

---

---

# **ADL x MLDS 2017 Fall HW2 - Video Captioning**

2017/10/28  
adlxmls@gmail.com

---

---

# Introduction

## Outline

- **Introduction : Video Caption Generation**
- **Sequence-to-sequence based model : S2VT**
- **Training Tips**
  - Attention
  - Schedule Sampling
  - Beamsearch
- **How to reach the baseline ?**
- **Format & Submission Rules**
  - Dataset
  - Rules & Format

# Introduction

## Video Caption Generation

- Input: A short video
  - Output: The corresponding caption that depicts the video
  - There are several difficulties including:
    - (1) Different attributes of video (object, action )
    - (2) Variable length of I/O
- ( In this task, video features will be provided )

# Introduction

## Video Caption Generation - Example

**Input**

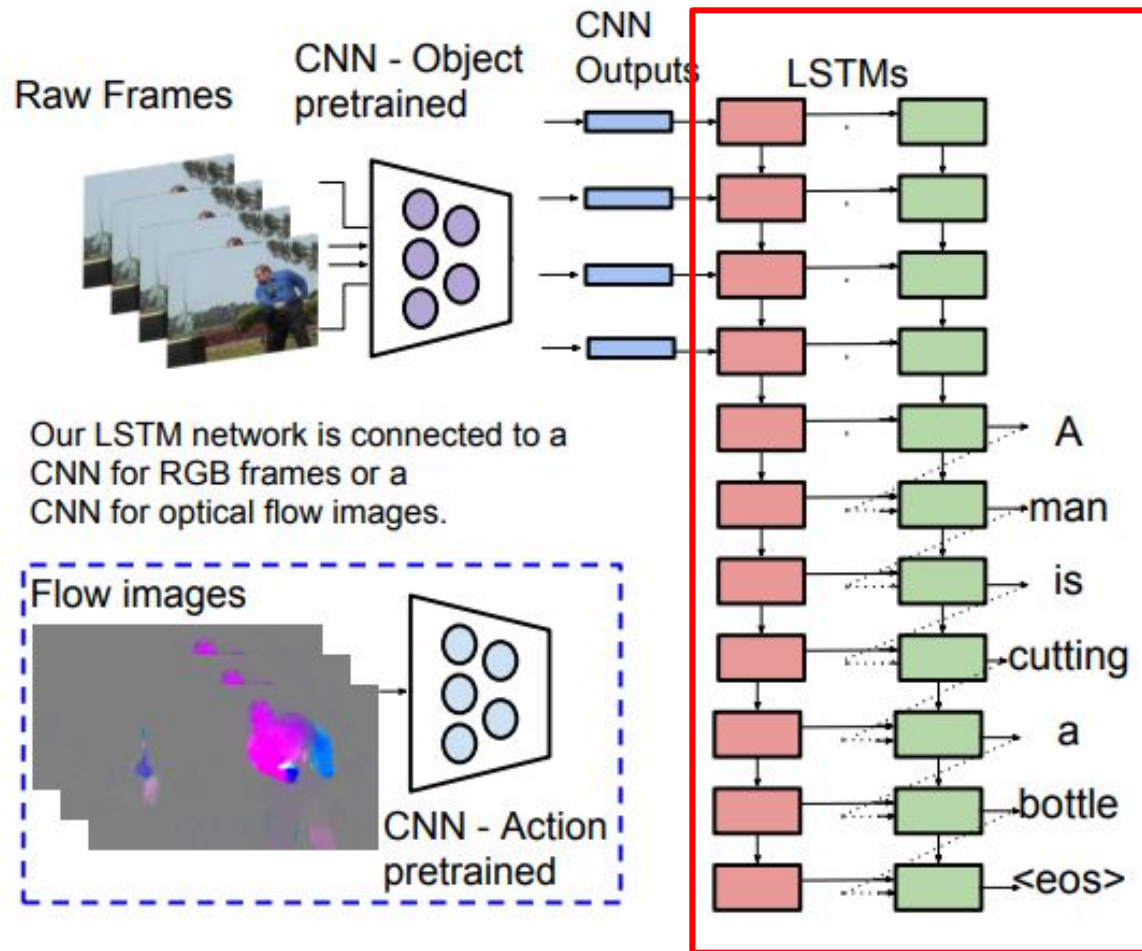


**Output**

a man is playing a  
song on the piano

# S2VT

## Sequence-to-Sequence Based Model: S2VT



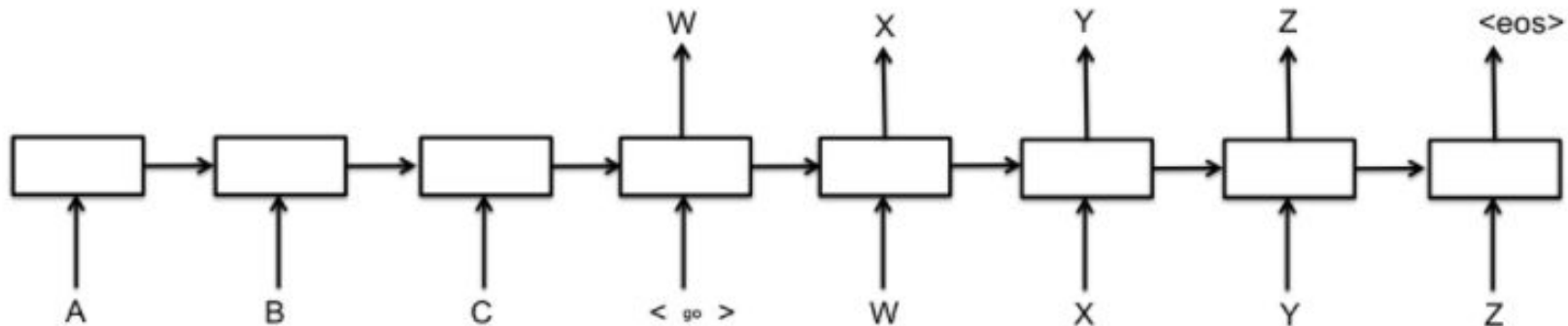
Refer to the following paper for detailed info:

<http://www.cs.utexas.edu/users/ml/papers/venugopalan.iccv15.pdf>

# Seq to Seq

## Two recurrent neural networks (RNNs)

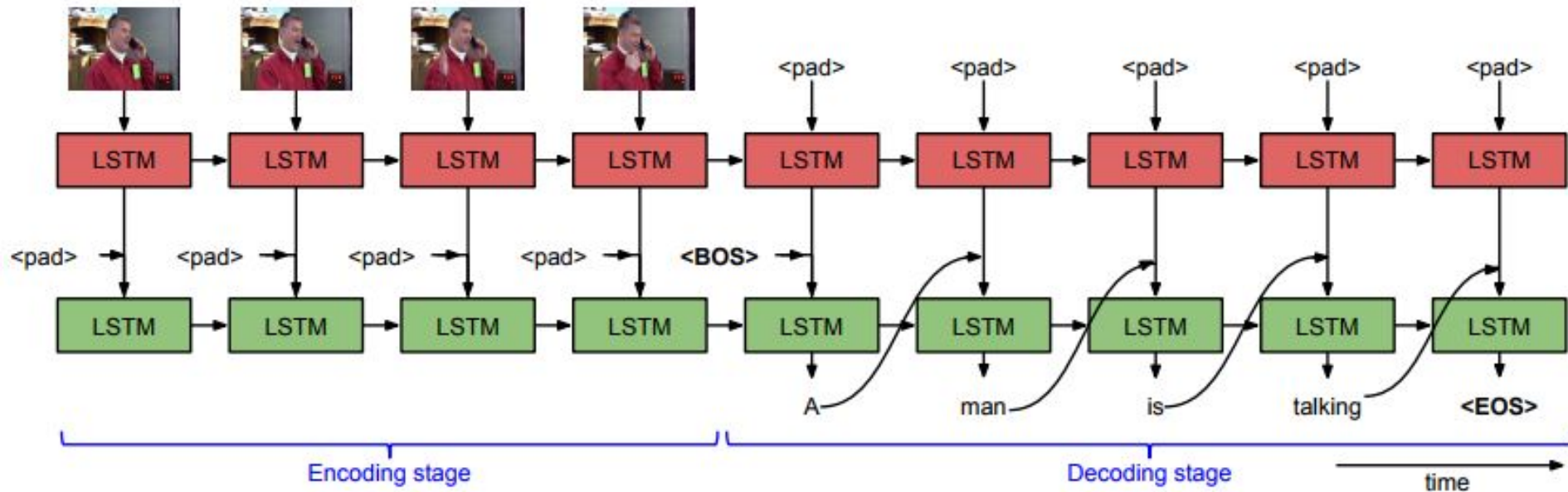
an *encoder* that processes the input  
a *decoder* that generates the output



<https://www.tensorflow.org/tutorials/seq2seq>

## Sequence-to-Sequence Based Model: S2VT

- Two layer LSTM structure



- **2 LSTM stacks**

upper one is for encoding  
bottom one is for decoding.

- **Encoding stage**

CNN features  $\rightarrow$  LSTM1  $\rightarrow$  output1

- **Decoding stage**

output1  $\rightarrow$  LSTM2  $\rightarrow$  word  $y_t$

- **Parameter sharing** between 2 LSTM stacks can help reduce the complexity



## Text Input

- One-hot Vector encoding

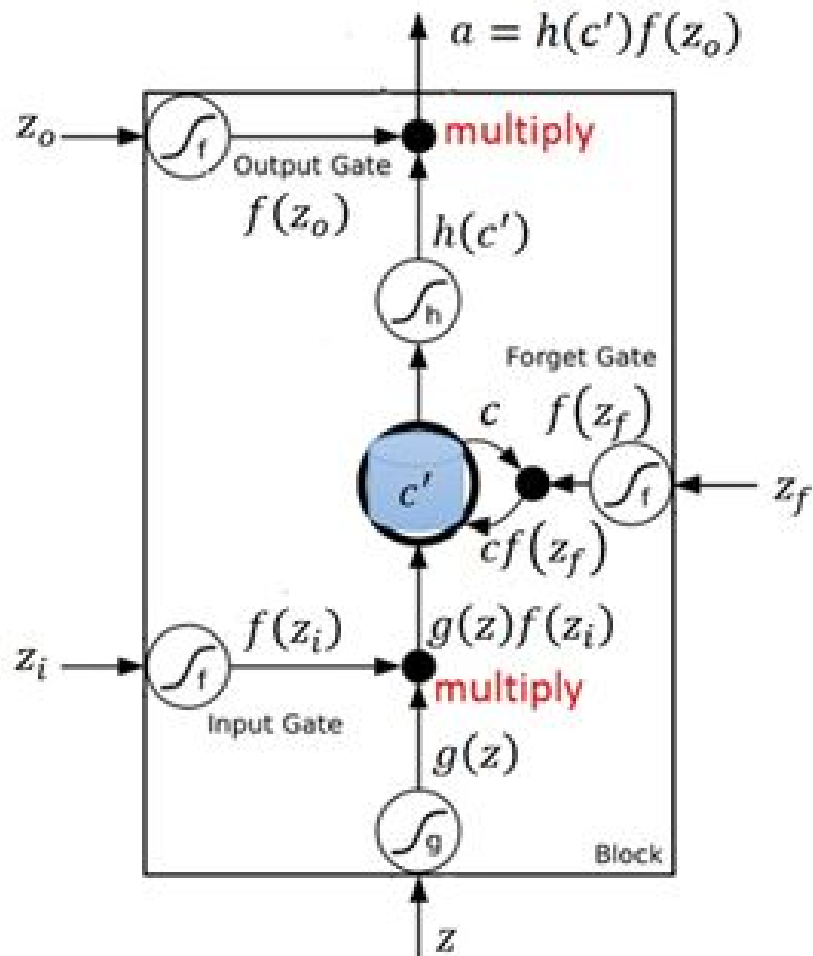
( 1-to-N coding, N is the size of the vocabulary )

- e.g.

neural = [0, 0, 0, ..., 1, 0, 0, ..., 0, 0, 0]

network = [0, 0, 0, ..., 0, 0, 1, ..., 0, 0, 0]

## - LSTM unit

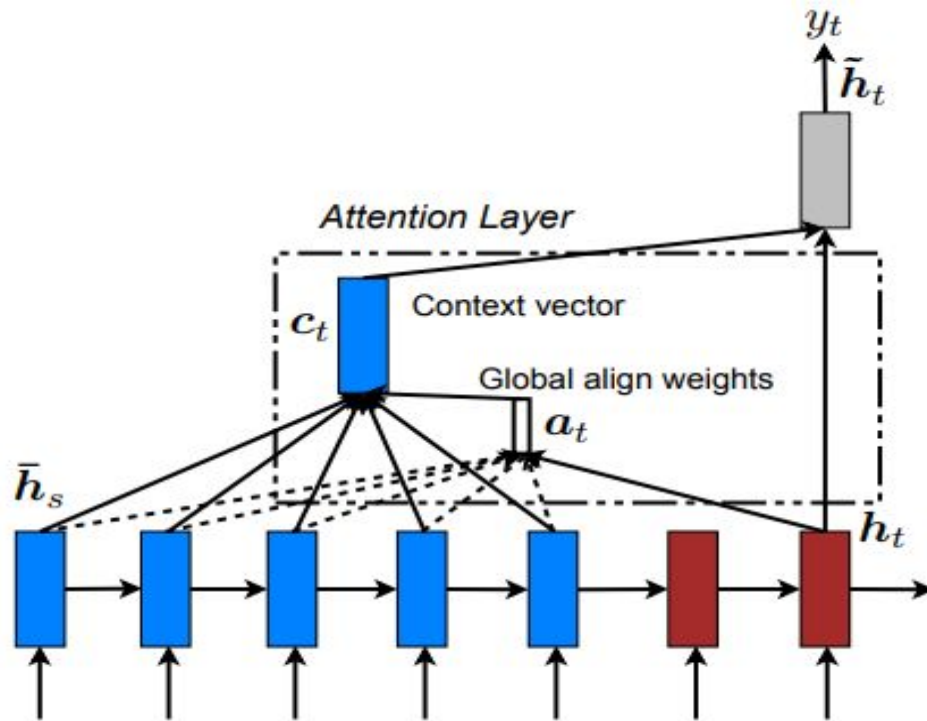


$$\begin{aligned}
 i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\
 f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\
 o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \\
 g_t &= \phi(W_{xg}x_t + W_{hg}h_{t-1} + b_g) \\
 c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
 h_t &= o_t \odot \phi(c_t)
 \end{aligned}$$

# Training Tips

- **Attention on encoder hidden states :**

Allow model to peek at different sections of inputs at each decoding time step

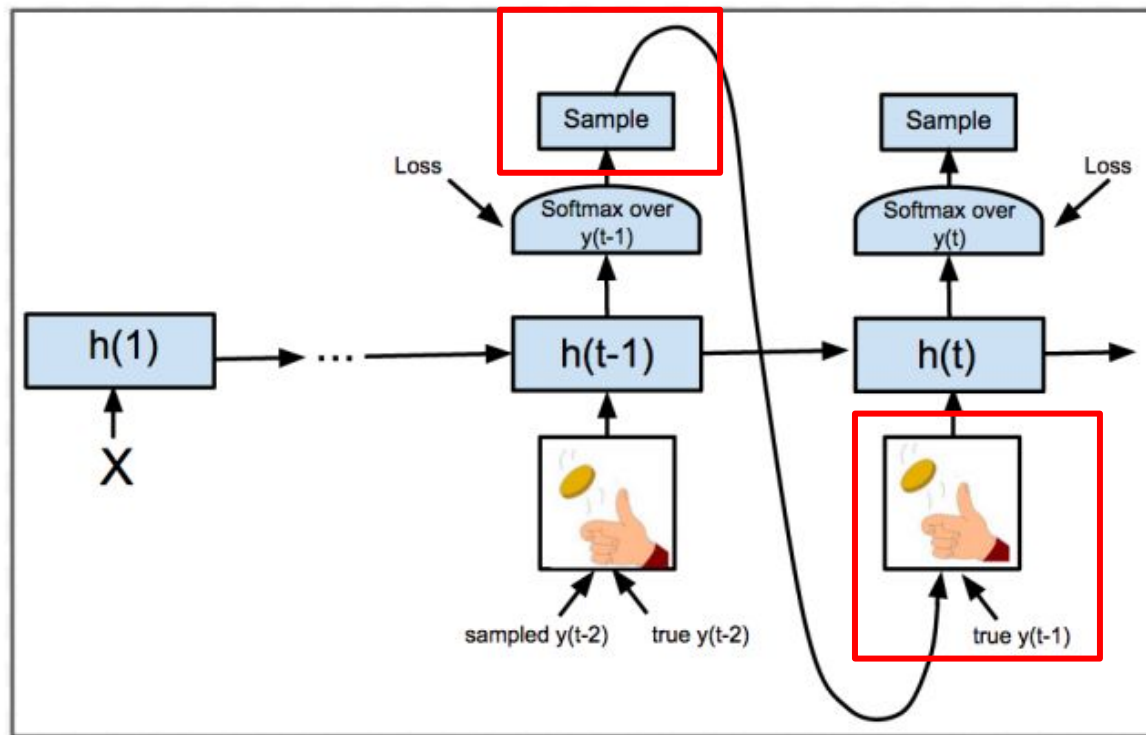


# Training Tips

- **Schedule Sampling :**

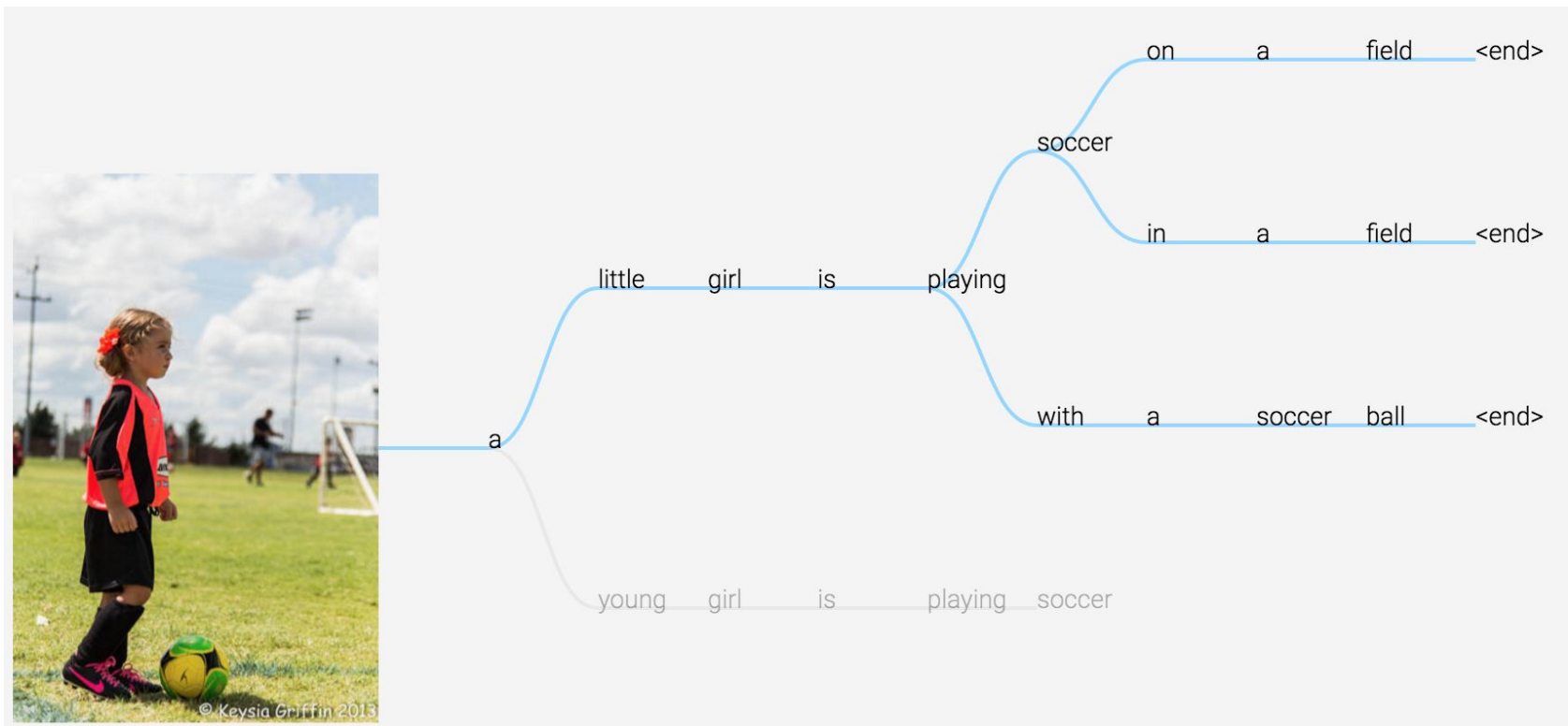
To solve “exposure bias” problem,

When training, we feed (groundtruth) or (last time step’s output) as input at odds



# Training Tips

## Beamsearch : keep a fixed number of paths



# Training Tips

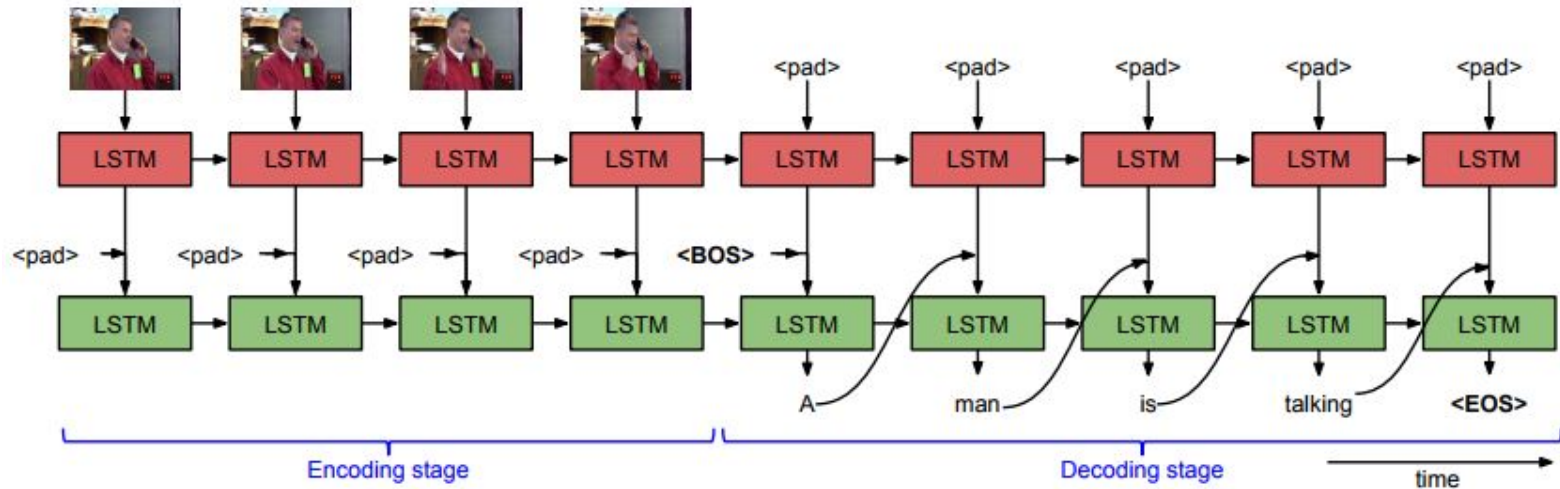
## Beamsearch

Demo

<http://dbs.cloudcv.org/captioning>

# Baseline

## S2VT model



- Training Epoch = 200
- LSTM dimension = 256
- Learning rate = 0.001
- vocab size = 3000
- AdamOptimizer
- Training time = 72 mins, by using 960 TX

**Baseline BLEU@1= 0.25 (Captions Avg.)**

## Evaluation - BLEU@1

Precision = correct words / candidate length

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

where  $c$  = candidate length,  $r$  = reference length

$$\text{BLEU@1} = \text{BP} * \text{Precision}$$



# Baseline

## Evaluation - BLEU@1



Ground Truth : *a man is mowing a lawn*

Prediction : *a man is riding a man on a woman is riding a motorcycle*

BLEU :  $1 * 4/13 = 0.308$

# MSVD Dataset

# Data

- **MSVD**
  - 1450 videos for training
  - 100 videos for testing
- **peer review**
  - several videos (more than 10)

- [Download Link](#)

```
.
├── bleu_eval.py
├── peer_review
│   ├── feat
│   └── video
├── peer_review_id.txt
├── sample_output_peer_review.txt
├── sample_output_testset.txt
├── testing_data
│   ├── feat
│   └── video
├── testing_id.txt
├── testing_label.json
├── training_data
│   ├── feat
│   └── video
└── training_label.json
```

**Kaggle**

**There is NO kaggle competition!**

# HW Rules

# Peer review

- Captions of video are hard to evaluate by machine.
- Students of ADLxMLDS should help review captions produced by other's model.

# Special Mission

- To encourage the students to start the homework as early as possible.
- Upload captions of some videos in testing set.
- Please upload your model to your github , and write `hw2_special.sh` to produce corresponding captions.
  - Usage : `hw2_special.sh [the data directory]`  
`[output_file]`
  - The format of output file should be the same as `sample_output_testset.txt`, but only consist of five lines.
- Period : 2017/11/6 0:00 - 2017/11/12 23:59
- [Link](#)



# HW Rules

- Please write shell script to run your code.
- There should be `hw2_seq2seq.sh`
- Please follow the script usage below:
  - `./hw2_seq2seq.sh $1 $2 $3`
  - \$1: the data directory,
  - \$2: test data output filename
  - \$3: peer review output filename
- Ex: `./hw2_seq2seq.sh myData/  
sample_output_testset.txt  
sample_output_peer_review.txt`

# HW Rules

- Please implement one **seq-to-seq model(or it's variant)** to fulfill the task
- Please also implement **attention** to fulfill the task
- Please use python with version  $\geq 3.5$
- Extra dataset is allowed to use.
- **Allowed packages include :**
  - PyTorch v0.2.0
  - Tensorflow r1.3 ( Tensor layer forbidden )  
<https://tensorlayer.readthedocs.io/en/latest/> (x)
  - Keras 2.0.7 ( Tensorflow backend only )
  - MXNet 0.11.0
  - CNTK 2.2
  - Numpy
  - Pandas
  - Python Standard Lib

# HW Rules

**If you use other packages, please ask for permission first !!!**

# Grading

# Grading Policy

- I. Baseline (4%)
- II. Peer review (4%)
- III. Special mission (2%)
- IV. Review other's results (2%)
- V. Report (6%)
- VI. Notice

# Grading Policy -- Baseline(4%)&Peer review(4%)

- Pass the baseline (4%)
  - Average bleu score should  $\geq 0.25$
- Peer review (4%) :For those passing the baseline, your score will be linearly graded, rounded to the 2nd decimal place.
  - Ex: if 100 people pass the baseline, you will get 3 points if you're at 25th place.
- We will run your code to make sure your performance passes the baseline.

## Grading Policy -- Review other's result(2%)

- Get 2% if you review other's result
- Rules will be announced after TAs successfully produce all the students' output.

## Grading Policy -- Special Mission (2%)

- Get 2% if you have submitted the form and upload models to your github.



# Grading Policy -- Report(6%)

- Do not exceed 4 pages and written in Chinese.
- Model description (2%)
  - Describe your seq2seq model
- Attention mechanism(2%)
  - How do you implement attention mechanism? (1%)
  - Compare and analyze the results of models with and without attention mechanism. (1%)
- How to improve your performance (1%)
  - Write down the method that makes you outstanding
  - Describe the model or technique (0.5%)
  - Why do you use it (0.5%)
- Experimental results and settings (1%)
  - parameter tuning, schedual sampling ... etc
- README : please specify library and the corresponding version in README

## Grading Policy -- Bonus(2%)

- TAs will select about 5 persons, according to both **creativity** and **performance** (top 10%, by the score of peer review) for introducing your model during the class
- If you are chosen, **you have to present** in order to get the bonus.

# Grading Policy -- Notice

- Please fill the [late submission form](#) first **only if you will submit HW late**
- Please push your code before you fill the form
- **There will be 25% penalty per day for late submission,** so you get 0% after four days
- You get 0% if the required script has bug.
  - If the error is due to the format issue, please come to fix the bug at the announced time, or you will get 10% penalty afterwards.

# Submission Rules

# Submission Rules

- Create hw2 directory under ADLxMLDS2017
- Under hw2, there should be:
  - report.pdf
  - **your\_seq2seq\_model**
  - hw2\_seq2seq.sh // should run your RNN model
  - model\_seq2seq.py and other necessary files
  - \*In model\_seq2seq.py should include your training codes.
- Please do not upload any dataset to Github (include external dataset)
- If your model are too big for github, upload to a cloud space and write it in your script to download the model
- Your script should be done within 10 mins (include preprocessing) excluding model downloading

# Deadline

Github code & report deadline: **2017/11/19 23:59 (GMT+8)**

# FAQ

# Q1: 使用的lib 限制

- **Allowed packages include :**

- PyTorch v0.2.0
- Tensorflow r1.3 ( Tensor layer forbidden )  
<https://tensorlayer.readthedocs.io/en/latest/> (x)
- Keras 2.0.7 ( Tensorflow backend only )
- MXNet 0.11.0
- CNTK 2.2
- Numpy
- Pandas
- Python Standard Lib

**If you use other packages, please ask for permission first !!!**



**Q2: 請問助教會跑training的程式嗎？**

A:

不會。我們所規定的十分鐘只包含testing。除非我們認為有必要就會請你們來跑training的code。

## Q3: 有推薦上傳model的平台嗎？

A:

dropbox, google drive都是大家常用的平台。不過推薦大家可以使用gitlab, 操作方法與github類似, 但是可以上傳大容量的檔案。

**Q4: peer review 底下的資料夾怎麼是空的？**

A：

屆時同學上傳model 後，在助教電腦上的資料夾才會有影片 及相對應得feature

## Q5: test set 的答案怎麼一起給了？

A：

因為沒有Kaggle, 方便大家validation 和測準確率, 因此也給大家testset 的答案。

**Q6: attention model助教也會跑嗎？**

A：

不會的，這部分請同學自己實驗並寫在report裡，同學只需要交自己最好的model。

## Q7: 助教會使用bleu\_eval.py 來測是否有過baseline嗎？

A:

bleu\_eval.py僅給同學參考用，在改的時候可能會再做修改，但計算average的方式會一樣的，請同學不要用程式的漏洞.....

## Q8: peer review 的 video 有幾個呢？

A：

屆時助教在跑model時，會將所有的影片標示在 peer\_review\_id.txt（助教的data資料夾底下），請同學利用這個檔案讀所有的video。

Eg：1.avi

feature 位置：peer\_review/feat/1.avi.npy

video 位置：peer\_review/video/1.avi

**Q9: data 裡的feature是怎麼抽的呢？**

A:

pretrain在ILSVRC的VGG19。

80\*4096維的feature, 是指每個影片抽80個frame, 每個frame有4096維feature。



## Q10: Average bleu score 是怎麼算的呢？

A:

對於每個影片，你的答案會對他的所有的字幕都算一次bleu score，平均後得到關於這支影片的分數。

將所有影片的分數取平均後，就是你的總bleu score。

詳細演算法請見 `bleu_eval.py`

# FAQ

- 有問題請利用TA hours、信箱或FB 社團，**請不要FB私訊助教！！**
- If you have other questions,
  - please contact TAs via [adlxmlids@gmail.com](mailto:adlxmlids@gmail.com)
  - post your questions on [facebook group](#)
  - go to TA office hours
    - 李佳軒 Thu 16:00-17:30 (電二531)
    - 蔡哲平 Wed 10:30-12:00 (電二531) (11/8(三) 助教要期中考, 請假)
    - 林圓方 Fri 9:30-11:00 (博理527)