Machine Learning Final Report
Dialogue Modelling Problem

Member:
林啟祐 R07946012/ 吳泓毅 R07946013/ 劉育宏 R07946008

Section 1: Introduction & Motivation

本次Final Project主要專注於自然語言處理的相關練習，此類應用在實務上可以廣泛運用在聊天機器人等功能的建置。通過語意的理解與學習，聊天機器人可以為客戶端提供準確的服務內容，並且針對客戶需求進行更即時的反饋與回應。通過導入聊天機器人優化企業的服務水平，將許多傳統模式中被動的運營環節轉化為更多主動觸及的方式，例如等待用戶反饋置換為主動引導用戶分享回饋。在這個段落會著重說明聊天機器人在實務上可以實踐的商業用途以及目前建置的商業模式。

商業用途

(用途一) 客戶服務
客戶服務是目前聊天機器人在商業用途中最為普遍的落地模式，有別於傳統客服人員，聊天機器人可以做到以下三點重大革新。

1. 維持24HR不間斷服務，能夠隨時隨地協助客戶獲取企業的解答服務。
2. 能達到即時回應的快速服務，避免詢答客戶長時間的等待。
3. 雖然短期需對導入聊天機器人付出一定的成本代價，然而從長遠的眼光來看，聊天機器人的建置可以有效的降低企業在人力支出的相關成本。

儘管人們往往擔心聊天機器人無法有效地提供客戶準確的應答內容，然而隨著自然語言處理技術的成熟，聊天機器人已經有能力滿足60%到80%的客服需求，相對也能優化企業的運營模式。

(用途二) 採集客戶資訊與消費偏好
除了被動的回應客服需求之外，聊天機器人可以收集客戶訊息進行更深入的分析與歸類，甚至可以主動提供用戶其他產品類別進行消費意願的侵入式探索。通過此類功能，聊天機器人在完成客服工作之外還能進一步幫助企業的CRM系統進行聯動整合，以達成精準行銷以及擴大Loyal Loop的目的。

(用途三) 導購服務
聊天機器人可以告知客戶商品在實體賣場的擺放位置，有效的減少客戶商品搜索的花費時間。在電商市場，聊天機器人可以主動推送商品促銷訊息，刺激客戶進行消費行為，或是根據客戶的行為模式給予可能感興趣的商品的推薦，並提升客戶在選購過程中的購買轉化率。

(用途四) 推廣促銷活動
傳統模式中，企業經常使用的促銷活動推廣方式包括簡訊、電子郵件或是紙本文宣的投遞來散播訊息，然而根據統計，傳統的營銷電子郵件被客戶打開閱讀的比例只有約1%左右。聊天機器人則擁有更大的彈性在對話中直接傳遞促銷訊息給消費者，也可以搭配手機APP等應用更有效率的將資訊傳遞到智能終端移動設備。

(用途五) 蒐集客戶評價
客戶的反饋意見對於企業在運營模式上對市場做出及時反應是至關重要的參考依據，然而實務上企業經常遭遇消費者對於填寫問卷缺乏行為動機，或是不願意花額外的時間給予意見評價。通過聊天機器人的應用，企業可以將冗長的問卷拆分為多條不同的簡訊或訊息進行意見採集，提升客戶參與意願。這些聊天機器人所引導的意見和評價將會是企業面對今日快速變動的市場進行敏捷調整的關鍵指引。

商業模式

(模式一) 原生內容和贊助內容
原生內容和贊助內容的相互搭配逐漸被運用在電商客戶攫取的應用上。比如消費者在一篇關於手機型號選擇的頁面上詢問聊天機器人關於某項手機功能重要性的問題，聊天機器人除了給予相應答覆之外，還能根據客戶的對話內容推送另一篇關於贊助企業熱銷手機的行銷內容。

(模式二) 強化線上銷售連結性
聯合廣告銷售已經成為企業在線上市場經常運用的一種商業模式。例如A和B兩家銷售同類型展品但是鎖定不同客戶群體的企業，通過聊天機器人強化兩家企業之間的連結合作。比如A企業所定為中低階市場，當A企業的聊天機器人理解客戶需要功能性更好的商品時，則自動推送B企業的網站連接將該客戶導向B企業的店鋪。此類共享流量的模式在電商市場尤為常見，也因為聊天機器人的應用，讓企業可以及時的將潛在消費者引導到適合的商店，把握更多的潛在商機。

(模式三) 進行收費性諮詢
人們都希望獲得專業可靠的好建議，同時也願意為好的建議付出相對合理的成本。然而傳統商業模式中，專業顧問的收費有時是一般消費者無法支付的金額，通過聊天機器人的建置，可以迎合經濟型客戶對專業諮詢的需求。例如以往法律諮詢需要向律師支付為數不斐的諮詢費用，聊天機器人可以針對一些基礎的法律問題進行回答，讓消費者有機會以更親民的價格獲取相關法律知識，同時也可以讓弱勢族群在面臨法律訴訟的過程中獲得更多的權益保障。

Section 2: Data Preprocessing/Feature Engineering

程式碼定義：

(1.) 0 -> first sample
(2.) message-so-far -> 前面問答的內容
(3.) option-for-next -> 其他選項
(4.) 斷詞 -> Spacy

Embedding作法:

- Speaker & Utterance每300個字Concade在一起
- 每句話以50個字做計算
- 隨機抽選10個可能的選項(沒有把所有都寫進去)
- 判斷其中1個正確，其餘9個選項判為錯誤。
- Pre-train的部分採用了FastText English word vectors released by Facebook.

- # crawl-300d-2M.vec - 2 million word vectors trained on Common Crawl (600B tokens)

- - 每次對話會有600個dimension (包括 Participant 300個dimension以及Word 300個dimension)。

-

Section 3: Model Description

我們的model 是由 Gated recurrent unit (GRU) 和 self-attention 構成。最開始的 model 是只用了 GRU，後來再加入 self-attention的機制。

GRU:
GRU 用了兩個 gate 來記憶和更新以前的資訊: reset gate 和 update gate。假設在這之前記憶的資訊(hidden state)是 h(t-1)，reset 現在的資料 x(t) 會和過去的 h(t-1) concate 起來在算出 reset gate 和 update gate 的值:
reset gate = sigmoid(W_r * [x(t), h(t-1)]),
update gate = sigmoid(W_r * [x(t), h(t-1)])

再將過去的資訊選擇性地跟現在的資料結合:
h'(t-1) = Hadamard_prod(reset gate, h(t-1)),
h' = tanh(W* [x(t), h'(t-1)])

最後得到的結果就是用update gate 將現在和過去的資訊做加權平均:
h(t) = update gate * h(t-1) + (1 - update gate) * h'

相比LSTM的三個gate，GRU的gate 較少不僅減少了參數的量，也增快了收斂的速度。

Self-attention:
Attention 的機制是為了解決傳統 RNN 在長距離的記憶問題，因為RNN 的input 還是sequence input，所以假如重要的資訊出現在句子頭尾的話，重要的資訊很可能被中間的資訊給覆蓋掉，而 attention 的機制就在於把焦點聚焦於重要的字上，如此一來model 更能捕捉到字義的連貫性。

在 self-attention 中，每個字除了一開始的 embedding 以外，還會另外有三種表達的方式，分別是 query, key, value，如果資料中的兩個字(a,b)有語意的連貫性，那這兩個字的 (a_query, b_key) 分數就會很高，最後再根據這個分數和 value 的值做加權平均輸出。藉此模型可以學到比較長距離的語意關係。

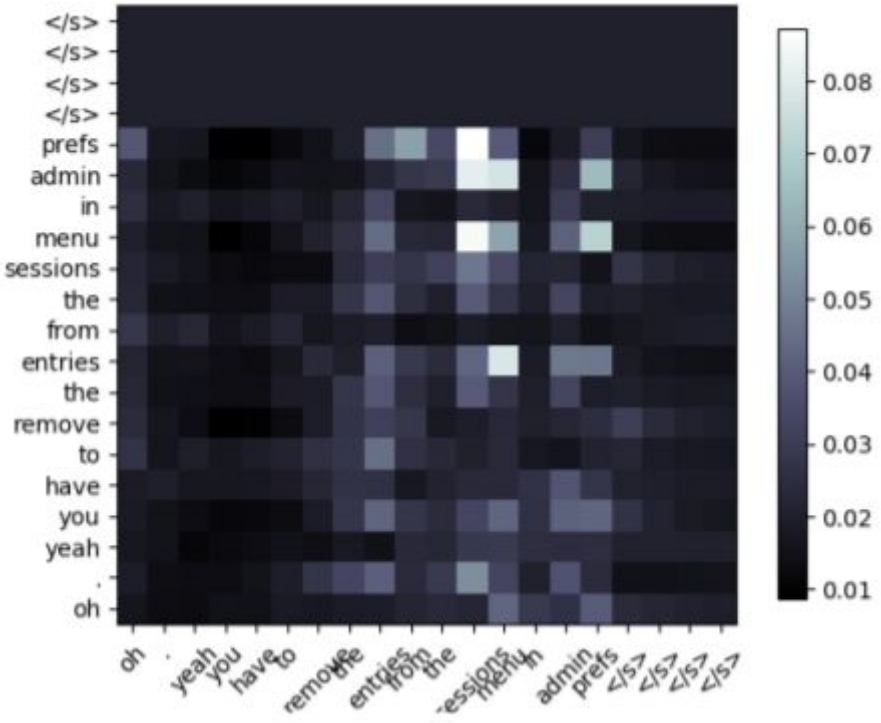
這裡的query, key, value 參數都是 model 在 training 的過程中訓練出來的，所以每一種(query, key, value) 的 pair 都可能代表字跟字的一種對應，所以當我們把這個 pair 增加時，model 就可以捕捉到文字中字跟字各種組合的各種意思，這個在原本的 paper 中又稱作 Multihead attention。

Section 4: Experiment and Discussion

在模型中我們目的在於比較對話和回答句的相似度，因此模型將對話和回答句皆利用GRU做sentence embedding，再接上一層bilinear linear得到分數，loss function 都是用 Binary cross entropy with logits loss，而最佳化都是用 adam。此外我們嘗試比較使用attention機制和不使用attention機制的實驗結果。

通過 simple baseline:
一開始我們有分 training 跟 validation 的資料，只過一層的 gru 還沒有接上 attention，然後每題的選項我們只抽1個positive、9個negative來作training，最好就作到0.24666，通過simple baseline，在training 的過程中，不確定是不是NLP model 的關係，model 大約在9到10個 epoch 之後就會 overfit 了，而 leaderboard 上的結果也跟 validation 的 recall 差不多。

Attention 的效果:
因為在回答對話的任務中，我們需要專注的重點不需要是整句對話，反而是專注在幾個重要的詞即可，剛好attention機制可以使對話和問答句專注在某幾個重要的詞上，使得模型能夠更有效的找到最適合的問答句。下圖是同一句話中字跟字重要度的權重。我們可以看到中間有一些自權重較高，代表model 認為這兩個字的字義上是有關係的。



再加上 attention 以後，我們model 的loss 也從最開始RNN的 9.80 降到 9.69。
比較 GRU 及 LSTM:
從下面這張圖也可以看出上一張所提到的，GRU 比LSTM用較少的參數，但卻可以在更短的時間達到更好的效果。

Simple_rnn	LSTM	GRU
recall@10	0.6416	0.582
GPU memory	2025MiB	2589MiB
Training/test speed	11:09/ 2:39 per epoch	29:19/ 5:31 per epoch

而我們最好的model 就是將 RNN 換成GRU，然後配上 self-attention，最後得到的 training loss 是9.53而在 Kaggle 上的成績則是0.3325。

Negative sample size:
一開始我們是先選一個比較小的 negative sample size 來作 training，原因是想減少training 的負擔，而且在 word2vec 中也是類似的概念，所以我們是先從 model 下手去改善結果，但是效果都沒有差很多，而最後結果從 0.2 跳到 0.3 的關鍵是把 validation 的資料一起拿去作 training，然後把每題100個 negative sample 全選；不過光從 training 和 validation 的 size 來看，validation 的size 也只有快到 training 1/20 ，所以比起加入 validation 的資料，還是應該歸功於把 negative sample 都納進來比較合理。
至於在 negative sample size 的選取上面，因為有每一題有100 個 negative sample，假如全部都納進來作 training， 那 training 的過程會跑很久(因為每一筆的資料都很大，所以為了要把資料放進 GPU中， batch size 不能開得太大)，再加上如果有GPU記憶體的限制，當 nagative sample size 增大時，在有限的 GPU 記憶體下，NN model 的層數也不能開得太高，因此限制了model 的複雜度；最後比起去調整Negative sample size 和 model complexity 的 tradeoff，為了模型的 robustness 我們選擇把 negative sample 全部納進來。

固定 Pretrained embedding:
以上的實驗都是把 Pre-trained embedding 當作embedding layer 的 initialization來作，我們還有測試把 embedding layer 固定，這樣每個字的 embedding 就不會在 training 的過程中被改到，不過這樣子做會導致 training loss 下降得太慢，雖然比較不會 overfit 但結果上也沒有比較好；有想過因為固定了 embedding layer 所以相對減少了 model 的複雜度，因此可以靠增加 GRU 的深度來把 model 的複雜度加回來，不過最後沒有執行，原因除了GPU 記憶體的限制外，還有當我們固定 Pretrained embedding 時，這個 embedding 就跟我們的dataset 沒甚麼關係，而且僅加深NN 的層數不算是非常結構性的改變。

Section 5: Conclusion

總的來說，NLP 的模型比起其他類型的 task 更容易 overfit，而這方面還是需要靠觀察 training 和 validation 的 error 來避免，而且由於語言中的同樣的字可能帶有在遇到不一樣的字最搭配時，可能會產生完全不同的意思，又或者有些辭意需要上下文來判斷，這個時候藉由RNN配上 attention 的效果就非常顯著。至於在 Embedding 上面，如果是用 Pre-trained 在不同 task 或不同 dataset 的 embedding 還是要重新讓 embedding fit 在目前的dataset 比較好，

Improvement and future work
從準確率上來看，我們的model 還沒有抓到太多語意時就先 overfit 了，所以猜測是model 的部分要做調整，猜測應該是要多增加 self-attention 的數量，讓他變成Multihead attention ；如果文本在龐大一些，那可以對目前的 dataset 直接作一個 embedding 這樣在training 的過程中就可以合理的把embedding 固定，進而對model 作更多的調整。

Application
針對現階段業界所使用的自然語言處理相關應用場域可以說是非常廣泛且效果卓著。從服務業的客戶服務支援，到金融業KYC分析對客戶進行優化推薦，到網路電商進行流量互通等方方面面都可以看到聊天機器人所發揮的重要作用。相信隨著技術領域的突破以及對市場教育的日漸成熟化，我們能持續看到聊天機器人或是其他自然語言處理相關的應用工具在產業典範轉移的洪流中扮演關鍵角色。

Reference

https://www.economist.com/business/2016/04/09/bots-the-next-frontier

https://www.shopify.com/enterprise/ecommerce-chatbots

https://www2.deloitte.com/nl/nl/pages/financial-services/articles/the-rise-of-chatbots-in-financial-services.html

https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/how-bots-algorithms-and-artificial-intelligence-are-reshaping-the-future-of-corporate-support-functions

https://mobilemonkey.com/blog/best-chatbots-for-business/

https://www.digiteum.com/chatbot-for-retail

Chung, Junyoung; Gulcehre, Caglar; Cho, KyungHyun; Bengio, Yoshua (2014). "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling". arXiv:1412.3555 [cs.NE].

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in Advances in Neural Information Processing Systems, 2017,

article:
Attention機制詳解(二)Self-Attention與
Transformer:https://zhuanlan.zhihu.com/p/47282410