Event Source Page discovery solution with Extremely Heuristic One-step search algorithm

黃允誠
107502520
資訊工程學系 三年 A 班
國立中央大學

***Abstract*—此報告旨在闡述搜尋演算法使用強大啟發函數的優劣，以及為何如此極端的啟發式搜尋演算法對這次主題有效**

Keywords—演算法、工人智慧、經驗法則、一步搜尋

I. 引言

此次主題主要以爬蟲程式解析網頁 HTML 來解題，解題過程中主要關注網頁上的各個連結。

而現代網頁上琳瑯滿目，連結非常多，且同網域中的子網頁間往往有大量連結能夠彼此跳轉。以圖論的概念而言，網站中各個子網頁構成非常密集的「稠密圖」。這個特性造成搜尋演算法到達一個節點後，其「視野」非常寬廣，幾乎涵蓋了整個網站中絕大部分的子網頁。

第二個重要特性則與主題本身有關，即以「活動主顯示頁面」為目標。活動公告為大部分網站的主要架設目的之一，因此絕大部分的網站在主頁當中必定會顯眼的放上通往該類網頁的連結，且連結上的提示文字通常是簡短而清楚的。因此只要找到網站常用來標示此類連結的關鍵字，或分析出網頁慣用的 HTML 編碼方式，搜尋演算法便可以相當輕易的從主頁面、甚至是任何頁面直接找到對應連結，並直接從 HTML 代碼中爬取目標網頁的網址。

綜上所述，以圖論的概念而言，此次主題的目標網頁在絕大部分情況下皆為網站主頁的直接相鄰節點。

II. 演算法基本策略

A. 敏感關鍵字

- Interests: 目標關鍵字。在文字中或子元素的 HTML 代碼中含有此類關鍵字的連結，極有可能為通往活動頁面的連結。此類關鍵字中，每個關鍵字有不同的優先權。
- Noise: 誤判關鍵字。含有目標關鍵字但同時含有此類關鍵字的連結，有極低的可能為通往活動頁面的連結。但絕大部分僅是通往跟活動有相關性的頁面，而非真正的活動總覽頁面，例如「歷年展覽史」。此類關鍵字無優先權之分。
- More: 補述關鍵字。又分為內文部分與網址部分兩類。當連結的內文含有內文關鍵字，且目標網址含有網址關鍵字，則該連結可能為部分活動預覽列表底部之「更多」按鈕，例如「查看所有展覽」。或者該連結通往活動頁面，但其內文以圖像取代，因此並無內文。

B. 連結權重

分析完一個網頁中每個可能連結的資訊後，演算法會將關鍵的資訊紀錄成該連結的權重。當所有網頁中的可能連結分析完畢，便根據權重依序詢問 API，直到確認答案或確認並未擷取到答案。權重是分為五層的階層式權重，由高至低列舉如下：

- 擷取原因權重: 極短內文，包含目標關鍵字且不含誤判關鍵字 > 「更多按鈕」類連結 > 不限內文長度，包含目標關鍵字且不含誤判關鍵字 > 無內文圖片連結 > 極短內文，同時包含目標關鍵字與誤判關鍵字
- 目標關鍵字權重: 對於擷取原因權重為第一、第三及第五類的連結，同類之間以包含的目標關鍵字之權重再行比較。
- 關鍵字位置權重: 若還是相同權重，則關鍵字出現的位置在內文中越靠前者優先選擇。
- 內文長度權重: 內文較短者優先
- 網址長度權重: 目標網址較短者優先

III. 程式碼梗概流程

A. 初始化

對於每個查詢網址，先分析其網站首頁網址並初始化資料結構。

B. 對網頁搜尋

透過爬蟲程式分別連上查詢網址及其首頁，接著爬取所有連結，而後對每個連結執行步驟 C.

C. 分析連結

依據基本策略連結進行分析。若符合可能為解答的條件便生成該連結對應的資料紀錄並存入列表中，否則丟棄連結。資料紀錄包含所有權重、內文及連結網址。

D. 提交答案

依據權重對連結列表進行排序後依序提交給 API。

E. 額外嘗試

若以上步驟尚未找出答案，則直接以首頁作為答案提出。若非答案，最終直接以查詢網址作為答案提出。執行此步驟之原因在下一部份詳細解釋。

IV. 問題解決

演算法設計開發過程中遇到許多繁瑣問題，此處不一一贅述。此部分主要列舉較大規模、高層次的問題，並提出最終使用的解決方案。

A. 無法找到目標連結

雖如引言所述，從絕大部分網站首頁皆能直接查詢到活動網頁連結，然而仍舊有少部份情況無法查詢到。深入了解後發現，並非啟發函數不夠強大，而是從首頁確實無法直接一步查找到對應連結。

出現此情況有特定模式，舉此次作業資料集 id 85 號查詢為例：查詢網址為「國立台灣史前文化博物館」官網中的一個子網頁。但透過程式自動分析首頁之後，我們發現此網站與許多藝文推廣公家機構官網共用網域。即我們分析出的「首頁」其實僅是作為這些機構官網共同的列表顯示頁面，而非我們真正目標網站的首頁。問題在於沒有一個單純的演算法可以確認應該在網址的何處切割以得到目標網站的實質首頁，此種情況下要想堅持一步搜尋，最簡單的作法便是改為從查詢網址搜尋。

除此之外，一步搜尋還有另一弱點：無法處理「0 距離目標」的情況。有些網站架構較簡單，本身便是作為活動列表的目的架設，因此該網站的首頁即被視為活動總覽頁面。另外當直接以活動總覽頁面本身輸入要求查詢時，此架構亦無法爬取該網址本身。以上兩個問題，分別以查詢網址與首頁網址作為答案便可解決。

B. 個別網站高頻重複誤判關鍵字

以此次作業資料集 id 92 號查詢為例: 程式執行過程中連續提交大量錯誤答案，且這些答案的內文皆為「有活動」。經了解該網站中有大量的連結 title 屬性皆被設置為「有活動」，其字串長度極短，對演算法而言為非常完美的答案。

此問題解決方案亦不複雜，在提交出錯誤答案的同時記錄該連結內文的前綴。若同樣前綴重複出現多次且連結皆非目標，便視為該網站的「個別網站高頻重複誤判關鍵字」，之後該網站中其他候選連結之內文若為相同詞墜，便直接忽略不再提交。

V. 討論與總結

從程式碼乍看之下或許會覺得稱此演算法為「搜尋演算法」稍嫌遷強。其中並沒有一般搜尋演算法的遞迴流程，且自稱啟發式搜尋卻沒有明顯的啟發函式。

實際上，第 II.部分闡述的策略即為此演算法之啟發函式。由於整個搜尋過程只有一步的距離，在這一步之內，用排序方式選擇出最可能是答案的鄰點，便是應用了啟發式搜尋的核心策略。

在引言中已經提過，此架構要表現出好的效能，有兩個必要條件：其一是問題本身給予「搜尋者」的視野必須非常廣，即在「稠密圖」上進行搜尋、其二是假定目標與起始點的距離為一。簡而言之，若問題本身的特性在一步之內便能有高度的 Completeness，此方法便會有驚人的效能。

除了一步搜尋的架構之外，高度的啟發也有利弊。以此次作業為例，需要深入分析各種規律及網頁開發者慣用的 HTML 編寫方式，過程十分艱辛。但是最終得到的模型相當強勢，此即為「工人智慧」。

演算法在這次作業中的查詢案例中，除了少部分架構較為特殊的網址，絕大部分的網址在幾次的提交便會命中答案。且許多網址皆是在第一次提交便為正確答案，意即直接在一步內找到最佳解。由於是人工分析規律，即使將此演算法拿來查詢更多其他網址，仍有把握維持一定程度的水平。最後再加上考量此次作業使用的驗證方案並不甚穩定，這樣看似稍嫌粗暴的策略，實則非常有效。

最終結果與驗證方案相關問題

經與助教討論後，紀錄結論如下。以下紀錄與當時討論之結論或有出入，起因於對演算法進行的更多改良導致程式提交答案之順序有變動。但由於此程式會輸出完整的紀錄檔案，本人可根據提交的網址對照得知助教認可 API 缺漏標記之答案對應的提交。

A. 原驗證結果

- 命中率 93%，命中成本 165，平均命中成本 1.774

B. 助教認可答案更正

- ID2: 應為第 1 筆提交
- ID6: 應為第 1 筆提交
- ID33: 應為第 2 筆提交
- ID38: 應為第 2 筆提交 (http / https 伺服器自動轉址)
- ID47: 應為第 8 筆提交 (伺服器自動轉址)
- ID51: 應為第 1 筆提交 (網址中含有日期極難標記)
- ID83: 應為第 1 筆提交
- 命中率 99%，命中成本 179，平均命中成本 1.808

C. 額外答案更正

以下紀錄與助教討論過後，額外發現有疑慮的答案驗證，皆為本人認為若詢問助教亦會得到認可的項目。詳情可對照提交作業中的「程式輸出紀錄」檔案，若沒有問題還請助教酌情參考。

- ID12: 應為第 1 筆提交 (.tw / .com)
- ID24: 應為第 1 筆提交。藝文網站之展覽資訊列表應優於最新消息。
- ID29: 應為第 1 筆提交。該網站的「消息活動」分為「最新消息」與「展會活動」兩個類別，最新消息未被標記為正確答案。該網站非藝文網站，最新消息中亦有許多重要活動資訊，且網址不加後綴時預設自動導引至此類別，而非展會活動。另外，該網站最新一筆活動停留在 2020 年初，因此是網站本身停止維護，而非活動資訊過期。
- ID37: 應為第 1 筆提交。該網站為教育機構官網，演講活動列表被標記為正確答案，但學生活動列表卻未被標記。
- ID48: 應為第 1 筆提交 (http / https 伺服器自動轉址)
- ID67: 應為第 1 筆提交。該網站為教育機構官網。雖然由網址觀察，第 1 筆提交之網址應為正確答案之子網頁。然而實際查看網頁內容，便會發現第 1 筆提交之網址其實顯示了所有該網站近期之活動資訊，應為正解，而被標記為正確答案之網址，卻導引至該教育機構之研發成果，且其中資訊最近一筆日期在 2020 年 6 月。
- ID70: 應為第 1 筆提交。該網站相當特殊，兩個不同的網址，卻顯示出一模一樣的內容。
- 命中率 99%，命中成本 167，平均命中成本 1.687