HW3-2

r06922115 鄭皓謙 - attention model
r06942119 林宗憲 - RNN model

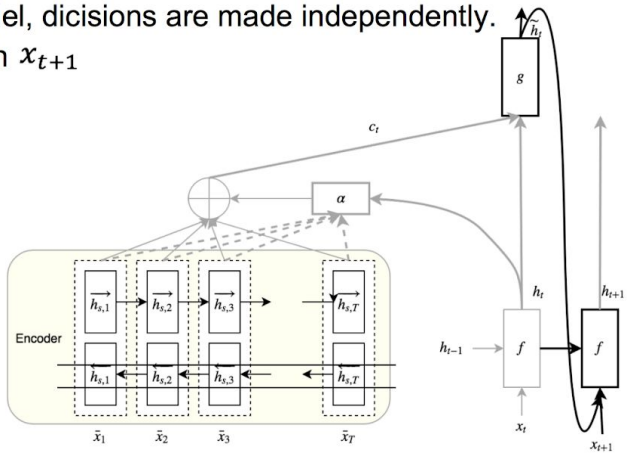
Attention model

Input Feeding

- In Luong's attention model, dicisions are made independently.
- $\tilde{h}_t$ are concatenated with x_{t+1}

$$h_{t+1} = f(h_t, x_{t+1}, \tilde{h}_t)$$

C_t



25

這是模型的架構，並不是特別從講義上的模型改的，只是借用講義的圖片方便呈現。
將bidirectional RNN部份去掉，然後用上一個時間點的context取代上一個時間點的hidden vector。

訓練細節

vocabulary size : 91484
encoder & decoder hidden size : 512
encoder $ decoder layer : 2
teacher forcing rate : 0.5
attention score function : dot(Wa, b)

由於忘記設想testing有unknown toke，所以testing使用random word取代。

Performance with attention

LM score: 28.551820176921225 > 5.0
Jaccard Distance score: 0.041322851167007645 < 0.25
Accuracy，POS : 0.8489 > 0.55, len : 0.9800 >= 0.98, rhyme : 0.9396 > 0.86

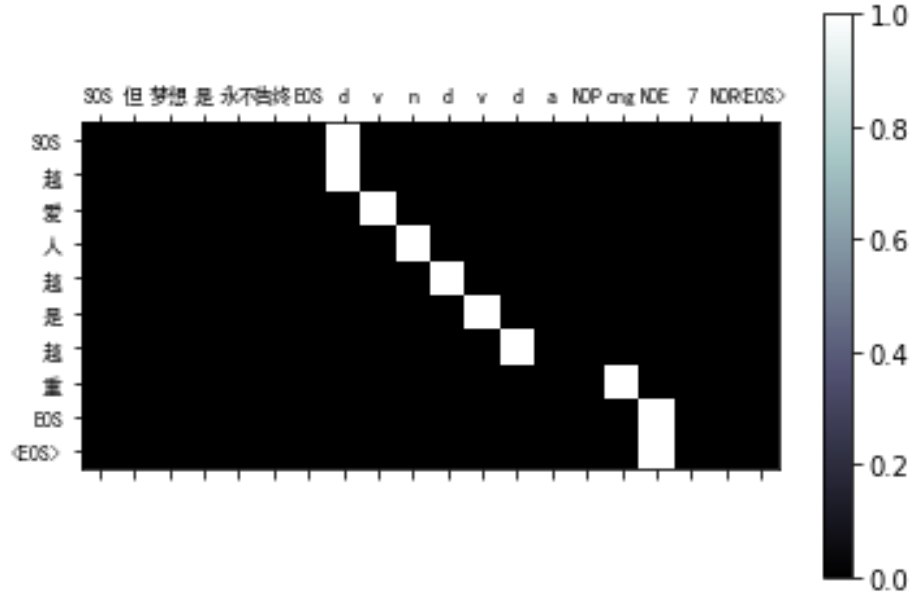
Performance without attention:
LM score: 24.995689776676436
Jaccard Distance score: 0.049450429472488344
Accuracy，POS : 0.7822, len : 0.9940, rhyme : 0.9036

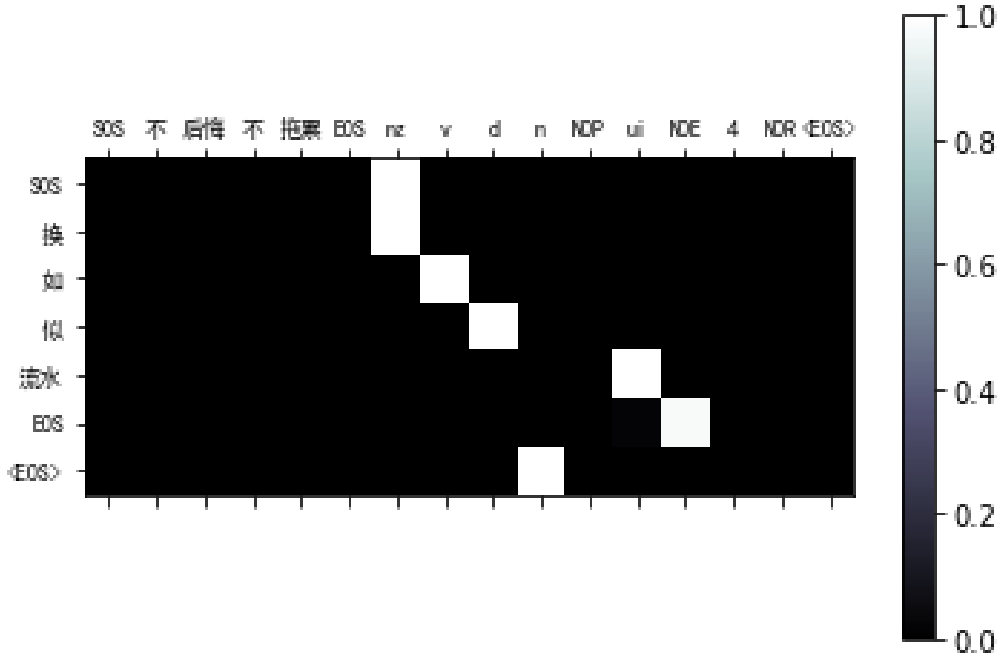
Visualize attention mechanism

Length control performance drop

加入attention mechanism之後，POS跟rhyme都上升了
可是length accuracy卻下降了，從attention的角度來看，predict eos時，並沒有明顯的分佈。
而pos跟rhyme則對應的相當清楚，EOS前一個字會直接對應到rhyme，而不是最後一個pos。

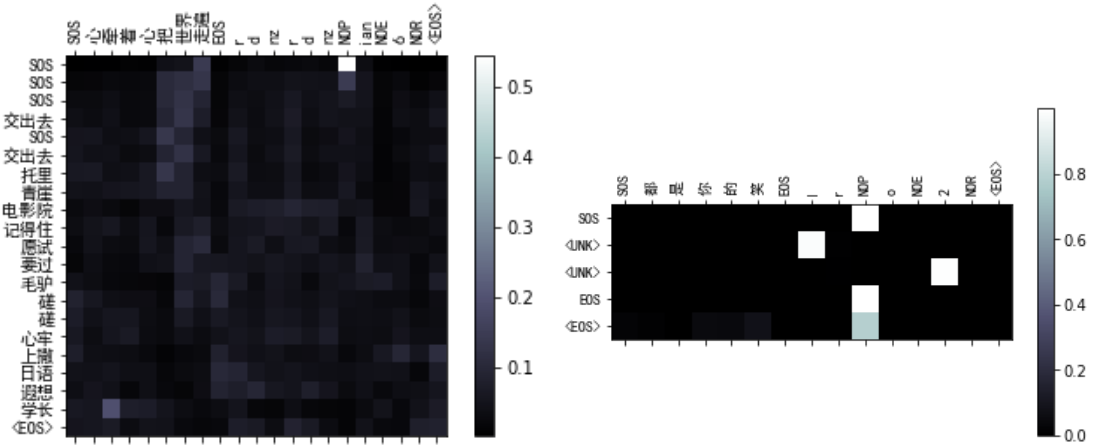
X軸為 輸入 Y軸為 輸出





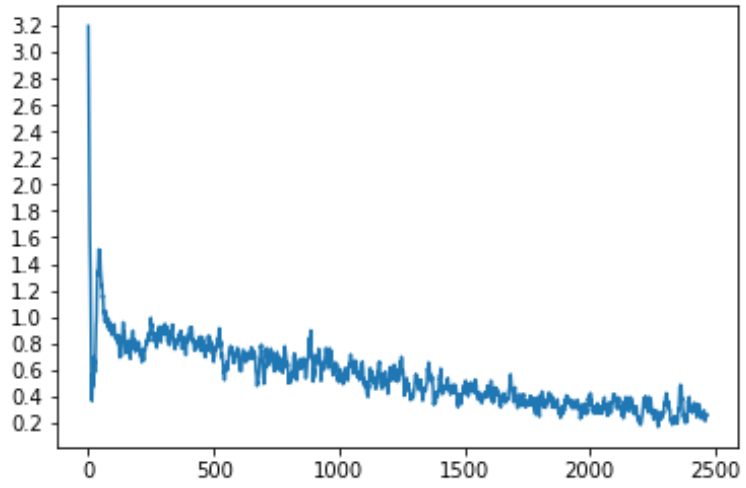
除了EOS沒有明顯的對應關係以外，發現attention weight集中在pos 以及 rhyme的部分。
於是觀察了attention weight在訓練過程中的變化

初期＆中期



attention weight傾向集中而不是分散的分佈，同樣透過計算在訓練過程中，分布的entropy，值越小表示分布得越集中

Ｘ軸為訓練的迭代次數，Ｙ軸是entropy

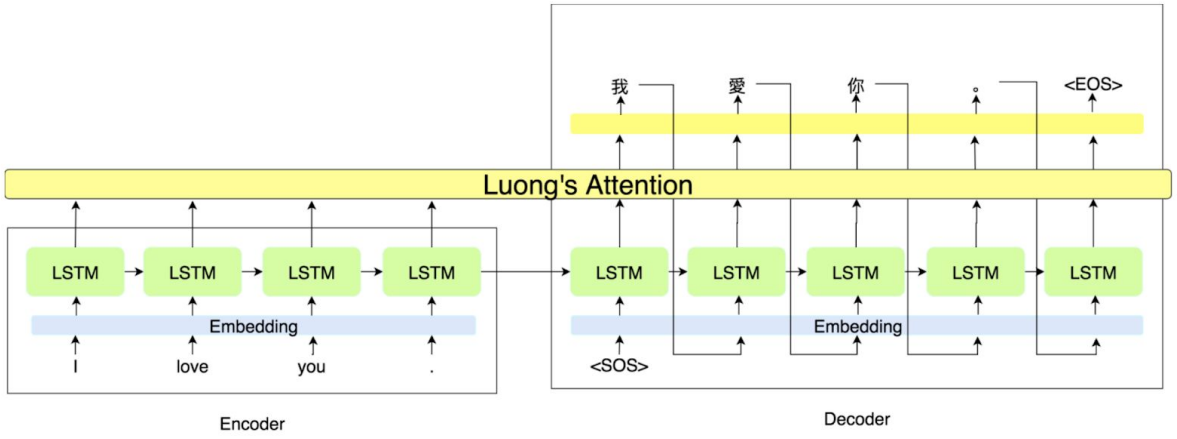


Why model pay attention to POS/Rhyme ?

當context只跟pos / rhyme有關的時候，模型是怎麼決定輸出的？
在只有RNN的seq2seq模型裡，模型就已經能夠控制POS/ rhyme，以及具備上下文相關的能力。
我們猜測是RNN的能力，足夠保留上下文的資訊，而attention加強了pos/ rhyme的對應能力。
為了剝奪RNN hidden state 傳遞資訊的能力，設計了一種新的模型。
encoder RNN只負責提供hidden state給decoder
decoder RNN只負責做attention，而attention的目標則是input sequence。如下圖

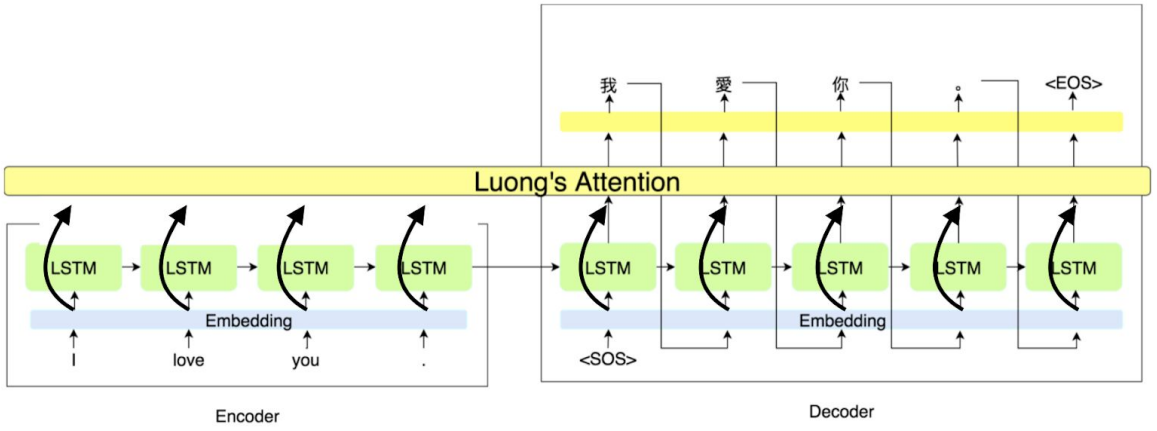
Linear Model

原先的attention示意圖



19

修改的示意圖

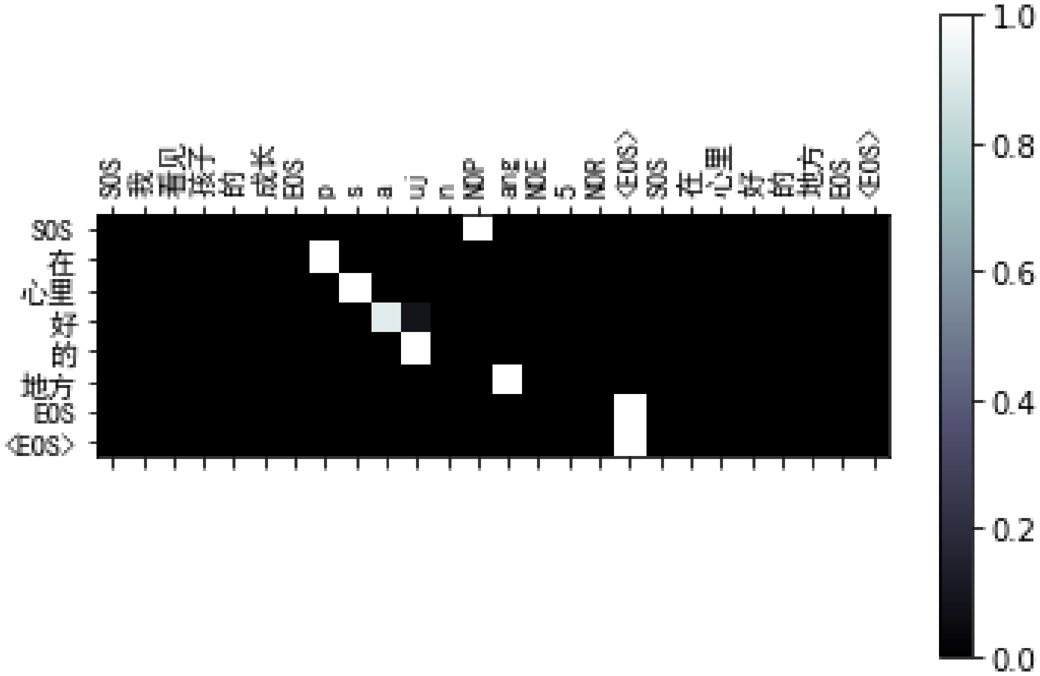


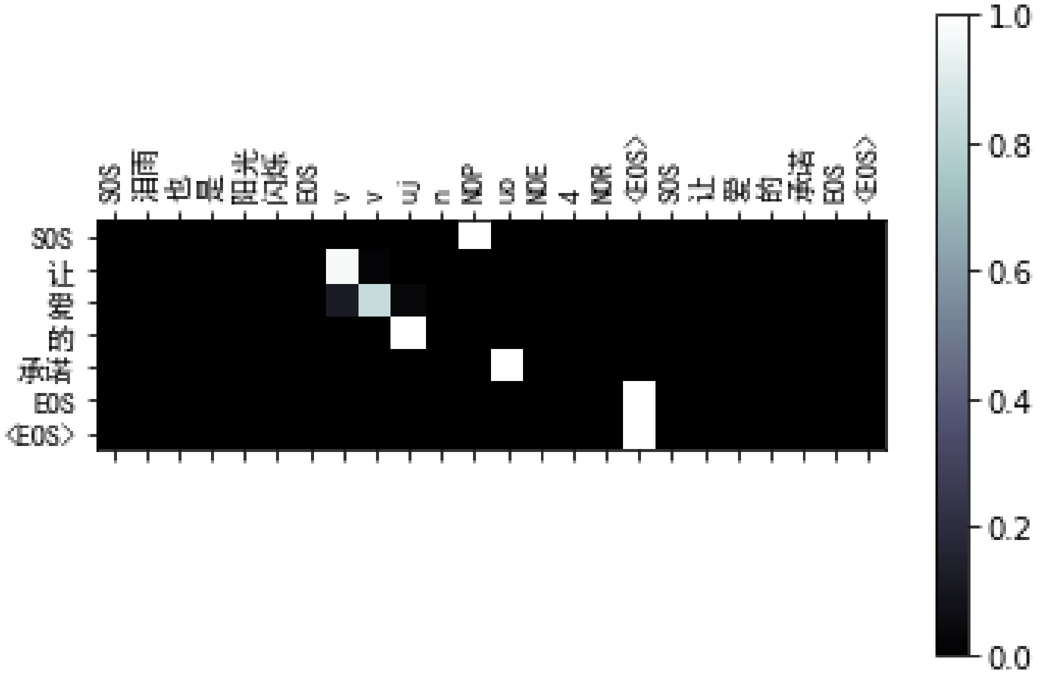
而模型只靠context以及X_i做輸出，如此一來，attention weight代表了參考了哪些輸入。
原先decorder的情況，sequence經過了RNN，所以不了解到底在X_t的時候，RNN還保留了多少X_0 到X_t-1資訊。
沒有RNN的hidden state，所以attention的範圍必須涵蓋decoder sequence去補足語言模型的能力。
為了保留位置的資訊，加入了positional encoding。

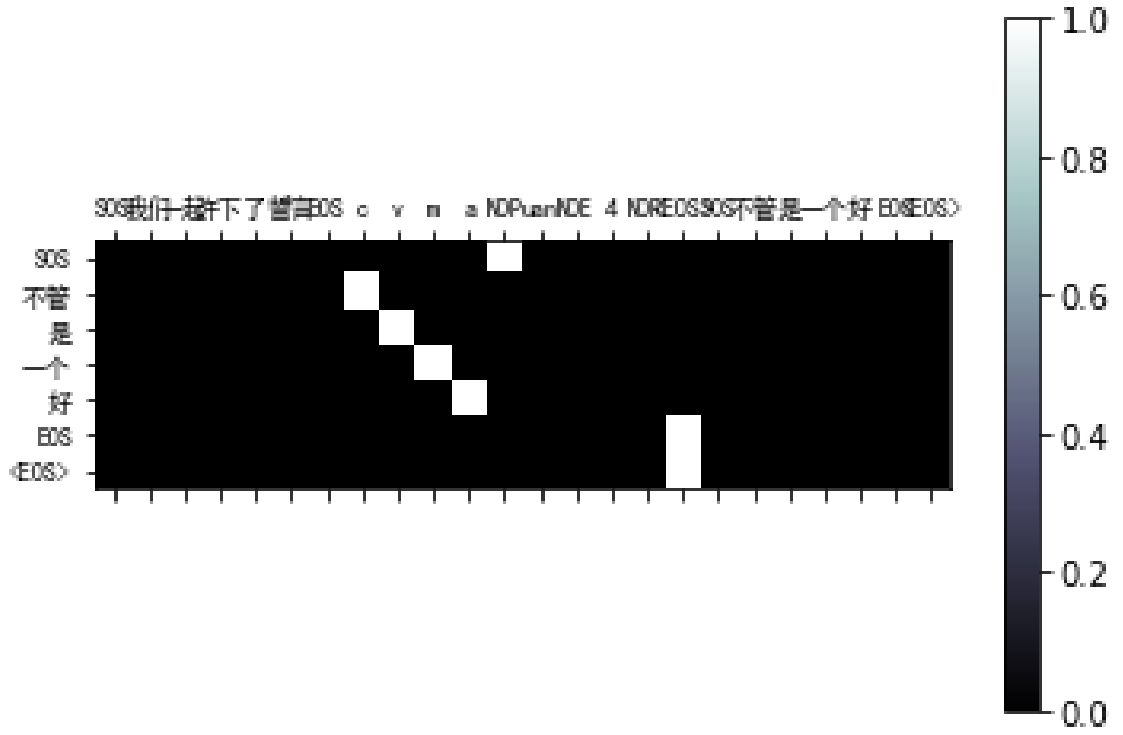
這個模型的表現並不好，只有POS的部份過了baseline
Performance with linear model:
LM score: 19.221698235416476 > 5.0
Jaccard Distance score: 0.07925910889219716 < 0.25
POS : 0.6780 > 0.55
len : 0.9688 < 0.98
rhyme : 0.5700 < 0.86

從attention visualization上來看，有些常出現的韻腳會對到，但是比較不常出現的韻腳

這是新模型的 attention visualization


SOS
在
心里
好
的
地方
EOS
<EOS>
1.0
0.8
0.6
0.4
0.2
0.0


SOS
让
爱
的
承诺
EOS
<EOS>
1.0
0.8
0.6
0.4
0.2
0.0



這些都是挑過比較好的輸出，當長度很長的時候，輸出的情況很糟糕，出現重複字詞。
而且從圖上可以看出，幾乎不看有關歌詞的部分，所以即便過了LM score，但是上下文的關係薄弱，而POS給予的資訊足夠讓模型輸出像樣的句子。

結論

Linear模型原先是希望能過baseline，如此一來便可以輕鬆解釋generate process。
但是效能上的失敗，讓他變成推測RNN能力的實驗。

在剝奪hidden state傳遞資訊能力的情況下，模型很難輸出上下文相關的句子。
推測attention mechanism增強了POS/Rhyme的對應。
而recurrent component則攜帶了上下文的能力，以及語言模型的能力。

改善方向

Linear model在訓練過程上，還有許多改進的地方，由於時間關係沒有太多嘗試。
而上下文關係，也許能夠透過分開兩個attention mechanism，分別對應control signal以及文本，漸少模型收斂的難度。