從常見問答集自動產生客服問答機器人

歐亭昀
資訊工程學系
國立中央大學
桃園，台灣
outingyun7897@gmail.com

周冠玲
資訊工程學系
國立中央大學
桃園，台灣
lily12457@gmail.com

陳弘軒
資訊工程學系
國立中央大學
桃園，台灣
hhchen@g.ncu.edu.tw

***Abstract*—本論文提出一個能從企業的常見問答集自動生成問答機器人的模型，並與微軟 Azure 雲端服務平台提供的 QnA Maker 做比較。我們從中央大學計算機中心取得使用者實際詢問服務台的文字紀錄及該中心網站上的常見問題集，以此做為實驗資料。實驗結果顯示，我們的模型相較於 QnA Maker 在正確答案排在前十名、平均正確答案排名、平均倒數排名、及平均折損累計都有相對優異的表現 (表三)，而我們的成果開源於 https://github.com/OuTingYun/Customer-Service-Chatbot-from-FAQ。**

1. 導論

客服中心經常是企業面對客戶的第一線。客服管道除了傳統的電話，如今還包含電子郵件、即時通訊軟體、臉書等方式。多元的通訊方式使得客服人員的工作負荷居高不下。在此同時，近年自然語言處理技術大幅進步，聊天機器人逐漸能模擬人類的對話模式與真實的人類交談。本論文即以此為切入點，欲實際驗證聊天機器人能否輔助客服中心完成客戶的需求。

為了使目標更聚焦，我們不要求聊天機器人能閒聊 (chit-chat)，僅要機器人回答與業務相關之詢問。同時，我們不處理語音轉文字的訊號處理問題，假設所有的對話均透過文字方式進行。為了方便企業使用，我們希望對話機器人僅需企業提供少量的資料即可回答相關問題。具體而言，企業僅需提供常見問答集，我們的對話機器人即可判斷使用者的問題符合常見問答集當中的哪個問題，並回覆相應的答案。

我們採用的模型同時包含了 Okapi BM25 (以下簡稱 BM25) [1] 搜尋相關性評分、Word2Vec 字詞嵌入表示法提取模型 [2]、及 Bidirectional Encoder Representations from Transformers (以下簡稱 BERT) 語言模型 [3]。為了方便其他研究團隊重製我們的實驗或擴充功能，我們將原始碼開源。我們從中央大學電子計算機中心取得使用者的實際詢問紀錄文字檔及常見問題集，[1]，並人工標註每個使用者的問題屬於常見問題集中的哪一題。我們將提出的模型與微軟雲端運算服務 Azure 的 QnA Maker [4] 服務做比較，實驗結果顯示：我們的模型在命中率 (Hit Rate)、平均排名 (Average Rank)、及平均倒數排名 (Mean Reciprocal Rank，簡稱 MRR) 三個指標的結果均優於 QnA Maker 所提供的服務，顯示本模型的優越性。

[1]https://www.cc.ncu.edu.tw/page/qna

2. 相關研究

本節回顧我們的模型所使用的幾項技術 (包括：TF-IDF 相關度計算、Word2Vec 模型、及 BERT 模型) 及聊天機器人的相關研究。

A. TF-IDF、Word2Vec 及 BERT 模型

搜尋引擎的結果排序是搜尋引擎中很重要的部分。排序的其中一個重要因素是使用者查詢的字串與網頁內容的相關度，其中，TF-IDF 及其延伸方法可能是最常見的相關度分數計算方式 [5]。這個方法的核心精神有二：其一，若搜尋的字串或子字串在某文件中出現的次數愈多，則該文件與搜尋字串愈相關；其二，若某字詞出現在大量的文件中，則該字詞難以有效分辨不同文件，故搜尋時應降低其重要性。在計算出一篇文章中每個詞彙的 TF-IDF 分數後，可將這些分數串成一個陣列，並計算不同文章所相應陣列間的餘弦相似度 (cosine similarity) 來代表文章間的相似度或搜尋字串與文章間的相似度。由於本研究的目標是判斷使用者的問句屬於官方常見問答集中的哪一個問題，因此可使用 TF-IDF 或其延伸方法判斷使用者的句子與常見問答集中的每一則問題之間的相似度，以此做為判斷的線索之一。然而，TF-IDF 搭配餘弦相似度的方式難以對同義字進行處理，例如：wardrobe 和 closet 分別是英國人和美國人對於櫃子的稱呼方式，但當使用者以 wardrobe 做為搜尋的關鍵字時，若僅採用上段敘述的方式將難以找到文章中僅包含 closet 而不包含 wardrobe 的文章。我們的方法中包含 TF-IDF 的一種著名延伸模型 BM25 相似度，細節將在第 3節中描述。

Word2Vec 模型是一個將詞彙轉化為固定長度向量 (稱為詞向量) 的方法，此方法能有效保留字詞間的語意關係或句法關係，其中最著名的例子可能是 king : queen ≈ man : woman [2]。採用此模型有多個好處：第一，能有效處理上段中敘述的「同義字」問題，原因是 Word2Vec 模型通常可讓同義字產生類似的詞向量，故搜尋 wardrobe 時，包含 closet 的文章也容易出現在搜尋結果列表的前段。第二，Word2Vec 採用自監督式學習 (self-supervised learning)，任何文章不需要再經人工標注即可做為輸入，故非常容易取得大量的訓練資料。第三，已有不少研究團隊公開經大量文件訓練而得到的 Word2Vec 模型，直接採用這些模型可減少我們的開發及訓練時間。然而，Word2Vec 模型有單詞歧義 (ambiguation) 的問題，即：同一個字詞在不同的語境

下可能有不一樣的意義 (如：「蘋果」可做為科技公司也可做為水果)，但 Word2Vec 只會對每個字詞給予單一個詞向量。我們的方法中包含 Word2Vec 模型來處理一部份同義字的問題，細節將在第 3節中描述。

BERT 是一個基於變換器的雙向編碼器表示技術 (Bidirectional Encoder Representations from Transformers)。給定同一個詞彙，BERT 會依上下文的用字遣詞給予不同的詞向量，因此沒有單詞歧義的問題。BERT 模型及其延伸或相關模型在許多自然語言任務取得優異的成果，使用預訓練 (pre-trained) 的 BERT 模型再針對下游任務進行微調 (fine tune) 已經幾乎成為今日自然語言任務的標準手法。BERT 的預訓練任務有兩個，分別為遮掩語言模型 (Masked Language Model) 和下一句預測 (Next Sentence Prediction)。我們的方法中將利用 BERT 模型為使用者的句子進行分類，細節將在第 3節中描述。

B. 對話機器人

對話機器人按其建置目的大致可分為兩大類：任務導向 (task-oriented) 及開放話題閒聊型 (open domain chit-chat) [6]。開放話題閒聊型機器人通常需要大量的對話資料搭配百萬個參數甚至數十億個參數的神經網路模型 (e.g., [7], [8])，不但訓練成本極高，也和我們的目標不同。任務導向的對話機器人則因為符合應用場域的對話紀錄較少，故通常需要混搭不同的技術 (e.g., 資訊檢索)、設置額外的限制 (e.g., 僅能按固定的模板回覆)、或採用額外的輔助資源 (e.g., knowledge base) 加以輔助 [9], [10]。

在 FAQ bot 這方面，過去的作品多半採用資料檢索的方式比對出與使用者的發問最相似的問題 [11]，但這些作品多半為特定情境設計，並非提供一套工具或平台自動從 FAQ 產生對話機器人。同時，我們發現微軟的 Azure 平台提供的 QnA Maker [4] 雲端服務與我們的計畫目標十分相似，二者均只需企業提供 Q&A 資料集即可自動生成對話機器人回覆使用者的問題，QnA Maker 採用資訊檢索技術搭配 WordNet 語義網，並額外以使用者在 Bing 的搜尋行為訓練句向量 (sentence embedding) 產生器。相較於我們開發的工具，QnA Maker 最大的優勢是使用的便利性，僅需透過瀏覽器即可體驗從 FAQ 自動產生的問答機器人；此外，QnA Maker 具備簡單的閒聊機制 (如：打招呼)。然而，高度封裝及自動化的介面使得企業難以客製化或擴充機器人的核心機制。

3. 對話機器人設計

A. 以 BM25 檢索及 Word2Vec 配對使用者問題與標準問題

我們將 FAQ 中出現的問題稱為「標準問題集」。我們計算使用者的問題與標準問題集中的問題的 BM25 分數，選出標準問題集中的哪些問題在用字上最接近使用者的句子。若使用者的句子為 q，標準問題集中的一個問題為 d_i，兩者間的 BM25 計算方式如式 1。

$$s(q, d_i) \propto \sum_{\forall w \in q} TF(w, d_i) \times IDF(w) \tag{1}$$

其中，$TF(w, d_i)$ 是 d_i 中的每個字詞 w 的 TF 分數 (如式 2 所示)，$IDF(w)$ 是 w 的 IDF 分數 (如式 3所示)。

$$TF(w, d_i) = \frac{f_{w,d_i} \times (k+1)}{f_{w,d_i} + k\left(1 + b\left(\frac{|d_i|}{|D|_{\text{avg}}} - 1\right)\right)}, \tag{2}$$

其中 f_{w,d_i} 是 of w 在 d_i 中的出現次數，$|d_i|$ 是標準問題 i 的字數，$|D|_{\text{avg}}$ 是每個標準問題集的平均字數，$k = 1.2$ 和 $b = 0.75$ 兩個參數值則採 Lucene 的建議。[2]

$$IDF(w) = \log\left(\frac{N - n_w + 0.5}{n_w + 0.5}\right), \tag{3}$$

其中 N 標準問題集的題數，n_w 是 w 的檔案頻率 (document frequency)。

除了使用 BM25，我們也採用 Word2Vec 來找到意義相近的詞彙。我們採用中文維基百科資料庫中的 20201020 的備份當作訓練資料，並使用結巴分詞器進行中文斷詞。[3]。

當使用者提出問題時，問題同樣經結巴分詞器切割，再將切割後的斷詞依序從 Word2Vec 模型找到三個相似詞，將得到的所有相似詞和原本的斷詞放入共同與標準問題集中的問題計算 BM25 分數做相似度匹配。

B. 以 BERT 配對使用者問句與標準問題

除了採用資訊檢索搭配 Word2Vec 技術，我們也將使用者問句與標準問題的配對設計成多元分類問題。我們採用 Google 官方的 BERT-Base-Chinese 預訓練模型，[4]並將中大計中官方問題中各個標準問題的文字內容當作訓練的特徵 (feature)，對應的題號是要預測的目標類別 (target class) 產生訓練資料，以此訓練資料微調 (fine tune) 模型。

在輸入使用者問句後，會先將此問題的特徵向量輸入我們的模型，此模型會依照輸入的特徵向量來判斷該問題最有可能對應的標準問題的題號，最後再將對應該題號的標準答案作為輸出。

C. 整合 BM25、Word2Vec、及 BERT 共同配對使用者問題與標準問題

在本論文中，我們以中央大學計算機中心的 FAQ 及使用者在 LINE 上的詢問做為實驗資料。初期的實驗 (細節將在第 4.3節中說明) 發現：給定使用者的問句，若將所有的標準問題按整合 BM25 與 Word2Vec 的方式 (方法一) 做配對並按分數由高至低排列，正確的標準問題通常會被排在前十名，採用 BERT 的分類模型 (方法二) 時，正確的標準問題出現在前十名的比例反而不如

[2]https://lucene.apache.org/core/8_9_0/core/org/apache/lucene/search/similarities/BM25Similarity.html

[3]https://github.com/fxsjy/jieba

[4]https://storage.googleapis.com/bert_models/2018_11_03/chinese_L-12_H-768_A-12.zip

方法一高。然而，一旦方法二能將正確的標準問題排在前十名，其排名通常非常前面，平均而言甚至高於方法一。換言之，當方法二給予某標準問題較高的配對分數時，我們應該傾向於相信此結果，如果方法二配對的最高分分數並不高，則應該相信方法一的結果。

有了上述的觀察，我們擬定了將兩個方法融合的策略：當使用者輸入問題時，會將使用者問題分別輸入到方法一和方法二得到結果一和結果二。先以結果二作為評判依據，假使結果二的分數高於特定分數，就以結果二作為最後的輸出。假使結果二的分數低於此分數，就以結果一做為最後輸出。

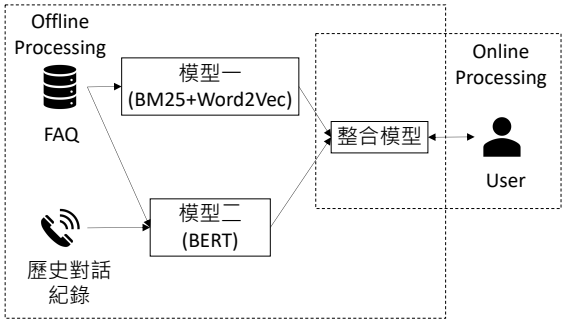

圖 1. 整體架構圖

最後的整體模型架構如圖 1所示。系統在線下產生模型一 (BM25+Word2Vec)、模型二 (BERT)、及兩者之整合模型，模型上線後，即可與真人進行對話。模型的原始碼公佈於https://github.com/OuTingYun/Customer-Service-Chatbot-from-FAQ。

4. 實驗

A. 實驗資料

本研究中所使用訓練資料的標準問題與答案，是採用中央大學計算機中心官網上常見問題中的 52 題問答。而測試資料的使用者問題與回答，則是從中央大學計算機中心 LINE 官方帳號中，挑出 54 題使用者問題，與管理者的回答。

實驗中，我們將使用者問題放入模型中，模型會回傳各個標準問題的配對分數，將標準問題按配對分數由高至低排列後，再根據正確答案的排名比較各個模型的優劣。

B. 評估方法

本論文使用四種檢驗指標，包括：平均排名 (Average Rank，簡稱 AvgRank)、Top-k 準確率 (top-k accuracy，簡稱 Acc@k)、平均倒數排名 (Mean Reciprocal Rank，簡稱 MRR)、及平均折損累計增益 (Average Discounted Cumulative Gain，簡稱 AvgDCG)。

若給定使用者的問句為 q_i，其相應的標準問題之題號為 r_i ($1 \leq r_i \leq Q$，Q 是標準問題集的題數)，$rank_m(q_i, r_i)$ 是模型 m 對問題 q_i 回傳的序列中 r_i 的排序，則平均排名其計算方式如式 4，數值愈小代表模型 m 的效果愈好。

$$AvgRank = \frac{1}{N}\sum_{i=1}^{N} rank_m(q_i, r_i), \tag{4}$$

其中 N 是使用者詢問的問題數。

Top-k 準確率指的是模型回傳的序列的前 k 名中包含相應的標準問題之題號的比例，故數值愈大愈好，其計算方式如式 5。

$$Acc@k = \frac{1}{N}\sum_{i=1}^{N} I(rank_m(q_i, r_i) \leq k), \tag{5}$$

其中 $I()$ 為指示函數 (indicator function)，當其參數為真時函數回傳 1，否則回傳 0。

MRR 是一種用來評估資料檢索查詢回應答案品質的衡量方法，其將標準答案在搜索結果中排名的倒數作為其準確度，再對所有問題取平均，MRR 表示式如式 6。

$$MRR = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{rank_m(q_i, r_i)} \tag{6}$$

最後，平均折損累計增益鼓勵將正確的題號放在回傳序列的前面，其表示式為式 7。

$$AvgDCG = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{Q} \frac{I(rank_m(q_i, r_i) = j)}{\log_2(j+1)} \tag{7}$$

C. BM25 + Word2Vec vs BERT

本節比較我們的方法一 (BM25 + Word2Vec) 及方法二 (BERT) 的效果，我們比較兩種方法回傳的排序的前十名的效果。其中，AvgRank@10、MRR@10、及 AvgDCG@10 均只有應被配對的標準問題排在前十名時才會被納入計算，若應被配對的標準問題排在第十一名或更後面，該次結果不會被納入此表中 AvgRank@10、MRR@10、及 AvgDCG@10 的計算中。

表 I給出了兩個方法的比較。從 Acc@10 可看出：方法一比較容易將正確答案排在前十名。然而，從其他幾個指標可看出：若只考慮正確答案真的出現在前十名的案例，BERT 會把正確答案排在較前面的位置。這項觀察讓我們提出第 3.3 節的整合模型。

	BM25+W2V	BERT
Acc@10 (愈大愈好)	**87.03%**	70.37%
AvgRank@10 (越小越好)	3.45	**2.42**
MRR@10 (越大越好)	0.47	**0.70**
AvgDCG@10 (越大越好)	0.60	**0.77**

表 I
模型一、模型二在前 10 名的評估結果

D. 與微軟 QnA Maker 的比較

微軟的 Azure 雲端平台所提供的 QnA Maker 服務與我們的問答機器人的目的相同，二者均能從常見問答集自動產生問答機器人。本節比較此商用軟體與我們的模型。

表 II 列出我們的模型與 QnA Maker 在各項指標的結果，我們的整合模型在各項指標均優於 QnA Maker。

另外，QnA Maker 有時候僅能回傳常見問答集中一部份的結果，若正確答案不在回傳的清單中，則我們無法正確計算一部份的指標。在我們的實驗中，有兩個使用者的問題出現這種情況，表 II中最後一列的 Missing 紀錄 QnA Maker 有兩次這種狀況，此時，QnA Maker 的 AvgRank、MRR、及 AvgDCG 我們給予所有可能結果的範圍，但即使給予 QnA Maker 最樂觀的估計，其結果仍不如我們的整合模型。

	BM25+ W2V	BERT	BM25+ W2V+BERT	QnA maker
Acc@3	57.41 %	55.56%	**62.96**%	53.70%
Acc@5	68.52 %	64.81%	**70.37**%	66.67%
Acc@10 (愈大愈好)	**87.03** %	70.37%	**87.03**%	85.19%
AvgRank (愈小愈好)	5.67	10.78	**5.50**	[5.89 - 7.05]
MRR (愈大愈好)	0.42	0.50	**0.53**	[0.435 - 0.434]
AvgDCG (愈大愈好)	0.55	0.60	**0.63**	[0.562 - 0.560]
Missing	0	0	0	2

表 II
三個模型的評估結果

5. 結論與未來展望

本論文介紹了一個從常見問答集自動生成客服問答機器人的模組，此模組整合了傳統資訊檢索 TF-IDF 的查詢方式及較近期的 BERT 語言模型。我們以中央大學計算機中心的官方常見問答集及使用者實際詢問官方 LINE 的文字紀錄做為實驗資料，實驗結果顯示：整合模型能有效地找到官方問答集中的哪些問題較符合使用者所敘述的問題。目前整合模型沒有”以上皆非”這個選項，每個使用者問題都能在官方問答集中找到較符合的問題。

若將我們的整合模型與微軟 Azure 雲端服務的 QnA Maker 商用軟體做比較，我們的整合模型有幾點優於 QnA Maker。第一，我們的方法第一次就回答正確的比例較 QnA Maker 高 (如第 4節所示)。第二，由於 QnA Maker 的高度封裝，開發人員不容易更改 QnA Maker 產生的聊天機器人的對話邏輯；我們公開原始碼於 GitHub，利於開發人員直接更換或擴充模組。第三，QnA Maker 為商用付費軟體。但 QnA Maker 也有幾個較優秀的地方：第一，QnA Maker 僅需網頁及簡單的電腦操作能力即可試用，門檻較低；第二，QnA Maker 具備簡單的閒聊機制，我們的模組目前完全不具備此功能。

目前各種技術所產生的聊天機器人其對話能力仍相當有限，特別是較長的對話中，機器人容易只專注於當下的句子而忽略了整段對話的脈絡。然而，本論文僅針對 Q&A 客服對話為目標，Q&A 客服對話通常較短且需求較明確，而對話中出現頻率少的字詞通常為判斷問題的關鍵，所以文章前後關係的重要度較低，因此，就現階段的對話機器人技術而言，或許 Q&A 客服對話是一個比較有機會自動化的應用。

我們目前僅用較為直觀的方法將兩個模型做整合，並未使用複雜的組合方式，因此在未來工作方面，我們希望能以加權整合的方式對兩個模型做合併，並且希望能在其他語言 (e.g., 英文) 上測試本模組。這個方向最大的難處在於資料的取得：英文的常見問答集來源很多，但真實的使用者詢問紀錄則很難得到。換言之，雖然我們能夠從英文的常見問答集自動生成對話機器人，但評估機器人的能力則比較困難，因為我們沒有實際的使用者詢問紀錄做測試。這方面或許可藉由招募實驗受測人員實際與機器人對話來做測試。

References

[1] S. E. Robertson and H. Zaragoza, “The probabilistic relevance framework: BM25 and beyond,” *Foundations and Trends in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.

[2] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in *International Conference on Learning Representations (ICLR), Workshop Track Proceedings*, 2013.

[3] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 2019, pp. 4171–4186.

[4] P. Agrawal, T. Menon, A. Kamel, M. Naim, C. Chouragade, G. Singh, R. Kulkarni, A. Suri, S. Katakam, V. Pratik, P. Bansal, S. Kaur, A. Duggal, A. Chalabi, P. Choudhari, S. R. Satti, N. Nayak, and N. Rajput, “QnAMaker: Data to bot in 2 minutes,” in *Companion of the Web Conference*. ACM / IW3C2, 2020, pp. 131–134.

[5] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. Cambridge University Press, 2008.

[6] D. Jurafsky and J. H. Martin, *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition, 2nd Edition*, ser. Prentice Hall series in artificial intelligence. Prentice Hall, Pearson Education International, 2009.

[7] S. Roller, E. Dinan, N. Goyal, D. Ju, M. Williamson, Y. Liu, J. Xu, M. Ott, K. Shuster, E. M. Smith *et al.*, “Recipes for building an open-domain chatbot,” *arXiv preprint arXiv:2004.13637*, 2020.

[8] D. Adiwardana, M.-T. Luong, D. R. So, J. Hall, N. Fiedel, R. Thoppilan, Z. Yang, A. Kulshreshtha, G. Nemade, Y. Lu *et al.*, “Towards a human-like open-domain chatbot,” *arXiv preprint arXiv:2001.09977*, 2020.

[9] T.-H. Wen, D. Vandyke, N. Mrksic, M. Gasic, L. M. Rojas-Barahona, P.-H. Su, S. Ultes, and S. Young, “A network-based end-to-end trainable task-oriented dialogue system,” *arXiv preprint arXiv:1604.04562*, 2016.

[10] H. Al-Zubaide and A. A. Issa, “Ontbot: Ontology based chatbot,” in *International Symposium on Innovations in Information and Communications Technology*. IEEE, 2011, pp. 7–12.

[11] B. AbuShawar and E. Atwell, “ALICE chatbot: Trials and outputs,” *Computación y Sistemas*, vol. 19, no. 4, pp. 625–632, 2015.