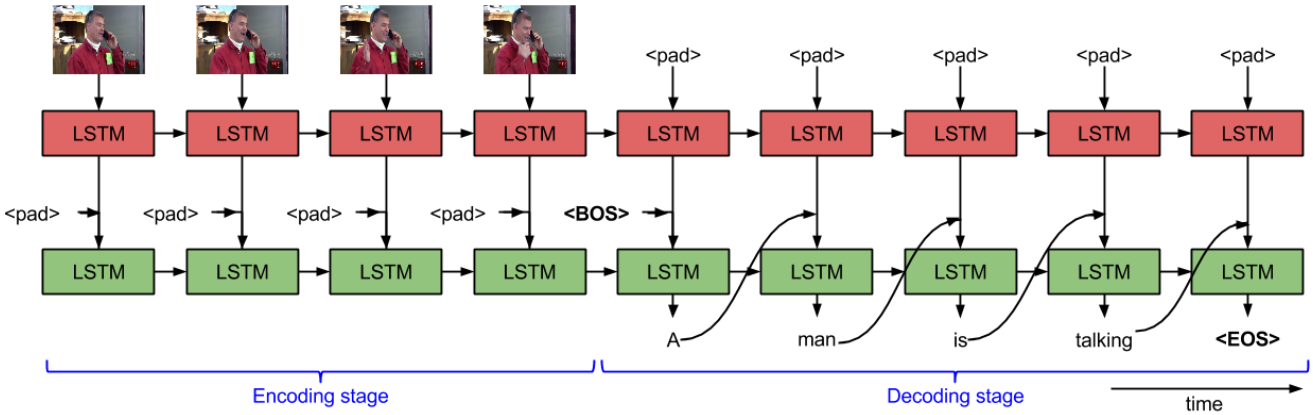
HW2-1 Video Caption

Model description (3%)

> Describe your seq2seq model :

Model Description

- 一般的S2VT，雙層dim512的bidirectional LSTM (with peephole)
- Input: 助教預先處理過得video feature。
- Output: LSTM後直接接Dense(50)後project到6000維的one-hot encoding。
  - 6068 ~= len(dict{word min frequency = 4})
- Training階段為Scheduled Sampling。
- Infering階段為BeamSearch。



Model Hparams

Contents	Parameters
Optimizer	Adam(0.001)
Clipping norm	5.0
Dropout rate	0.5
Batch Size	10
Rnn cell	LSTM with peephole
Max_length	12
Beam Search width	5

How to improve your performance (3%) and why do you use it (1%)

> (e.g. Attention, Schedule Sampling, Beamsearch...) Write down the method that makes you outstanding (1%)

BeamSearch

- 我們使用了BeamSearch，參數(beam width = 10)。
- 使用`tf.contrib.seq2seq.BeamSearchDecoder`。
- 原因目的就同Beamsearch出生的意義一樣，為了抓取機率最高的句子。

Scheduled Sampling

- rate = `min ( 0.5 + 0.7 * now_iteration / max_iteration , 1)`
  - 0.5 遞增向上至 1 後滯留在 1 。(1表示只看自己輸出。)
- 使用`tf.contrib.seq2seq.ScheduledEmbeddingTrainingHelper`。
- 使用原因為一開始生出來的句子很糟 $\rightarrow$ 轉向feat ans， 後段生出來的句子能看，為了使infer一致$\rightarrow$ 轉向feat output。

Dropout

- 在LSTM外面包裝Dropout Wrapper
  - status/input/output 在training階段都是 drop rate = 0.5。
- 因為Ensemble幾乎能避免Overfitting，並且能有效extract新feature。

Gradient Clipping

- 因為RNN的error surface很陡峭，
  因此使用gradient clipping (5)來避免劇烈的變化，
  使得loss噴成Nan，而無法收斂。

Analysis and compare your model without the method. (1%)

- 不使用Gradient Clipping有很大的機會會噴到Nan，因此他的**bleu@1**都是0。
- 將BeamSearch的Beam width調到1 (退化成Greedy)時， **bleu@1**大約是0.58~0.60左右飄動。
  - 比BeamSearch的performance還要差一些些，但沒有很多。
- 不使用Scheduled Sampling:
  - Sampling rate = 0 (退化成只看答案去train) : **bleu@1** ：0.56。
  - Sampling rate = 1 (退化成只看上個時間點去train) : train很久還沒收斂就kill掉了。

Experimental results and settings (1%)

> parameter tuning, schedual sampling ... etc

- train by words -> train by snippets

- 想辦法將caption的句子，以一個片語為單位
  - e.g.: naive(word) : `a|man|is|playing|the|guitar`
    $\rightarrow$ snippet cut: `a man|is plaing|the guitar` 。
- train出來的**bleu@1** 介在 0.58~0.62左右飄動，效果並不是這麼的好。我們猜測有可能的原因是因為，切成snippet後，`the guitar`和`a guitar`會被判成不同的字。這件事並沒什麼大不了，但是有可能會因為`a guitar`出現次數過少，而把`a guitar`判成`<UNK>`，使本來的`the|guitar`能夠對應到`a|guitar`的機會消失了。
- 還有一個有可能做壞的可能是：判成snippet後，原本的word dimension從6000倍增到12000，維度增加，使得model比較難train起來。(因為有些word cluster反而因為我們的snippet cutting而貝拉開了。)

- lemmatization + adjustment: (詞性還原 + 人工句義還原):
  - 先使用**nltk**的包，將原本句子中，相對重要的字句提取，做成句干後丟給model去train。
    - e.g.: `a man is playing the guitar` $\rightarrow$ `man play guitar`
  - predict時，用我們的英文知識+`nltk.pos_tag`(詞性判別)，拼湊句子。
    - e.g.: `man paly guitar` $\overset{N\text{前加入冠詞}}{\longrightarrow}$ `a man play the guitar`
      `a man play the guitar` $\overset{Ving}{\longrightarrow}$ `a man is playing the guitar`
  - **bleu@1**雖然可以很快的就train到0.64~0.65 ，可是由於這個作法有點小髒，因此放棄使用。

Job Assignment

- b05902127 全部的2-1還有Report。
- b05902013 全部的2-2還有Report。