從搜尋引擎到文字探勘（上）

資料處理與分析是當代IT顯學，在大數據當道的時代，讓我們來探討搜尋引擎與文字探勘的應用

文/ 王建興 | 2014-09-04 發表

資料探勘的目的，就是希望從大量的資料中挖掘出有用的資訊。因此之後，將資料探勘的技術應用在商業上，希望找出一些發現或規則，可以運用於商業經營之中，包括商業決策或商務規則，以便帶來更多的商業利益。

因此，從資料探勘的技術出發，漸漸演變，成為所謂的「商業智慧（Business Intelligence）」領域。而在近年，伴隨著「大數據（Big Data）」的這個口號喊的震天價響，現在的軟體系統能夠儲存的資料量愈來愈大、計算能力也愈來愈高，從「海量級」的資料中，能夠透過探勘技術挖掘出的資訊、甚至是商業規則，也就更具吸引力。

文字探勘技術與搜尋引擎的崛起

文字探勘（Text Mining）被視為是資料探勘（Data Mining）的一環，其中有個關鍵的差別，在於傳統資料探勘所處理的資料，都是「結構性」的資料，也就是說，資料本身具有明確的結構，例如，像是一個固定結構的表格，每個欄位有其明確的定義及值。而資料探勘技術中的演算法，則是以這些結構性的資料為輸入，經過演算過程之後計算得到結果。但文字探勘不同於資料探勘的地方，則在於它的原始輸入資料，都是沒有特定結構的純文字，這些文字的內容，都是用人類的自然語言所寫成的，所以，無法直接套用資料探勘的演算法，來計算出些什麼有意義的東西。

在我們生活當中，除了具結構性的資料，也有相當大量的文字資料，像是每天的新聞、人們在Facebook、Twitter、微博上所發表的近況更新、部落格文章、專利文件等等。這些自然語言文字型的資料中，同樣蘊藏可觀、極具潛力的「礦產」，也就是有價值的資訊，等著我們用資訊技術去開採。這就是文字探勘技術及應用所希望達成的目標。

另一方面，大約在十幾年前，搜尋引擎的技術開始普及了起來，除了有一些商業公司開始釋出搜尋引擎的實作，授權其他公司使用之外，也有開放原始碼的專案持續的開發並釋出，其中最有名的，莫過於Apache Project 的Lucene了。

隨著開放原始碼的搜尋引擎專案的壯大，除了讓開發社群愈來愈輕易就能開發運用搜尋引擎能力的應用系統之外，也讓世人更容易得窺搜尋引擎設計之堂奧。同時，在開放原始碼的社群裡，許多以搜尋引擎為核心的程式庫及系統，也得以建立在Lucene的基礎上，陸續問世。像是基於Lucene、提供分散式索引搜尋能力的Solr，還有提供 crawling──即俗稱「爬網頁」系統的Nutch、……等等。這些在 Apache 之下的開放原始碼專案，以Lucene為核心，形成了一個生態系，也為想要運用搜尋引擎機制的開發者提供了各種便利的工具。

最初，開發者對於全文搜尋引擎的需求，就是很典型的在索引各輸入文件之後，能夠搜尋出符合給定查詢關鍵字的文件。但是，其實當我們將文件送至搜尋引擎進行索引之後，就建立起進行許多文字探勘應用時所需的基礎。搜尋引擎不只可以提供全文搜尋的功能，更同時幫我們把做文字探勘時需要的一些基礎建設都一併建立好了。我們可以在全文搜尋引擎的基礎上，發展許多文字探勘的功能。

文字探勘與搜尋之間的關係

為什麼說全文搜尋引擎在建立文件索引的同時，也幫文字探勘建好了基礎建設呢？因為建立文件索引的過程，基本上就是在將沒有結構的文字資料，轉換成為結構性的數值性資料，而且還要易於查詢。

概念上來說，搜尋引擎的索引結構是將每個文件以及該文件中所含有的「詞」，表示成為一個向量。例如，所有的文件中總共有 n 個可能詞，從w1到wn，那麼就可以用一個 n 維的向量來表示一個文件。在這種表示方式下，若一個文件被表示為向量 V，其中的第i個元素為 v(i)，則 v(i) 的值，為詞wi的權重。

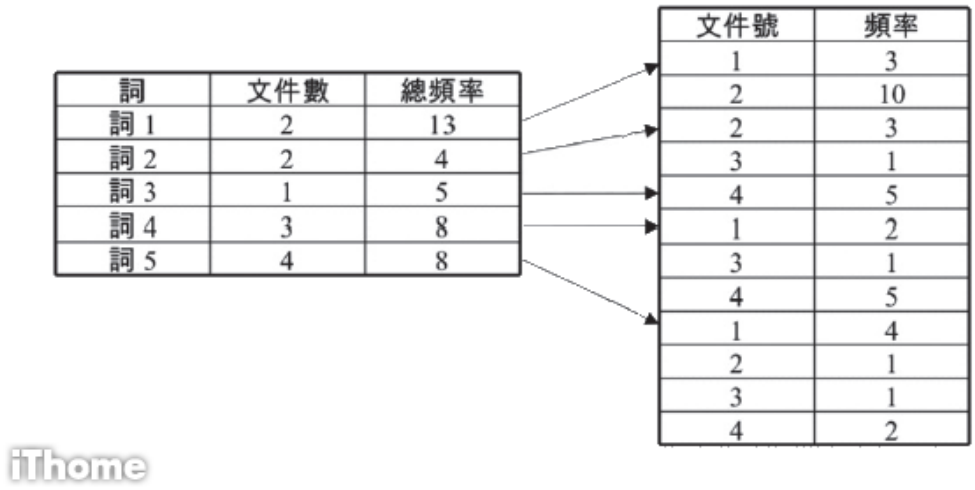
至於權重應該使用什麼表示方式呢？最直覺的方式就是使用該詞在文件中所出現的次數。因此，倘若我們有四份文件，這五份文件裡總共出現了五個不同的詞，那麼就分別將它們表示成為四個向量，每個向量中的元素，其權重為該對應之詞出現於該文件中的次數（**如圖1**）。

文件號	詞 1	詞 2	詞 3	詞 4	詞 5
1	3	0	0	2	4
2	10	3	0	0	1
3	0	1	0	1	1
4	0	0	1	5	2

圖1：我們以表列的方式來呈現5個詞在4份文件當中的權重。

當然，這樣子的結構純粹是概念上，實際上如果用一個很大的矩陣來儲存表示每份文件的向量，那麼無論在空間上、或是在時間上，都不會是一個有效率的表示方式。因此，在概念上，又演化成一種我們稱為反向索引（Inverted Index）的結構。請注意，這仍然是一種概念上的表示方式，實務上，會有更有效率的儲存結構。

如**圖2**所示，反向索引所需的空間比單純的矩陣表示方式少多了，因為若用向量來表示文件，那麼，向量中的大多數元素其值可能都是零，因為並不是每份文件都會出現大多數的可能詞。但採用反向索引的表示方式後，所用的空間可以降低。此外，你可以留意到，反向索引的結構也有利於查詢。



詞	文件數	總頻率
詞 1	2	13
詞 2	2	4
詞 3	1	5
詞 4	3	8
詞 5	4	8

文件號	頻率
1	3
2	10
2	3
3	1
4	5
1	2
3	1
4	5
1	4
2	1
3	1
4	2

圖2：將5個詞在不同文件出現頻率的權重關係結構呈現上，改以反向索引的方式呈現。

而且，我們從圖示中左方的表，可以看出來它是以查詢為主的設計，從這張表可以一目了然，得知每個詞出現的總文件數及總頻率，而左方表格的每一列，都有一個連結，連結至右方表格的某一列，這代表著左方表格中的詞所出現的第一份文件，因為我們知道該詞總共出現了幾個文件，所以，我們可以沿著連結所指向右方表格的列，往下找出所有出現過該詞的文件。

舉例來說，詞 4 總共出現在三份文件裡，連結至右方表格中的第一列是文件 1，接著是文件 3 及文件 4。透過右方的表格，我們除了可以知道該詞出現在那些文件，也能得知該詞在各文件出現的頻率。

對搜尋引擎來說，這樣的結構可以滿足最基本的需求。因為當輸入關鍵詞之後，搜尋引擎便可以拿關鍵詞去比對反向索引左方的表，接著沿著指向右方表格的連結，找出含有該詞的所有文件。若關鍵詞有多個，可以綜合多個查詢結果做集合運算，即可得到最後的結果。

以上，便是搜尋引擎所建立結構的基礎概念，但這只能做到找出符合關鍵詞的文件，並不能提供搜尋結果優劣的排序。此外，這樣的索引結構和我們想談的文字探勘又有何關聯呢？就讓我們在下一回繼續介紹了。

相關報導請參考「從搜尋引擎到文字探勘（中）」「從搜尋引擎到文字探勘（下）」

作者簡介

王建興

目前在一家網路應用軟體公司擔任技術長的工作，專長是物件導向設計以及Internet應用系統的開發。他過去的研究興趣包括：點對點網路、分散式網路管理、行動式代理人、感知網路。從企業應用軟體系統，到個人行動裝置上的應用，他都有一些開發的經驗。並且對於網路創業及網路應用的發展趨勢，持續保持高度的關心。

更多 iThome相關內容

- 宏碁探勘社群文字掌控品牌聲譽
- 大資料時代，保險公司最怕的對手不是同業而是Google

- Google AI今天中午將出戰南韓圍棋好手，YouTube將實況轉播
- 從搜尋引擎到文字探勘（中）
- 科學家分析Twitter推文評估颶風災損，效果更勝模型
- Gartner：預測型的進階分析工具竄起，2018年全球過半企業將搶要用

熱門新聞

【MIS必看】WannaCry勒索病毒猖獗！上班第一天如何處理病毒未爆彈？TWCERT/CC教你6步驟自保
2017-05-14

如何躲過WannaCry勒索蠕蟲風暴？週一上班先不要開電腦，照著這些方法做
2017-05-14

XP也有救！WannaCry 2.0勒索軟體肆虐，微軟破例再釋Windows XP修補
2017-05-14

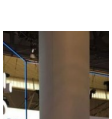
別哭！被WannaCry加密的檔案有機會救回
2017-05-16

WannaCry企業自救術
2017-05-19

微軟譴責美國政府監控政策引發WannaCry災情!
2017-05-15

史上最大WannaCry勒索蠕蟲，到底有沒有WannaCry 2.0變種？
2017-05-14

WannaCry的幕後攻擊者是誰？新證據指向北韓
2017-05-16

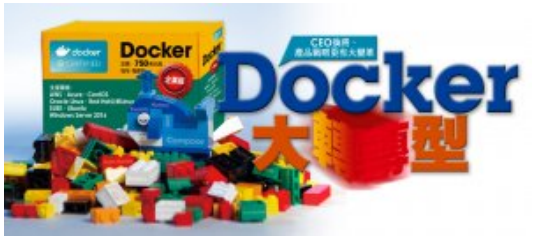


思科營收連六季負成長，準備再裁1100名員工
2017-05-18

XP用戶中了WannaCry先別慌，資安業者有解

2017-05-19

專題報導



Docker大轉型

SD-WAN崛起，企業重新定義連外網路

2017超融合應用伺服器採購大特輯【一線大廠篇】

企業Chatbot新機會

強化資安意識，鞏固惡意郵件進逼的最後防線

更多專題報導