

資料科學基礎與實務Part II 專題

人工智慧於 模擬人類 對話系統設計

組員



李瓊妮 D0813259

周天愛 D0845026

藍于婷 D0845265

謝語宸 D0845323

蔡佩蓉 D0813157

專題目標



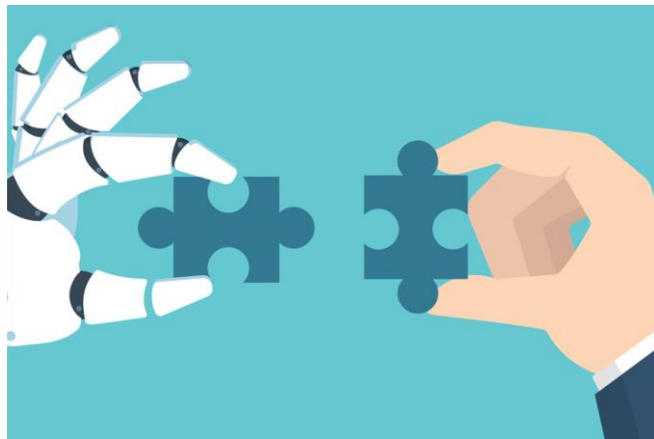
以自然語言處理及專家系統為架構去設計一套仿人類的對話系統，以 C 語言來設計對話程式，資料庫的建構為程式核心。

在輸入對話的同時，系統會判斷單字詞及文句是否為資料庫內所設定相符合的資料，如果樣式比對成功就會輸出對應的回答句，模擬人類之對話反應來設計資料庫，設計出具有個人化及互動導向的對話系統。

研究背景與動機



現代電腦網路科技的進步下，人工智慧已融入人類社會的生活中，為了協助人類或者取代人類的人工智慧設計也都開發出無數產品，人們的日常生活中大量地使用人工智慧的技術產品，卻不了解相關的技術議題和影響力，而未來是否會發展出具有強人工智慧的技術，會不會誕生出不受人類的控制的思維，要去克服的問題很多，等著我們去解決。



美國 MIT伊麗莎(Eliza)程式



論文主要參考外國 MIT 人工智慧實驗室所研究的伊麗莎(Eliza)程式。

程式核心為樣式對比(pattern matching)，內部預先設定好對話反應樣式(dialog patterns)，讓使用者在使用時從對話互動中有如同真人在對話的反應錯覺，對話樣式的資料庫設計為研究重點，而 Eliza 程式所使用的程式語言為 LISP而程式設計概念是以模仿為主要核心。



一段簡易的對話與 ELIZA 互動模式的結果



C

Well, my boyfriend made me come here.

Your boyfriend made you come here.

R

ELIZA

C

I am unhappy.

Do you think coming here will help you not to be unhappy?

R

ELIZA

研究方法與目標



模擬人類對話系統之設計作為主要的研究，收集與人類對話之特性 相關及自然語言處理的文獻，去設計一個具有互動及獨立意識之對話系統

接下來會介紹

研究
理論

系統架構
及
設計

模擬
及
測試

結
論



研究理論



自然語言是什麼



自然語言(Natural Languages)就是人類平常使用在讀、說、寫的語言文字，用來交流對話的行為包含聲音、語辭等等都是屬於自然語言的範圍。

自然語言處理(Natural Language Processing)

簡稱NLP，可以用來處理自然語言架構的人工語言系統

系統是以電腦數據資料庫來設計，用於轉化自然語言或人工語言令電腦可使用和了解相對應的代表含意，會運用到很多人工智慧與學習行為相關領域的概念

自然語言處理-2



自然語言處理整體包括了自然語言理解和自然語言生成兩個部分

自然語言理解:電腦能處理及理解自然語言背後的意義

自然語言生成:以自然語言來表達給定的意圖、思想等

自然語言處理和人類對話中的各個層次上通常存在的各式各樣的歧義性或多義性 (ambiguity) , 造成轉換困難。

設計一個中文為資料庫的自然語言系統



從處理文字的形式上可以將中文字（包括標點符號等）組成的字元串和自然語言的形式（字元串）之間當作是一種**多對多**的關係。

問題產生:就算是字形字數一樣的一段字元串，在不同的敘述或不同的語境下，可以分析成不同的詞串、片語串等並具有不同的解釋。

解決:為了**消解歧義**需要極其**大量資料庫**和進行**推理的規則樹**，此方法為如何有效地處理消除歧義的辦法之一。

自然語言處理的主要範疇



單字詞的分類:

單字詞(word)具有獨立語意或扮演特定語法功能，是最小語言單位，由詞性分類賦予語句中每個字詞適當之詞性符號或標記的過程，則稱為詞性標記 (part-of-speech tagging ，或稱 POS tagging)

詞義的分歧:

解釋

舉例 :buy

購買

I bought Fruits and vegetables for \$1000.

相信

He isn' t going to buy his argument.

自然語言處理的主要範疇



句法的模糊性:

一個句子可能會剖析 (Parse) 出多棵剖析樹 (Parse Tree) ，而我們要仰賴語意及前後文的資訊才能在其中選擇一棵最為適合的剖析樹。

語言行為與計畫:

在對話中回答的樣式有很多種，因此找到**最佳解釋**是設計重點。



專家系統



專家系統架構



專家系統被稱為基於知識資料庫的系統

必須具備三要素:

(一) 知識資料庫 (Knowledge-Base)

(二) 推論器 (Inference Mechanism)

(三) 介面 (Interface)

知識資料庫(Knowledge-Base)



資料庫系統的用途主要為儲存人類所擁有的相關知識

資料庫包含兩種類型:

- (1)知識本身所表示的意涵及分析的結果
- (2)人類所特有的經驗、判斷力與直覺這類的非一般知識

推論器(Inference Mechanism)



層次:

利用任意檢索(Blind Search)或啟發式檢索(Heuristic Search)嘗試尋找最有可能的答案

控制策略:

前推式:從已知的條件中尋找答案，利用資料逐步推出結論

回溯式:從已知的條件中尋找答案，利用資料逐步推出結論

雙向式:結合前推及回溯的邏輯策略

推論器會根據知識庫、使用者的問題及複雜度來決定適用推論層次

介面 (Interface)



主要功能:提供相關資料的輸入與輸出

分為三個主要部分：開發者介面、使用者介面、系統介面

開發者介面用途在方便協助系統發展者進行**知識粹取**、**知識庫與推論器的編輯與修訂**，並能對專家系統進行測試、記錄，並說明系統運作的過程、狀態與結果，目前設計的系統程式還屬於此介面應用上

專家系統優點



1. 可用性提高可使性能或供應量提升
2. 專家系統可多重且快速製造。
3. 降低成本，即提供專業知識的成本及電腦取代人類。
4. 可靠的專家系統可設計於解決高風險作業。
5. 知識整合方面為有系統性、多領域專業且大量資訊，保存也完善、永久。

專家系統缺點



1. 缺乏因果推論知識
2. 當專家系統遇到未知的情況發生，無法進行排除
3. 設計及研發是耗時且須要多位專家來共同合作。
4. 基本上內部的知識資料皆為簡單的淺層知識為了方便專家系統的編程及使用。

專家系統與一般電腦系統功能比較



	專家系統	一般電腦系統
功能	解決問題、解釋結果、進行判斷與決策	解決問題
處理能力	處理數字與符號	處理數字
處理問題種類	多屬準結構性或非結構性、可處理不確定的知識、使用於特定的領域	多屬結構性、處理確定的知識

馬可夫模型

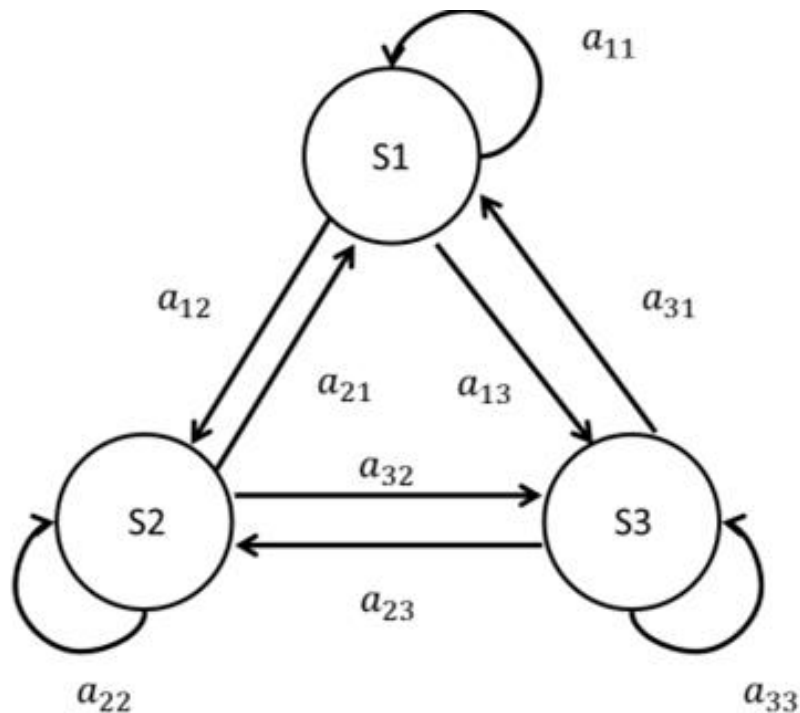


狀態: S1、S2、S3

路徑:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

出發點的路徑加總為1，因為每個狀態都可對應出 N 條路線，單一路線只連向某個單一一個狀態，這 N 條路線的數值都可用機率計算及表示



模式匹配 Pattern Matching



將一整個字串看成一個整體，**String Matching** 為程式設計中的「字串匹配」處理，假設現在設定兩個字串 **S1** 和 **S2**，找出 **S1** 當中是否有一段字串正好是 **S2**，並且找出位置。

S1 意指 `text`(一段文字或文章)

S2 意指執行 `pattern`(單一字詞)

可以想作是從長篇文字 **S1** 之中搜索小段文字 **S2**。

舉例



單詞比對：

S1：今天天氣不錯!

S2：天氣

比對結果：今天天氣不錯!

||

天氣



S1: 今天天氣不錯!

S2: 天氣

比對結果:今天天氣不錯!

||
天氣
(X)

S1: 今天天氣不錯!

S2: 天氣

比對結果:今天天氣不錯!

||
天氣
(X)

S1: 今天天氣不錯!

S2: 天氣

比對結果:今天天氣不錯!

||
天氣
(O)

窮舉法演算法



我們使用的為**順序窮舉法**即逐一比對字元

窮舉方法：順序窮舉 / 排列窮舉 / 組合窮舉

窮舉法的步驟：

1. 列出相符的所有可能解。
2. 根據條件逐一判定。
3. 滿足條件即得到一個解。



系統架構 與設計



專家系統架構

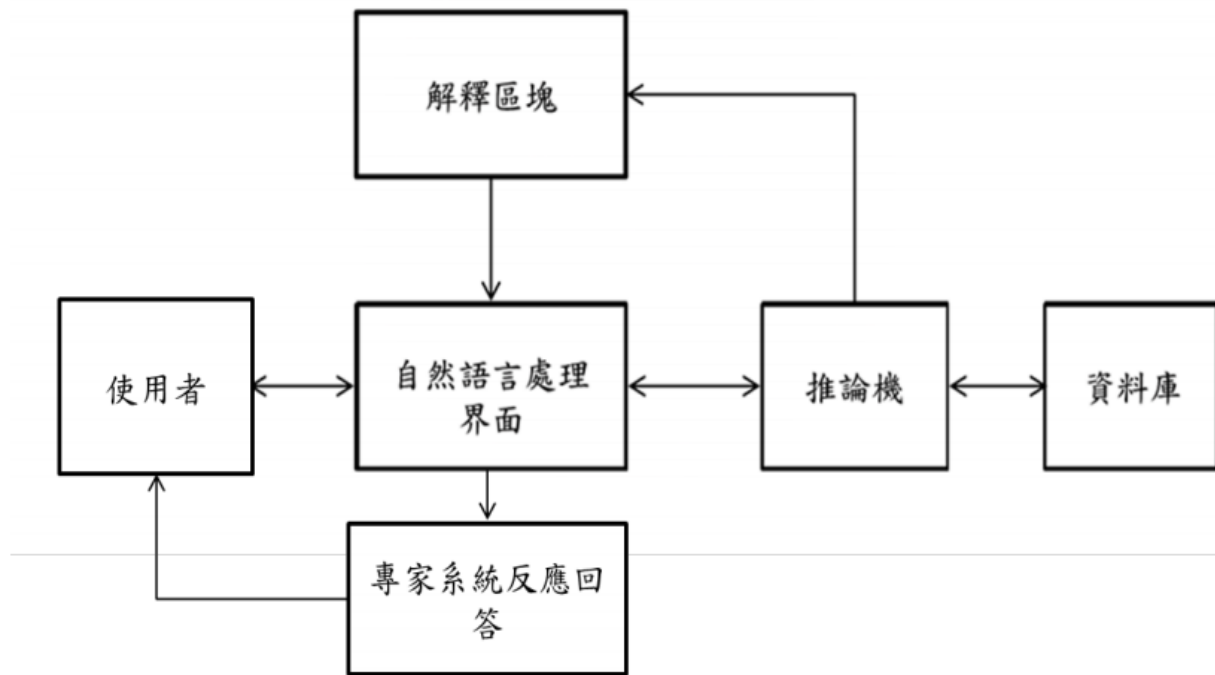


圖 12. 專家系統架構[18]

對話系統語言設計



最大的困難就是人類語言具有複雜性及多樣性的特質，因此需要**簡化**所選定的語言

以人類對話的文句型態來簡化表示，一般句子可分為很多種詞性作排列組合

舉例如下： 句子詞性排列分析 代號

主詞+動詞+受詞 S+V+O

主詞+形容詞 S+A

主詞+動詞+形容詞 S+V+A

資料庫設計



資料以不重覆的方式來儲存許多有用的資訊，讓使用者可以透過檢索、排序、計算、查詢等方法，來做有效率的管理並運用的資訊。

先設定關鍵字或範例文句的資料庫，在使用者輸入對話文句後，可以比對出符合資料庫內的設定後，輸出相應的回答模式，應用自然語言處理來設計，並對對話人物的改變作主詞轉換。

第一階段以單字詞建構資料庫

第二階段以對話文句建構資料庫

第三階段以回答文句建構資料庫

第一階段以單字詞建構資料庫

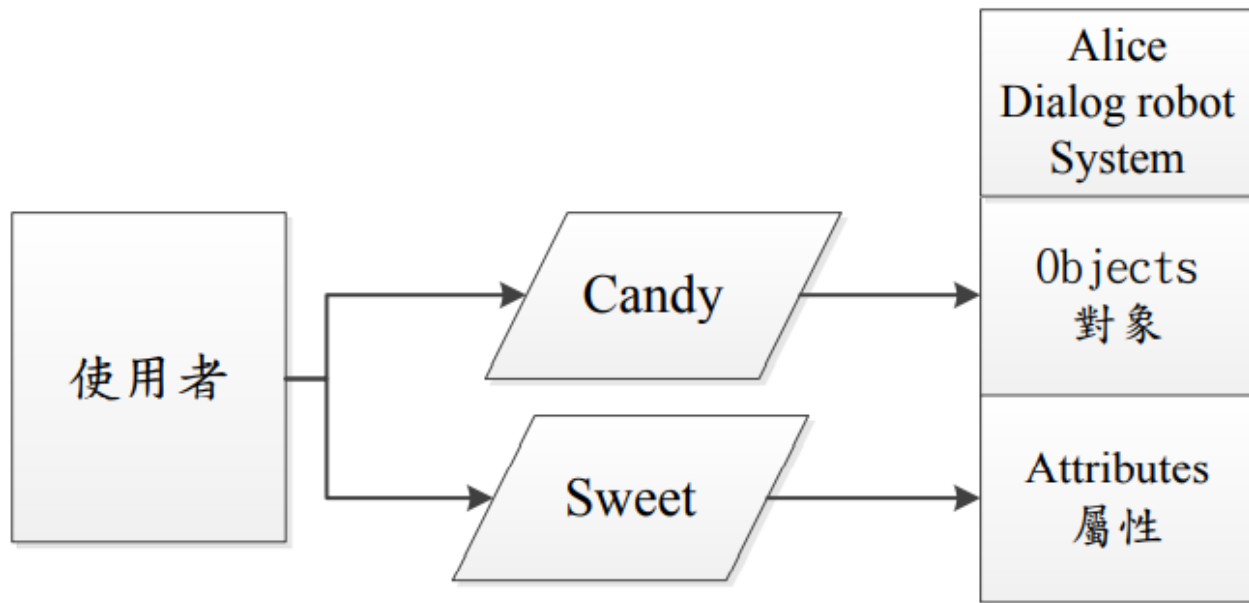


圖 13. 資料庫設計概念圖

第二階段以對話文句建構資料庫



設定一些簡單會話文句，盡量簡化會話詞彙以方便系統辨別

EX:

+S 要+O(若+S 得到+O 怎樣?)

(為什麼+S 要+O?)

如果你輸入我想要吃冰淇淋經由比對這規則(+S 要+O)的結果，+S 對應到我，O 對應到吃冰淇淋，所以此樣式比對有成功，再根據資料庫隨意選擇一個反應樣式句子輸出

第三階段以回答文句建構資料庫



假設輸入英文文句：How are you?

輸出回應是 I am fine 。

輸入中文文句：你喜歡吃甚麼水果？

輸出回應是水果是甚麼？能吃嗎？

回答可以多種設定，加入隨機取值或其他方法來增加模糊情感的錯覺

字串對比



strstr:是針對兩個輸入參數做比對，**strstr** 會先將第一次比對成功的 **pointer** 回傳，也就是如果要找尋 **How do you do?** 字串中的 **do**，函式會回傳第一次比對成功的 **do pointer**，而並非回傳最後一個比對到的。

strchr:用來找尋第一次比對成功單一字母符號，也是一樣回傳該指標，當需要比對單一字元時較常使用此指令，如比對主詞時只需抓取你我他等單一字元時使用。

strcmp:比較 **s1** 與 **s2** 兩個字串是否相等，較長對話句比對時會用到，有時以陣列方式設計可比對每個位址的字元。

strncmp:比較 **s1** 與 **s2** 兩個字串前 **n** 個字元是否相等，特殊比對用途，當關鍵詞需要取出時使用。

推論引擎設計



把知識表達成解決問題的簡單規則，規則可以定義成 **IF-THEN** 結構來使用

IF 部分對給定的資訊或事實

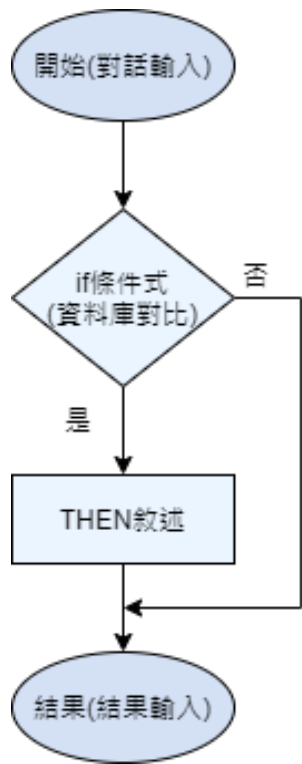
THEN 部分為相對應的事實

任何規則都包含兩個部分：**IF** 部分為前項(前提或條件)，**THEN** 部分為後項(結論或事實)

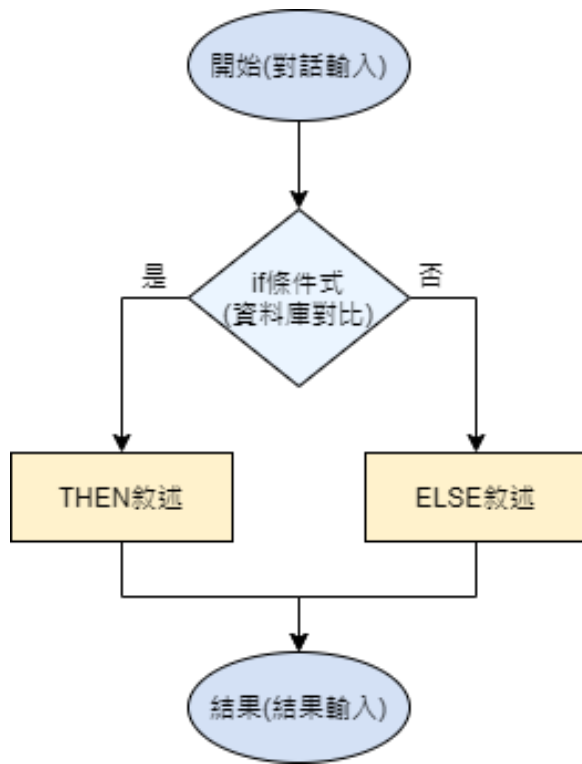
例子：**IF** 當紅綠燈亮綠燈 **THEN** 那我們可以開車往前行。

IF 當紅綠燈亮紅燈 **THEN** 那我們必須停車等紅燈。

推論引擎設計



IF-THEN規則



IF-THEN-ELSE規則



模擬及測試



比對方式



本論程式會抓出比對字串中**關鍵單字比對資料庫**後，套用設計的規則樹內去執行，篩選出相對應的反應回答句子。

比對方式分三種：

- (一) 字元長度比較
- (二) 字元符的代碼比較
- (三) 陣列儲存位置方式逐一比較

比對結果與自行設計之反應句資料庫有關。

Eliza是對**單字進行比對**，抓取關鍵單字去進行比對，如有比對成功，就回傳並回應對應文句。

程式設計

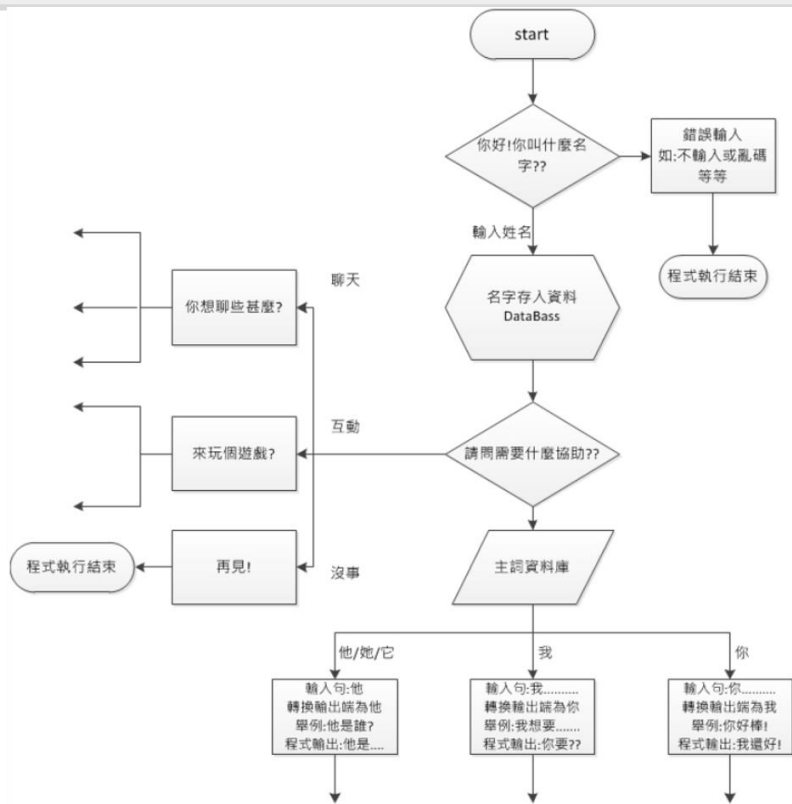


優先分類對話領域，再分類其中的單字資料庫，之後再對於不同種類的對話字詞，分類為各個獨立的字詞資料庫。比對到關鍵字時回傳資料庫內容，比對此文句結構是否有相同結構在完整文句資料庫內，如有比對成功，就回應使用者回答樣式資料庫內容。

程式設計片段



系統程式的片段流程圖



片段程式-1



片段指令
(pos 指令用來比對關鍵字單字詞)

```
cin.getline(X,20);

pos1=strstr(A,X);
pos2=strstr(B,X);
pos3=strstr(C,X);
pos4=strstr(*D,X);

if(pos1==A)
{
cout<<"虛擬:"<<pos1<<"要聊些甚麼?"<<"\n";
cout<<name<<":";
cin.getline(Y,100);
pos4=strstr(*D,Y);
```

片段程式-2



(計算字串長度及比較字元符)

當程式比對到關鍵單字詞時回傳資料庫內容，繼續比對整句對話是否有相同結構在整句的文句資料庫內如果比對使一樣的字元內容及字數立即輸出設定好的回答樣式資料庫內容，回應使用者且繼續對話樣式比對直到結束。

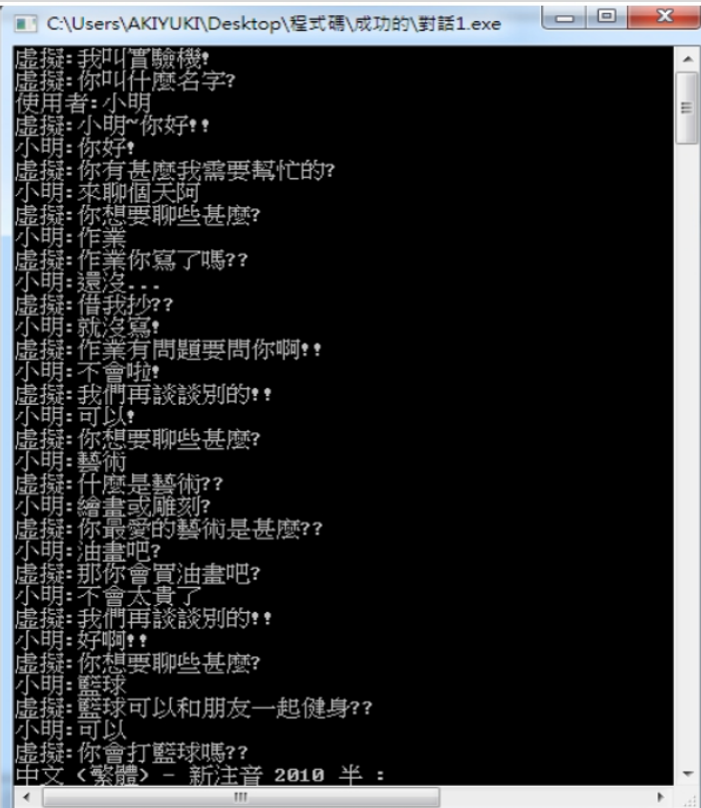
```
if( strA[n]< str1 )
{
    //" 字串不相等" ;
    //cout <<"比對失敗"<< endl;

    //cout <<"我不清楚你在說什麼!!"<<endl;

}
else if( strA[n] > str1 )
{
    //" 字串不相等"
    //cout <<"比對失敗"<< endl;
    //cout <<"我不清楚你在說什麼!!"<<endl;
}
else
{
    //cout <<"比對符合!";
    cout<<"JC:"<<strB[n]<<endl;
}
```

實作

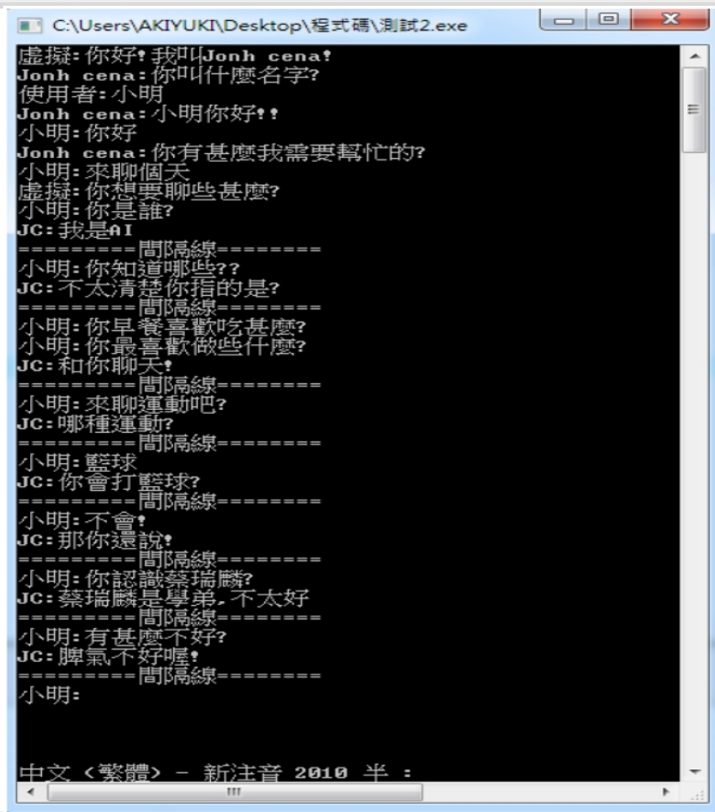
圖為關鍵字詞資料庫執行結果



```
C:\Users\AKIYUKI\Desktop\程式碼\成功的\對話1.exe
虛擬: 我叫實驗機!
虛擬: 你叫什麼名字?
使用者: 小明
虛擬: 小明~你好!!
小明: 你好!
虛擬: 你有甚麼我需要幫忙的?
小明: 來聊個天阿
虛擬: 你想要聊些甚麼?
小明: 作業
虛擬: 作業你寫了嗎??
小明: 還沒...
虛擬: 借我抄??
小明: 就沒寫!
虛擬: 作業有問題要問你啊!!
小明: 不會啦!
虛擬: 我們再談談別的!!
小明: 可以!
虛擬: 你想要聊些甚麼?
小明: 藝術
虛擬: 什麼是藝術??
小明: 繪畫或雕刻?
虛擬: 你最愛的藝術是甚麼??
小明: 油畫吧?
虛擬: 那你會買油畫吧?
小明: 不會太貴了
虛擬: 我們再談談別的!!
小明: 好啊!!
虛擬: 你想要聊些甚麼?
小明: 籃球
虛擬: 籃球可以和朋友一起健身??
小明: 可以
虛擬: 你會打籃球嗎??
中文 <繁體> - 新注音 2010 半 :
```

圖為完整對話執行結果

(包含關鍵字詞比對、對話領域
的分類選定及輸入句的整句對比
資料庫和對話反應樣式輸出)



```
C:\Users\AKIYUKI\Desktop\程式碼\測試2.exe
虛擬: 你好! 我叫Jonh cena!
Jonh cena: 你叫什麼名字?
使用者: 小明
Jonh cena: 小明你好!!
小明: 你好
Jonh cena: 你有甚麼我需要幫忙的?
小明: 來聊個天
虛擬: 你想要聊些甚麼?
小明: 你是誰?
JC: 我是AI
===== 間隔線 =====
小明: 你知道哪些??
JC: 不太清楚你指的是?
===== 間隔線 =====
小明: 你早餐喜歡吃甚麼?
小明: 你最喜歡做些什麼?
JC: 和你聊天!
===== 間隔線 =====
小明: 來聊運動吧?
JC: 哪種運動?
===== 間隔線 =====
小明: 籃球
JC: 你會打籃球?
===== 間隔線 =====
小明: 不會!
JC: 那你還說!
===== 間隔線 =====
小明: 你認識蔡瑞麟?
JC: 蔡瑞麟是學弟, 不太好
===== 間隔線 =====
小明: 有甚麼不好?
JC: 脾氣不好喔!
===== 間隔線 =====
小明:
中文 <繁體> - 新注音 2010 半 :
```



結論





- 中文因為編碼問題就要考慮到字元字串長度，目前抓去的規則還是在字數、字元上的比對，屬於文字形體的比對，而詞意的判斷上還需要再改進及設計，中文在設計比對上詞意難度很高。
- 設計上定位為助手的角色協助使用者，以對話互動方式加入做選擇。
- 在與使用者聊天的同時，系統可以抓取對話中特殊表達情緒與關鍵字的反應樣式，並透過用字遣詞來加強對話語氣，增加人類對電腦的互動意願。

未來展望



1

增加更多功能的互動，例如：互動遊戲、接收/傳送訊息、語音等等有發展性的功能新增。

2

資料庫增大及推論規則的處理複雜度增加，使對話樣式可以更多選擇及多樣化。

3

系統設計加入語意及詞性的分析，使用類似雲端運算處理的演算法來執行。



參考文獻





[1] A.M.Turing, "Computing Machinery and Intelligence", Mind 49, pp.433-460 , 1950.

[2] E. Robert , G. Robert and G. Beber , "Parsing the Turing Test. Philosophical and Methodological Issues in the Quest for the Thinking Computer," Springer Science, pp.3-301, 2008.

[3] M. Agarwal and S. Goel, "Expert System and it's Requirement Engineering Process," Recent Advances and Innovations in Engineering (ICRAIE) ,pp.1 - 4, 2014.

[4] S. Kang , Youngjoong and K. J. Seo , "A Plan-Based Dialogue Model Using a Discourse Stack for Intelligent Human-Robot Interface Development," Advanced Language Processing and Web Information Technology International Conference on,pp.162-167 , 2008



[5]L. Caviglione and W. Mazurczyk , “Understanding Information Hiding in iOS,”
Computer,pp.62 – 65,2015.

[6] 徐恩普，歐陽崇榮 編譯，人工智慧-如何使用 C 語言來發展人工智慧，松崗 電腦圖書資料有限公司，1987 年9 月。

[7] 孫家麟 編著，專家系統之理論與程式設計，松崗電腦圖書資料有限公司，1987 年12 月。

[8] L.P. Kaelbling, M.L. Littman and A.R. Cassandra, "Planning and acting in partially observable stochastic domains ," Artificial Intelligence (Elsevier) 101 (1–2) , page. 99–134, 2013



[9] H.H. Bui, S. Venkatesh and G. West , “Policy recognition in the abstract hidden markov model. ,” Journal of Artificial Intelligence Research , vol. 17, p. 451– 499, 2002.31

[10] 謝政勳，廖琬洲，李聯旺 編譯，人工智慧：智慧性統導論（第三版），全華圖書股份有限公司，2012年3月。

[11] 蔡金翰，語音對話系統和對話策略之研究，國立交通大學電信工程學系，碩士論文，2005年。

[12] 朱育德，基於字詞內容之適應性對話系統，國立中央大學資訊工程學系，博士論文，2006年7月。



[13] M. Li and C. Feng , "The use of artificial intelligence in the information retrieval system epoch-making changes in information retrieval system," Information Management and Engineering (ICIME), The 2nd IEEE International Conference on , pp.310 - 313 , 2010.

[14] D. Bolton(2000), A project to build better versions of Eliza in C,C++ and C#, 檢自 <http://cplus.about.com/od/Eliza/a/A-Project-To-Build-Better-Versions-Of-Eliza-In-C-Cpp-And-Csharp.htm>.

[15] . 劉秉鈞(2016) 。人工智慧於模擬人類對話系統設計 。逢甲大學自動控制工程學系碩士論文

介紹影片連結



<https://www.youtube.com/watch?v=qz8BmDJAkYA>



報告結束
謝謝大家

