Information Retrieval and Extraction Project 1 Report

Name and ID

鍾興寰 b04901058 蘇軒 b04203058 謝宏祺 b04902043 劉家豪 b04504042

Division of work

一人各實作一種方法

蘇軒 25% 鍾興寰 25% 謝宏祺 25% 劉家豪 25%

Text Preprocess

本次專案的文字資料來自於網路平台，因此使用者的打字習慣會較為隨性，而且可能會附上超連結。有鑑於此，我們需要先針對文字做一系列嚴謹的前處理。前處理的項目包含以下幾種：

- Lower case: 將所有的英文字母轉成小寫
- Url removal: 利用 regular expression 將 url pattern 全部移除
- Punctuation removal: 移除所有的標點符號
- Lemmatization: 將所有的英文詞做 normalize。比方說，動詞全部轉現在式，名詞全部變單數，be 動詞全部用 be 代稱等等。因此 “I used to eat noodles everyday. However my taste is getting boring…” 這句話會變成 “i use to eat noodle everyday however my taste be get bore”

Methods

以下為我們嘗試的四種方法的說明，包含 siamese CNN, bilateral matching, BM25 以及 siamese CNN + RNN。

- Siamese CNN

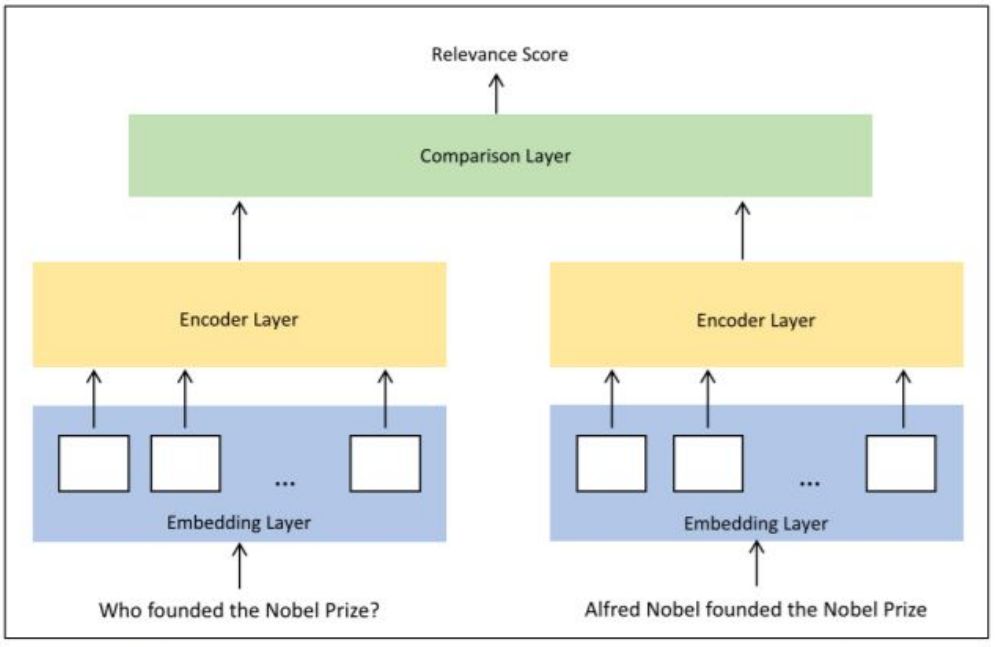
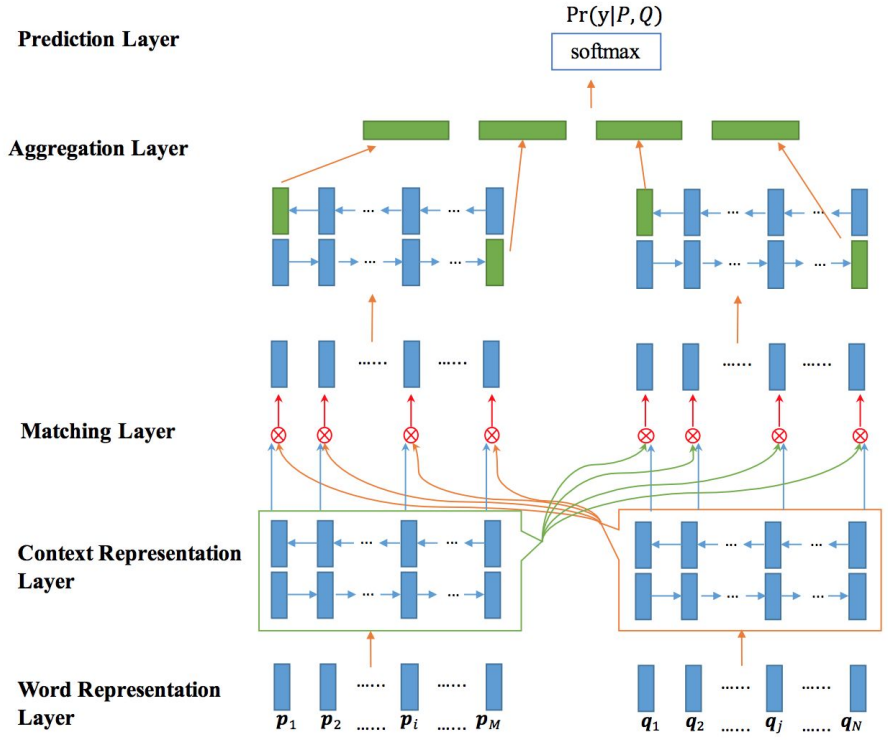


Figure 2: The general architecture of a Siamese model. The same encoder is used to generate the vector representations for the input sentences.

Siamese Model 參考的是 “A Review on Deep Learning Techniques Applied to

Answer Selection” 這篇裡提到的 Siamese architecture。Embedding部份我們採用的是gensim的Word2vec，訓練資料為所有的training data，每個word vector輸出的維度是100維。接著Encoder的部分我們採用的是CNN作為我們的Encoder，將question, comment分別輸入至兩個CNN model(架構相同，參數不同)，中間有加入normalize的layer。不share參數的原因是，我們覺得在question以及comment所使用的語氣以及字詞可能會不相同，因此我們不讓這兩個model share相同的參數。接著我們將兩個CNN output concat起來作為Comparison部分的輸入，這裡不使用 add 而使用 concat 的原因是add可能將question以及comment的feature混合或是中和掉。而Comparison的部分使用的是全連接層，最後透過sigmoid輸出 [0,1] 區間的分數。

- Bilateral Matching



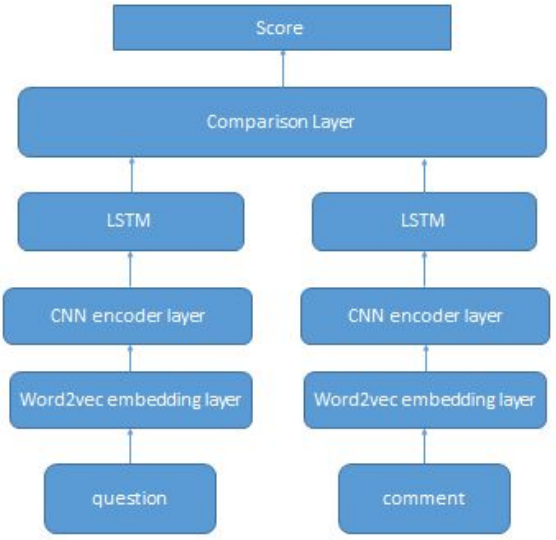
Bilateral matching 參考 “Bilateral Multi-Perspective Matching for Natural Language Sentences” 這篇 paper 的做法，稍微簡化後實作。這個方法類似 siamese network 的架構，但多了將兩個 input sequence 做 matching 的步驟。Question, comment pair（或 question, question pair）各自輸入到 embedding layer，變成兩個詞向量序列 word representation layer。接著再將詞向量序列各自輸入雙向 GRU，每個 timestep 都會輸出向量，形成 context representation layer。然後將 question 的 context representation layer 裡的一個 timestep qi 和 comment 的 context representation layer 的整個序列做 matching。Matching 的方式為將 qi 和 comment 序列裡的每個 timestep ci 做 element wise 相乘再取平均，即可得到 matching 後的 qi。除此之外，也將每個 ci 和整個 question 的 context representation layer 做 matching。最後會有兩個 matching 過的序列，將這兩個序列各自輸入雙向 GRU，再輸入一個 feed forward network 後過 softmax 輸出結果。

- BM25 model

$$\mathcal{B}_{i,j} = \frac{(K_1+1)f_{i,j}}{K_1\left[(1-b)+b\frac{len(d_j)}{avg\_doclen}\right]+f_{i,j}} \qquad sim_{BM25}(d_j,q) \sim \sum_{k_i[q,d_j]} \mathcal{B}_{i,j} \times \log\left(\frac{N-n_i+0.5}{n_i+0.5}\right)$$

bm25是1970 80 年代提出的傳統方法， 在實作上有一定的performance，常被拿來當作基本的baseline，因此我們選擇此方法來當傳統方法的代表。 我們實作上用去除標點符號的testing data算出各參數後， 利用上述公式，計算出query 和 comment 的分數。K1 和 b 是可調整的常數， N是comment總數， 而ni是存在term ti 的comment總數，fij 是 term frequency。在三個task中， K1 = 1 , b = 0.75 。

- Siamese CNN+RNN



Siamese CNN+RNN 整組討論出來的結果，參考自Bilateral Matching還有Siamese Model 方法類似siamese network的架構，先將traing data分成sentence經過gensim的word2vec Embedding成128維的vector，再經過CNN的encoder 把 question 跟 comment 分別經過兩個架構一樣、參數不一樣的CNN model 接著將生成的 feature map 分別經過單方向的LSTM最後再concatenate起來後用sigmoid function來輸出[0, 1]的分數。理由和siamese CNN相同，因為擔心 question 和 comment 的 語法還有用詞不一致所以特別用兩個獨立的network structure 訓練出兩組不同的參數來testing。

Evaluation

Subtask A	Siamese CNN	Bilateral Matching	BM 25	Siamese CNN+RNN
Accuracy	0.6169	0.7358	0.5568	0.6564
F1-score	0.5880	0.7351	0.6445	0.6227
MAP	0.8026	0.8365	0.6271	0.7790

Subtask B	Siamese CNN	Bilateral Matching	BM 25	Siamese CNN+RNN
Accuracy	0.7705	0.8034	0.7920	0.4682
F1-score	0.0288	0.1128	0.3880	0.2844
MAP	0.2649	0.3200	0.4170	0.2833

Subtask C	Siamese CNN	Bilateral Matching	BM 25	Siamese CNN+RNN
Accuracy	0.6428	0.5778	0.7102	0.6119
F1-score	0.0786	0.0857	0.1096	0.0773
MAP	0.0805	0.1168	0.0985	0.0837

Discussion

- NN 在 taskB 上均沒有取得很好的成績，此原因我們推斷是因為在 subtaskB 方面的 data 較 A, C 來的少上許多，導致我們在訓練 model 的時候沒有足夠的資料量來支撐複雜的架構，進而導致嚴重的 overfit，所以 MAP 的分數才會如此的低落。
- Bilateral Matching 有別於 siamese 的架構，它在代表兩個句子的 context representation sequences 還沒合併之前就先做 matching 的動作，加強兩個 input 的交互作用並試圖讓整組神經網路根據這個交互作用關係調整。除了 subtask B 以外，他的 MAP 表現是所有方法中最好的。
- BM25 在TaskB表現比其他方法好的原因推測是因為他利用term frequency， 而相關問題間多半會有幾個相同的關鍵字，因此能達到一定程度的表現。至於其他task較複雜，所以光靠term frequency 等參數無法達到太好的表現。
- Siamese CNN+RNN 想法源自於 Bilateral Matching 與 Siamese CNN 原本預計效果會優於上面各個NN的結果，可是MAP效果的表現卻一般般，可是在 Subtask B與 Subtask C 方面相較於Siamese CNN 有明顯的改善而相較於Bilateral Matching 卻低了許多，原因可能源自於在RNN的實作方面只有用單方向的LSTM 而 Bilateral Matching 卻用 bidirection input 。除此之外，我們也可以從這裡發覺在 Information Retrieval 方面有無使用RNN的效果還是明顯的比CNN還要來的顯著。
- 在進行data process 的時候，為了讓每一個input vector 的維度相同，所以我們使用 pad_sequence來做sentence的刪減及補齊，方法是將word數量超過50的sentence 從後刪減到50個字，而數量不到50的則補zero vector 至 50個。我們發現很有可能因為這樣把sentence中超過可是重要的字刪除掉了，才導致MAP 表現不佳。經過討論後，在對於超過數量超過50的sentence 應該用sliding window 的方式來處理，如此一來不只可以增加資料量，也可以避免上述情形。

Conclusion

這次專案我們嘗試了四種方法。其中包含了傳統的 probabilistic model BM25 與神經網路架構 RNN、 CNN，同時也針對 Siamese network 架構做了一些變化的嘗試 （bilateral

matching）。挑選這些方法的目的是為了綜觀古今，了解過去方法的限制，並站在巨人的肩膀上發揮創意。然而，我們的方法在未來仍然有許多可以進步的空間。比方說可以嘗試不同的 bilateral matching 算法，或是設法化簡神經網路的架構以減緩 over-fitting。經過這次專案，我們對 natural language matching 這個領域有更深刻的認識。