

Final Project Proposal

一、Basic Information

Team Name: **KPCOF (KoPingChen Of First)**

(排行榜隊名: NTU_b05901111_KPCOF (限20字))

Team Members:

B05901111 電機三 陳建成 (隊長)

B05901040 電機三 蔡松達

B05202061 電機三 陳威旭

B05901192 電機三 張晁維

Topic: **Intent Retrieval from Online News**

二、Problem Study

我們認定此題目是以文本分析為核心，因此關於文字的處理方面，我們大約看了以下幾篇paper：

- Universal Sentence Encoder (<https://arxiv.org/pdf/1803.11175.pdf>)
這是我們在BERT之前第一個看到的，也是因為簡報中有提及，一個與Doc2Vec目的相似的sentence encoding的方法。不過雖然有成功載下code，但基於官方文檔上表示僅支援英文（雖然用中文丟下去跑得出來，但是有斷詞或沒斷詞結果差異甚大，我們並不清楚其背後到底是怎麼處理我們的中文string的，因此目前也暫時不採用）。
- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (<https://arxiv.org/pdf/1810.04805.pdf>)
本篇是一個近來相當知名的language model (based on transformer，在5/30堂上有提及) BERT模型的第一次proposal，由於其在於許多NLP的task上面都有不錯的表現，因此有嘗試去應用它於我們的模型架構中。只不過配合起來的實質效果還在調整中。
- Distributed Representations of Sentences and Documents (<https://arxiv.org/pdf/1405.4053v2.pdf>)
這篇是我們前期主要嘗試的實作方法 (Gensim Doc2Vec)，理論方面大約簡略讀過，主要用在實作面上居多。
- TF-IDF (<https://arxiv.org/pdf/1902.09875.pdf>)
這篇是後面嘗試TF-IDF之後想要參考的部分，可能會採取其中適合我們題目的部分來實作。

三、Proposed Method

目前已經嘗試並實作數種方法於本次題目如下：

1. Naive Method : Doc2Vec & NN

將十萬筆新聞資料經過jieba中文斷詞後，利用gensim的Doc2Vec模型訓練得到各新聞內文的embedded vector與訓練資料和測試資料query的embedded vector，並以訓練資料內relevance作為訓練標籤，將訓練資料query的embedded vector和對應新聞內文embedded vector串接後再做翻轉作

為訓練資料輸入神經網路模型，輸出維度為四的向量並假設每個維度的值表示模型預測為該 relevance 的機率，最後計算模型輸出與訓練標籤交叉熵。

排序方式為先比較模型對各新聞內文對該 query 的輸出向量傾向 relevance，同 relevance 內再比較模型對該 relevance 預測的機率高低。

以此方式實作之結果皆不甚理想，上傳平台評估結果約在 0.001 至 10^{-5} 左右，除去 jieba 斷詞效果與 Doc2Vec Embedding 效果不佳的可能性以外，直接將內文與 query 串接作為訓練資料可能不是一個很好的方式，需要再構思生成訓練資料的方式與最佳的訓練模型架構。

2. Naive Method : Doc2Vec

將十萬筆新聞資料經過 jieba 中文斷詞後，利用 gensim 的 Doc2Vec 模型訓練得到各新聞內文的 embedded vector 直接利用 Doc2Vec 的相關性比較，得到 0.01 左右的正確性得分。

3. rule-based BOW : exhaustive research

由於上述做法無法有太好的成效，我們便想到利用比較直覺的方式進行實作，首先對於 100000 筆新聞資料的內文直接進行搜尋，若是出現測資中的關鍵字才納入後續的考量範圍（以 if-else 暴力法搜尋是否有相同的字詞存在，而此處的關鍵字暫時不將立場是贊成或反對納入考慮），並且關鍵字出現越多次便給予該新聞越高的得分，最終 20 筆測資分別可以得到一些相關的新聞（各自約有 500~5000 筆新聞），接著針對被納入考量範圍的新聞進行立場判斷，若是與測資立場相同則再加上更多分數，若立場相反則扣分。兩次的給分結束後便進行排序，得分越高的關聯性越大，越低的關聯性越小，可以得到評估結果 0.13 左右的得分。

4. TF-IDF Bag of words

將十萬筆新聞資料經過 jieba 中文斷詞後，利用 scikit-learn 的 CountVectorizer 把每個句子轉成 Bag of words，接下來再經過 TfidfTransformer 的 transformer 分別計算每個詞語的 TF-IDF 值，將 BOW 的量值 weighted by TF-IDF，因此可以根據每個詞語的重要性去找出用詞分佈相似的文章，進而進行分類。現階段過 simple 的部分還尚未應用 Training data，只是單純使用 cosine similarity 作比較，接著會使用 NN 經過 dense 做 training 當作對於每一個維度（也就是每個個別的詞語）進行重要性參數的調整。

以上唯一通過 simple baseline 的只有 TF-IDF，因此接下來會以此為基礎進行改進。另外雖然先前其他如 Doc2Vec、BERT 之類的還 train 不起來，但是調整 model 架構或許可以得到更好的結果，例如把輸入 concatenate 起來再進行 fine-tuning 之類的嘗試。

四、Reference

- Universal Sentence Encoder (<https://arxiv.org/pdf/1803.11175.pdf>)
- BERT: Pre-training of Deep Bidirectional Transformers for Language (<https://arxiv.org/pdf/1810.04805.pdf>)
- Distributed Representations of Sentences and Documents (<https://arxiv.org/pdf/1405.4053v2.pdf>)
- TF-IDF (<https://arxiv.org/pdf/1902.09875.pdf>)