

誠實包子

B04000000 羅章倫、B04000000 羅章倫、B04000000 羅章倫、B04000000 羅章倫、B04000000 羅章倫、B04207003 廖永賦

簡介

內隱聯結測驗 (IAT)

眾所皆知，**人們所講的未必是他們內心的真實想法**。理解真實的想法和表達出來的想法之間的差異，是心理學家相當感興趣的議題，而內隱聯結測驗 (Implicit Association Test, **IAT**) 就是心理學家用來偵測這個差異的工具之一。

現在，讓我們進行一個假想實驗。在電腦螢幕上，會依序出現一張「人臉」或一個「名字」，你的任務是在最短的時間內判斷出現的「人臉」或「名字」屬於**男性**或是**女性**。若是屬於男性的「臉」或是「名字」，你需要盡快舉起**左手**；若是屬於女性的「臉」或是「名字」，則要盡快舉起**右手**。在完成剛剛一系列的判斷作業後，你要再進行一次幾乎相同的作業，只是這次出現的「臉」如果屬於**男性**，你要舉起的是**右手**而非左手；如果是**女性**，則要舉起**左手**。換言之，這次在看到**男性的臉和男性的名字** (或女性的臉和名字) 時，要舉起的是不一樣的手。

第二次的判斷作業會比起第一次難上需多，因為在人們的概念網絡中，同性別的臉和名字之間有相當強的聯結，因此在看到這兩種具有強烈連結的刺激時，做出相同的反應 (例如，看到男性的臉或名字時，皆舉起左手) 是比較容易的。而這些判斷作業的難度，則會顯現在作業的反應時間上——第二次判斷作業所耗費的時間會比起第一次還要長。IAT 的設計即是透過比較這兩次判斷作業的反應時間，來推測在受試者的心中，特定的概念 (e.g. 「男臉」或「女臉」) 是否與特定的屬性 (「男性化名字」或「女性化名字」) 有較強的連結。

IAT 測得是人們心中 (前意識)，特定概念與特定屬性之間的 (相對) 聯結強度。我們因此能透過 IAT 推測人們對特定概念的**好惡** (特別是人們不願說或沒意識到的好惡或刻板印象)，例如，我們可以將前述的假想實驗中的「男臉」代換成「民進黨」、「女臉」代換成「國民黨」、「男性化名字」代換成「具正面意義之詞彙」、「女性化名字」代換成「具負面意義之詞彙」。此時，若受試者較為喜歡「民進黨」，其心中對於「民進黨」與「具正面意義之詞彙」應有較強的聯結，而此聯結應能反映在 IAT 所測得的反應時間上。

設計概覽

我們開發的遊戲，**誠實包子**，即是根據 IAT 的理論與實驗 (Greenwald, McGhee, & Schwartz, 1998) 所設計的。**誠實包子**所欲測量的是臺灣人心中對於「民進黨」與「國民黨」的相對偏好，因此，延續 IAT 的設計，我們的遊戲由 5 個區塊所組成 (見下圖)。這 5 個區塊中，第 1、2、4 區塊的目的是為了讓受試者練習將特定反應 (按下鍵盤的 E 或 I 鍵) 與特定概念 (民進黨、國民黨) 或屬性 (正、負面詞彙) 進行配對，以排除反應時間測量上的系統性誤差。第 3 個區塊及第 5 個區塊則是實際測量受試者反應時間的地方。若受試者在第 5 個區塊上的反應時間上比起第 3 個區塊還快，則說明其心中對於「國民黨」與「正面詞彙」有較強的連結，因而推論其政治傾向偏向國民黨；反之，若第 3 區塊較快，則會有推論其偏向民進黨。

實驗區塊 (依順序)	1	2	3	4	5
目的	概念-反應 配對	屬性-反應 配對	概念-屬性 連結測驗1	(反向) 概念-反應 配對	概念-屬性 連結測驗2
實驗指示	民進黨相關：左手按鍵 國民黨相關：右手按鍵	正面意義：左手按鍵 負面意義：右手按鍵	概念、屬性刺激交錯出現 民進黨相關：左手按鍵 正面意義：左手按鍵 國民黨相關：右手按鍵 負面意義：右手按鍵	民進黨相關：右手按鍵 國民黨相關：左手按鍵	概念、屬性刺激交錯出現 民進黨相關：右手按鍵 正面意義：左手按鍵 國民黨相關：左手按鍵 負面意義：右手按鍵
刺激流程範例 (刺激 → 反應) (出現順序由上至下)	蔡英文 → 左 蘇貞昌 → 左 吳敦義 → 右 陳其邁 → 左 朱立倫 → 右	虛偽 → 右 真誠 → 左 謙虛 → 左 厭惡 → 右 廢物 → 右	真誠 → 左 吳敦義 → 右 厭惡 → 右 蔡英文 → 左 謙虛 → 左 陳其邁 → 左 虛偽 → 右 蘇貞昌 → 左	朱立倫 → 左 陳其邁 → 右 蔡英文 → 右 吳敦義 → 左 蘇貞昌 → 右	虛偽 → 右 蘇貞昌 → 右 謙虛 → 左 朱立倫 → 左 厭惡 → 右 吳敦義 → 左 真誠 → 左 陳其邁 → 右

系統設計

使用者在進行 IAT 的時候，是在刺激 (特定概念或屬性之圖片與文字) 呈現之後進行反應 (按下 E 或 I)。因此，實驗刺激可說是組成 IAT 最重要的成份，我們因而將「實驗刺激」操作化成 Python 中的物件，以「實驗刺激物件」為導向作為設計的基礎。因此，縱使使用者進行在操作的時候，看似有 5 個實驗區塊，但對於程式來說，每個新畫面的出現其實就只是一個新的「實驗刺激物件」— 前端 (browser) 在接收到後端 (Python) 所傳來的「實驗刺激物件」(透過 JSON 格式在 Python 與 JS 之間傳輸) 後，會依據這個物件的不同屬性決定要如何在網頁上呈現這個「實驗刺激物件」。同時，前端會在使用者按下按鍵後，將「使用者答對與否」以及「從刺激出現到按下按鍵所需之反應時間」記錄在「實驗刺激物件」中，並將這個更新後的物件傳回後端以利後續資料分析與處理。在使用者回答最後一個刺激後，後端在接收到資料後會計算出實驗區塊 3 和 5 的每題平均反應時間 (答錯則忽略不計)，以決定使用者的政治傾向，最後，再將此結果傳回前端呈現結果。

實驗刺激物件

「實驗刺激物件」在 Python (後端) 中是由自定義的 class Stim() 所生成。為了使前、後端能溝通 (前端使用 JS 後端使用 Python)，後端在將「實驗刺激物件」送給前端之前，會將其轉為 JSON 格式的字串，而前端則會將接收到之 JSON 字串轉換成 JS 的物件，並且，在收集到使用者的反應之後，將這個物件再轉換成 JSON 字串以送至後端。

一個「實驗刺激物件」含有下列屬性：

- block: 該刺激所屬之**實驗區塊**
 - 1, 2, 3, 4, 5: 對應至上圖 **1 至 5 之實驗區塊**
 - 0, 6: 開始畫面或結束畫面。此二值用於告知前端目前接收到之刺激並非「真的實驗刺激」，而是遊戲的開始或結束畫面。
 - 01, 12, 23, 34, 45: 1 至 5 之**實驗區塊間的指導語**。這 5 個值用於告知前端目前接收到之刺激並非「真的實驗刺激」，而是實驗區塊之間的指導語。
- content: 實驗刺激的內容，可能是「正、負面詞彙」、「圖片檔的相對路徑」或是「結果回饋」。若 block 非 1 至 6，則前端不會理會此內容。

- answer: 正確答案 ('left' or 'right'), 該刺激呈現後, 使用者需左手按鍵或右手按鍵才能答對。若 block 非 1 至 5, 則前端不會理會此內容。

前、後端通訊 : WebSocket

相較於比較常用於前、後端溝通的 HTTP 協定, 我們的設計使用的是比較新的 WebSocket 協定。使用 WebSocket 的好處在於其讓前端與後端之間的資料交換變得更加簡單, 因為在 WebSocket 的協定下, 前、後端的通訊是雙向的。換言之, 後端能在不經前端的同意下 (不需 GET 或 POST), 直接將資料送至前端。

下圖是使用 `__main__.py` 中使用 `websockets` 與 `asyncio` 套件所撰寫的後端通訊邏輯 (已簡化)。在此段程式碼之前, 程式已先處理了「實驗刺激物件」出現的順序, 並將這些物件依序儲存於 `stim_lst` 當中。後端與前端傳送這些「實驗刺激物件」的 API 寫在 `experiment()` 這個 `asynchronous` 函數裡。可以看到這個函數的內部由一個 `for loop` 組成, 因此每一次迴圈開始時, 後端就會主動將一個「實驗刺激物件」送至前端 (line 6-8)。在送出物件後, 程式執行到第 11 行的地方時, 就會「卡住」, 等待前端將更新過後的物件傳回後才會繼續執行。換言之, 執行這個程式所需的時間, 是由每次在前端進行實驗的使用者所決定——若每次看到刺激後, 他都很快地按下按鍵, 後端的 `for loop` 就會跑得比較快。

```
1 stim_lst = [Stim1, Stim2, Stim3, ...]
2
3 async def experiment(websocket, path):
4     for i in range(len(stim_lst)):
5
6         # Send stimulus to client
7         sending = stim_lst[i]
8         await websocket.send(sending.toJSON())
9
10        # Receive response from client
11        res = await websocket.recv()
12        msg = json.loads(res)
13        sending.rt = float(msg['rt'])
14        sending.correct = str2bool(msg['correct'])
15        stim_lst[i] = sending # Save received result
```

實驗刺激物件出現順序

<如何決定刺激順序 (演算法)>

前端呈現

前端使用 JavaScript 所寫成, 其工作主要在 (1) 接收「實驗刺激物件」後, 決定如何呈現; (2) 呈現刺激後, 收集使用者的反應情形 (「作答正確與否」與「反應時間」); (3) 將更新過後之「實驗刺激物件」傳回後端。(2) 與 (3) 的工作相對簡單, 這裡以說明 (1) 的設計為主。

上文提過每個「實驗刺激物件」皆有 block 這個屬性。對於後端程式來說，block 屬性並無太大意義，因為其設計是以「實驗刺激物件」為導向，其只須確定物件出現的順序即可。但對於前端，每個「實驗刺激物件」的 block 屬性決定了其會如何呈現 — 若一個「實驗刺激物件」的 block == '3'，則這個刺激會被呈現為實驗區塊 3 中的樣式 (民進黨的按鍵提示出現在左、國民黨在右等)；若 block == '12' 則這個刺激則會被呈現成「指導語」(提醒使用者下個實驗區塊要怎麼進行)。



團隊成員分享

參考資料

Greenwald, A. G., McGhee, D. E., & Schwartz, J. L. (1998). Measuring individual differences in implicit cognition: the implicit association test. *Journal of Personality and Social Psychology*, 74(6), 1464-1480.