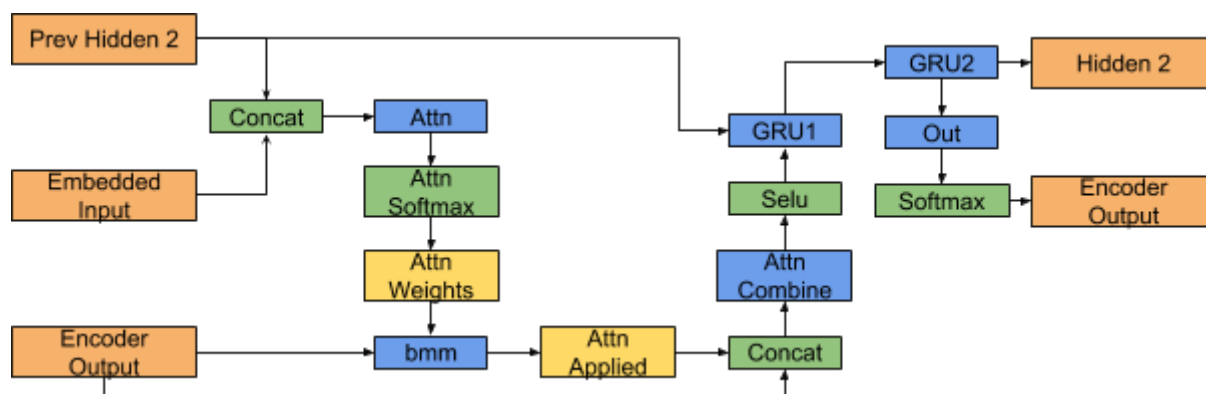


## MLDS Homework 2-2 Report

這一題實做的model為seq2seq的attention版本，attention只採用第二層GRU的hidden，每次隨機從dataset取100句，每個batch更新一次，詳細的參數設定以及Attention架構圖為以下：

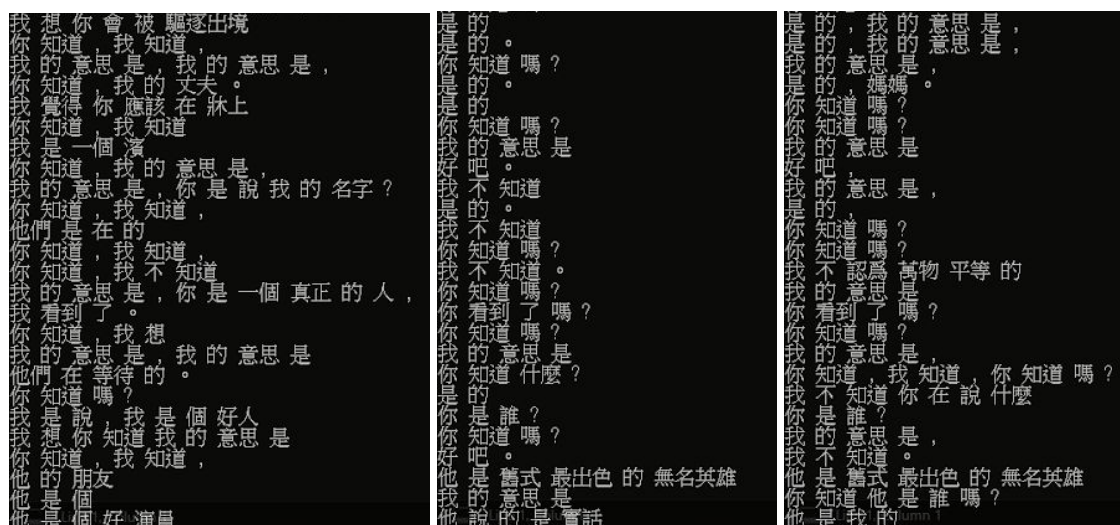
- dictionary size = 71000/6800
- max length = 15
- batch size = 100
- encoder/decoder hidden size = 800
- encoder/decoder gru = 2 layers
- encoder/decoder optimizer = Adam
- learning rate = 0.001
- epochs = 100000~150000



在training中我們比較了兩種變數對於model的影響，分別為dictionary size以及teacher forcing rate。dictionary size比較分別移除frequency<5以及frequency<150的詞，分別得到約71000字以及6800字。teacher forcing rate則比較constant值1.0以及0.5。另外在testing過程中分別比較有無beam search的差異。在有beam search的testing中我們發現預測出短句的機率較高(原因是因為短句後面皆為{PAD}的機率遠高於其他詞)，因此我們加了length normalization，將{EOS}後面預測出來的字的機率不計算，只以預測句中到{EOS}的長度除{EOS}前的機率log相加。結果為下表：

	Beam Search	Length Normalization	Perplexity(<100)	Correlation(>45)
largeDic highTF	No	/	8.886030	0.55451
	Yes	No	8.224528	0.51542
		Yes	7.736847	0.52743
smallDic highTF	No	/	10.113701	0.53234
	Yes	No	8.191511	0.49922
		Yes	7.510231	0.50701
LargeDic lowTF	No	/	59.103105	0.42009

	Yes	No	31.204263	0.47535
		Yes	34.374392	0.44825
smallDic lowTF	No	/	48.116261	0.45997
	Yes	No	23.086570	0.50935
		Yes	26.151728	0.47528



## README:

### Dependencies :

- pytorch 0.3.1
- numpy 1.14.1

### training與testing指令 :

- training: `python3 2-2/train_2gru_attn.py [clr_conversation.txt] [word_min_freq] [teacher forcing ratio]`
- testing: `python3 2-2/test_2gru_attn.py [word_dict.txt] [encoder model] [decoder model] [test_input path] [test_output path] [is_beam_search] [is_beam_normalize]`

### best result 指令 :

- training: `python3 train_2gru_attn.py 5 1.0 => output: encoder.pt decoder.pt word_dict.txt`
- testing: `python3 test_2gru_attn.py word_dict.txt encoder.pt decoder.pt test_input.txt test_output.txt 0 0 => output: test_output.txt`

分工表	
r06725008 郭毓棠	2-1 model/train、2-1 report
r06725005 郝思喬	2-2 model/train、2-2 beam norm、2-2 report
r06725020 劉冠宏	2-1 attention、2-2 attention、2-2 beam search