Similar Page Detection

專題編號：102-CSIE-S025
執行期限：101 年第 1 學期至 102 年第 1 學期
指導教授：王正豪 教授
專題參與人員： 99820324 鄭仕東
99820325 古宇辰

一、摘要

處在一個資訊爆炸的時代，「如何透過關鍵字去搜尋我們所要的資料」已經是我們當下最關注的問題之一了！然而，透過搜尋引擎的支持，我們總是能獲得成千上萬筆的結果。但是，在這茫茫的「網頁海」當中，又有多少是屬於內容一樣只是「換湯不換藥」地放在不同網站的呢？本專題就是以此為出發點，幫使用者過預先找出重複內容的頁面，並提醒使用者哪些可能為重複地內容，以幫助使用者節省在於找資料的時間與體力。然而本專題最後的呈現方式是會做成一個 Google 瀏覽器套件的型態。

關鍵詞：Copy Detection、Near-Duplicate、Google Chrome Extension。

二、緣由與目的

使用搜尋引擎搜尋需要的資料是每個人每天都會做的事情，但是在搜尋的過程中，常常會遇到重複內容不斷出現在各個網頁中的討厭情況。看似不同的搜尋結果，點進去才恍然大悟發現這是已經看過的內容，這種惱人的事一而再，再而三地反覆發生在我們的生活中，不僅浪費時間去點擊每個連結，更大幅耗損使用者的精神嚴重困擾著大家，實有解決之必要。

因此，我們便以此考量做為本專題的出發點，藉由「預先擷取各連結的主要內容，並做相對應的比較」，來達到不僅大幅節省使用者搜尋資料的時間，更可降低使用者搜尋時所耗費的體力。

隨著 Google Chrome 近年來使用率在世界各地大幅提升，因此我們選擇將我們的專題透過實作出一個 Google Chrome Extension 的方式來與以 Java Web 的架構所形成的後端做一個結合並呈現。

三、實做重點

首先，為了方便檢視本專題，此系統的架構圖如圖 1 所示：

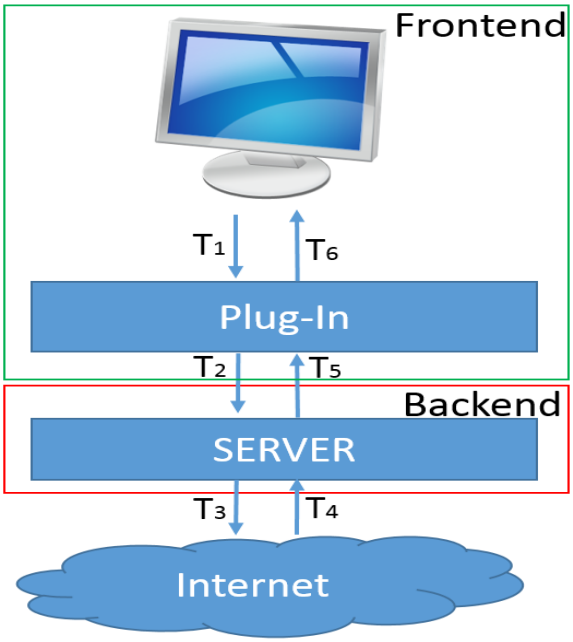

圖 1、系統架構圖

這張架構圖所呈現的是對於這個系統大致架構做一個簡單的描述。由此可知，我們的系統分成兩部分，分別為「Client(前)端」以及「Server(後)端」。在此，我們將對這兩部分的實作重點並且對照著代碼做詳細地說明：

- **Client端功能實作：**

T1：當User使用套件時，會呼叫套件上的前端程式(JavaScript)做擷取「此頁面所有連結」的功能。

T2：將T1完成後的資料，做一些進一步處理以及包裝成JSON的格式，再由Ajax

的方式將此POST到Server上的Servlet做後續的處理。

T6：Server完成工作後會將「相似對」回傳給前端。此時，套件便要針對此資料，去做剖析，並將出現在「相似對」的相對應資料去做一個呈現的動作。而這個步驟完成了之後，也就是這個系統的一個完整流程就已經算是結束了！

Server端功能實作：

T3，T4：抓取文章。當套件傳入JSON時，會先將JSON解譯成對應的物件，並將此物件中「所有連結」節取出來。得到某搜尋頁面上出現的所有網址後，再利用HTML Parser(jsoup)網路爬蟲程式去一一抓取每筆連結裡的主要內文。抓取後並將文章一一存入此user的List。

T5：此為本系統主要區塊。首先，拿到T4 處理過後的List之後，將其主要文章一一互相經過SimHash演算法去做比較。不僅僅只有比對此頁所有連結的主要內文，假若使用者有搜尋過其他頁，還得將此頁的這些主要內文送去與其他頁面的做一一比對。透過這樣的交互比對之後，最後將會以「相似對」的形式回傳，以表示「某一筆連結的內文與某另外一筆可能是重複的」。並且將此「相似對」包成物件，再以JSON的形式送回前端接收，完成後端的工作。

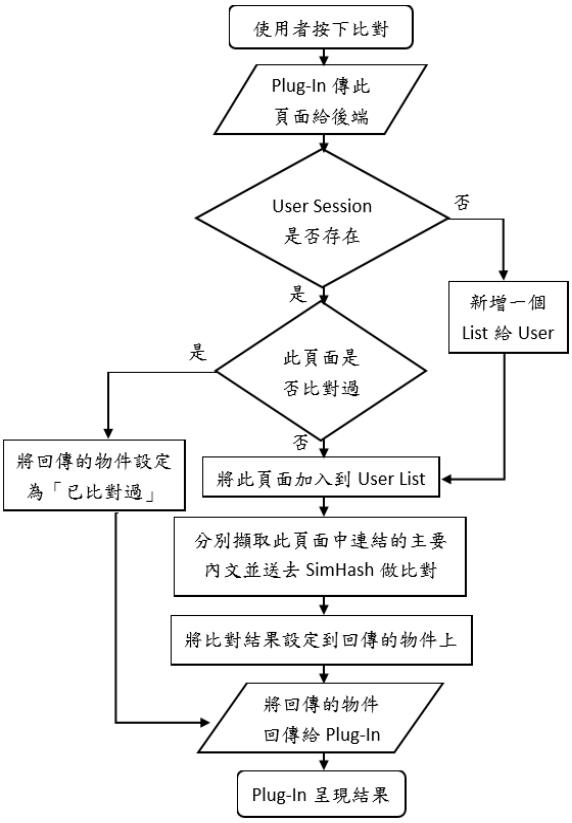

圖 2、系統流程圖

其中，SimHash 演算法實作的細節：接收到主要文章後，會先對文章進行斷詞，並計算詞頻，接著把文章斷詞所產生的單詞送去 hash，hash 所產生的結果再加上權重(詞頻)，最後，加總所有經過加權的結果即得到代表此文章的 fingerprint。算出此文章與另一篇要比對文章的 hamming distance 即可知道相似程度。

介紹完了我們的實作重點之後，我們以下用一個完整的「系統流程圖」來對本系統做一個最後的總結。如圖 2 所示。

四、未來展望

首要目標就是縮短等待時間，造成等待的原因有二，一是抓取文章，二是比對文章。由於抓取主要文章的速度是取決於各網站的回應時間，所以很難在數秒鐘內完成，最有效的方法是使用多台電腦作為 Server，即可大大縮短等待時間。比對文章方面則可以利用 Multi-Thread。

再來就是提升比對準確度，可能會依照文章特性使用不同的演算法，例如，超過五百字的文章適用於 SimHash。

參考文獻

[1] Jenq-Haur Wang and Hung-Chi Chang, "Exploiting Sentence-Level Features for Near-Duplicate Document Detection," Proceedings of the Fifth Asia Information Retrieval Symposium (AIRS 2009), pp. 205-217, 2009.

[2] Gurmeet Singh Manku, Arvind Jain and Anish Das Sarma, "Detecting Near Duplicates for Web Crawling" In Proc. WWW 07 Proceedings of the 16th international conference on World Wide Web Pages 141-150, 2007

[5] Googel Chrome Extension：http://developer.chrome.com/extensions/index.html