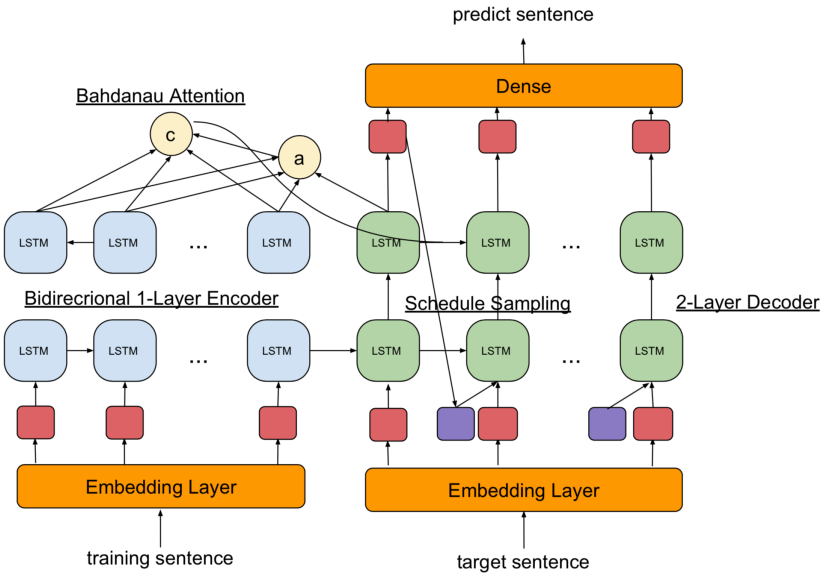
# MLDS Homework 2-2 Report

b05902127 劉俊緯 b05902013 吳宗翰

## Model Description




如圖：

- Embedding Layer: word dim (約50000) -> 1000
- Encoder: 一層Bi-directional LSTM cell，hidden units:256 (use_peephole=True)
- Decoder: 雙層LSTM cell，hidden units: 256 (use_peephole=True)
- Use Normalize Bahdanau Attention Method
- Use Schedule Sampling when training

## How to improve your performance

### Schedule Sampling

1. Training trick 在作業中，我們使用tensorflow的 `ScheduledEmbeddingTrainingHelper` 其中，sample probability：$max(0.3, \frac{\text{curr\_step} \times 0.7}{\text{total\_iteration}})$
2. Why should we choose this
    - 由於在這個task中，字典裡面中文字很多，而且本身的dataset又比較糟(句子之間可能本來correlation就比較低)，本來就比較難學習，因此特別怕上課所提到「一步錯，步步錯」的狀況。
    - 另外選擇$0.3$為threadhold的原因是，經過實驗發現當sample probability太大的時候，loss又升回4以上了，所以作罷；$0.7$倍的linear上升則是實驗出來好像比較好
3. Compare with the model without the method
    - 在沒有Schedule Sampling的時候，由於我的batch size只有開到32，因此在每個step他都能fit的不錯(loss蠻低的)，不過就連在training set上面decode的結果都蠻差的(就別說validation set或testing set了)
    - 承上，所謂表現結果不好就是發現他更常出現重複說同樣的話的狀況

### Attention Method

1. Training Trick
   使用 Normalize Bahdanau Attention Method (參考tensotflow NMT教程)
2. Why should we choose this
   在這個task中，一開始發現在沒有Attention的時候超級難收斂(至少跑到10000個iteration回答的東西還是很怪)，因此就考慮了Attention，希望他可以比較早的回答出make sense一點的句子
3. Compare with the model without the method
   - 承上，在沒有Attention的時候，loss下降的很慢，也比較難回答出make sense的句子；然而加上Attention以後，收斂的速度快很多
   - 另外一個Issue就是在我這個有Attention的model下，回答出的句子很容易會有「我、你、他」等等我不希望的重複主格，因此如果還要提升performance的話，感覺要如投影片所說的在Attention的地方加上regularization項 (這裡只是猜想，因為時間關係這次作業沒時間驗證ＱＱ)

## Experimental results and setting

### Data Preprocessing

1. training大致就是使用第$i$行去predict第$i+1$行
2. 有人工的清掉局部奇怪字符，並且拿掉過多的句子
3. min_word_frequency = 10, dictionary大約五萬

### Setting

1. Loss function: tensorflow sequence loss (其實就是cross entropy)
2. Optimizer:
   - Adam, learning rate: 0.001
   - Gradient clipping (5)，避免loss噴到nan
3. iteration: 20,0000
4. batch size: 前半是32，後半是64

### Network Structure

1. 架構如上model discription
2. Attention method + Schedule Sampling如上所述
3. initializer:
   - embedding layer: xavier
   - LSTM cell: glorot_normal
4. Dropout: 0.1 (兩層RNN之間)

### Heuristic

1. predict的時候會自動去除連續的單詞(應該只有一點點)
2. predict的時候，送入Query問句給Schedule Sampling，發現在perplexity就變差了，不過correlation的performance有變好(估計是train爛了QQ)
3. predict的時候在某些condition下(人工判斷句子爛成一定程度)，會有一定機率回答原句(人眼看

大概有一成多一點的資料)

**備註**

1. 在沒有做heuristic的時候，perplexity可以過，不過correlation大概0.35多一點
2. 在做完heuristic後，perplexity差不多壓線(0.96)，correlation也提升到0.53左右，人眼看一下句子好像沒有很好，日常句子拿下去實驗會發現句子架構還好，但相關度不是很高
3. 分工表：2-2為吳宗翰負責