

隊名：深的不要不要的

- **Environment (1%)**

- OS 2016 x86\_64 GNU/Linux
- CPU Intel(R) Core(TM) i7-4790K CPU @ 4.00GHz
- Memory
- GPU NVIDIA Corporation GM200 [GeForce GTX 980 Ti]
- Libraries:

Tensorflow 1.0

Nltk 3.2.2

Genism 1.0.1

Pandas 0.19.2

- **Model description & reward functions (2%)**

- *Word-embedding*

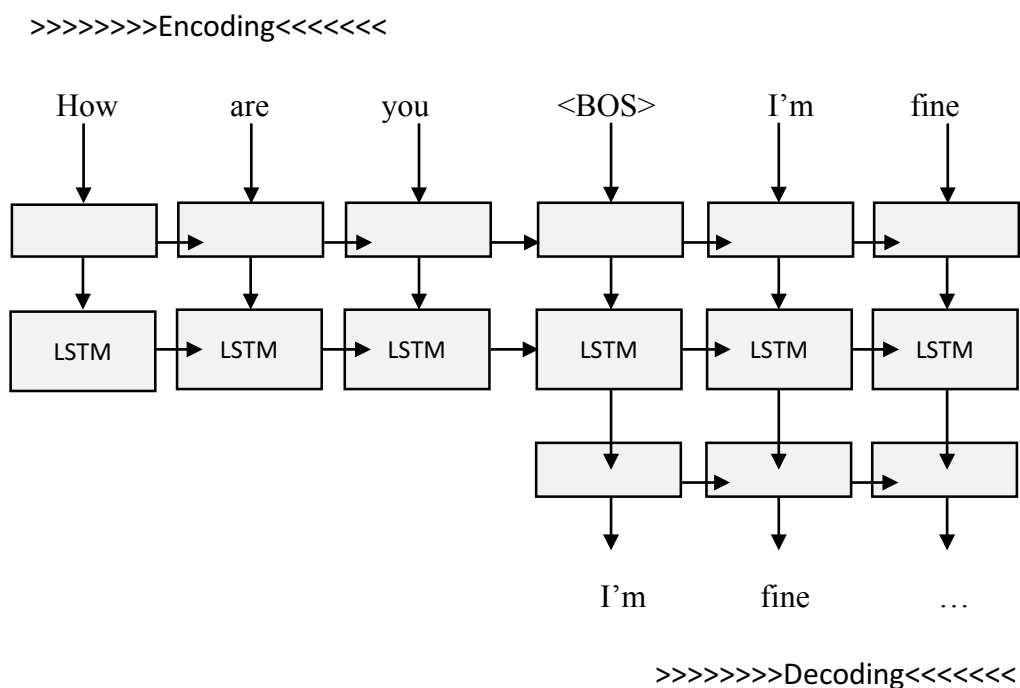
➔ *Source*: Cornell movie dialogues

取前 20000 個高頻率單字，300 維

- *Model*

➔ *Sequence to Sequence*

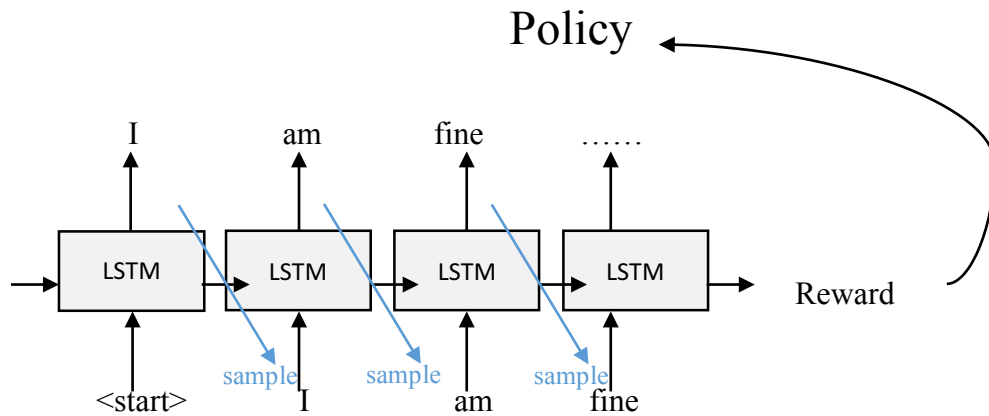
模型最後在 output 字時，我們嘗試了兩種方法，第一種方法是取機率最大的那一個字（Model 1），第二個方法是以 Sample 的方式，取 multinomial 中的詞句。



## ➔ Reinforcement Learning

RL 的模型如下圖所示，Reward Function & Policy:

我們將 BLEU score 當成 RL 的 reward，若預測出的詞語中有 'don't know' 或是 <unk>，reward - 0.1



- **How do you improve your performance (3%)**

1. 我們先 train 了一個 greedy 的 Sequence 2 Sequence model。
2. 嘗試使用 RL 減少 I don't know，但結果完全爛掉。
3. 在 train Sequence2Sequence 的時候對每一步 RNN 都做 Sampling。
4. 只對 RNN 的第一個 step 做 sampling 其他都做 argmax。

- **Experiment settings and observation (3%)**

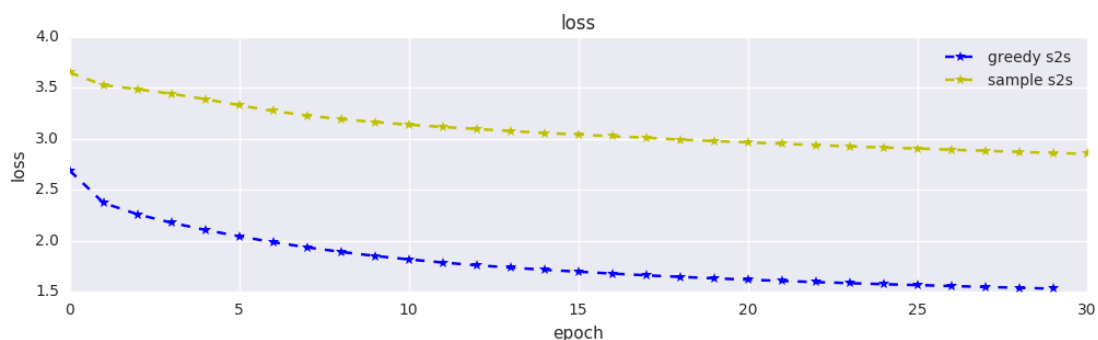
- *Parameters:*

max\_seq\_len = 20 , voca\_size = 20000 , embed\_size = 300 , rnn\_size = 256 , n\_layers = 3 , n\_epoch = 150 , batch\_size = 32 , learning\_rate = 1e-3

- 時間（一次 epoch）：

➔ 約 10 分鐘

- *Loss*



- *Observation:*

1. Greedy 的 Sequence to Sequence 收斂速度很快，結果差強人意，但還是有完整句子。
2. Reinforcement Learning train 不太起來，推測是因為沒有做好 seq2seq 的 sampling 所以在 train RL 的時候，sample 不出好句子。reward 會跟 loss 相乘所以 loss 應該會下降，但因為那句 target 沒有貢獻 loss，所以 network 不會往那個 target 做 gradient
3. 在 train Sequence2Sequence 的時候對每一步 RNN 都做 Sampling，結果不佳會產生類似 I. I. I... You, you, you. 的句子。推測是因為這些詞在每個 step 都很常出現。
4. sampling 之後在用沒有 sample 的 model train 效果還不錯。會有比較多元的句子，也不會奇怪的句子，例如 I. I. I... You, you, you.

- **Team division (1%)**

r05521608 土木碩一 任精瑋: model design  
r04521603 土木碩二 魏仰廷: model analysis  
r05631018 生機碩一 曾秋旺: model trial