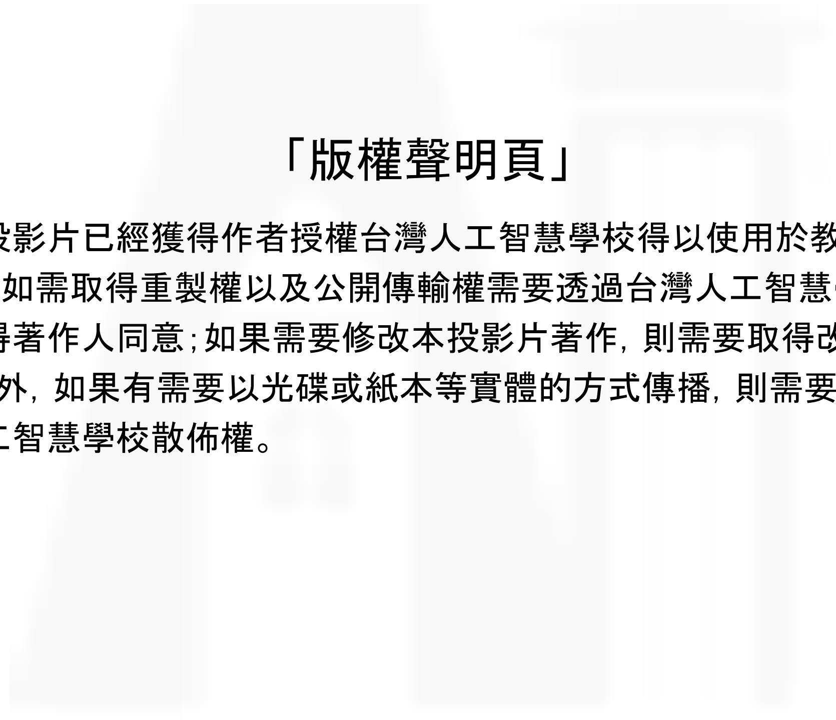
自然語言處理與文字探勘

陳縕儂 & 教研處



「版權聲明頁」

課程內容

講師投影片
資料與投影片
影片播放列表
程式碼：~/courses-tpe/NLP

1. 文本特徵建模練習
2. NLP with RNN
3. BERT fine-tune

Code / Data 放在 hub 中的 courses 內

- 為維護課程資料，courses 中的檔案皆為 read-only, 如需修改請 cp 至自身的環境中
- 打開 terminal, 輸入
  - [台北班]

    ```
    cp -r courses-tpe/NLP/part4 <存放至本機的名稱>
    ```
  - [新竹班]

    ```
    cp -r courses-hsi/NLP/part4 <存放至本機的名稱>
    ```
  - [台中班]

    ```
    cp -r courses-txg/NLP/part4 <存放至本機的名稱>
    ```

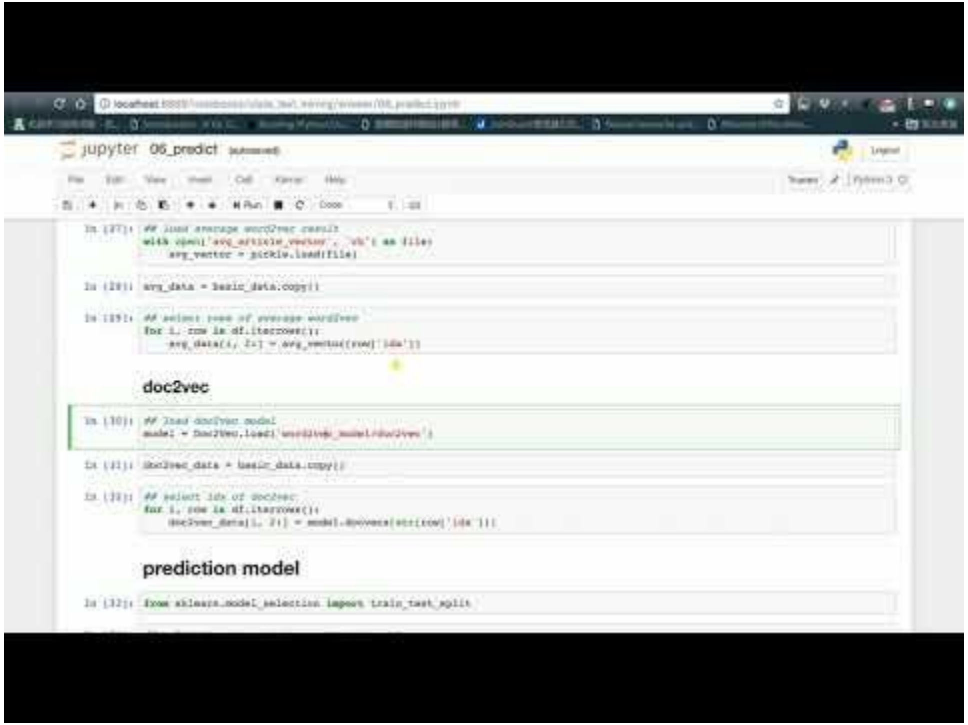
程式實作

文本特徵建模練習

程式練習時間

- 06_predict.ipynb
  - 定義 y: 發文的推文多 or 噓文多？
    - 絕對差異定義？> 20
  - 利用 bag of words, TF-IDF, average word2vec, doc2vec 各做一組 prediction model, 比較哪一組 features 最好

程式解說


jupyter 06_predict (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3
Run
Code

```
In [27]: ## load average word2vec result
         with open('avg_article_vector', 'rb') as file:
             avg_vector = pickle.load(file)

In [28]: avg_data = basic_data.copy()

In [29]: ## select rows of average word2vec
         for i, row in df.iterrows():
             avg_data[i, 2:] = avg_vector[row['idx']]
```

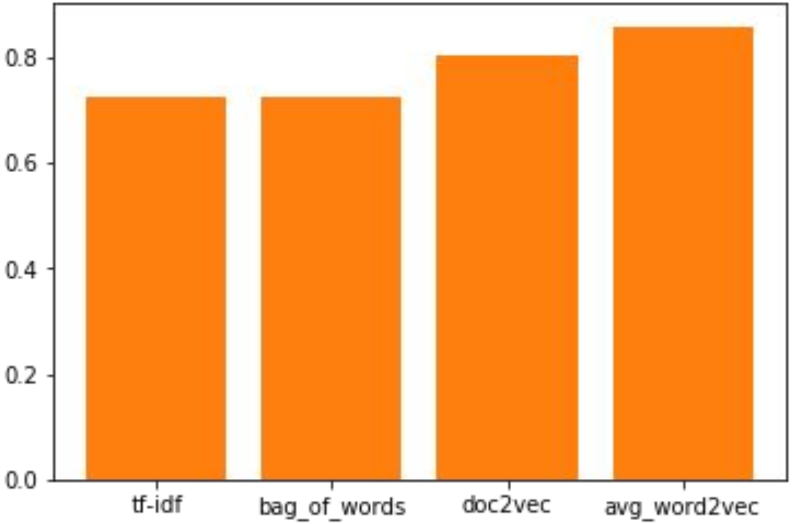
doc2vec

```
In [30]: ## load doc2vec model
         model = Doc2Vec.load('word2vec_model/doc2vec')

In [31]: doc2vec_data = basic_data.copy()

In [32]: ## select idx of doc2vec
         for i, row in df.iterrows():
             doc2vec_data[i, 2:] = model.docvecs[str(row['idx'])]
```

prediction model

```
In [33]: from sklearn.model_selection import train_test_split
```

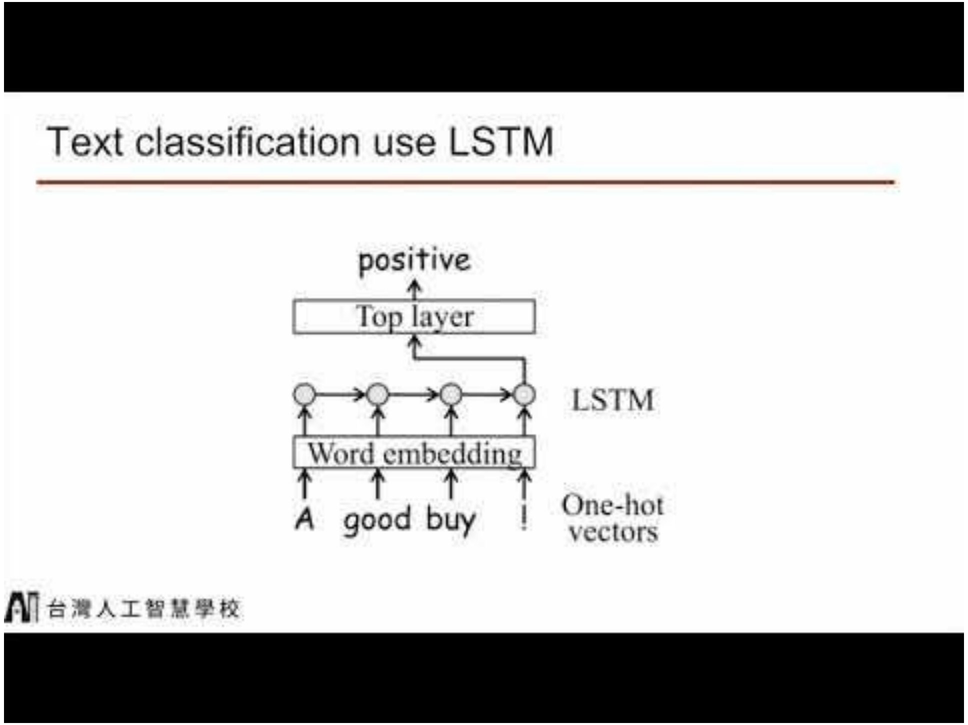
Each model's AUC



程式實作

NLP with RNN

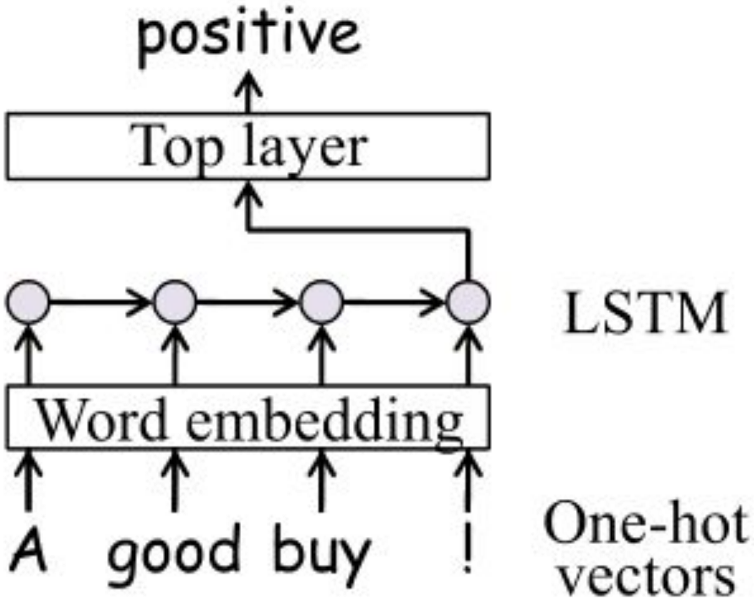
建模流程



Text classification use LSTM

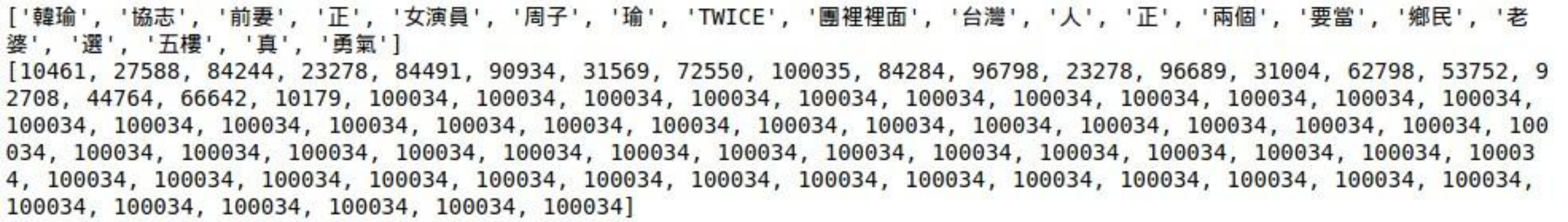
台灣人工智慧學校

Text classification use LSTM



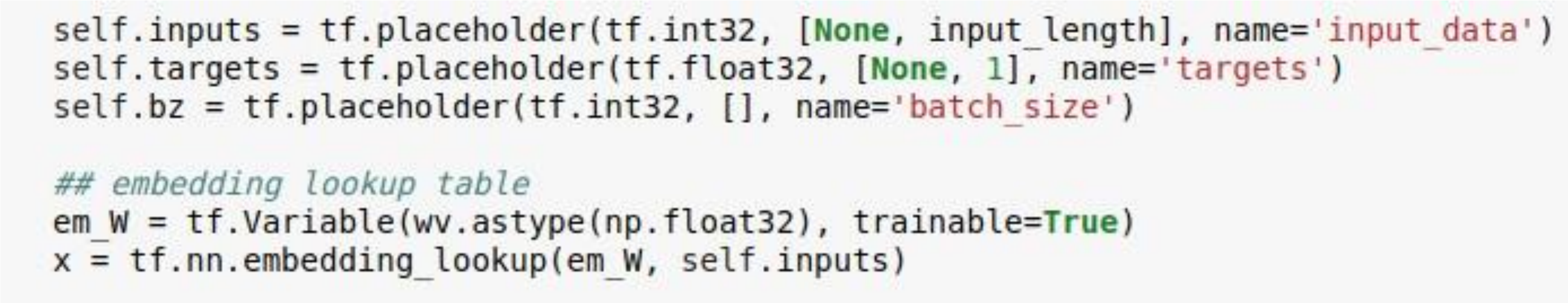
how to feed text data into LSTM network ?

- 先把 text data 轉換成 id 格式
  - word to id
- 不在 word vector 的字，用別的 id 代表
  - e.g. len(word)+1



```
['韓瑜', '協志', '前妻', '正', '女演員', '周子', '瑜', 'TWICE', '團裡裡面', '台灣', '人', '正', '兩個', '要當', '鄉民', '老
婆', '選', '五樓', '真', '勇氣']
[10461, 27588, 84244, 23278, 84491, 90934, 31569, 72550, 100035, 84284, 96798, 23278, 96689, 31004, 62798, 53752, 9
2708, 44764, 66642, 10179, 100034, 100034, 100034, 100034, 100034, 100034, 100034, 100034, 100034, 100034, 100034,
100034, 100034, 100034, 100034, 100034, 100034, 100034, 100034, 100034, 100034, 100034, 100034, 100034, 100034, 100
034, 100034, 100034, 100034, 100034, 100034, 100034, 100034, 100034, 100034, 100034, 100034, 100034, 100034, 10003
4, 100034, 100034, 100034, 100034, 100034, 100034, 100034, 100034, 100034, 100034, 100034, 100034, 100034, 100034,
100034, 100034, 100034, 100034, 100034, 100034]
```

embedding lookup table

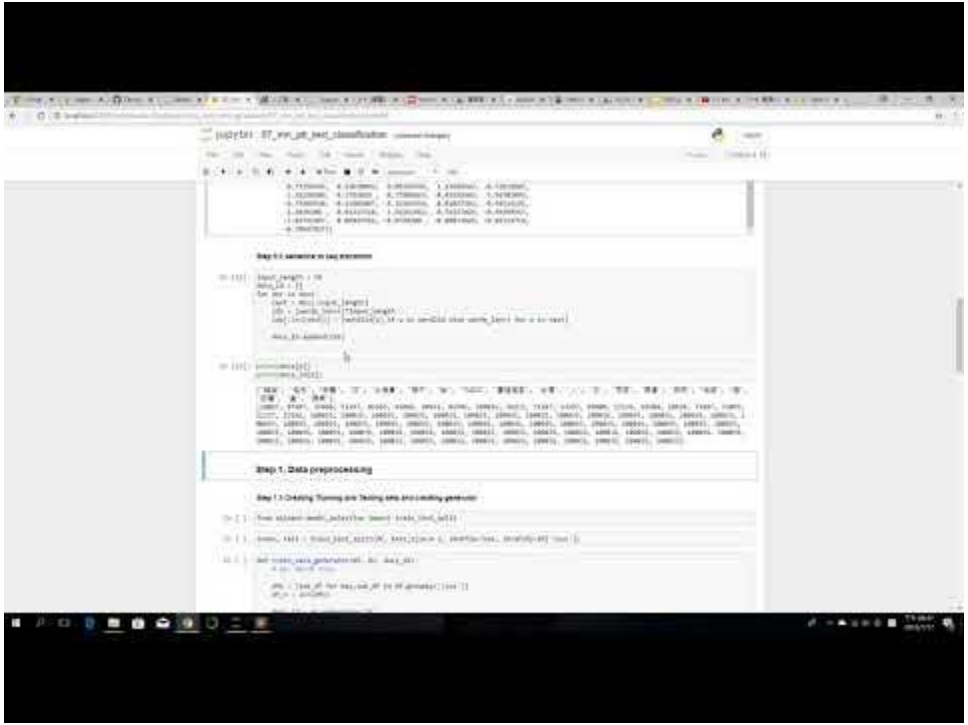


```
self.inputs = tf.placeholder(tf.int32, [None, input_length], name='input_data')
self.targets = tf.placeholder(tf.float32, [None, 1], name='targets')
self.bz = tf.placeholder(tf.int32, [], name='batch_size')

## embedding lookup table
em_W = tf.Variable(wv.astype(np.float32), trainable=True)
x = tf.nn.embedding_lookup(em_W, self.inputs)
```

程式練習時間

- 07_rnn_ptt_text_classification.ipynb
    - 練習如何把 text data 轉換成 id 格式
    - 練習建構 embedding lookup table 接 LSTM network
    - 可嘗試
        - fine tune word vector (trainable = True)
        - 不 fine tune word vector (trainable = False)
        - randon initialize embedding lookup table

程式解說

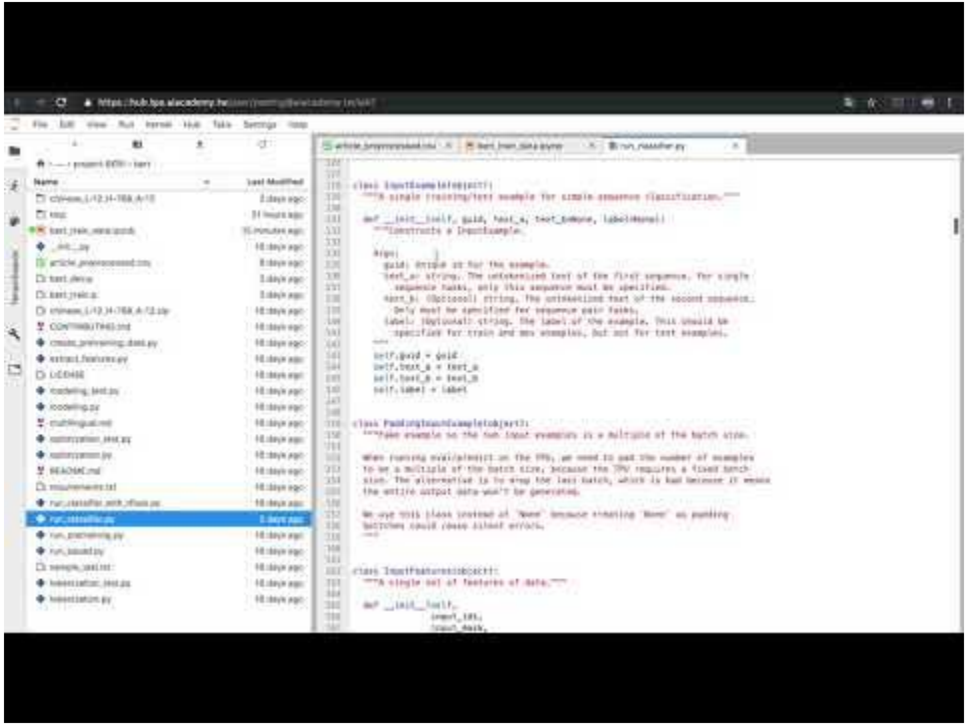


Step 1. Data preprocessing

程式實作

BERT fine-tune

BERT fine-tune



Name	Last Modified
chinese_L-12_H-768_A-12	2 days ago
tmp	21 hours ago
bert_train_data.ipynb	15 minutes ago
`__init__.py`	18 days ago
article_preprocessed.csv	8 days ago
bert_dev.p	3 days ago
bert_train.p	3 days ago
chinese_L-12_H-768_A-12.zip	18 days ago
CONTRIBUTING.md	18 days ago
create_pretraining_data.py	18 days ago
extract_features.py	18 days ago
LICENSE	18 days ago
modeling_test.py	18 days ago
modeling.py	18 days ago
multilingual.md	18 days ago
optimization_test.py	18 days ago
optimization.py	18 days ago
README.md	18 days ago
requirements.txt	18 days ago
run_classifier_with_tfhub.py	18 days ago
run_classifier.py	3 days ago
run_pretraining.py	18 days ago
run_squad.py	18 days ago
sample_text.txt	18 days ago
tokenization_test.py	18 days ago
tokenization.py	18 days ago

```python
class InputExample(object):
  """A single training/test example for simple sequence classification."""

  def __init__(self, guid, text_a, text_b=None, label=None):
    """Constructs a InputExample.

    Args:
      guid: Unique id for the example.
      text_a: string. The untokenized text of the first sequence. For single
        sequence tasks, only this sequence must be specified.
      text_b: (Optional) string. The untokenized text of the second sequence.
        Only must be specified for sequence pair tasks.
      label: (Optional) string. The label of the example. This should be
        specified for train and dev examples, but not for test examples.
    """
    self.guid = guid
    self.text_a = text_a
    self.text_b = text_b
    self.label = label


class PaddingInputExample(object):
  """Fake example so the num input examples is a multiple of the batch size.

  When running eval/predict on the TPU, we need to pad the number of examples
  to be a multiple of the batch size, because the TPU requires a fixed batch
  size. The alternative is to drop the last batch, which is bad because it means
  the entire output data won't be generated.

  We use this class instead of `None` because treating `None` as padding
  battches could cause silent errors.
  """


class InputFeatures(object):
  """A single set of features of data."""

  def __init__(self,
               input_ids,
               input_mask,
```

歡迎挑戰 kaggle text classification task

- Toxic Comment Classification Challenge
- Mercari Price Suggestion Challenge
- Spooky Author Identification
- Personalized Medicine: Redefining Cancer Treatment
- Quora Question Pairs
- ...

還不夠嗎？可以看看別的主題

- N-Gram
  - Modelling Natural Language with N-Gram Models
  - 自然語言處理中的 N-Gram 模型介紹
- Topic model
  - Begineer guide to topic modelling in python
  - Topic modelling with scikit-learn
  - Latent Dirichlet Allocation
  - LDA 數學八卦
  - 手刻版 topic model