MLDS HW4 Report

組別：我才是真的Baseline
組員：f03942038鍾佳豪 / r05942102王冠驊 / d05921018林家慶 / d05921027張鈞閔

Environment

OS: Ubuntu 14.04.5 LTS
CPU: Intel i7-5930K
GPU: GeForce GTX 1080 , GeForce GTX TITAN X
Libraries: tensorflow, numpy, pandas, argparse

Data Preprocessing

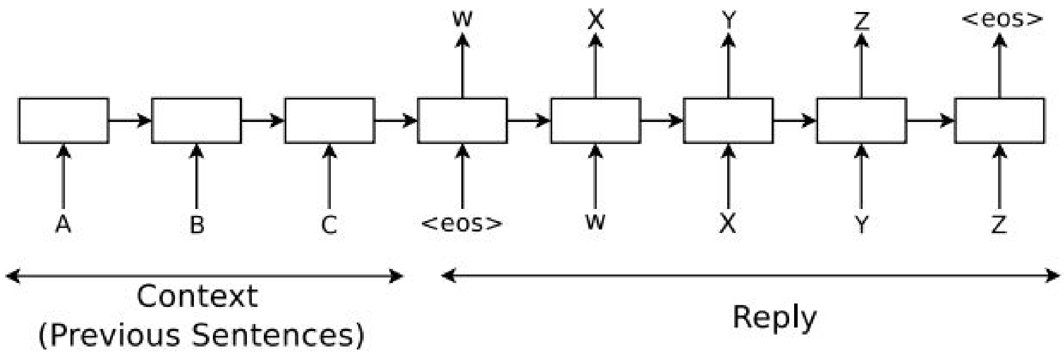
我們所考慮的訓練資料集包括 Cornell Movie-Dialogs Corpus、OpenSubtitles 以及 Twitter_en。前兩個資料集都包含大量的電影台詞，OpenSubtitles 的資料量比 Cornell Movie-Dialogs Corpus 大上許多。 Twitter_en 則是 Marsan-Ma/ChatCorpus 從 Twitter 上所蒐集的資料集。此外，Twitter_en 內的句子比較口語化，不如電影台詞來的正式。最終，我們只選用了 OpenSubtitles，因為其資料夠多、句型又比較正式，較適合用來訓練 chatbot。

我們將資料集中相鄰的每兩個句子(或台詞)分別當成訓練模型的 Input Sequence 及 Output Sequence。為了加快訓練的速度，我們利用 maxLength 這個參數限制我們所取用的句子長度，並且利用 vocabularySize 這個參數限制最終我們所建立的字詞字典的大小。我們在在字詞字典中加入 <pad>, <go>, <eos> 以及 <unknown>，其中 <pad> 用於補齊不同台詞間的長度差異、<go> 與 <eos> 分別代表一句台詞的開始與結束，而 <unknown> 則用於取代不存在字詞字典的字詞。

Model

A. Seq2Seq Model

我們實作的 Seq2Seq Model 是參照 A Neural Conversational Model [1] 所提出的想法，其基本架構如下：



其中，Context 為使用者的問題(Input Sequence)，Reply 則是機器的回答(Output Sequence)。模型包含了一個 Encoder 以及一個 Decoder，Encoder 會讀入 Input Sequence、Decoder 則負責解出 Output Sequence。Encoder 跟 Decoder 都有 Embedding Layer 將文字轉成 Word Vector。Decoder 有 Project Layer 將 Output Vector 轉回文字。詳細的設定如下：

Encoder: 3-layer GRU with 256 hidden units、Decoder: 3-layer GRU with 256 hidden units、Word Embedding size: 256、Object Function: Sampled Softmax、Optimizer: Gradient Descent Algorithm (learning rate = 0.5)、Max Gradient Norm: 5、Vocabulary Size: 100000、Batch Size: 64。

我們亦參照 [2-4]所提出的方法來改善 Seq2Seq Model ，其中包括：Attention Mechanism [2]、Bidirectional Encoder for First Layer [3]、Residual Connections [3]、Scheduled Sampling [4]。我們會在實驗中探討這些方法帶來的改善。

B. Reinforcement Learning

根據 Deep Reinforcement Learning for Dialogue Generation [5] 的觀察，若單純使用 Seq2Seq Model 訓練一個 Chatbot，會發現模型常常產生"I don't know"這種較為安全的回答，使得與 Chatbot 的對話較為枯燥乏味。因此，我們使用 Reinforcement Learning來改善這個問題。我們依照 Jiwei Li [5] 以及 Marsan-Ma [6] 的作法進行 Reinforcement Learning。

我們所考慮的 Reward 包括：r1 = Ease of answering : 所生成的句子不能是常出現的句子，r2 = Information Flow : 連續生成的句子必須要有變化，r3 = Semantic Coherence : 連續生成的句子必須要有語意上的一致性。依照[5]的設定，加權為 Reward= 0.25*r1+0.25*r2+0.5*r3。

Experiments

Exp1: 比較 Seq2Seq Model 加上 Attention Mechanism、Bidirectional Encoder for First Layer 及 Residual Connections 的結果。Training steps: 60000。

	Seq2Seq + Attention (Proposed by [2])	Seq2Seq + Attention + Bidirectional encoder + Residual connections (Proposed by [3])
Training Perplexity	9.57	9.13
Testing Perplexity	9.55	10.45
Hi	what is you doing ?	hi , you are a good man
Bonjour.	what ?	hello , you .
How are you?	what is you doing ?	how are you ?
My name is Etienne.	what is you doing ?	you know it ' s a good idea .
What is your name ?	what is you doing ?	you ' re a man .
How old are you ?	what ?	00 .
Are you conscious?	no .	yes .
Where do you want to go ?	here .	you ' re in the right place .
Where are you ?	it ' s a good time .	you ' re in here .
I'm going to kill you!	get up !	you ' re going to kill me !
What is your best memory ?	it ' s a good idea .	you know it ' s a good idea .
How much is two plus two ?	0 .	000 .
Do you have a girlfriend?	no .	yes .
Goodbye!	goodbye .	goodbye !
Good night.	good night .	good night .

* Testing Perplexity 為所有 Buckets 的平均值。

在訓練的過程中，我們發現第二個模型收斂的比較快。訓練到一定的次數後，兩個模型的 Perplexity 都收斂到差不多的數值，Training 及 Testing Perplexity 都在 9~11之中跳動。

從實際的回答上，我們可以看到第一個模型似乎還沒有學好；而第二個模型所產生的回答較為理想。可見加入 Bidirectional Encoder 來處理雙向的資訊、以及加入 Residual Connections 有助於 Chatbot 模型的訓練。此外， Perplexity 似乎不太能真正反映 Chatbot 的好壞。我們也可以看到模型生成的回答都偏短、安全。

Exp2: 利用 Scheduled Sampling 將實驗一中的模型都再訓練 10000 Steps。在每個 Step，有 3/4 的機率會將 Decoder 生成的 Output 當成下個時間點的 Decoder Input；有 1/4 的機率會將正確的 Target Input 當成 Decoder 的 Input。

	Seq2Seq + Attention (Proposed by [2])	Seq2Seq + Attention + Bidirectional encoder + Residual connections (Proposed by [3])
Training Perplexity	14.78	14.30
Testing Perplexity	10.77	10.53
Hi	what you	you ' you
Bonjour.	what .	you .
How are you?	what you	you '
My name is Etienne.	you ' re .	you is
What is your name ?	what you	you .
How old are you ?	what you	you '
Are you conscious?	what you	you '
Where do you want to go ?	you ' re not .	you .
Where are you ?	what you	you '
I'm going to kill you!	you ' re not	you !
What is your best memory ?	what .	you '
How much is two plus two ?	what you	three is four .
Do you have a girlfriend?	yes .	you .
Goodbye!	what you	you !
Good night.	good night .	good night .

我們可以看到我們實作的 Scheduled Sampling 會導致模型學壞。原因可能是因為我們是在"每個 Step" 來決定是否要使用 Decoder 生成的 Output 當成下個時間點的 Decoder Input。也就是說，整個 Batch 都是用同一種方式訓練。因此，我們的模型反而會逐漸、且不管上下文，都只生成一些常出現的字詞或標點符號，如"what"、"you"。論文[4]中所提出的 Scheduled Sampling 應該是要在每生成一個字的時候都隨機選擇一次。

Exp3: 利用 Reinforcement Learning 調整實驗一的兩個模型。考慮 100 個句子，計算 Reward，並更新模型。

	Seq2Seq + Attention (Proposed by [2])	Seq2Seq + Attention + Bidirectional encoder + Residual connections (Proposed by [3])
Hi	is you , on what on on on . on	you you
Bonjour.	is . you , my	you , . , you
How are you?	is , m m m m m m m m m m m m m	you you in
My name is Etienne.	you you m . you you . that that that . that this	i ,
What is your name ?	s . . s you . you . you	she . , you you

	you this .	
How old are you ?	what on . on this . this	you you in
Are you conscious?	is . you you , this ' this . this	you you s you
Where do you want to go ?	you you on on . what what what us us . us us this .	i in

我們實作的 Reinforcement Learning 也讓模型學壞。可能的原因為 1) 我們只使用單一的 Model 進行 Reinforcement Learning，並未像論文[5]讓兩個 Model 互相對話，2) 考慮的句子太少，Learning Rate 又設定的太大，導致模型更新太快而壞掉，3) Reward 設計或實作得不好，導致模型沒有朝正確的方向更新。

Conclusion

經過實驗的測試，我們認為 Seq2seq Model + Attention + Bidirectional Encoder for First Layer + Residual Connections 是表現最好的模型。然而，單純利用 Seq2seq Model 的 Chatbot 所生成的回答真的是偏短、安全，這也許是 overfitting 的結果。我們也認為 Perplexity 似乎不太能真的反映 Chatbot 的好壞，真正的評估還是要經過人的判定。也因此，Reinforcement Learning 是一個方法可以考慮人的感知，以改善 Seq2seq Model 用於 Chatbot 的效能。

我們也發現 Scheduled Sampling 必須是要在每生成一個字的時候，都隨機選擇一次才可能會有效果。對於 Reinforcement Learning，在我們的實作上沒有帶來改善，尚須進一步修正。

Reference

[1] Oriol Vinyals and Quoc Le, “A Neural Conversational Model,” arXiv:1506.05869.
[2] Dzmitry Bahdanau et al., “Neural Machine Translation by Jointly Learning to Align and Translate,” arXiv:1409.0473.
[3] Yonghui Wu et al., “Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation,” arXiv:1609.08144.
[4] Samy Bengio et al., “Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks,” arXiv:1506.03099.
[5] Jiwei Li et al, “Deep Reinforcement Learning for Dialogue Generation,” arXiv:1606.01541.
[6] Marsan-Ma’s github: https://github.com/Marsan-Ma/tf_chatbot_seq2seq_antilm

Team Division

f03942038 鍾佳豪	Model implementation (Bidirectional encoder、Residual connections、Scheduled sampling、Reinforcement learning); Experiment; Report
r05942102 王冠驊	Model implementation (Seq2Seq model、Attention Mechanism); Experiment; Report
d05921027 張鈞閔	Data Processing; Experiment
d05921018 林家慶	Data Processing; Experiment