTM Spam detection architecture
model2 = Sequential()
model2.add(Embedding(vocab_size, embeding_dim, input_length=max_len))
model2.add(Bidirectional(LSTM(36, dropout=0.2, return_sequences=True)))
model2.add(Dense(1, activation='sigmoid'))
model2.compile(loss = 'mean_absolute_error', optimizer = 'adam', metrics=['accuracy'])

# Training
early_stop = EarlyStopping(monitor='val_loss', patience=2)
history = model2.fit(training_padded, train_labels, epochs=30,
                     validation_data=(testing_padded, test_labels),callbacks =[early_stop], verbose=2)
```

03-模型建構— Bidirectional-LSTM Model

■ 超參數調整過程

Embedding	(500,16,50)
Bi-LSTM (n,dropout)	(20,0.2)
dense	(1,sigmoid)
loss_function	binary_crossentropy
optimizer	adam
epochs	30
loss	0.1871
accuracy	0.9443

Embedding	(500,32,50)
Bi-LSTM (n,dropout)	(20,0.2)
dense	(1,sigmoid)
loss_function	binary_crossentropy
optimizer	adam
epochs	30
loss	0.1633
accuracy	0.9564

Embedding	(500,32,50)
Bi-LSTM (n,dropout)	(24,0.2)
dense	(1,sigmoid)
loss_function	binary_crossentropy
optimizer	adam
epochs	30
loss	0.1791
accuracy	0.9507

Embedding	(500,32,50)
Bi-LSTM (n,dropout)	(36,0.2)
dense	(1,sigmoid)
loss_function	binary_crossentropy
optimizer	adam
epochs	30
loss	0.1319
accuracy	0.9636

Embedding	(500,32,50)
Bi-LSTM (n,dropout)	(36,0.25)
dense	(1,sigmoid)
loss_function	binary_crossentropy
optimizer	adam
epochs	30
loss	0.1958
accuracy	0.9492

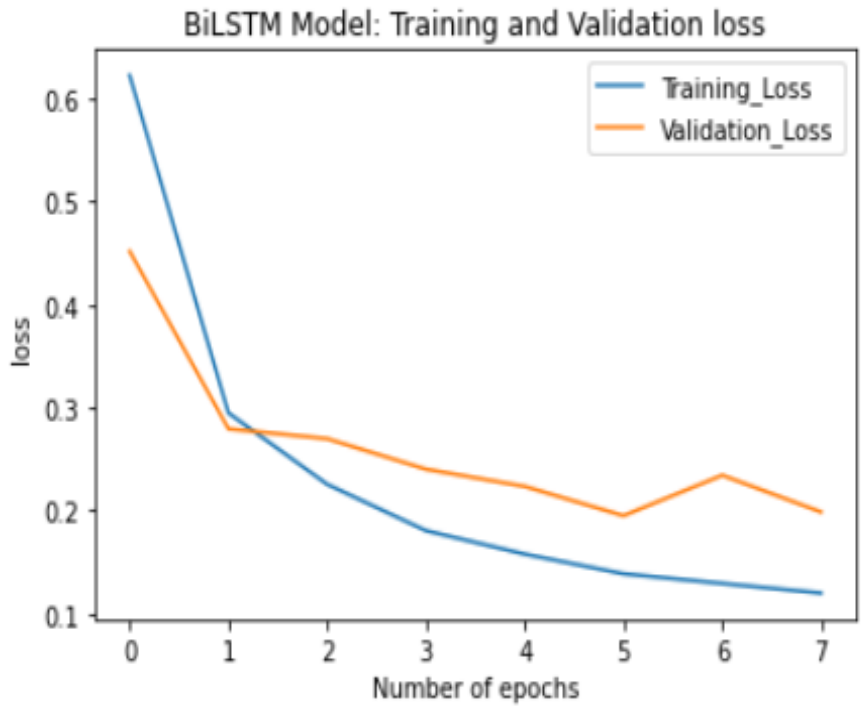
Embedding	(500,32,50)
Bi-LSTM (n,dropout)	(36,0.2)
dense	(1,sigmoid)
loss_function	mean_absolute_error
optimizer	adam
epochs	30
loss	0.0533
accuracy	0.9569

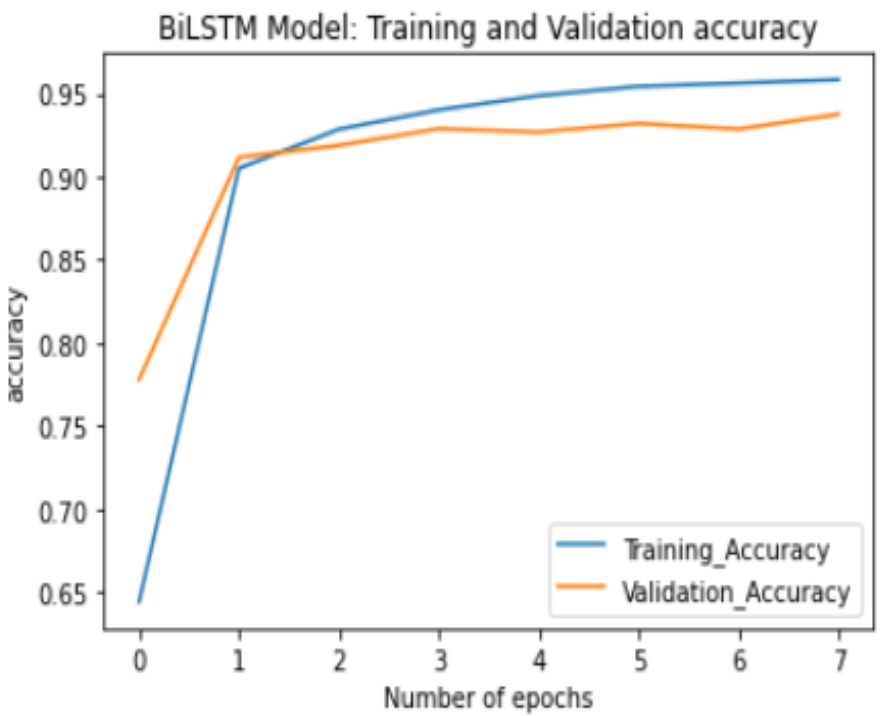
Embedding	(500,32,50)
Bi-LSTM (n,dropout)	(36,0.2)
dense	(1,tanh)
loss_function	binary_crossentropy
optimizer	adam
epochs	30
loss	0.1492
accuracy	0.9558

Embedding	(500,32,50)
Bi-LSTM (n,dropout)	(36,0.2)
dense	(1,relu)
loss_function	binary_crossentropy
optimizer	adam
epochs	30
loss	0.1871
accuracy	0.9404

03-模型建構— Bidirectional-LSTM Model

■ 模型訓練集與測試集之損失&準確率





03-模型建構— 模型準確率比較

	Dense	LSTM	Bi-LSTM
Training Process	✓ Embedding ✓ Dense ✓ Dropout ✓ Loss_function ✓ Optimizer	✓ Embedding ✓ LSTM ✓ Dense ✓ Loss_function ✓ Optimizer	✓ Embedding ✓ Bi-LSTM ✓ Dense ✓ Loss_function ✓ Optimizer
資料前處理	0.9431	0.9243	0.9443 ⓥ
最佳化模型	0.9532	0.9357	0.9636 ⓥ
Improvement	+1.01%	+1.14%	+1.93%

04

結論&發展

總結、未來展望

04-結論— 總結 & 未來展望

■ 比較 Dense、LSTM、Bi-LSTM 三種類神經網路模型，Bi-LSTM模型表現最佳

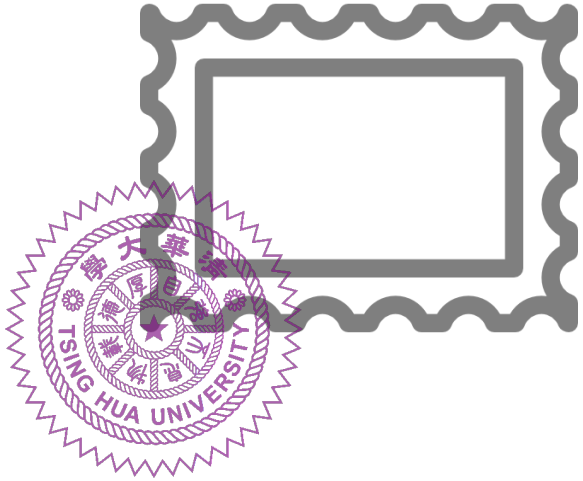
✓ 透過模型應用，可有效區分垃圾郵件與有效郵件，減少人工識別所耗費的時間

未來改善

- ✓ 增加資料量
- ✓ 超參數調整
- ✓ 嘗試不同模型

未來應用

- ✓ 文本分類
- ✓ 情感分析
- ✓ 文本生成
- ✓ 語言翻譯



Thanks for Listening