

# 以FOMC發言預測利率走勢

Crawler:邱子軒

Database:鄭元甦

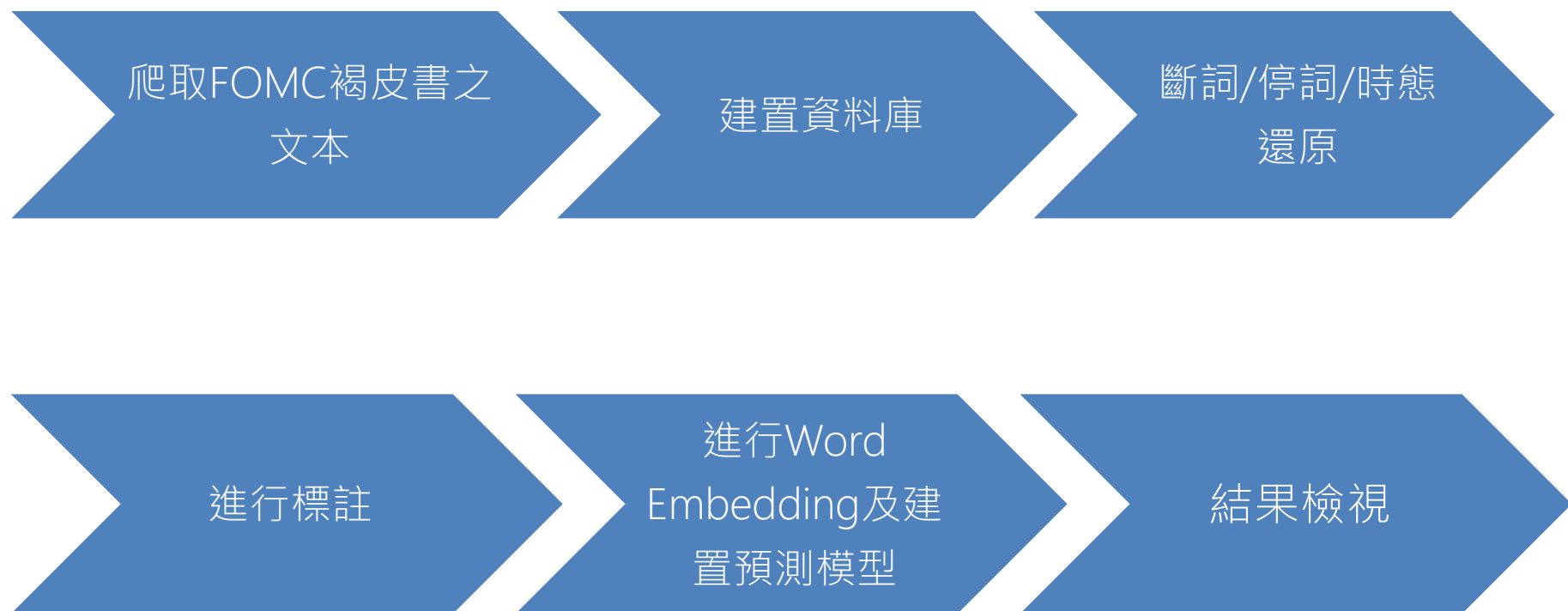
Word\_Embedding:余柏叡

Prediction\_Model:黃煒翔

# 專案說明

- 目標：  
以NLP之方式建置利率決議預測模型，運用FOMC褐皮書對FED利率決議進行預測
- 什麼是FED？  
A: FED即聯準會，是美國的中央銀行，可以決定國家貨幣政策，貨幣政策的決定(升息、降息、維持)攸關整體金融市場之環境。FED為國際金融市場最具影響力之央行，眾多金融資產以FED之利率作為訂價依據，故許多金融機構相當重視預測FED利率決議。
- 什麼是FOMC？  
A: FOMC為聯準會進行貨幣決策前的討論委員會，決策必須在經濟成長與通膨之間取得平衡

# 專案流程



# 流程一：網路爬蟲

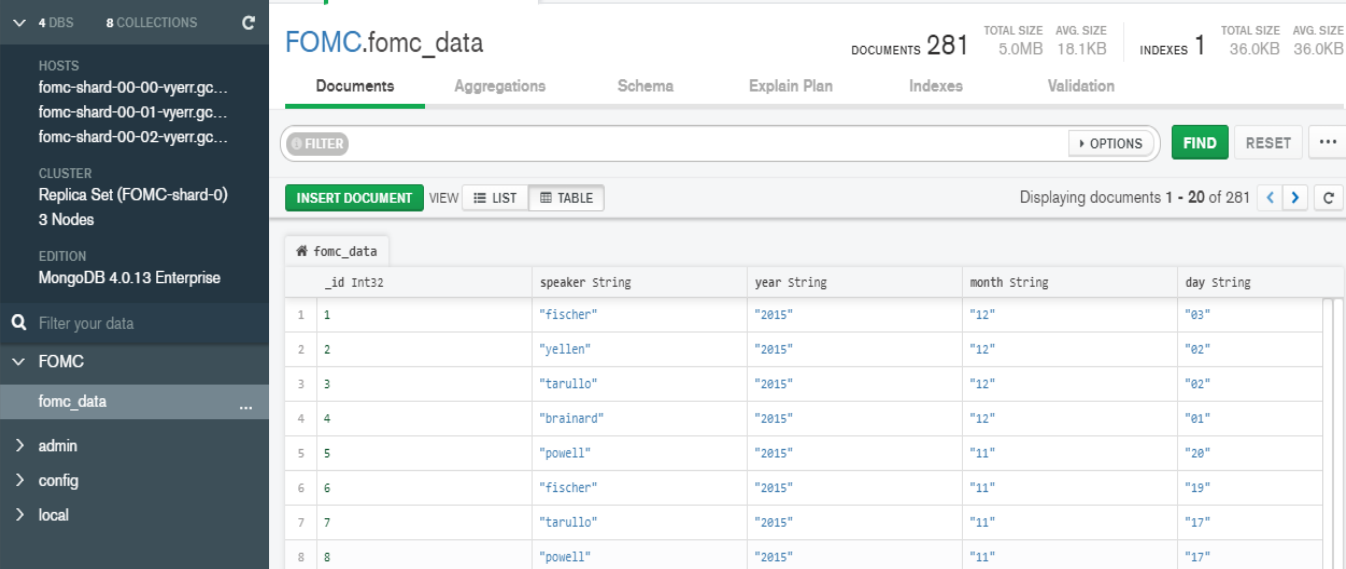
- 說明：  
我們使用python BeautifulSoup套件進行資料整理。首先必須克服的問題是如何有效大量且短時間一次下載所有的文檔。這必須要去研究聯邦公開市場委員會網站的HTML代碼。之後再把下載的檔案轉成json檔方便建立資料庫。

```
matching=[K+x for x in matching]↓  
aa=len(matching)↓  
for i in range(0,aa):↓  
    url = matching[i]↓  
    page = requests.get(url).text↓  
    soup = BeautifulSoup(page)↓  
    p_tags = soup.find_all('p')↓  
    p_tags_text = [tag.get_text().strip() for tag in p_tags]↓  
    sentence_list = [sentence for sentence in p_tags_text if not '\n' in sentence]↓  
    sentence_list = [sentence for sentence in sentence_list if '.' in sentence]↓  
    article = 'NEW ARTICLE' + url + '.join(sentence_list) + 'END ARTICLE' ↓  
    my_json_string = json.dumps(article)↓  
    with open('2019', 'a+', encoding='utf-8') as f:↓  
        json.dump(article, f, ensure_ascii=False, indent=4),
```

示範碼

# 流程二：建置資料庫

- 說明：  
本組使用非關聯式資料庫(MongoDB)來儲存爬蟲結果。因非關聯式資料庫較關聯式資料庫具儲存彈性；且文本檔案較大，使用非關聯式資料庫在讀取上較具效率。建立資料庫的過程除了在MongoDB建立帳號並且設定存取權、安裝之外，在本地使用MongoDB Compass以便確認資料庫是否能夠正常連線及確保爬蟲的結果能夠順利地儲存在資料庫內。



The screenshot displays the MongoDB Compass interface for the 'fomc\_data' collection. The left sidebar shows the database structure, including hosts, cluster information, and a list of collections. The main area shows the 'Documents' tab with a table of 281 documents. The table columns are: \_id (Int32), speaker (String), year (String), month (String), and day (String). The first 8 documents are visible, showing a sequence of speakers and dates from 2015.

	_id Int32	speaker String	year String	month String	day String
1	1	"fischer"	"2015"	"12"	"03"
2	2	"yellen"	"2015"	"12"	"02"
3	3	"tarullo"	"2015"	"12"	"02"
4	4	"brainard"	"2015"	"12"	"01"
5	5	"powell"	"2015"	"11"	"20"
6	6	"fischer"	"2015"	"11"	"19"
7	7	"tarullo"	"2015"	"11"	"17"
8	8	"powell"	"2015"	"11"	"17"

資料庫顯示

# 流程三一詞性還原(Lemmatization)

英文單字會因時態、單複數不同而變化，若不處理會造成文字探勘研究的偏誤，例如 the performance looks good 和 the performance is better than last year 兩句話的 good 和 better 是比較級關係，卻會被當成兩個不同的單字

- 使用套件：NLTK + Stanza(美國 Stanford大學開發之語言處理套件)
  - 以 it' s better than before 為例
  - NLTK: it 's good than before
  - Stanza: it be better than before
- Stanza 無法處理形容詞之詞性還原、NLTK不夠細緻，縮寫(ex. 無法處理 It 's)
- 目標：  
went/ goes → go  
cars → car  
better → good

# 流程三—斷詞(Segmentation)

先進行各種文本預處理，例如透過人工標記的方式保留完整片語、去除符號及stop words，使結果更精確

- 使用套件：NLTK
- 斷詞預處理：去除符號及stopwords 後，在保留片語的前提下將句子斷成單詞
- 以 *However, there are a lot of companies doing this!* 為例

處理  
順序

- 詞性還原後的句子: *however, there be a lot of company do this!*
- 去除符號及 stopwords : *however there a lot of company do this*
- 保留片語進行斷詞 : *however, there, a lot of, company, do, this*
- 若不保留片語語意會不精準 : *however, there, a, lot, of, company, do, this*

得到一  
串詞的  
list 以進  
行後續  
分析

- Stop words 定義(Stanford) : some extremely common words which would appear to be of little value in helping select documents matching a user need are excluded from the vocabulary entirely. These words are called *stop words* .

# 流程四：標註

- 說明：  
以人工方式針對每一文句進行標註，標註類別有三：Pos、Neu、Neg。因文本數與句數不多，於是四人交互檢視標註結果，倘若分歧嚴重將詢問業主意見。

# 流程五：建置模型

- 說明：  
詞嵌入使用word2vec進行訓練，我們可以利用這模型得出各詞之向量值。本組將訓練完成的word2vec模型，排序出各狀態關鍵詞之最相關的單詞。用線性內插法(兩端為100%升息、100%降息)，得出估計之升息或降息或維持機率。



# 結果檢視

- 說明：  
實際值本組採用2019年各次聲明發布時，美國公債期貨市場上的殖利率所隱含之升降息機率(數據來源：Bloomberg、財經M平方)。今年的預測方向大致與市場預期相同，估計機率值和實際機率值之差距都維持在33%之內。

	12/11	10/30	9/18	7/31	6/19	5/1	3/20	1/30
Calculated Possibility (估算值，負數表示 降息機率)	-22.82%	-22.63%	-21.83%	-24.38%	-28.11%	-31.12%	-34.12%	-28.45%
Market Implied Possibility (實際值，負數表示 降息機率)	-8.90%	-22.90%	-43.80%	-56.50%	-20.80%	-10.00%	-2.00%	1.30%
估計誤差	-13.92%	0.27%	21.97%	32.12%	-7.31%	-21.12%	-32.12%	-29.75%

# 改進方向

- 套入更多更新機器學習方法降低誤差值
- 納入媒體新聞看法，擴充樣本
- 觀察改為分類預測之結果