應用序列到序列生成模型於情感型聊天機器人之研究

曹光中一，淡江大學資訊管理系研究所

指導教授：魏世杰

摘要

受益於深度學習演算法的發展，商業聊天機器人中應用深度神經網路的方法變得愈發普及。在 2018 舉辦的 NTCIR-14，CECG 子任務內，傳統生成式聊天機器人中加入了情感分類的元素，旨在針對同一問句，為不同情感標籤生成不同的回應。受此啟發，本文主要研究的生成式聊天機器人，透過深度學習的架構，使用 seq2seq 模型進行訓練，並以情感標籤為之分類。最後，本文將透過便利抽樣的方式對 30 位使用簡體中文的微博用戶進行調查，以此來評測聊天機器人在加入情感標籤條件下與普通生成模型的表現差異。研究發現以情感標籤為分類的模型，在多個面向的評估上都略勝傳統模型。

關鍵詞：聊天機器人、自然語言生成、深度學習、情感分析

壹、導論

一、研究背景

近年來微博微信等新興 SNS（Social Networking Services）飛速崛起，每天都有數以億計的對話在網路上傳播。在這些對話中又有著極大一部分來自於聊天機器人。

從回應答句的角度來看，可以將聊天機器人大致分成兩類：

（一）檢索式聊天機器人：回答是預設的，其中會使用到規則引擎、正則表達匹配或是使用深度學習預先訓練好的模型，从已有的知識庫中檢索一句最優答句進行匹配。

（二）生成式聊天機器人：不依賴也不使用已存在的回答句，可以自動產生新的答句。這種模型在訓練的過程中要求較大量的時間和語料，而語料通常包含 post 和 response 兩個部分。時下生成式聊天機器人大多會使用長短期記憶模型（Long Short-Term Memory，LSTM）和循環神經網路（Recurrent Neural Network，RNN）來訓練生成模型。這種方法早先在機器翻譯的領域中取得過非常好的成效。

目前而言，聊天機器人的商業化程度已有相當的規模，但受制于傳統聊天機器人大多是檢索式聊天機器人，預設的回答漸漸無法滿足企業與客戶溝通的需求。

二、研究動機

美國雲端通訊公司 Twilio 在 2016 年的報告中指出，有接近 89%的用戶更希望透過訊息（messaging）的方式跟企業或商家直接溝通，而不是電話或者郵件。但是只有 48%的企業去實現了這種溝通的可能。

現在，生成式聊天機器人也許能為企業和客戶溝通建立一座新的桥梁。前文提到的傳統聊天機器人（檢索式聊天機器人）有一些缺陷，而生成式聊天機器人所倚賴的巨量資料和運算時間，在當今科技飛速的發展下也已然不是很大的問題。在 Tensorflow、Keras、Pytorch 等開源深度學習框架，以及強悍的 GPU[1]設備支持下，使用深度學習訓練生成式模型的時間，已經被壓縮在一個完全可以接受的範圍內。此外，在這個資訊爆炸的年代，想要在網路上收集到大量資料用於此種研究，也不再是一件難事了。於是如何才能夠提升生成答句的質量，成為了生成式聊天機器人研究的重點。

在 NTCIR-14[2]，CECG 子任務中釋出的資料集里，除了問答對之外，還包含了句子（post & response）本身的客觀情感標籤，共分成 6 類（Other,Like,Sadness,Disgust,Anger，Happiness）。這個資料集啟發我，將情感標籤也加入生成模型的訓練，透過類神經網路，模型將會學習到類似自然人類高情商（Emotional Quotient）的概念，從而生成更合適的自然語言表達。

1.GPU：圖形處理器（graphics processing unit），雖然 GPU 在遊戲中以 3D 彩現而聞名，但它們對執行分析、深度學習和機器學習演算法尤其有用。GPU 允許某些計算比傳統 CPU 上執行相同的計算速度快 10 倍至 100 倍。

2.NTCIR-14： (NII Test Collection for IR Systems) ，針對亞洲語種的跨語言資訊檢索會議，2019 年為第 14 屆。

三、研究目的

如上文所述，本研究將在序列到序列的生成中，加入情感標籤為參考要素，使用 Pytorch 的開源架構來實作生成式聊天機器人，以實現聊天機器人在大部分情況下，能夠透過問句及其情感，來生成更相關且更合適的回應。

為了讓實驗結果更有可信度，本研究會將部分的測試問答句，通過簡單抽樣的方式設計成問卷，交由簡體中文微博[3]用戶，從語言流暢度，問答相關度和情感表達度三個方面進行評測，以全方位展示實驗的結果。

四、論文架構

本論文共分五個章節，前述的部分為導論，闡述本研究的背景，動機及目的。其後第二章是文獻探討，將討論與本研究相關的各個技術，其在發展過程中產生的變化與影響。第三章將會對本研究的研究方法加以介紹。第四章將會說明本研究的實驗環境，實驗設計和結果。最後會在第五章總結本研究的實際成果與研究限制，並提出未來可能得以延伸的研究方向。

3.微博： 在這裡指新浪微博，是中國大陸地區使用者最多的微部落格服務網站。至 2018 年，微博的平均每日活躍用戶量超過兩億。

貳、文獻探討

2.1 NTCIR-14

NTCIR 是一個針對資訊架構（Information architecture，IA）技術的競賽，其中包含問題回答，資訊檢索，資訊萃取和文本摘要等。NTCIR 最早由國立情報學研究所（NACSIS）和日本學術振興會（JSPS）聯合贊助，並於 1998 年開始籌備，並最終在 1999 年成功舉辦首屆 workshop。經過二十年的發展 NTCIR 已然成為一項國際重要賽事，它設置了一系列基於中文，日文，英文三種語言的評估任務，目前舉辦至第十四屆。

在 2016 年，第十二屆 NTCIR 中首次設置了中、日兩種語料的評測新任務，即短文本對話任務（Short Text Conversation，STC），這也是目前國際上唯一的“語言開放域”對話方面的評測比賽。

第十三屆 STC 競賽中，來自搜狗公司和清華大學的冠軍隊伍，ZhaoH（2017）等人使用 S2SAttn 模型，在對話的生成上取得了不錯的效果。這種模型是結合注意力機制的 seq2seq 深度學習模型，表現上來說會優於檢索式模型。

在今年的比賽中 ,依然有中文評測任務 ,而其中的短文本對話任務 NTCIR14-STC3，在上一屆引入基於深度學習的生成模型對話的任務設置之後，又結合了情感標籤的元素，提出基於情感標籤的對話生成任務（Chinese Emotional Conversation Generation，CECG）。

我們團隊有幸參加了此次 NTCIR 14-STC3 的 CECG 子任務，也正是本次的比賽啟發我使用情感標籤分類，作為提升句子生成質量的一種管道。

2.2 檢索式聊天機器人

2.2.1 樣板式

樣板式聊天機器人的技術原理是經過人工在聊天庫中設定一些對話場景，然後根據不同場景塑造相對應的對話範本，並根據每一個問題設計其可能會出現的答案。基於這個技術的優點是精確性高，缺點是人工工作量大，可擴展性差，不同的場景要有不同的設定。大名鼎鼎的對話機器人 Siri 就是使用這樣的技術生成的，相比於其他技術的對話機器人其精確性非常高。

2.2.2 規則式

規則式聊天機器人的回應句也是提前設定好的。它的技術類似於搜索引擎，在製作一個這樣的聊天系統之前，需要創建一個聊天對話庫並建立索引，根據輸入的問題，在聊天對話庫中進行搜索查詢來進行模糊匹配，並根據我們預先設定的規則計算相對匹配度最高的問句，取得其索引值，將索引值相對應的答句返回給用戶。要提升這一類聊天機器人，很大程度上倚賴語料庫的擴增和搜尋匹配演算法的提升。

2.3 生成式聊天機器人

近年在自然語言處理（Natural Language Processing，NLP）領域中出現了很多新的模型，而在生成式聊天機器人中大部分都是在序列到序列（Sequence-to-Sequence，Seq2Seq）模型框架下進行改進的。這種框架的好處主要有兩點，一是在未來的擴展上有很大空間，二是生成模型的泛用性也可根據訓練語料來調整。以下主要介紹本研究中所使用的生成式聊天機器人相關的幾個技術。在詞表示方面有詞嵌入（word embedding），深度學習模型方面有 RNN，循環閘單元（Gated recurrent units，GRU），基於注意力機制（Attention Mechanism）的循環閘單元等等。

2.4 Sequence-to-Sequence 模型

Seq2Seq 可以簡單看做是兩個 RNN 組成的模型，一個編碼器到解碼器（Encoder－Decoder）結構的網路，它的輸入是一個序列，輸出也是一個序列，編碼器的作用是將一個可變長度的序列轉成固定長度的表示向量，即圖 1 中的 c。而解碼器將這個固定長度的表示向量 c 變成可變長度的目標的序列。在實際應用中，seq2seq 活躍在機器翻譯，對話生成，文本摘要等領域。因為要求輸出要與輸入相關，所以模型本身要能夠記憶輸入序列的訊息。下面將介紹 seq2seq 模型的運作原理。

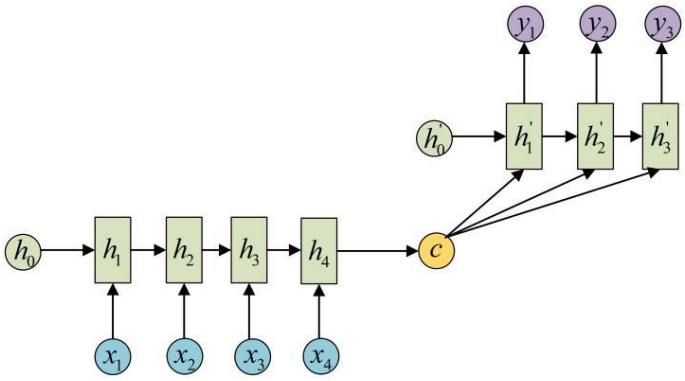

圖 1、Seq2Seq 模型示意圖

資料來源：（Luo[4]，2018）

2.4.1 深度學習

深度學習是經過構建大量訓練數據和大量隱藏層，學習更有用特徵，以達到提高分類或預測準確性的目的的機器學習模型。深度學習與傳統淺層學習的主要區別在於模型結構的深度，深度學習模型結構的深度通常會有 5 層以上。此外，在深度學習中，特徵學習的重要性被明確的強調，即經過一層層的特徵變換，將原始空間中的特徵表示轉換到新的特徵空間，使得模型分類與預測更加的簡單。

4.Luo：《Seq2seq 模型及注意力机制》,URL: https://zhuanlan.zhihu.com/p/36440334

學習特徵的方法有兩種：基於人工規則和使用大數據。前者適用於提升一些規則已知的學習，後者則可以更好地體現出數據本身所要表達的資訊。深度學習是一種端到端的結構，以實際例子來說可以從數據訓練直接到最終結果，不需要額外的特徵提取環節，或者也可以理解成所有的特徵都包含在隱藏層中。

深度學習有許多經典的模型，像是深層信念網路模型（Deep Belief Networks，DBNs），受限玻爾茲曼機(Restricted Boltzmann machines，RBMs)，卷積神經網路（Convolutional Neural Network，CNN），循環神經網路等。

2.4.2 循環神經網路

RNN 是為了解決前饋人工神經網路（Deep neural network，DNN）存在著無法對時間序列上的變化進行建模的問題（如自然語言處理、語音識別、手寫體識別），出現的另一種神經網路結構，循環神經網路。

RNN 循環神經網路，顧名思義其每一個節點的輸出，也會成為下一節點的輸入，即一個序列的輸出會與前面所有的輸出有關。具體的表現形式可以理解成整個網路會對前面的部分資訊進行記憶，即隱藏層之間的節點將會是有連接的，並且隱藏層的輸入通常會包括原始輸入和上一時間點的輸入。

從圖 2 可以看到，神經元自身的輸出可以在下一個時間點作用在自身，以實現循環（即圖中右側的環），並且隱藏層之間的節點互相連接。

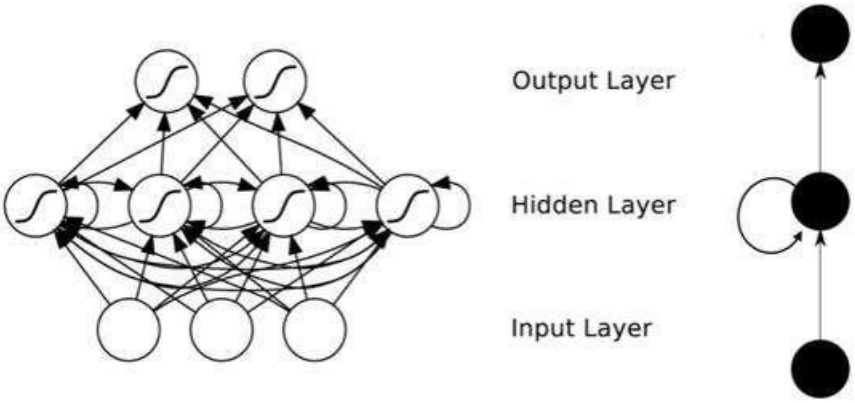

圖 2、RNN 模型示意圖

2.4.3 LSTM 和 GRU 單元

由於梯度消失問題，簡單 RNN（Simple RNN，S-RNN） 很難高效訓練。在反向傳播的（backpropagation）過程中，前面環節的誤差信號（梯度）消失的很快，於是就無法提取到更前面的輸入信號，使得 S-RNN 在捕獲大範圍的依賴上效果不佳。Long

short-term memory (LSTM) 結構是 Hochreiter and Schmidhuber （2017） 設計來解決梯度消失問題的，他們首創地引入了門閘機制。

S-RNN 結構中每層都複用相同的矩陣 W，這將導致梯度計算包含多次的矩陣 W 的乘積運算，這很容易導致梯度值消失或者爆炸，門閘機制解決這個問題的主要方法就是避免單個矩陣的重複乘積計算。

門閘機制，即通過門閘向量，動態的控制對當前記憶狀態的讀取，即可以被當前記憶狀態和輸入所控制，且其行為是可以學習的。該機制保證了和記憶部分相關的梯度，即使經過很長的時間跨度，也能保持較高的值，避免了梯度消失。

當訓練 LSTM 網路時，Jozefowics （2015） 強烈推薦總是將遺忘門的 bias 部分初始化接近 1。

雖然 LSTM 結構效果很好，但由於結構相對複雜，導致難以分析，且運算成本高。Gated recurrent unit (GRU) 是 Cho（2014） 作為 LSTM 替代品提出的。 GRU 雖然基於門閘機制，但減少了門閘的數量，並取消了切分 memory 的思路。

圖 3 中左側，i，f，o 分別表示輸入門，遺忘門，輸出門，c（cell state）代表某一時刻的隱藏狀態，這個隱藏狀態的計算由當前的輸入和上一時間點的隱藏狀態決定。由圖中可以看到 LSTM 單元的輸入會有三個，x 當前輸入，最上端的 c 隱藏狀態代表長時記憶，h（hidden state）代表短期記憶。GRU 是 LSTM 的一種變體，它把 LSTM 中的三個控制閘門數量減少到兩個，合併輸入門和遺忘門成為更新門，同時也合併了 c 和 h，為一個變數。GRU 的兩個門 r 和 z 代表重置門和更新門，r 來控制需要保留多少之前的記憶，z 來控制需要從前一時刻遺忘多少資訊。因此 GRU 只會有兩個輸入和兩個輸出，比 LSTM 更加簡潔。

目前 GRU 在語言模型和機器翻譯任務重表現很好，雖然相較 LSTM 孰優孰劣尚未有所定論，但可以肯定的是使用 GRU 可以有效降低訓練模型的時間成本。

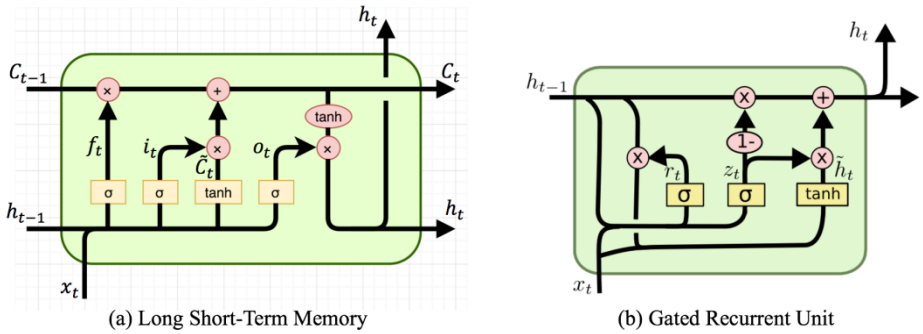

圖 3、LSTM 和 GRU 模型示意圖

參考來源：（Colah[5]，2015）

5.Colah：《Understanding LSTM Networks》,URL: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

2.5 詞嵌入

詞嵌入（Word Embedding）是指透過特定算法將詞語轉換成向量編碼的表示法，表示方法主要有 one-hot，N-gram，分散式表示(distributed representation)、共現矩陣等。

在早年很多機器翻譯的任務中，詞表示會使用單熱編碼（One-Hot-Encoding）。

單熱編碼是一種稀疏向量表示法，它將需要表示的元素總數作為向量的統一維度，每一個元素的對應向量，只有在特定的某一維表示為 1，其餘均為 0。這種詞表示法的優點是簡單，缺點也顯而易見。當語料的數量較大時，使用這種詞表示法容易產生維度災難。此外，這種詞表示法中每個詞都是獨立的，因此也無法計算詞之間的相似度。

本文中使用到的是另一種分散式詞表示法。該方法以深度學習為基礎，將文本中的每個詞訓練成不同的短向量，并將它們集中在一個向量空間中。在這個空間里有距離的概念，如餘弦相似度（Cosine similarity）。這樣一來，我們就可以用詞之間的距離來表示相關度。

最常用的詞向量訓練法有兩種：CBOW 和 Skip-Gram。兩者的原理類似，實現的方式相反，在此不多贅述。

2.6 注意力機制

在序列到序列模型中，固定長度向量承擔編碼輸入序列的整個“含義”的負擔。由於語言的差異，這是一個非常難的問題。想像一下兩個幾乎相同的句子，二十個單詞長，只有一個單詞不同。編碼器和解碼器都必須細緻入微，以便將這種變化表示為空間中略微不同的點。

Bahdanau 等人（2014）在論文中提出注意機制。通過給編碼器提供一種“注意”部分輸入的方式來解決這個問題，而不是依賴於單獨的特定向量。解碼器可以根據輸入是句子的不同部分這一觀點，來考慮生成輸出。

Luong 等人(2015)在 Multiplicative attention 中提到了如下三種注意力的計算方法，本文會採用第一種。

$$\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \begin{cases} \boldsymbol{h}_t^\top \bar{\boldsymbol{h}}_s & \textit{dot} \\ \boldsymbol{h}_t^\top \boldsymbol{W}_a \bar{\boldsymbol{h}}_s & \textit{general} \\ \boldsymbol{v}_a^\top \tanh\left(\boldsymbol{W}_a [\boldsymbol{h}_t; \bar{\boldsymbol{h}}_s]\right) & \textit{concat} \end{cases}$$

注意力的計算與解碼器中的另一個前饋層有關。該前饋層將使用當前的輸入和隱藏狀態來創建新的向量，該向量與輸入序列的大小相同（即是固定的最大長度）。如圖 4 所示，該向量通過 softmax 處理以創建注意力權重，該注意力權重乘以編碼器的輸出以創建新的上下文向量，然後用於預測下一個輸出。

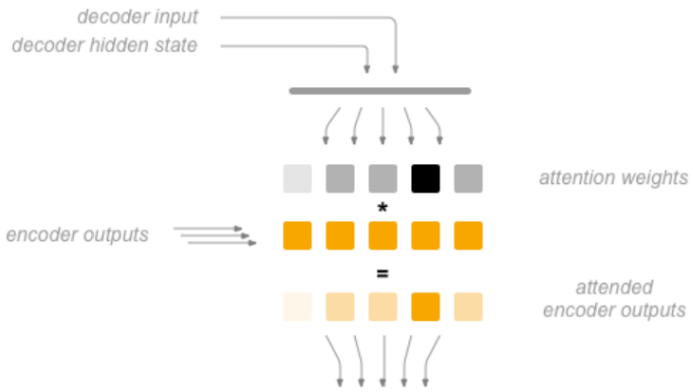

圖 4、注意力計算示意圖

2.7 基於注意力機制的 Seq2Seq 模型

將 Seq2Seq 模型拆解為兩部分理解，圖 5 中的上半部分是編碼器（Encoder），下半部分是解碼器（Decoder）。編碼器將許多輸入編碼到一個向量中，並從一個向量由解碼器解碼為多個輸出，因此可以擺脫序列順序和長度的約束。編碼序列由單個向量表示，在理想情況下，這一向量可以被理解為整個序列的“意義”。

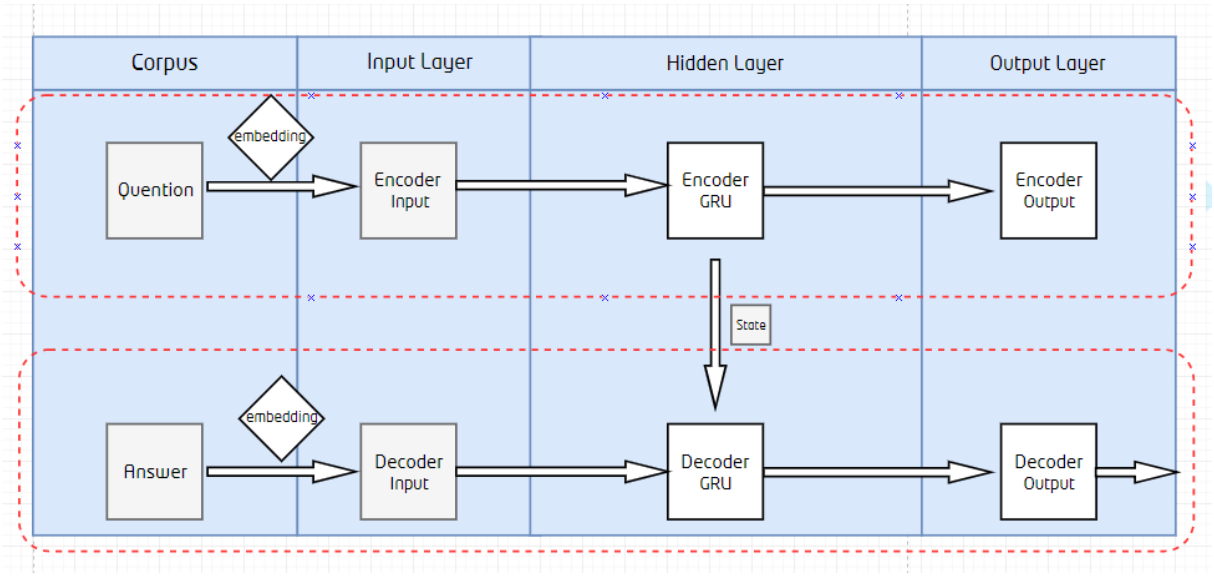

圖 5、Seq2seq 系統架構圖

2.7.1 GRU-Encoder

Seq2Seq 網路的 Encoder 本質上是一個 RNN，它為輸入句子中的每個詞輸出一個向量和隱藏狀態，並將隱藏狀態用於下一個輸入字。GRU 結構可以用來解決梯度消失的問題。圖 6 是本研究中採用的編碼器結構。

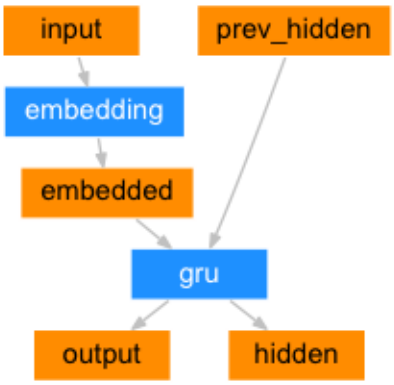

圖 6、Encoder 示意圖

參考來源（Pytorch tutorials[6]，2017）

2.7.2 GRU-Decoder with Attention

一般的注意力計算來自 Decoder 的隱藏狀態（h_t）和 Encoder 的狀態（h_s），經由正規化後最終會得到一個總和為 1 的值。a_t 代表下式中的 attn_applied。

$$a_t(s) = align(h_t, \bar{h}_s) = \frac{exp(score(h_t, \bar{h}_s))}{\sum_{s'} exp(score(h_t, \bar{h}_{s'}))}$$

Luong 等人（2015）提出了一些 Global Attention 的模型，來改進傳統注意力模型，他們之間的區別在計算注意力得分的方式（詳見 2.6）。本研究中用到的得分計算方式是 dot，顧名思義是兩個狀態之間簡單的乘積。

因此，我們的 Decoder 就是在和 2.7.1 中提到的一樣的 RNN 之後再插入這個 Attn 計算模組來計算注意力，並將這些權重應用到 Decoder 的輸出，以獲得上下文向量。圖 7 是本研究中所採用的解碼器結構。

6.Pytorch tutorials：Translation with a Sequence to Sequence Network and Attention, URL: https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html

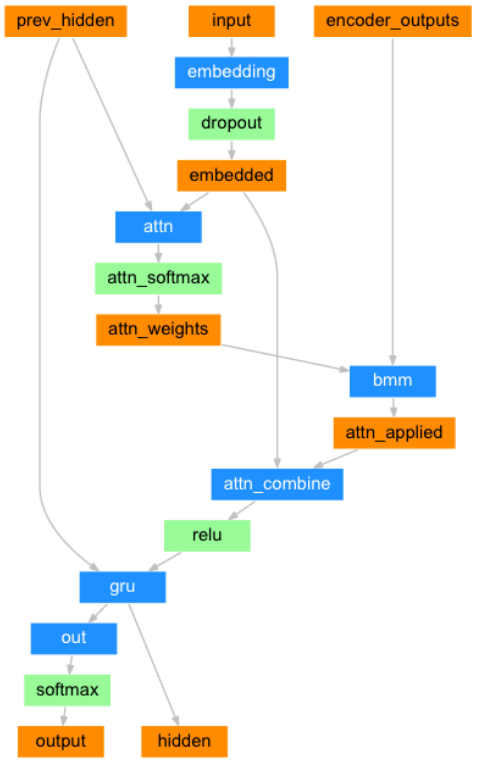

圖 7、Decoder with Attention 示意圖

參考來源（Pytorch tutorials，2017）

參、方法介紹

3.1 問題定義

在早年商用聊天機器人的研究中，我們不難發現一個現象，即在問句的回答中，很容易出現讓人覺得尷尬的回應。所謂尷尬指的是該回應合理，卻不合情。比如當一位顧客在反應購買的商品有問題時，不排除客服機器人會發生以下這種情況。

問：我買到商品為什麼是壞的？

答：壞了就算了吧。

以主觀判斷，問句屬於憤怒的情感表達，若回應是消極的情感，難免會讓人覺得不舒服。

如上所述，聊天機器人在實務中可能會產生不恰當情緒的回答。為了解決這一問題，我們建置了一個基於 seq2seq 模型的聊天機器人系統，並加入情感標籤元素，讓系統可以在特定的情感標籤下生成對應的答句，以提升聊天機器人對話的質量。

3.2 資料集觀察

為加入情感標籤元素，我們對實驗資料集進行了整理。在第十四屆競賽中，官方釋出了共計 60 萬筆問答對，每個問句和答句都匹配了一個情感標籤。

資料集的格式如下所示：

```
[[[post,post_label],[response,response_label]],[[post,post_label],[response,response_label]],...].
```

以資料集中第一筆資料為例：

[['现在 刷 朋友 圈 最大 的 快乐 就是 看 代购 们 各种 直播 。 。 。 。 。', '5'],
['卧 槽 我 也 是', '4']]

圖 8 中橫軸的 Q1 到 Q5 分別表示情感標籤為喜歡（Like），悲傷（Sadness），厭惡（Disgust），憤怒（Anger），快樂（Happiness）的問句，縱軸表示該情感問句對應不同情感答句的數量分佈。

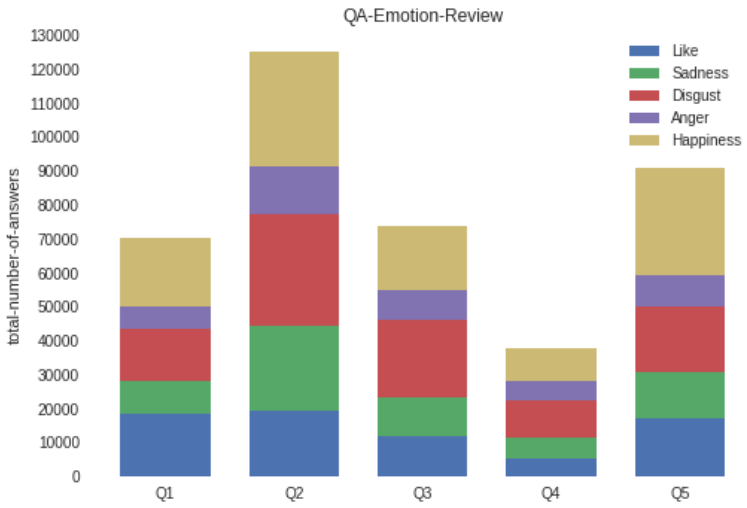

圖 8、針對不同問句情感統計得到的答句情感分佈圖

官方提供的資料中，對句子進行情感分類的情感分類器是使用一個 Bi-Lstm 神經網路進行訓練的，除了上面提到的五種情感，其實還有一種模糊的情感分類稱為其他（Others）。為了得到更直覺的統計效應，我們在進行統計時刪去了情感標籤為其他的資料。基於基本統計量的分析，針對每一種問句情感標籤，我們認為數量降序排行前三的回應是合適的。表 1 是統計得到的問句與答句合適情感匹配表，在對每一個問句情感我們都為它匹配了三個合適的答句情感。

表 1、問答句情感匹配表

問句情感	合適答句情感
Like	Like，Disgust，Happiness
Sadness	Sadness，Disgust，Happiness
Disgust	Like，Disgust，Happiness
Anger	Sadness，Disgust，Happiness
Happiness	Like，Disgust，Happiness

3.3 系統架構

本研究的系統架構如圖 9，可分成兩部分來看。第一部分是訓練不含情感標籤元素的傳統生成模型，以下簡稱 M1，第二部分是本研究提出的情感型生成模型，以下簡稱 M2。M1 是對照組，M2 是實驗組，實驗組是五個由不同情感標籤的資料集分別訓練的。模型的本質都是 Seq2Seq 神經網路，透過 M1 和 M2，會對同樣的問句產生兩種答句，最後再由人工的方式，對抽樣後的兩種答句進行評估。

首先我們刪去了情感標籤的元素，為 M1 的訓練做準備。M2 包含 5 個子模型，所以我們根據有效情感標籤（不含 Others），將資料集分成 5 份。M2 大致上可以看成 5 個小模型的整合，這五個小模型一樣也都透過本文所提到的 Seq2Seq 模型訓練，得到 5 個情緒模型，按照 Like，Sadness，Disgust，Anger，Happiness 的順序，記作 $M2_{Like}, M2_{Sadness}, M2_{Disgust}, M2_{Anger}, M2_{Happiness}$。每一個小模型都只能夠生成特定一種情緒的回答。

M1 和 M2 在演算法的應用上並無區別，只是 M1 中使用的語料會在透過前處理之後，將所有情感標籤去除。為了滿足 M1 與 M2 的對比實驗的科學性，兩個實驗使用到的訓練參數（RNN 層數，隱藏層數，學習率，迭代次數）與訓練資料集的數量將完全一致，最終結果展示時用到的語料也將相同，以達到更好的參照效果。

在對 M1 和 M2 生成的回應句的評估上，將會採用問卷的形式為 30 位受測人員進行測試。從 NTCIR 競賽的測試資料集中，我們準備對每種情緒隨機抽樣 4 句，共 20 句問句對應的生成答句來進行實驗。

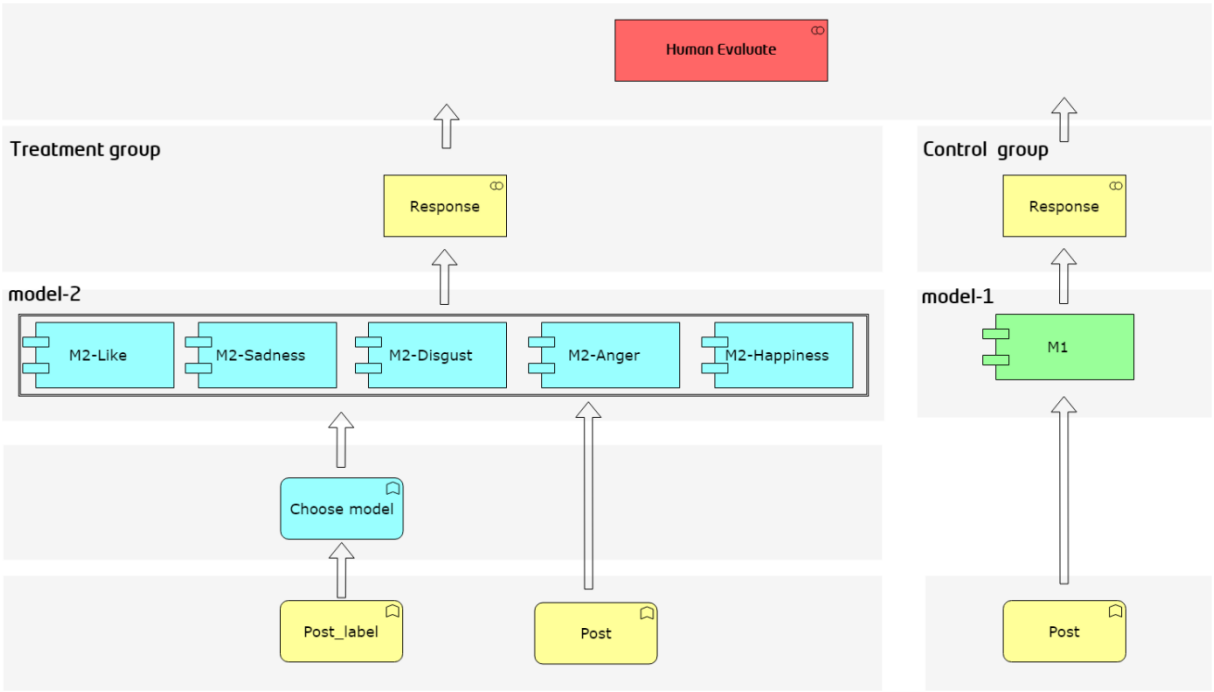

圖 9、系統架構圖

圖 10 展示了訓練資料的實際使用流向。對收集到的 60 萬筆訓練資料，我們從中隨機選取了五分之一作為 M1 的訓練資料。M2 則使用除去無效情感標籤之後，剩下的 46 萬餘資料，按照情感標籤的區別分別訓練。

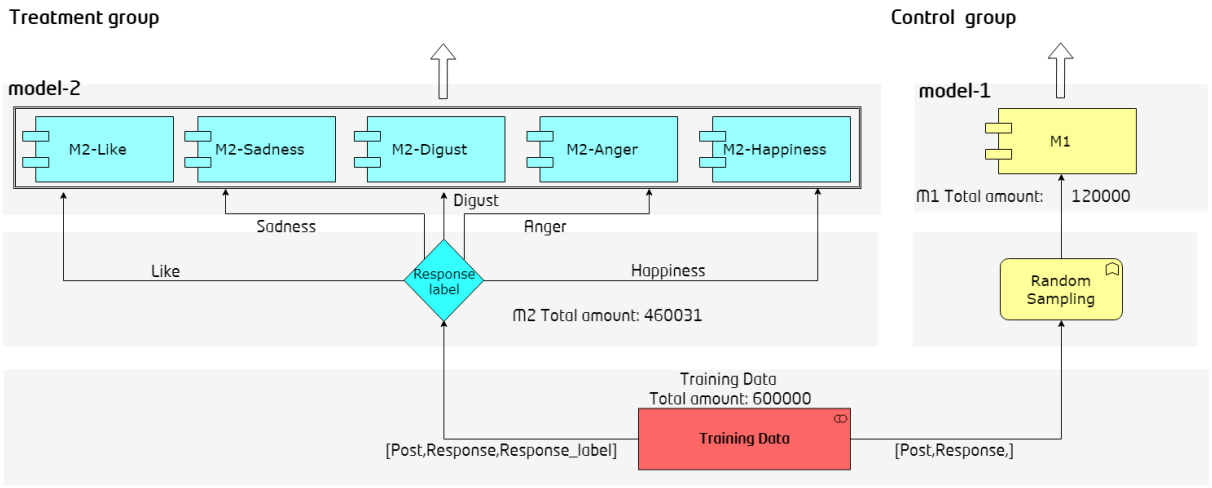

圖 10、訓練資料流向示意圖

M1 的輸入會是一個序列，輸出也是一個序列。M2 的輸入則可以看作是序列和情感標籤的組合。M2 輸入的情感標籤會對應到本文 3.2 中提到的匹配表，產生三種答句的候選情感，而這三種情感中，再隨機選取一種的情感模型作為真正要使用來生成答句的模型，以增加回答的多樣性。

肆、實驗設計與結果

4.1 實驗環境

以下將對本研究的環境進行介紹。基於成本和時間效益的考量，本研究的主要進程均在 Google Colab（Colabortary）上進行。Google Colab 是 Google 推出的免費 GPU 伺服器，預裝 Jupyter Notebook 環境，可以和 Google Drive 聯結，並且完全在雲端運行。圖 10 是 Google Colab 的環境配置圖。

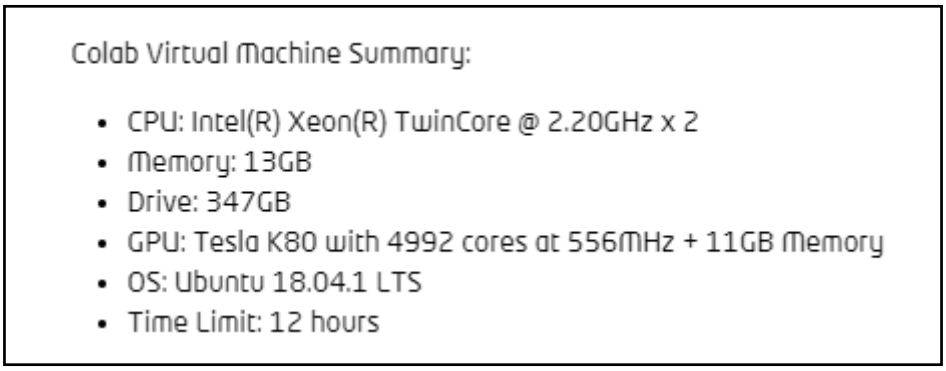

圖 10、Colab 環境配置圖

除了上述的硬體環境，本研究還用到一些 Python 套件，這些套件也在一定程度上推動了這次實驗的進度。表 2 是本研究中用到的套件列表。

深度學習框架	資料框	矩陣運算	詞向量	敘述統計	視覺化
Pytorch	Pandas	Numpy	Gensim	NLTK	Pyplot

表 2、使用套件表

本研究中使用到的語料來自 NTCIR 大會官方。在第十四屆的比賽中，大會方提供了共計 60 萬筆簡體中文微博用戶的問答對（post & response），問答對中每一筆資料還有一個基於句子本身的客觀情感標籤。情感標籤分類器是一個簡單 Bi-Lstm 模型，它接受來自 NLPCC 情緒分類挑戰任務的資料集訓練。情感標籤共分六類（Other,Like,Sadness,Disgust,Anger,Happiness）。

此外，我們還搜集了第十三屆比賽中用到的資料集，其格式與上述資料集相同，共計約 112 萬筆（1,119,207），斷詞後不重複的詞數有約 9.9 萬筆（98,627）。表 3 是為本研究收集的資料集之對照表。

表 3、實驗資料集對照表

資料集	問答數（筆）	詞數（個）	問句基本統計量（詞）	答句基本統計量（詞）
2017 train_data	1,119,207	98,627	max:33.11 median:8.5 min:1	max:33.14 median:10.7 min:2
2018 train_data	600,000	93,969	max:83.17 median:13.9 min:0	max:377.11 median:7.4 min:0

受限於實現和硬體設備，本次實驗中真正使用到的資料是 NTCIR-14,2018 年的共計 60 萬筆的資料集，其中前 12 萬筆用於 M1 的訓練。M2 則使用除情感標籤 0（即 Others）之外的共計 460031 筆資料訓練。

4.2 實驗設計

在模型訓練時期，M1M2 都使用相同的訓練參數進行訓練（GRU 層數=1，隱藏層數=512，學習率=0.0001，迭代次數=5000）。

問卷部分，本研究乃以微博和簡體中文的使用者作為受測對象，以探討應用兩種不同模型的文本生成質量。30 位受測對象中除了 19 位在台灣交換的，來自中國大陸地區的碩士在讀學生，也委請了 11 位年齡介於 18 歲到 54 歲，且使用微博兩年以上的用戶，以增加樣本的豐富性。受測人員將會從語言流暢度（fluency），問答相關度（coherence）和情感表達度（emotion）三個面向對生成句進行評估，且三個面向的權重賦值相等，針對每一個面向的評估都是從兩種回應句中擇優選擇，優勝句計 1 分。對三個面向有如下定義：

語言流暢度：指答句在表達時的邏輯，語法是否符合常識規範

問答相關度：指答句與問句所表達的內容是否相關

情感表達度：指答句所內涵的情感表達針對問句是否合理或合適

從每個評量面向得分的計算方式如下：

$$S_i = \sum_{q=1}^{20} \sum_{h=1}^{30} a_i(h,q) \qquad i \in [fluency, coherence, emotion]$$，S_i 的值域為 [0,600]

其中，$a_{fluency}(h,q)$ 表示，第 h 人在第 q 個問句對語言流暢度這個面向的給分（0 或 1），因此 $S_{fluency}$ 代表 M2 在語言流暢度的總得分。同理 $S_{coherence}$，$S_{emotion}$ 分別表示 M2 在問答相關度和情感表達度的得分。

最終 M2 總平均得分計算方式如下：

$$Avg_{M2} = \frac{(S_{fluency} + S_{coherence} + S_{emotion})}{3}$$

綜上，當從三個面向都由任一方的生成回應勝出時，會獲得本問卷滿分 60 分。

4.3 實驗結果

本次問卷共計回收 30 份，表 4 是問卷的結果統計。

表 4、問卷結果統計表

	$S_{fluency}$	$S_{coherence}$	$S_{emotion}$	Avg
M1	255	199	247	233.67
M2	345	401	353	366.33

在語言流暢度方面，M1 得 255 分，M2 得 345 分。問答相關度方面，M1 得 199 分，M2 得 401 分。情感表達度方面，M1 得 247 分，M2 得 353 分。就總平均得分而言，模型 1 得 233.67 分，模型 2 得 366.33 分。

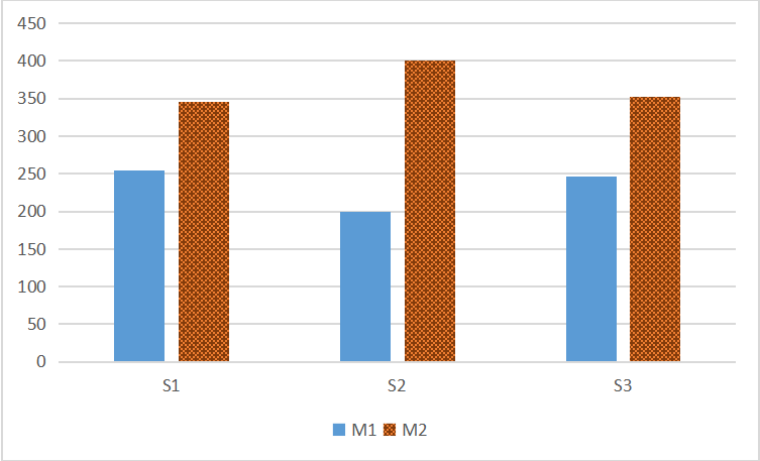

圖 11、問卷實驗結果對比圖

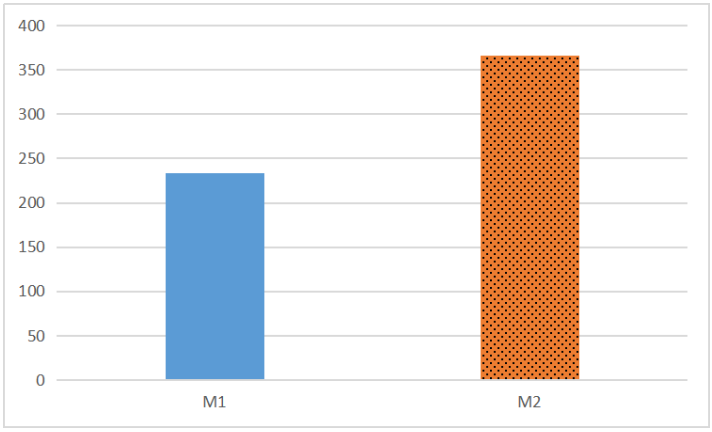

圖 12、問卷實驗結果對比圖（總平均分）

從圖 11，圖 12 可以觀察到，在三個面向上，M2 的結果都略優於 M1 的結果,三個面向中，優勝表現最顯著的是問答相關度方面，其次是情感表達度，而語言流暢度雖排在最後，但也有一定程度的領先於 M1，整體來說，M2 優於 M1。

在實驗早期的預想中，由於 M2 使用的資料集都由情感標籤特別進行分割，因此我們猜測情感表達度（$S_{emotion}$）方向應該會得到最顯著的優勝表現，而 $S_{fluency}$ 和 $S_{coherence}$ 的部分應該與 M1 的表現不相上下。從實際呈現的結果上來看，$S_{emotion}$ 確實體現了一定的差異，但 $S_{coherence}$ 的回饋令人意外。初步猜測其原因來自 M2 的訓練使用分割後的資料集，在詞表示上可能不如 M1 全面，導致生成序列中會用到大量和輸入序列中相同的詞，因而使受測人員產生相關度更高的感覺。受時間限制，實驗組和對照組的模型都還沒得到充分的訓練，因此 M1M2 都還有提升的空間，目前取得的結果也尚待更進一步的實驗分析。

伍、結語與未來發展

本研究建置了一個基於 Pytorch 深度學習框架的 seq2seq 文本生成系統，並加入情感標籤為分類要素，初步實現了情感型聊天機器人的構建。在三個面向的衡量指標中（語言流暢度，問答相關度，情感表達度），本研究所使用的系統表現均勝過傳統深度學習聊天機器人，為未來商用聊天機器人的建設提供一個可以參考的方向。

受限於硬體設備和時間的因素，實驗中使用的模型訓練參數並非最佳，因此系統的表現任留有進步的空間。此外，少數情感標籤下的語料（如 Anger）並未被充足利用，未來也許能針對某些特定情感標籤生成回應，以供特殊需求的聊天機器人系統使用。

在 NTCIR 大會官方提供的訓練資料集中，其對句子的客觀情感標籤的判別準確度也有待提升，若對句子的情感判別能夠更加準確，也許本文中提到的模型其表現會更加亮眼。

陸、參考文獻

1. Bahdanau,D., Cho, K.,Bengio,Y. “Neural Machine Translation by Jointly Learning to Align and Translate”, ICLR, 2016.
2. Cho,K., Merrienboer,B., Gulcehre,C.,Bahdanau,D., Bougares,F.,Schwenk,H.,Bengio,Y. “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation”, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) ,2014 .
3. Zhao,H.,Du,Y.,Li,H. “SG01 at the NTCIR-13 STC-2 Task” ,NTCIR-13, 2017.
4. Jalaj Thana. Python Natural Language Processing , Birmingham,Munbai, Packt publishing Ltd.,2017.
5. jozefowicz et al. “Exploring the Limits of Language Modeling” ICML, 2016.
6. Kim Y. “Convolutional neural networks for sentence classification”. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) , Pages 1746-1751, 2014.
7. Shang, L., Sakai,T., Li,H., Higashinaka,R., Miyao,Y., Arase,Y.,M.Nomoto. “Overview of the NTCIR-13 Short Text Conversation Task.” NTCIR-13, 2017.
8. Luong,M., Pham,H., Christopher D. Manning. “Effective Approaches to Attention-based Neural Machine Translation” Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing , Pages1412-1421, 2015.
9. Goyal,P., Pandey,S., Jain,K. Deep Learning for Natural Language Processing: Creating Neural Networks with Python. Apress publishing, 1st ed, 2018.
10. Sutskever,I., Vinyals,O., Le,Q. “Sequence to Sequence Learning with Neural Networks”,NIPS'14 Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, Pages 3104-3112, 2014.
11. Wu, Y., Wu,W., Xing,C., Zhou,M., Li,Z. “Sequential Matching Network: A New Architecture for Multi-turn Response Selection in Retrieval-based Chatbots”. arXiv:1612.01627, 2017.
12. Zaccone,G.Getting started with tensorflow., Birmingham,England, Packt publishing Ltd. ,2016.