MLDS 2017 Spring
HW2 - Seq2seq & Attention

mldsntu2017@gmail.com

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

Video Caption Generation

- Input: A short video
- Output: The corresponding caption that depicts the video

(There are some demos later!!)

- There are several difficulties including:

(1) Variable length of I/O

(2) Distinct attributes of videos

(In this task, CNN features will be provided!)

Video Caption Generation - Example

Input:

Output:

a man is playing a song on the piano .

Sequence-to-Sequence Based Model: S2VT

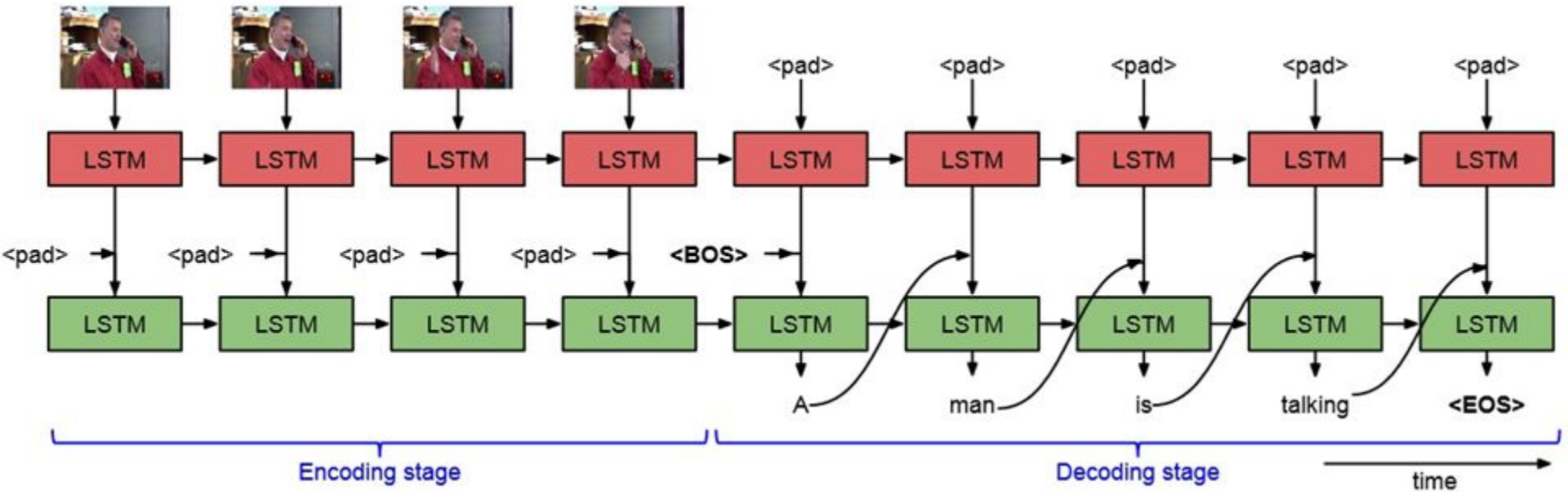
- You can refer to the following paper for detailed info:

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

- The original task includes CNN and LSTM, but you can only focus on the structure of LSTM, since the CNN features will be provided.

Sequence-to-Sequence Based Model: S2VT

- The LSTM structure



Sequence-to-Sequence Based Model: S2VT

- There are 2 LSTM stacks, the upper one is for encoding, and the bottom one is for decoding.
- Encoding stage:

  features→LSTM1→state h_t

- Decoding stage:

  state h_t→LSTM2→word y_t

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

(Remember to use variable scope in Tensorflow!)

Sequence-to-Sequence Based Model: S2VT

Text Input

- One-hot Vector encoding

(a.k.a. 1-to-N coding, where N is the size of the vocabulary)

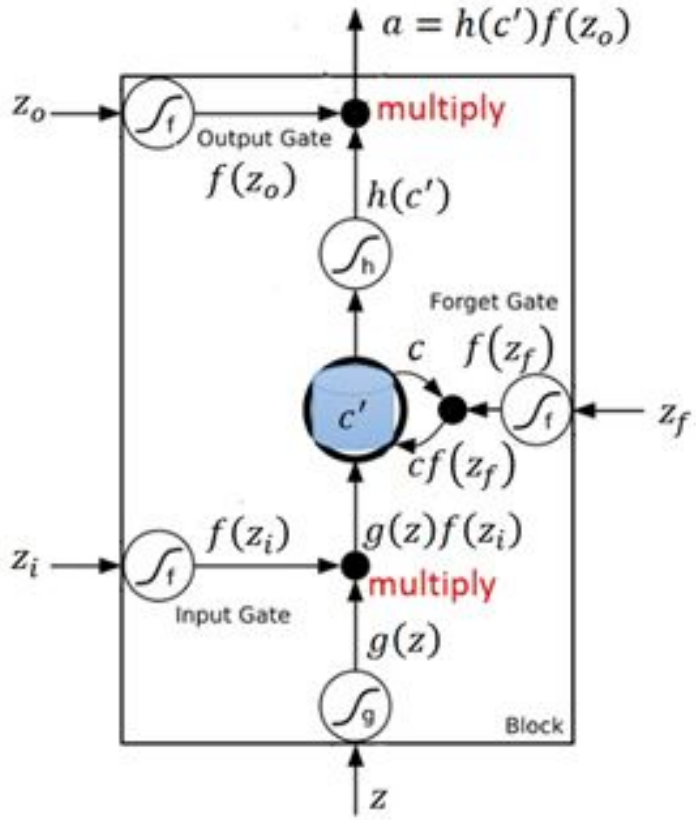
- e.g.

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

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

Sequence-to-Sequence Based Model: S2VT

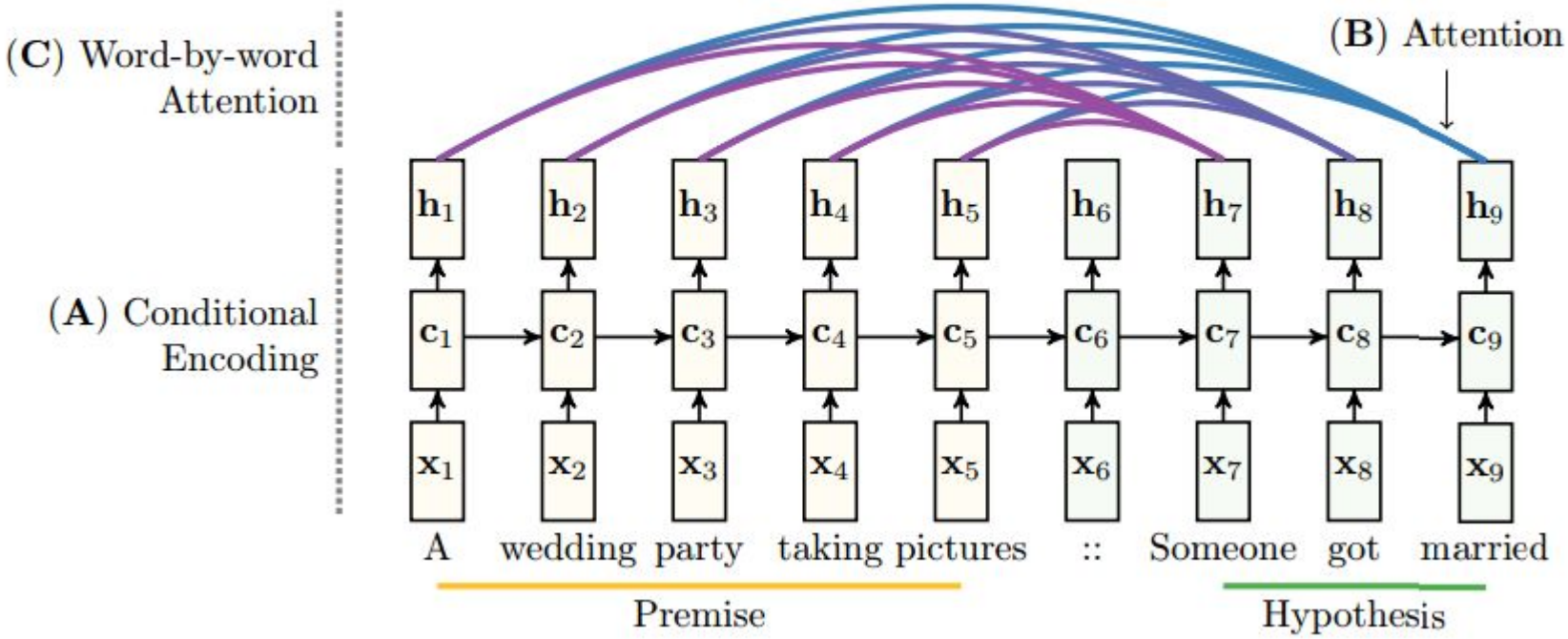
- LSTM unit



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

Training Tips

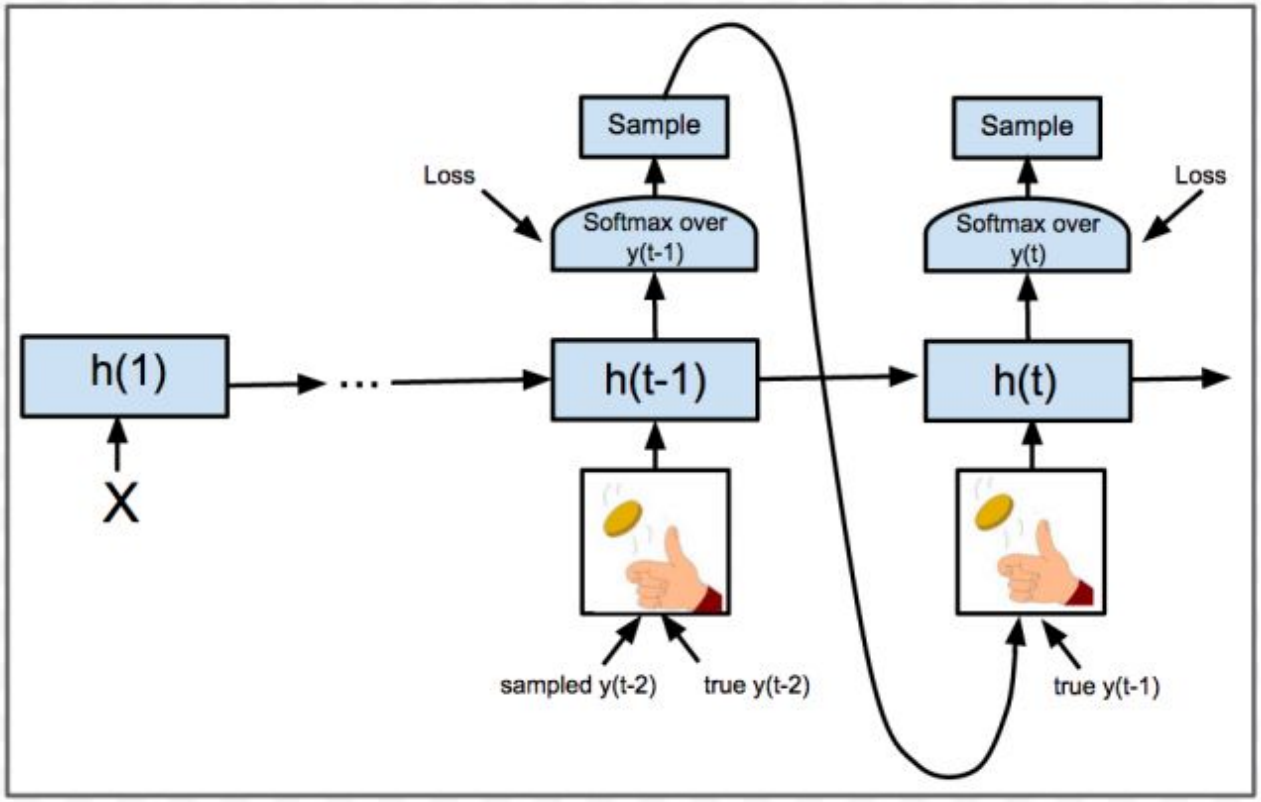
Attention Model



https://arxiv.org/pdf/1509.06664.pdf
http://www.aclweb.org/anthology/D15-1166

Training Tips

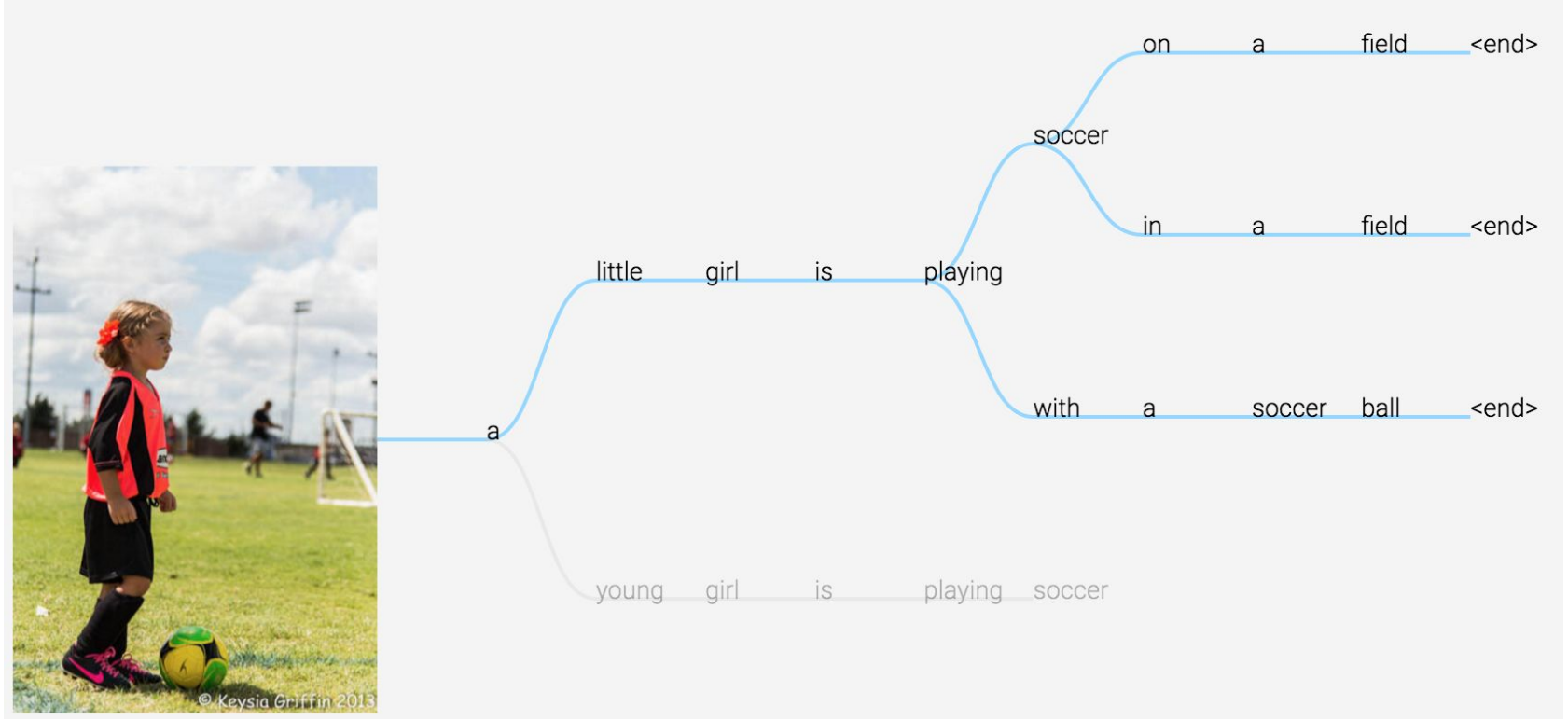
Schedule Sampling



https://arxiv.org/abs/1506.03099

Training Tips

Beamsearch



Training Tips

Beamsearch

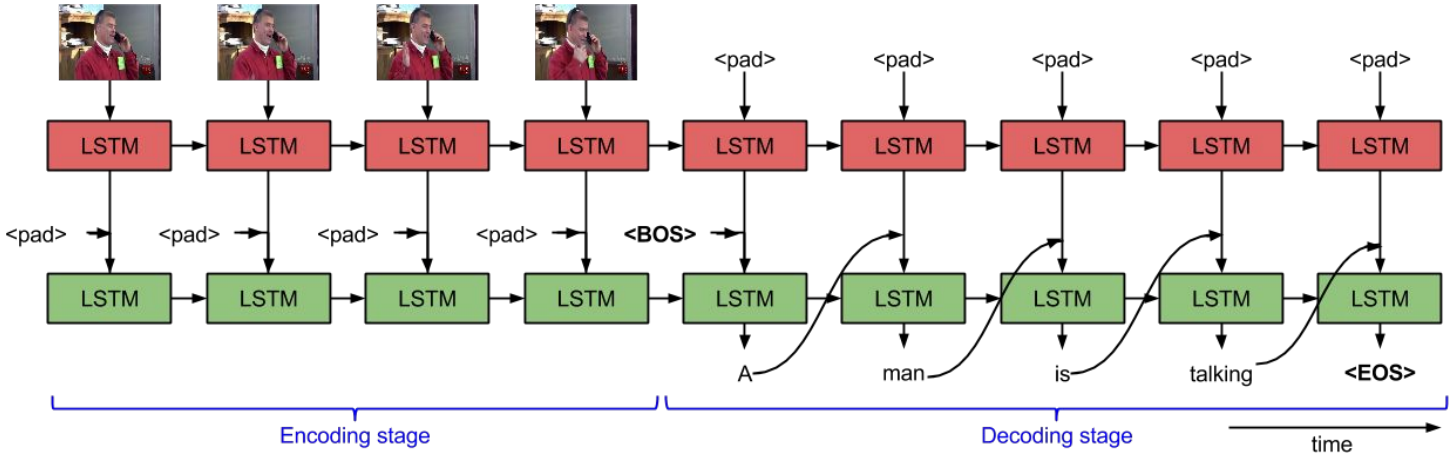
Demo :

http://dbs.cloudcv.org/captioning

Normal beamsearch : (Groups=1)

How to reach the baseline?

S2VT model :



- Training Epoch = 2000
- 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.)

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

Evaluation - BLEU@1

Precision = correct words / candidate length

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

where c = candidate length, r = reference length

BLEU@1 = BP * Precision

http://www1.cs.columbia.edu/nlp/sgd/bleu.pdf

How to reach the baseline?

Demo :

Ground Truth : ***a person is slicing a tomato***
Prediction : ***a man is slicing a tomato pieces***
BLEU@1 : 1 * 5/7 = 0.714

How to reach the baseline?

Demo :

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

Homework 2 package

MSVD Dataset :

- training_data/ : 1450 films's frame feature
- testing_data/ : 50 public testing film's frame feature
  - *Dimension of each frame = 80*4096*
- training_label.json : 1450 films's id and corresponding captions
- testing_public_label.json : 50 public testing films's id and captions
- testing_id.txt : the example file will input to testing script

Evaluation tool :

- bleu_eval.py
  - usage : python bleu_eval.py <candidate_sentence> <reference_sentence>

Download link : http://speech.ee.ntu.edu.tw/~yangchiyi/MLDS_hw2/MLDS_hw2_data.tar.gz

Submission on Github

- Only **Python** with **Tensorflow r1.0** (TAs will run your code in Tensorflow-only environment).
- Deadline: **4/27(Thu.) 23:59:59 (UTC+8)**
- **MLDS2017/hw2** should contain all the things you use.
  Ex. **run.sh**, model, **report.pdf**, etc.
  If some files are too big, upload to your cloud and write a script to download them. Remember to call the download script in run.sh.
- run.sh must output in **10 minutes**.
- Usage:
  bash run.sh [testing id file] [feature path]

Output format

- Command : ‘ bash run.sh <testing_list.txt> <feature path> ‘
  - Your captions of videos should be saved as a json file named ‘ **output.json** ’
  - Format of *output.json* : list of dictionary with two keys ‘caption’ and ‘id’
  - Your caption should be separated by the space among words and no need to output ‘<EOS>’ or ‘ . ‘
  - *output.json* should be placed at the **same path** as *run.sh*

```
[ {"caption": "I am hungry", "id": "xxxxxx"},
  {"caption": "MLDS hw2", "id": "oooooo"},
  {"caption": "I wanna sleep", "id": "------"} ]
```

- Every steps including downloading your model and testing process should be written in *run.sh*

Submission on Github

- You must create two branches **master** and **best**.
- **master** stores the model by **using only S2S with attention** (external data is not allowed)
- **best** stores your best model. If your best model is the same as your baseline model, just copy all files in master to this branch.
- The format is mentioned in the previous slide.
- You have to specify the best performance you achieved by using only S2S with attention in the experiment part of your report.

Grading Policy (30%)

- Format 2%
- Code 4% (S2S 2% + attention 2%)
- Baseline 4% (public 2% + private 2%)
- Report 15%
- 各組互評 5%
- Bonus
  - 互評前三名 (5% 3% 1%)
    - You have to present how you beat other teams to get bonus.
  - 限時任務 1%

What report should cover?

- Environment (1%)
  - Ex. OS, CPU, GPU, Memory, libraries you used and version, etc.
- Model description (3%)
- How do you improve your performance (5%)
- Experiment settings and observation (5%)
- Team division (1%)
- No more than 4 pages
- Please written in Chinese (unless you don’t know how to type Chinese)

Other Policy

- Late policy: 30% off per day late afterwards.
  [Delay form will be announced afterwards]
- No plagiarism is allowed.
- Use the given data only, except for pre-trained word embedding (you should specify the source in your report).