IRIE Final Project Report

蘇軒　b04203058
劉家豪 b04504042
謝宏祺 b04902043
鍾興寰 b04901058

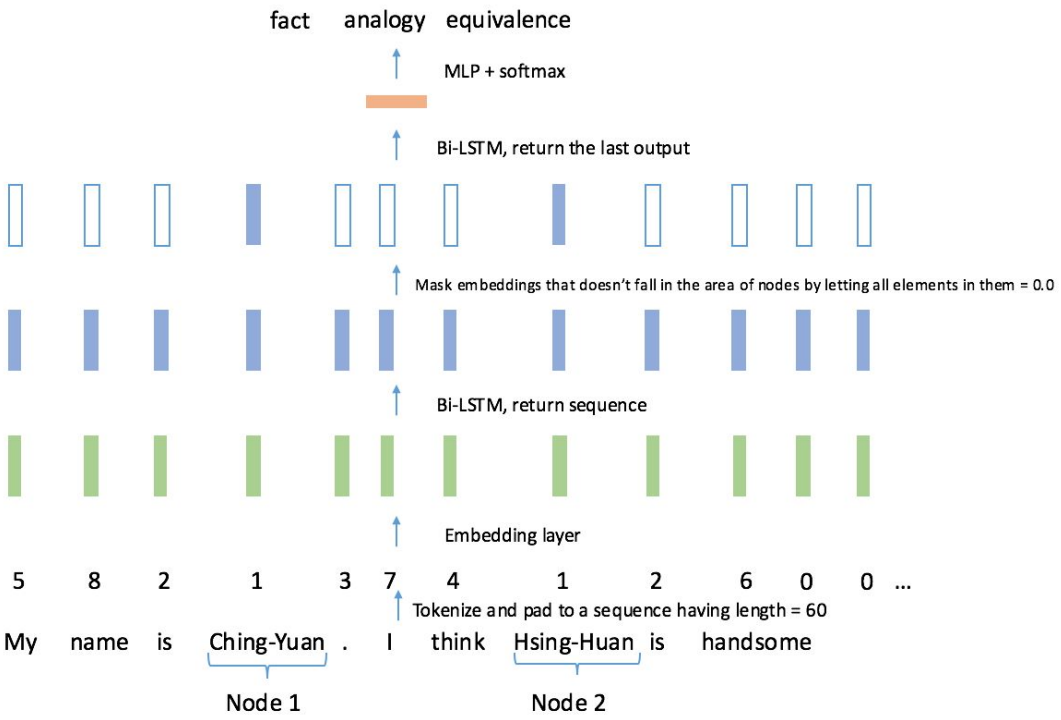
Division of work

蘇軒 25% 劉家豪 25% 謝宏祺 25% 鍾興寰 25%

Methods

Task 1 - Given tokens (without nodes information), predict the edge of given pair.

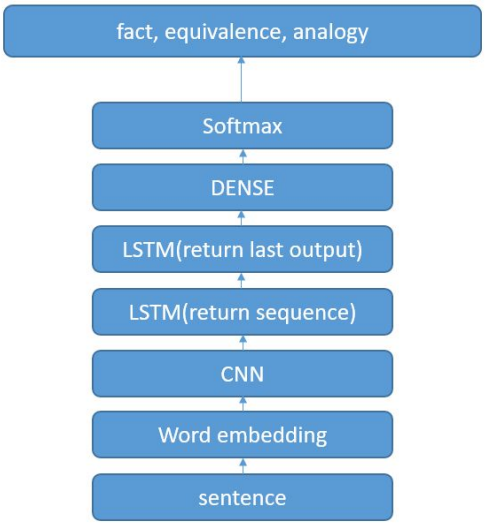
1. Bi-LSTM + masking contextual embeddings



上圖是此方法的架構。首先我們先將輸入句子的每一個字做 tokenize，然後將整句話 pad 到長度為 60 的序列。接著將此序列輸入 embedding layer，得到一個 worddim * 60 的輸出。在這個方法中使用的 embedding 為 pre-trained glove 200d，因此 worddim = 200。接著再將該輸出序列輸入下一層雙向 LSTM (hidden size = 64)，這個 LSTM 在每一個 timestep 都會輸出，因此 Bi-LSTM 層會輸出一個 128 * 60 的 contextual embedding sequence。
除了圖片裡顯示的輸入句子以外，這個模型還會吃另一個輸入 - position sequence。Position sequence 是由 0 跟 1 組成的 list，0 代表該字落在 node 的範圍外，1 則代表該字落在 node 的範圍內。就上圖而言，position sequence 會是 [0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0 ….]。這個 position sequence 會過另一層 128 維的 embedding layer，0 對應到的 embedding 裡的元素

都是 0.0，1 對應到的 embedding 裡的元素都是1.0，輸出的會是一個 128 * 60 的 position embedding sequence。
將 contextual embedding sequence 和 position embedding sequence 做 element-wise 的相乘，即可得到 masked contextual embedding sequence。由於我們是要預測兩個 node 之間的關係，只要不是在要預測的 node 所對應的 embedding 都會被歸零，資訊都會被消滅。最後我們將 masked contextual embedding sequence 過第二層雙向 LSTM (hidden size = 32)，並且只在最後一個 timestep 輸出一個 64 維的結果。將該輸出輸入三層 fully connected layer（第一層 32 個 neuron，第二層 16 個 neuron，第三層 3 個 neuron），再接一個 softmax layer 即可得到兩個 node 的 relation 在 {fact, analogy, equivalence} 中的機率分佈。
訓練過程中使用的 loss 為 categorical cross entropy，learning rate = 0.005，optimizer 使用 Adam。

2. CNN + two-layer LSTM



上圖是這次 Task 1 的架構，用和 Task 2 的 two-layer Bidirectional LSTM 同樣紀錄 node 位置的方式（稍後會有圖文詳述），簡單來說就是將每個 word embedding concatenate 4 維的 vector 以表示句字中每個字與 node 的距離，與 task 2 只是差在沒有Node information的情況下就沒有辦法在最後面加上 node_label 的 one hot encoding。之後把 sentence padding 成 (50, 260) 的 vector 後再經過一層 CNN 得到 vector 在將他餵進 return sequence = True 的 LSTM，得到的 output sequence 再餵入下一層 LSTM，得到的一維 vector，得到句子中每一個字跟字的關係後，在過 Dense 以及最後的 Softmax 層來做 classification，訓練過程使用 adam 來最佳化並且使用 categorical crossentropy 來當作 loss。

Task 2 - Given tokens and nodes information, predict the edge of given pair.

1. Siamese model with CNN encoding

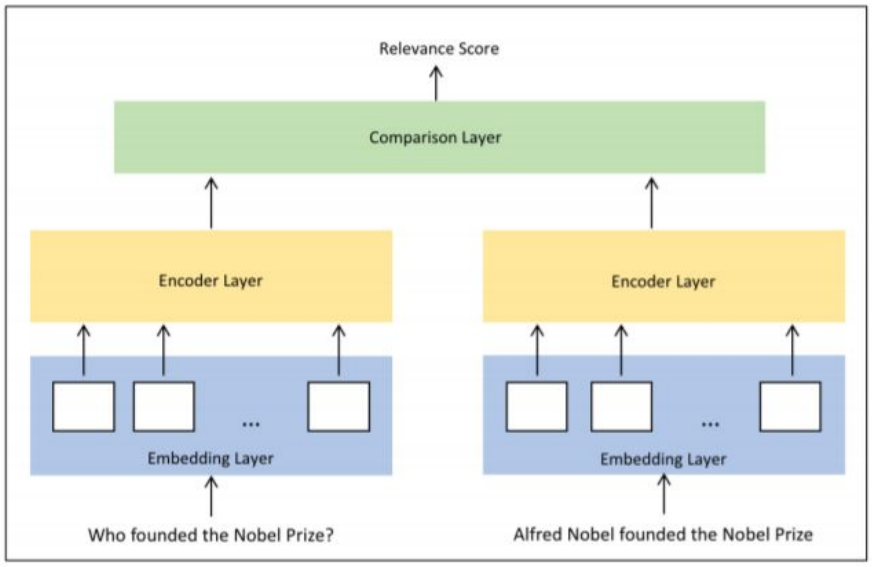
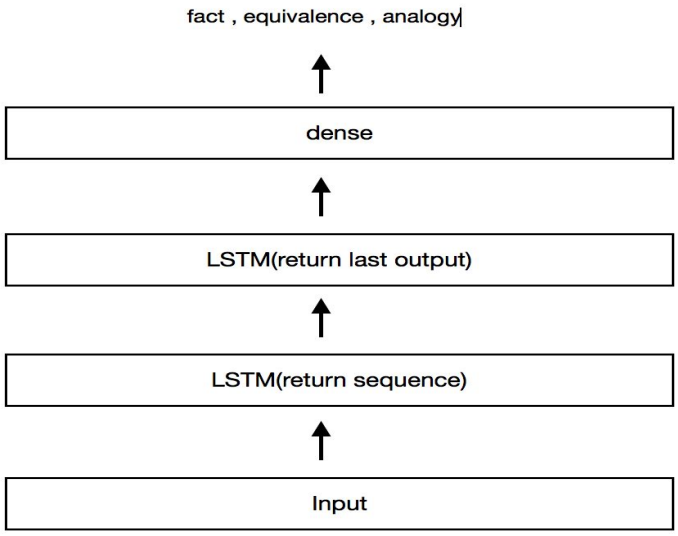


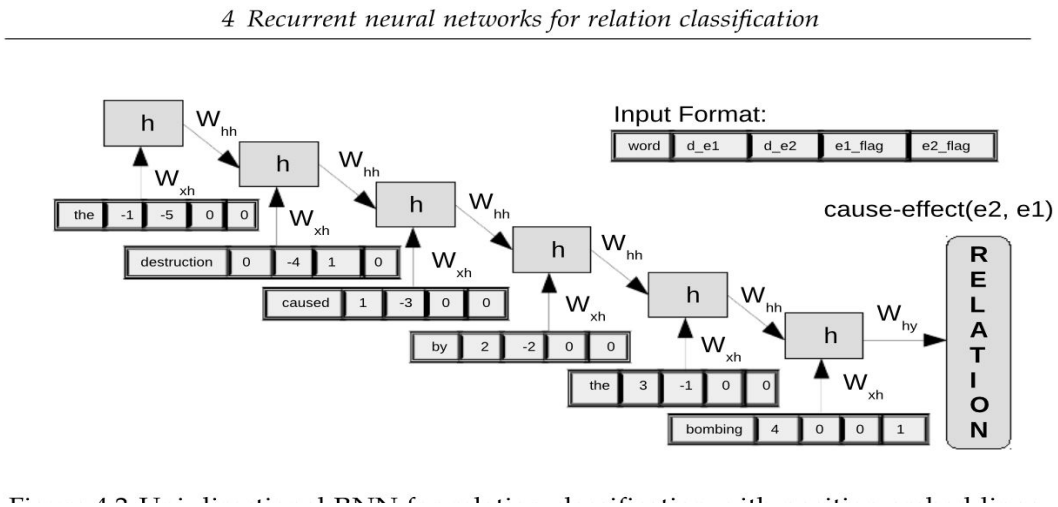
Figure 2: The general architecture of a Siamese model. The same encoder is used to generate the vector representations for the input sentences.

上圖是此方法的架構，這裡我們採用的是跟上次差不多的架構。首先我們一樣是用 gensim 先去做 word2vec 的 pretrain，再用 train 出來的 model 去將 input 轉化成 vector。然後兩個 input 分別是兩個 node 的，這裡 node information 我們選擇是將他 pad 在 token vector 之後。兩個 node 會經過不同的 Encoder Layer (架構相同，參數不同)，之後在 Concat 起來做後去做 softmax 分別對應到三種 edge feature 的機率 output。
Encoder Layer 使用的是三層的 Conv1D (filter分別是16/32/64) 後接一個 BatchNormalization 層。
訓練過程中使用的 loss 為 categorical crossentropy，optimizer 使用 Adam，learning rate = 1e-5, decay = 1e-7，validation_set 為 training data 的10%。

2. two-layer Bidirectional LSTM



基本上我的架構如上圖所示，input vector 餵進 return sequence = True 的 LSTM， 得到的output sequence再餵入下一層LSTM，得到的一維vector，包含了整個句子的資訊。再將此vector丟入一層dense layer 預測relation。比較特別的地方是我的input vector有經過處理。首先利用所有data 建出一個256維的word2vec model，把每個span都換成256維的vector，然後再256維後面再concatenate一些node的資訊，如下圖，



d_e1代表當前這個字和要比對的第一個span的距離，以圖片中第一個字為例，the 在要比對的第一個字destruction前面一個，所以d_e1 = -1， 同樣的，the 在要比對的第二個字bombing前面5個，所以d_e2 = -5。而e1_flag和e2_flag是用來標記要比對的字，可以發現destruction(e1_flag , e2_flag) = (1,0)， bombing(e1_flag , e2_flag) = (0,1)， 其他字都是(0,0)。除了此四維資訊，我們又在後面加上node_label的onehot encoding(19維)。

經過上述處理，我們的span vector變成一個279維(word2vec + 距離標記 ＋node_label)的向量。我們把一個句子padding成input 長度50，所以餵進LSTM的就是一個(279,50)的array。

Evaluation

Task 1 - Given tokens (without nodes information), predict the edge of given pair.

	Bi-LSTM + masking contextual embeddings	CNN + two-layer LSTM
precision	0.964	0.893
recall	0.950	0.882
f1-score	0.958	0.888

Task 2 - Given tokens and nodes information, predict the edge of given pair.

	Siamese model with CNN encoding	Bidirectional two-layer LSTM
precision	0.720	0.993
recall	0.603	0.974
f1-score	0.638	0.981

Discussion

- 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。我們後來有發現 sentence 最大的長度達到 169 這表示我們並沒有收集到全部的pair去train，有可能因此遺失一些重要的資訊，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。
- 經過此次 project 後我們發現在Task 2 上 Siamese CNN 的效果較 Bidirectional two-layer LSTM 的效果差，經過討論之後，我們推估還是因為在語句處理上面，還是太依賴 RNN 這種有前後先後關係的 model 來處理語句上文法還有文字之間的關係，所以 CNN 之效果才會較差。

- 在 data preprocess 上面這次我們沒有將文字進行 stemming 或是 lemmatization 之類的處理語句，經過討論後，如果之後加上上述這些前處理之後，做出來的效果應該會在往上一層。

- 在這次 project 中，我們必須要輸入一個句子以及兩個 node 的位置，最後輸出一個三選一的答案。因此我們的模型必須要能夠吃 node 的位置，否則這就只是一個 sentence classification 的模型，而非 relation classification 的模型。我們有提出兩種在模型中表示 node 位置的方法。方法一就是以 position sequence 直接對 contextual embedding layer 做篩選，將不是在 node 範圍內的 contextual embeddings 歸零。方法二則是在每個詞向量的尾端加四維紀錄 node 的位置。在 task 1 中 Bi-LSTM + masking contextual embeddings 就是採用方法一，CNN + two-layer LSTM 則是採用方法二，雖然後者的 performance 較低但兩者採用不同種類的的神經網路，並不能直接斷言第二種位置表示法比方法一差，只能斷言 CNN 或第二種位置表示法其中至少一者使得 CNN + two-layer LSTM performance 較 Bi-LSTM + masking contextual embeddings 低。至於哪一種位置表示法較好，需要更多實驗才能證明。

Conclusion

這次 project 是我們首次處理 relation extraction 的問題。句子當中詞與詞的關係往往跟句型結構頗有關聯，例如：「RISC-V 是一個免費的 open source project，但免錢的可能更貴，就好比冰冰免費幫高雄拍觀光影片一樣。」這句話當中「RISC-V」跟「冰冰」之間存在類比關係，可以從「A1 …. A2，就好比 B1 … B2」這種句型當中萃取出來。我們目前使用的模型還是 RNN 與 CNN 系列，主要還是以序列以及相鄰關係處理句字，還沒辦法有效處理結構化的句型。未來如果要在處理 relation extraction 的話，我們可以嘗試從 syntactic parsing tree 或 dependency graph 這類輔助資料下手，或許可以萃取出更精確的 relation。