# Linguistic and python

吳沛馨、翁珮瑜

## 一、動機

由於組員都來自中文系,我們有一個共同的經驗是,大一時學習了一整年的語言學。語言學是一門需要觀察及歸納的學科,需要分析大量的語料,有語音、語義、語法和語用這四個不同的大主題。

而語音學中,有很多題目是要尋找語音的變化,觀察某兩個音位的分布環境,根據其前後所接的語音環境,分析出這兩個音位的關係。若是出現在相同環境之中,兩個音位是「獨立音位」,若是出現在不同環境之中,則需要觀察其前後語音的性質,若經過歸納發現,某音位只會出現在特定性質的語音之前或之後,代表這個音位分布範圍特定,說明在特殊情況下,會使原本的音位改為此音位,兩音位稱為「互補分布」。

學習語音學時,我們花大量的時間在看語料上,親手寫出一個個音位的 分布環境,寫好後再進行歸納整理,將重複的環境去除,整理好後分析兩個 音位之間的關係。大量整理語料分析某兩個音位的互相關係,成為我們那段 時間的日常。

分析、歸納和整理,這些都是語言學的基礎功夫,如果將來想要踏足語言學的領域,觀察、統計、分析等技能不可或缺。因此,在學習到 python 的統計功能時,我們便思考如何將它運用在語言學上,簡化我們分析的過程。

由程式統計,給予我們語料的分布環境,我們再進行最後的分析步驟, 如此可以節省大量依靠人力整理的時間,是我們研究的初衷。

# 二、研究範例題目

5.		8 - 3730	70.700					
t/_a								
<i>Б</i> _	d. 何謂同化(assimilation)?依方向,同化又可分為哪幾種?依音質的變							
-2	化,又有哪些同化類別?							
-a	2. 何謂異化(dissimilation)?異化的結果可能會有哪三種音變?							
_0	3. 請看後面美國亞利桑納州印地安帕帕戈人(Papago)的語料(語料取							
- D	自 Kisseberth and Kenstowicz, 1976:37) : d/_ 4							
1 - 0	tatai	腱	∫inig	移動嘴唇	figitog	「思考」	1_#	
C1-41.	_ \ tatal	男男	∫ikpan	工作	3iwhiadag	到達	-a	
)/ - 4	_ i tamš	口香糖	daswua	堆積	∫ilwin	摩擦	-0	
- 00	Thohnto	贬低	doaʒida	跟蹤	zuhki	雨	-0	
1	tokih	棉花	zigos	去頭皮	∫iposid	品牌	70	
+1-7 [ \1 / 1	pialcodsid	辦到	zuni	乾果	zusukal	蜥蜴	7/-1#	
1 -0-1-0	Juagia	網織袋	dakpon	滑倒	∫ukma	黑暗	Hu-	
t/っても]/葉	Tuful	雞	do?ag	Ш		111	F) - W#	
	請問:					10/1>	(S) /- [thigh]	
	a. [1]/[J]和 [d]/[3]呈現了怎樣的分布情形?					1917	[四]/其巨情形	
	b. 有何規	b. 有何規則可以解釋這種分布?這種規則是同化規則還是異化規則?とある(と) ままっ						
4.請看後面的語科:後高元音時,舌面會拍升,連带使都近的音光的升,正面後音系面							大部分、正の変もするの	
	第一人稱		是门门门	5 6 4 (0) 1	40 11 61 6	T AND CAR I D	क्षेक्रे हिता हुई स	
	kumil	洗手	kumil			January Le c	are the transfer of the transf	
	zufrel	拌坊	zufrel	ul		1-11是基	在刑司。但是詞尾是統一	
	hundan	+2-4-	hund	and .	No. of Concession, Name of Street, or other party of the last of t		- HER IS A TATE OF THE	

(圖一)

## 資料來源:108學年度蘇建唐教授《中國語文通論》講義

我們從眾多習題之中挑選了一個題目作為問題(圖一),是美國亞利桑那州印地安帕帕戈(papago)的語料,題目要我們分析\t\和\ʃ\、\d\和\ʒ\的關係,在此以第一題做為範例,第二題的操作方法一模一樣,便不多加贅述。

在語言學的實例中,大多是因為音位之後的差異造成音位的改變,因此 我們設計程式時先以分析音位後面的環境為主,做簡單的初步分析與判斷, 我們的目的是讓學習語言學的新手可以減省花費在重複作業上的時間。無論 是語言學或是程式設計,我們都是從一個新手開始慢慢學習,如果能將學會 的東西學以致用,同時也將便利的工具交棒給下一位新手,雙方都可以進行 更有效率的學習。

而在真正實行時,遇到了一個難點,是 python 無法讀出\\這個音位,因此權變之下,我們使用\b\暫代\\\的位置。以下是我們的語料: tatai、tatal、tams、tohnto、tokih、buagia、bubul、binig、bikpan、bigitog、bilwin、biposid、bukma,共十三筆資料需要分析,我們想要 python 解決的問題是\t\和\\\兩者的分布環境。

# 三、 程式碼解析

## (一) 思考過程

從動機上來看,耗費時間的理由之一,是有重複的環境分布,因此我們使用在這堂通識課「資訊的邏輯思考」上教過的各種 python 統計功能的知識。首先是 dictionary 的技巧,將關鍵字出現的次數進行統計,顯示出各個關鍵字的名稱與次數,再運用 sort,讓關鍵字依照次數由大至小一一排列,使版面更清楚簡潔。另外,我們想要的結果只有關鍵字而已,並不需要次數,因此我們另外上網尋找只呈現 dictionary 前面的關鍵字,而不要後面的次數的方法。找到這三個步驟的解答,理論上可以解決我們的問題,接下來是實際運行的結果。

#### (二) 程式碼

```
☐ GEC.py 🖸
      import sys
     def count alphabets (in str):
           alphabet_count_dict = {}
           for alphabet in in_str:
              if str not in alphabet_count_dict.keys():
                   alphabet_count_dict[alphabet]=1
                   alphabet_count_dict[alphabet] = alphabet_count_dict[alphabet] + 1
          return alphabet_count_dict
    Edef sortdic(indic):
          sortlist=[]
           sortlist=sorted(indic.items(), key=lambda pair:pair[1], reverse = True)
 14
           return (sortlist)
    □def Convert(tup, di):
          di = dict(tup)
           return di
      str2 = ""
    for line in sys.stdin.readlines():
           s = line.strip()
           index = 0
          str = ""
 24
           for alphabet in s:
               str +=alphabet
               if (alphabet == "t" ):
 26
                   index = len(str)
                   stri = alphabet + s[index]
str2 += stri + " "
 29
     a = str2.strip()
     v = a.split("
      alphabet count dict = count alphabets(v)
      sortL=sortdic(alphabet_count_dict)
      diction={}
      sortlist2=Convert (sortL, diction)
      keylist=list(sortlist2.keys())
      print(keylist)
```

(圖二)

#### (三) 程式介紹與結果

「工欲善其事,必先利其器」,在程式上依然如此。因此我們將需要重複執行的動作用函式先定義清楚,使程式碼不至於冗長一串,dictionary 的用法是讓它可以將讀到的重複環境進行歸類與計算的動作,(圖二第2行至第9行)。再定義 sort,將次數由大到小排序整理(圖二第10行至第14行),最後是定義一個函式,可以將排序好的資料,轉換成可以隨意編輯的狀態(圖二第15行至第17行),需要這個函式是因為由 dictionary 統整出的資料會是無法隨意編輯的 tuple,而我們要將 tuple 轉成 list 的狀態,才可以將後面的次數資料刪除。

將工具都準備好後,開始準備讀取資料(圖二第20行),將讀取的資料去掉最後面可能會有的空白(圖二第21行),再建立一個參數(圖二第22行)與一個空字串(圖二第23行),接著根據題目要求建立迴圈,將字母通過迴圈一個一個加進空字串裡,如果讀取到\t\這個音位的時候,原有的參數將到此為止的單字長度記錄下來(圖二第25行),由於python在字串位置的讀取上是從0開始,因此,到\t\為止的單字長度,會等於\t\之後下一個的字母所在位置,設置一個新的變數,是\t\加上後一個字母(圖二第28行),如此一來,就找到了\t\的分布環境(圖二第29行)。

找到分布環境後,用事先定義好的函式將環境分類、排序、轉換,最後只取我們所需要的環境分布就好(圖二第36行),因此結果會從重複次數多到小排列顯示,接著將原本輸入lt\的位置改成\b\(圖二第26行),再讓程式從頭開始執行,因為程式碼一模一樣,所以不另外再附上圖片,實際運行後結果為:[ta,to]和[bu,bi](如下圖三),與當時我們手動進行的結果完全一致(如圖一)。

```
Anaconda Powershell Prompt (anaconda3)

(base) PS C:\Users\User> D:
(base) PS D:\> cd '.\GEC file\'
(base) PS D:\GEC file> type .\lin.txt | python '.\GEC final.py'
['ta', 'to']
(base) PS D:\GEC file> type .\lin.txt | python '.\GEC final.py'
['bu', 'bi']
(base) PS D:\GEC file> _
```

#### (圖三)

觀察這四個環境,再運用語言學課堂上學習到的知識,可以得知\u\和\i\都是發聲時舌頭位置較高的高元音(元音即母音),由此可以歸納出,\\\只會出現在高元音之前,而\t\則出現在其他環境,因此兩音位為互補分布,\t\在高元音前會變成\\\的發音,在其他情況時仍然維持\t\不變。

## 四、 結語

在思考如何解決問題的過程中,我們運用了老師教的方法,也學會上網搜尋資源,將自己的問題一步步剖析後,發現一個問題之中可以再拆分成其他問題。我們所寫出來的程式碼,也是根據從分解出來的源頭問題著手,以我們的問題為例:第一步問題,是如何找到音位後的環境,第二步則是分類,第三步排序,第四步是取分布環境,按部就班的解答每一步驟,就是程式的語言,也是我們在這堂課學習的邏輯思考。

最後,在看到程式結果後,我們只需要從這些結果之中進行分析,節省 許多觀察、手寫和人工計算的時間,我們也學到如何使用程式來解決原本需 要人工耗時計算的問題,這些重複進行的大量作業,可以用更快捷的工具簡 化負擔,讓研究者將心力投注於資料歸納後的分析上。

# 五、 工作分配

設計程式:吳沛馨、翁珮瑜

錄製影片:吳沛馨 報告撰寫:翁珮瑜