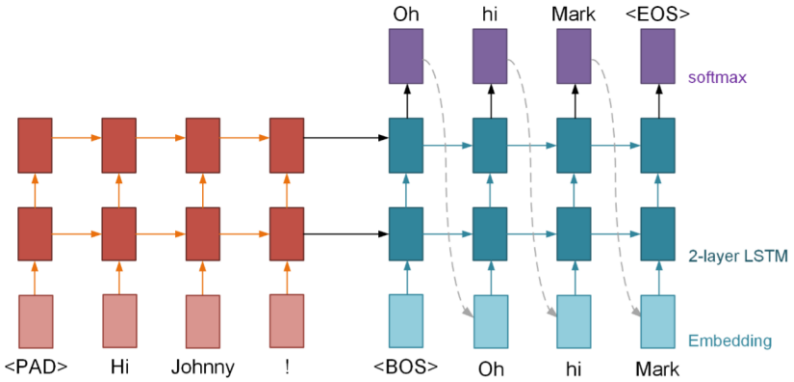
MLDS HW2-2 Report: Chatbot

組員：b04901060 電機三 黃文璁　分工：code: [2-1, 2-2], report: [2-2-1]
b04901003 電機三 許傑盛　code: [2-1, 2-2], report: [2-1-1, 2-1-2, 2-1-3]
b04901096 電機三 蔡昕宇　code: [2-2], report: [2-2-2, 2-2-3]

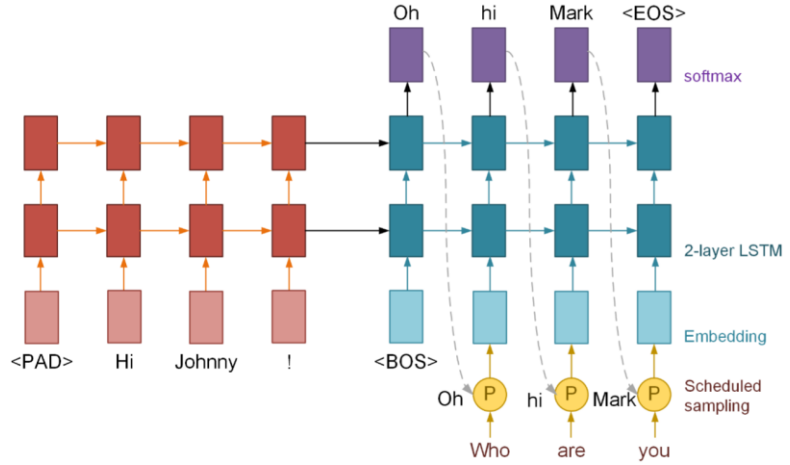
2-2-1 Model Description

本次作業 2-1 實作的 seq2seq 改進方法為 scheduled sampling。

基本上 seq2seq 模型本身和一般的 seq2seq 較爲類似，如下圖：



不過 training 過程中會加入 scheduled sampling 來讓訓練過程更容易收斂，如下圖：



其中 P 為選擇以 ground truth 為下一個 timestamp 之 decoder input 的機率，這個部分我們將於後面討論。至於模型和訓練過程的其他參數為：

Number of training sequence:	50000
Word embedding:	one-hot
Batch size:	64
Optimizer:	RMSprop (lr=1e-3)
Loss:	Cross entropy
RNN cell:	LSTM
RNN layers:	2
RNN dropout:	0.2

2-2-2 How to improve your performance

1. Methods that makes you outstanding

這次的實驗是應用 scheduled sampling 在我們 chatbot 的 training process 中，其中我們實驗了四種不同的 scheduled sampling 方法，再分別和不使用 scheduled sampling 的訓練過程比較。因此需要設一個 threshold，判斷 model 使用自己學到的字或是給正確的 target，以下列出四種改變 threshold 的方法分別爲：

（1）constant

Constant method 我們將 threshold 設為 constant，在實驗的部分，我們分別使用 threshold 為 0.2, 0.8 來做實驗，並與其他方法一併比較。

除了 constant method 以外，其他方法的 threshold 都會 decay，但 decay 的方式不同。我們觀察約 100 個 epoch 作右，大概比較能夠產出像是有文法的句子，因此我們大致上將 threshold 過了 100 個 epoch 設到很小。以下列出其他三種的 decay 方式

（2）linear decay

threshold = max(1e-8, 1 - epoch/100)

（3）exponential decay

threshold = max(1e-8, 0.98^(epoch))

（4）inverse sigmoid decay

threshold = max(1e-8, 10/10+e^(epoch/10))

此實驗是希望透過不同的方法，可以讓 model 不完全是硬記下 training data 的資料，而可以透夠自己生成的文字去學習。

2. Why do you use it?

由於這次作業 2-2 所用到的訓練資料較差，畢竟資料中相鄰兩句的關聯性就算是由我們來預測都很困難，我們認爲這種關係較弱的資料需要額外的輔助才能成功訓練出東西，故我們使用 scheduled sampling 來加強字與字之間的關聯，testing 上可能有較好的效果。

然而，我們不能完完全全的使用 teacher forcing 的方法，將有可能使得 model 只學到該筆資料的資訓而無法 generalize。我們在實驗中也分 4 種方法，目的是為了探討不同的方法的收斂的方式與訓練出來的結果之間有什麼樣的關聯，文字跟文字間的關聯性是否建立。

3. Analysis and compare your model without the method

本次實驗結果中，我們先討論有無 scheduled sampling 結果的差異性。首先，我們將 threshold 設為 0.8，視為一個有使用 scheduled sampling 而不是完全 teacher forcing 的一個方法，結果如下：

#49 Input: 這 一切 根本無法 逃避 ！

constant(0.8): 然後看了一個新感覺

teacher forcing: 你不知道該怎麼做的

#1049 input: 跟 某 對 年輕 的 新婚夫婦

constant(0.8): 也許這樣這是個問題

teacher forcing: 比塔兒幹嘛比你喝酒奶酒

#8015 input: 關於 電視 裏面 的 報道 , 不要 相信 你 看到 的 . 也 不是 說 絕對 .

constant(0.8): 哦天啊這個我教我的孩子

teacher forcing: 如果你們說出現在說我是誰的話

先不論有沒有回答到問題，或是有與 input 對話的感覺，先可看出有些這兩個方法結果雖然有些差異，但似乎都有合理的句子的樣子，也沒有中間會出現不應該出現的字的感覺，從字面上的差異並不大。有時候 scheduled sampling 的結果好很多，像是：

#4073 input: 你 上去 桑拿 按摩 就 有錢 ?

constant(0.8): 有我有一塊錢啦
teacher forcing: 謝你,頭

可以看到 scheduled sampling 的結果有提到“錢“字，與 input 較有關聯，讓我們結果品質有稍微的提升。然而，這兩個方法也可能 output 出來的句子很短，可能生成的句子裡只幾個字，像是：

#1263 input: 你 不是 來 露營 的

constant(0.8): 嗎?
teacher forcing: 好吧

有時候會有這樣不理想的結果，可能我們在 post-process 也把一些生成的重複冗字刪除，導致結果只有一兩個字，也可能這句話他並沒有學到任何的對話。但我們發現在 scheduled sampling 較容易出現這種狀況，且那幾個字剛好是在容易在結尾出現的。但整體而言，使用了 scheduled sampling 之後，結果似乎並沒有較差，以下有更多實驗的結果。

2-2-3 Experimental results and settings

本次實驗結果中，我們將上述提到 4 種不同的 scheduled sampling 的方法做實驗。在 training data 中隨機挑選幾組對話來觀察 4 種方法的不同。另外加入完全使用 teacher forcing 的結果作為比較。

#12 Input: 我 無能爲力

constant(0.2): 可是不是我真理的情負責
constant(0.8): 但是很多重要
linear: 得受我動
exponential: 利的道樣
inverse sigmoid: 我不來逼走。
teacher forcing: 不對,你明可以

#6759 input: 我 想 做 的 只是 給他些 建議 .

constant(0.2): 而果你覺到你擁有一切的
constant(0.8): 我知道我不能把他們解釋
linear: 我並不或如此羅,
exponential: 就你這麼來不要的
inverse sigmoid: 只要我們看到的
teacher forcing: 我也知道,非常抱歉,

#8015 input: 關於 電視 裏面 的 報道 , 不要 相信 你 看到 的 . 也 不是 說 絕對 .

constant(0.2): 當他肯指使變來發生婚,結婚.
constant(0.8): 哦天啊這個我教我的孩子
linear: 是真的,我是喜歡的
exponential: 無許上生沒有人這些的人都不有的
inverse sigmoid: 你想在我見我一次了,
teacher forcing: 如果你們說出現在說我是誰的話.

由於這次的訓練資料比較差，有些前文不對後文的句子，使得 model 容易學到一些不太合乎常理的對話，而且資料量大。因此，為了加快訓練的速度，且同時避免學習過多 bias 的對話，我們隨機取部分的 training pair 來訓練這個 model。

而本實驗主要觀察 scheduled sampling 實作方式有什麼不同。由上面的幾個結果做比較，從中很明顯的可以看出如果沒有 scheduled sampling，而是全部使用 teacher forcing 的話，解出來的句子比較像是對方或是自己會接的話，字跟字之間也相對比其他方法來的有關聯性。再來是比較兩種 constant 的結果，threshold 分別為 0.2 和 0.8。0.8 則有比較大的比例使用 target data，反之 0.2 則使用自己預測的字當作下一個 input。在 threshold 為 0.8 的情況下訓練出來的結果似乎比較像是一個能看得懂的句子。而 0.2 的狀況下，字跟字的之間的關聯有建立起來，但詞彙之間的關聯性似乎關聯性不大，讓結果是一個看不懂的句子。

相較於 constant method，其他的方法都會在一定的 epoch 之後會讓機器學習自己產生的字。由上面的結果來看，在沒有 teacher forcing 之下，句子開始變樣，使得字跟字之間的關聯性開始降低。推測這樣的訓練資料讓他學到的字詞本身的上下關聯性其實並不緊密，可能出現字出現的時間很突兀的感覺，三者方式的結果似乎差不多。

Beam Search

在這次的實驗中我們也加入 beam search 作為討論。以下的解果是以 teacher forcing 的 model 去跑 beam search。Beam 為 1 即是沒有跑 beam search 之結果

#1 Input: 那 一點 也 不難

Beam = 1: 嗯, 你想要溫暖的歌

Beam = 3: 作一段旅行, 廳, 上校, 你不能回家。這次機票, 你可

#6883 Input: 我 甚至 沒有 注意 他 .

Beam = 1: 但是如果你們這樣做, 我不知道

Beam = 3: 他犯罪!所能我會進去那樣說的, 你要把他扔掉你去

我們可以看到，加入 beam search 之後，都有可能解出更長的句子，可能在句末每個字串的機率跟 <EOS> 其實不相上下，可能會解出後面的字，句子生成出來看起來都比沒有使用 beam search 來的長。透過此，我們希望能夠解出可能沒被解出來的字句，或是被隱藏好的字句。但很多時候，句末解出來的字並不一定好，有可能生成出一堆冗字，像是：

#9432 Input: 只是 一個 偉大 的 時間

Beam = 1: 我們想要大家都不能拿到

Beam = 3: 你不是需想說, 我不能讓他看的。我不是在乎。我是

可以看到 beam search 找到較多不必要的字。且同時我們觀察到 beam search 出來的文字關係其實相對沒有改善很多。以下面同樣的例子來看，也將 beam 調高結果一併觀察：

#7383 Input: 由 那些 真實 的 經歷 了 那些 可怕 時刻 的 當事人 來演 .

Beam = 1: 真的記錄是這些人都不是說什麼?

Beam = 3: 他沒來做好多, 這麼大概別人來看着子吧!對啊!

Beam = 5: 再來像她一直做看上去我來一張吧!?對了, 老鼠下牀

觀察到也有相同的情況，beam search 沒有加強其生成的字之間的關聯，且與上述說明相同，句末會出現一些冗字。綜合而言，由這個訓練資料得到的結果得倚靠 teacher forcing 來強迫機器學習到真正的字詞，若不用，threshold 仍要提高一些讓他可以使用學到更多正確資訊。