

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

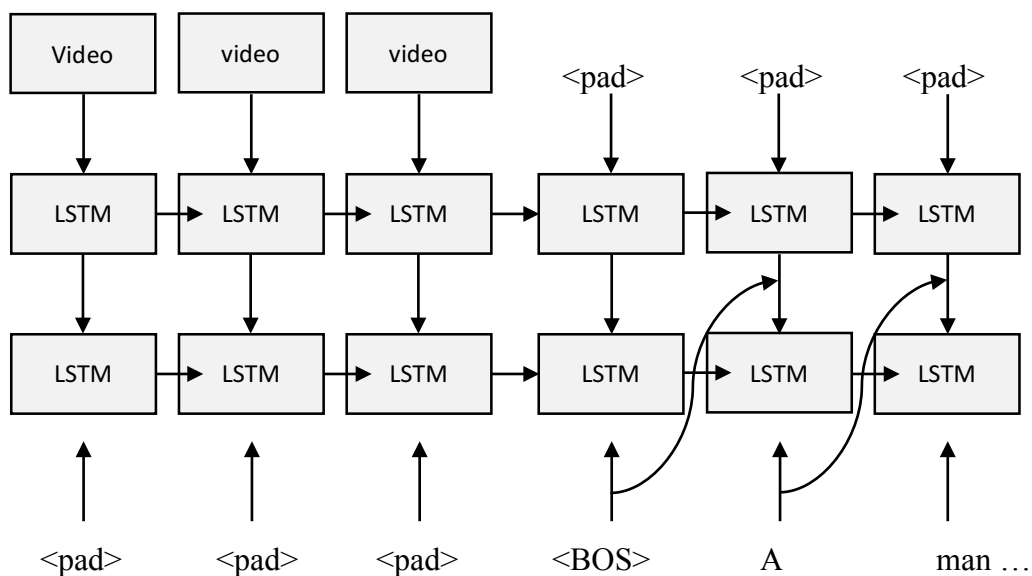
- **Model description (3%)**

- *Word-embedding*

➔ Pre-trained Word-vector GoogleNews-vectors-negative300.bin  
取 3000 個單字，300 維

- *Sequence to sequence*

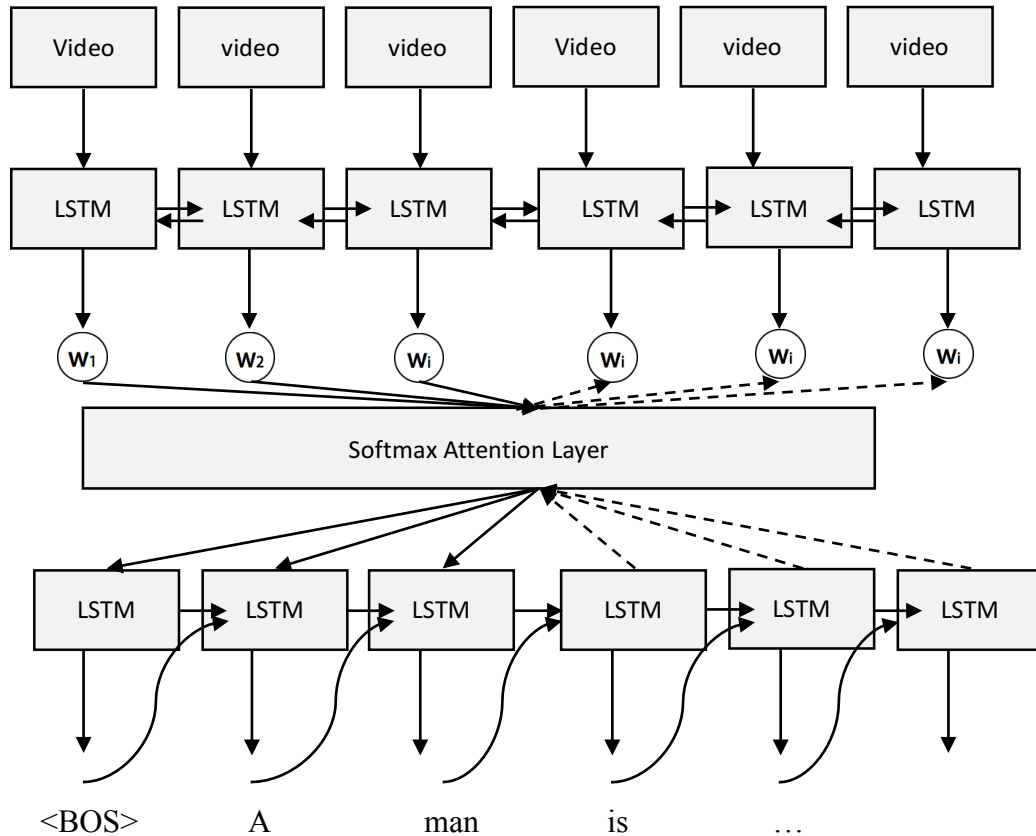
>>>>>>>Encoding<<<<<<<



>>>>>>>Decoding<<<<<<<

○ *Attention model*

>>>>>>>Encoding<<<<<<<



>>>>>>>Decoding<<<<<<<

• **How do you improve your performance (5%)**

1. S2S (model description 中) 由 LSTM 自己 output 出來的 train, 50 個 epoch 可得到 test data BLEU score: 0.27 (50epoch)
2. 把第二層 LSTM 的 output 當成第二層 LSTM 的其中一個 input, 改成 true caption 當成第二層的 input, Loss 加快, 且有完整句子, 但常常文不對題。
3. schedule sampling, 也就是 true caption 跟 lstm 的 output, 二者擇一隨機作為 input, 雖然會有奇怪的語句, 但較能配合影片內容。  
test data BLEU score: 0.29 (50epoch)
4. 加入 mask  
減少 train model 時一直遇到 padding 的機會, loss 會較正確

## 5. attention

一樣會有奇怪的語句，但在辨識影片上能更明顯

test data BLEU score:0.29 (50epoch)

### • Experiment settings and observation (5%)

#### ◦ Parameters:

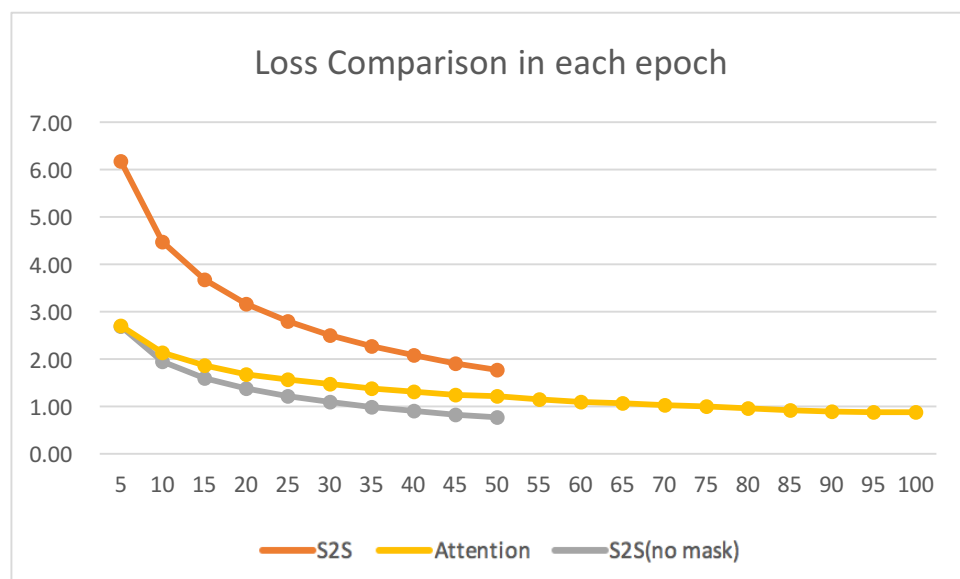
1. S2S (no mask) parameter: vocabulary size = 3000 / hidden dimension = 500 / batch = 32 / learning rate = 0.001 / epoch = 50
2. S2S parameter: vocabulary size = 3000 / hidden dimension = 500 / batch = 32 / learning rate = 0.001 / epoch = 50 / mask length = 20
3. attention parameter: vocabulary size = 5000 / hidden dimension = 500 / batch = 50 / learning rate = 0.0001 / epoch = 500 mask length = 20

#### ◦ 其他實驗：

1. S2S 上下兩個 LSTM 都是 2 layers，但 train 不起來。
2. 增加 schedule sampling: BLEU score: 0.29 (50epoch)
3. dimension 從 256 增加到 500: BLEU score 0.26~0.28

#### ◦ 時間（一次 epoch）：

1. S2S(no mask) 、S2S 約 240-250 秒
2. attention 約 250-300 秒



- **Team division (1%)**

r05521608 土木碩一 任精瑋: model + optimize

r04521603 土木碩二 魏仰廷: preprocess data + report

r05631018 生機碩一 曾秋旺: parameter-experiment