Machine Learning Final Project Report

- 新聞立場檢索技術獎金賽

指導教授：李宏毅
隊名：NTU_b05901063_39得第一4
b05901063 黃世丞
b05901025 王鈺能
b05901176 楊采綸

1. Introduction & Motivation :

我們這組的題目是「新聞立場檢索技術獎金賽」。

具爭議性議題的新聞一直是閱聽人關注與討論的焦點，例如：美國牛肉開放進口、死刑廢除、多元成家等。不論是政治、經濟、教育、兩性、能源、環保等公共議題，新聞媒體常需報導不同的立場。若能從大量的新聞文件裡，快速搜尋各種爭議性議題中具特定立場的新聞，不但有助於人們理解不同立場對這些議題的認知與價值觀，對制定決策的過程而言，也相當有參考價值。

我們的目標是開發一搜尋引擎，找出「與爭議性議題相關」且「符合Query的立場」的新聞，並依照相關程度由高至低排列。

2. Data Preprocessing/Feature Engineering :

- 使用jieba將query和新聞內文進行斷詞
  我們先刪除長度過短(字數小於30)和完全沒內容或廣告的文本，接著使用jieba的繁體中文辭典，再加上和20個query相關的專有名詞，例如：二段式左轉、十八趴.....等等，以此辭典進行斷詞，再依據自定義的stopword list，例如：標點符號、的、和.....等等，把沒有意義的字去掉，作為新的corpus

- 使用gensim的Word2vec產生辭彙的embedding向量
  利用上一步得到的corpus，用gensim Word2vec model訓練，得到每個字的word embedding向量

- 使用gensim的Doc2vec產生整個文本的embedding向量
  和word embedding類似，用gensim Doc2vec model訓練，得到每篇文本的Document embedding向量

- 人工增加query查詢的辭彙
  我們利用domain knowledge，增加query的詞彙，例如：國民黨是贊成服貿的，因此我們在該query加入「國民黨」這個詞，提升retrieval效果

- 將Bert以網站提供的訓練資料進行fine-tune
  Bert的run_classifier.py可以判斷兩個句子的語意關係，關係可以是「entailment(蘊含)」、「neutral(中立)」、「contradiction(矛盾)」三者其中之一。
  我們將訓練資料中label為3的資料標示為entailment，label為0的資料標示為contradiction後，先對XNLI(一個包含中文資料的語意判斷資料集)進行fine-tune，再對訓練資料二次fine-tune，希望能依靠Bert找出內容類似但是立場(語意)相反的新聞。

3. Methods :

- End to end model

  我們嘗試了三種架構，(1) 先將news內文丟進去gensim word2vec train好embedding，再將query和title接起來當作input丟進由兩層Bi-LSTM加一層DNN的model中 (2) 不接起來而將model改為由兩個小model並

聯的形式 (3) 覺得標題可能不太夠，嘗試加入內文一起train，model改為三個小model並聯。我們發現這個任務使用end to end model的話會train不起來，可能是因為官方給的training data太少而testing又必須在十萬個新聞中找到好的，硬用這些data訓練會讓model自己學到很奇怪的東西，表現也完全比不上直接用word embedding去算相似度。

- Doc2vec

  使用gensim的Doc2vec計算新聞內文和query的相似度，作法為將preprocess後的文本作為training corpus丟入doc2vec model，訓練完後將query做分詞到model中計算前300相似的文本輸出。

  此作法相較前一作法有改善(但仍然很差)，觀察其輸出發現的確部分query有抓到相關的文章，但大部分都抓到的內文都是很短的文本(很短的文本剛好幾乎都是廣告、樂透等等，因此後來才決定preprocess時把長度太短的刪掉)，結果仍不夠好。

- TF-IDF

  使用gensim的TfidfModel計算新聞內文和query的相似度，可以直接過simple baseline，再多調整後(見Retriever)可以過strong baseline

- bm25

  使用gensim.summarization.bm25計算新聞內文和query的相似度，表現比單純使用TF-IDF略好，加上query expansion也可以再提升0.03~0.05左右的分數

- Latent Semantic Indexing (LSI)

  使用gensim的LsiModel計算新聞的latent topic(新聞-words matrix的eigen value)，然而可能因為新聞太多太雜又互相有些許相關，因此沒有如預期地把各類新聞分很開。

- Retriever

  這是我們各種方法的集大成者

  1. TF-IDF pivot document length normalization

     由於長的文章容易包含更多關鍵字，因此若不做調整，TF-IDF偏好選到長的文章，因此要做pivot document length normalization，給予長的文章penalty(詳見[3])，gensim model可以直接調，我們設pivot point為所有文章的平均長度、斜率為0.2，這樣可以使retrieval結果變好

  2. 更多Query expansion

     我們利用domain knowledge手動加入我們認為合適及不合適的query單字，調整query word vector

3. Pseudo Relevance Feedback(PRF)

   我們生成第一批搜尋結果，並假定前N項搜尋結果與query相關(pseudo relevant)，將這些文章出現次數前K多的word乘以weight W當作新增的query加入query vector內，對corpus做re-rank，將re-rank的前300項作為最後的結果

4. 手標training data

   我們看我們找出來的前300筆新聞，額外標記「立場相同(3)」和「立場相反(0)」的新聞，作為新的training data

- Whoosh

  Whoosh是個現成的中文檢索API，只要將欲檢索的document加到索引中，再將query用結巴分詞丟進去searcher去對每個建立好的索引算分數即可。

- Add Training Data

  我們原本只使用label為3跟0的資料，是3的就把他排到很前面、是0的就直接刪掉，後來連label為2和1的都直接加進去只不過順位稍微調低一點，MAP分數就大幅上升。我們覺得應該是因為top300裡面也可能抓到很多實際上分數是0的news，則加入label1, 2的話就會把這樣的news擠掉，讓表現變好。

4. Experiment and Discussion :

- BERT XNLI Processor判斷語意
  由於TF-IDF檢索是靠關鍵詞出現的頻率來計算分數、無法分辨新聞的立場，故高機率讓立場完全相反的新聞出現在很前面的排名。
  而Bert官方文檔有個fine-tune的task是在判斷不同句子間的語意關係是contradiction, neutral或是entailment，我們希望在檢索出前500名相關的新聞後，再透過丟文檔進fine-tune過後的Bert model中來判斷他和query是否是立場相同的，若是相反的話就刪掉。
  - 先用XNLI的dataset fine tune BERT，再去predict query和新聞之間的關係(--max_seq_length=128 \ --train_batch_size=32 \ --learning_rate=5e-5 \ --num_train_epochs=2.0)

    ```
    INFO:tensorflow:***** Eval results *****
    INFO:tensorflow:  eval_accuracy = 0.76546186
    INFO:tensorflow:  eval_loss = 0.605138
    INFO:tensorflow:  global_step = 24543
    INFO:tensorflow:  loss = 0.6051433
    ```

    fine tune完後，我們把每個query前500名的新聞切成最大長度為128的片段，並且將其和query對應作為新的testing data丟進BERT中predict。由於一篇新聞被切成幾個片段，故判斷新聞跟query立場是否相同時，我們是用count來判斷，若一片段contradiction分數最高則count+=1；entailment則count-=1；neutral則不變。若最後count為正表示立場相反，刪掉該新聞。

然而實際實驗結果發現大部分的pair都會被判定成neutral，造成以count來判斷立場的方法不可行。
我們認為這是因為XNLI的training data其實是由一些比較簡單、簡短的語句組成、兩句的重疊性也算高，而我們後來丟進model中的data都非常之長，用詞和文法較為複雜、跟query的重疊性也不算太高，再加上原本較簡單的task eval_accuracy也只到0.76，還是有些會判斷錯，造成BERT predict的結果不算太好。

好吧 你 在 市场 上 买 新车	你 在 找 一 辆 新车 吗？	entailment
白 点点头	白人 摇头了	contradicto

通姦在刑法上应该除罪化。	文化部长龙应台日前指出通姦罪至今存于刑法之中，是个台湾应该慎思改弦易辙之道的问题。
通姦在刑法上应该除罪化。	话题的起因，来自于我国首次依据联合国的规格提出国家人权报告，经主动邀请国家专家审视之后，专家们提出了多项建议，其中包括国家以刑罚干预通姦行为的政策应予检讨。
通姦在刑法上应该除罪化。	关注社会文化的龙部长，于是有感而发。
通姦在刑法上应该除罪化。	龙应台的发言，引起了正反各种回响，政府部门之中，法务部对于通姦除罪化似乎态度保守，行政院会中之讨论，则是莫衷一是。
通姦在刑法上应该除罪化。	此项议题非自今日始，在不同的民间女性权益团体之间，似也还未形成一致的看法。
通姦在刑法上应该除罪化。	如今话题重开，反对除罪化者有质疑此事与文化部何干，龙部长不该吹皱一池春水者；我们则以为，此事关系国家刑事政策走向，终须有个正确的判断，通姦除罪化的时刻到了。
通姦在刑法上应该除罪化。	这件事或许与法务部的职责关系更大，但是确实也与社会文化密切相关；时至今日，性别平等虽然已成金科玉律，但也不要忘记通姦是前性别平等社会的产物，在刑法是用来维持人伦

- 先用XNLI的dataset fine tune BERT，再用官方給的training data去fine tune
  我們這次實驗是沿用上一個的步驟，不過參數有稍微修改(--max_seq_length=512 \ --train_batch_size=6 \ )，為了讓文章不要被分太多段，我們將max_seq_length為512(BERT的最大值)，而因為運算資源的問題（只有一張1080ti），batch_size只能設成6，否則會OOM。
  不過因為我們的learning rate和epoch沒有調整，最後出來的eval_score沒有太好。

- Retriever 調參數

參數/MAP分數	0.3258403	0.3275964	0.3198263	0.3256541	0.3278729	0.3259687	0.3266943
expand_size	40	50	50	50	50	44	50
origin_weight	2	2	2	2	2	2	2
add_query_weight	16	32	32	32	32	32	32
train_weight	8	4	4	4	4	4	4
expand_weight	4	2	2	2	2	4	4
train_expand_word_num	100	100	50	50	100	140	150
feedback_expand_word_num	100	300	200	200	300	444	250
iterate	1	1	2	1	1	1	1
slope	0.2	0.2	0.2	0.2	0.3	0.2	0.2

expand_size	PRF 當作相關的文章數量
origin_weight	原本query在expand及feedback時乘的weight
add_query_weight	query expansion在expand及feedback時乘的weight
train_weight	training data在expand及feedback時乘的weight
expand_weight	PRF在expand及feedback時乘的weight
train_expand_word_num	在training data取前幾重要的字數
feedback_expand_word_num	在前expand_size新聞取前幾重要的字數
iterate	做PRF的次數
slope	pivot document length normalization的斜率

整體來看，調整各個參數的影響其實都很小，除了增加PRF的次數會讓performance明顯下降外，其他參數增加會減少並沒有什麼明確的正反相關。
調整weight部分，我們認為自己手加的query是最準的，因此給予最大的weight，training data中取出來的字也很重要，而PRF的字可能會比較不準，因此weight相對較小。
加入query的字數不能太多或太少，太少抓關鍵字的效果不明顯；太多可能反而會抓到很多不相關的字。

- 負向的Query
  把原始的query加入一些「負」的辭彙，希望能減少搜尋結果中出現這些辭彙的機率。舉例來說，釋昭慧相當反對博弈在台灣合法化，因此出現釋昭慧的新聞通常是反對的立場，我們就希望搜尋結果盡量不要出現釋昭慧。實作的方法是query expansion時，將TF-IDF向量中負向辭彙所在的維度的分數乘以-1，再和原本沒有負向辭彙的向量作加權平均。這樣處理過後的TF-IDF向量，在負向辭彙的TF-IDF分數都會是負的，造成含有負向query辭彙的新聞在計算cosine相似度時會有penalty。

- Retriever使用另一種算TF-IDF的公式

  原本的TF-IDF公式是用smartirs的L+t+c，後來發現gensim可以自己調整想要的TF-IDF公式，故我們就改成右下圖的公式，不過perfomance沒有

Term frequency		Document frequency		Normalization	
n (natural)	$tf_{t,d}$	n (no)	1	n (none)	1
l (logarithm)	$1+\log(tf_{t,d})$	t (idf)	$\log \frac{N}{df_t}$	c (cosine)	$\frac{1}{\sqrt{w_1^2+w_2^2+\ldots+w_M^2}}$
a (augmented)	$0.5+\frac{0.5\times tf_{t,d}}{\max_t(tf_{t,d})}$	p (prob idf)	$\max\{0, \log \frac{N-df_t}{df_t}\}$	u (pivoted unique)	$1/u$ (Section 6.4.4)
b (boolean)	$\begin{cases}1 & \text{if } tf_{t,d}>0\\ 0 & \text{otherwise}\end{cases}$			b (byte size)	$1/CharLength^{\alpha}, \alpha<1$
L (log ave)	$\frac{1+\log(tf_{t,d})}{1+\log(ave_{t\in d}(tf_{t,d}))}$				

CharLength: the number of characters in a document.

tf	is the term's frequency in document
qtf	is the term's frequency in query
N	is the total number of documents in the collection
df	is the number of documents that contain the term
dl	is the document length (in bytes), and
$avdl$	is the average document length

Okapi weighting based document score: [23]

$$\sum_{t\in Q,D} ln\frac{N-df+0.5}{df+0.5}\cdot\frac{(k_1+1)tf}{(k_1(1-b+b\frac{dl}{avdl}))+tf}\cdot\frac{(k_3+1)qtf}{k_3+qtf}$$

k_1 (between 1.0–2.0), b (usually 0.75), and k_3 (between 0–1000) are constants.

  比較好。

5. Conclusion :

- 由於訓練資料非常不足，只有4000多筆，而且品質參差不齊，因此所有ML方法的效果都非常糟糕，或許未來訓練集擴大以後可以再嘗試使用ML的方法。
- 傳統的TF-IDF方法效果較好，在只使用最基本的term frquency x inverse document frequency即可以過simple baseline，然而其缺點是沒有像ML一樣有一個系統性調整參數的方式，必須trial and error一個個嘗試可能可行的方法及參數。我們嘗試了pivot document length normalization(針對長度做較長的新聞降低分數、較短的反之), query expansion(加相關的字進入query), pseudo relevance feedback(取第一次query前幾名的結果當作相關加入第二次query)，他們分別都對retrieval有幫助。
- 直接把training data的
- 資料加入，不論按照321的順序放到輸出結果的前面，雖然放2和1很不直覺，但放了之後結果卻會好非常多。

6. Reference :

[1] Bert: https://arxiv.org/pdf/1810.04805.pdf
[2] Universal Sentence Encoder: https://arxiv.org/pdf/1803.11175.pdf
[3] TF-IDF: http://singhal.info/ieee2001.pdf
[4] Gensim Word2vec: https://radimrehurek.com/gensim/models/word2vec.html
[5] Gensim Doc2vec: https://radimrehurek.com/gensim/models/doc2vec.html
[6] Gensim tfidfmodel: https://radimrehurek.com/gensim/models/tfidfmodel.html
[7] Whoosh: https://www.jianshu.com/p/127c8c0b908a
[8] Introduction to Digital Speech Processing: http://speech.ee.ntu.edu.tw/DSP2019Spring/