本體論架構應用於文章詞彙相似度之比對

程彥儒
中正大學電機工程學系
d95415007@ccu.edu.tw

劉立頌
中正大學電機工程學系
aliu@ee.ccu.edu.tw

黃皇賓
中正大學電機工程學系
g94415037@ccu.edu.tw

摘要

資訊檢索為近年來廣泛應用的技術，為了增加檢索的準確度和效率，文章分類和以文找文成為近年來熱門主題。其主要方法是利用文章的詞彙做比對找出文章相似度。若只從字面比對有同義詞和一詞多義問題，需利用語意做比對處理。使用知識基礎的語意比對方法，受限於知識來源並需要大量人力成本；使用統計基礎的語意比對方法，只能找出相關詞而非同義詞且需收集處理大量文章。本論文利用本體論之元素延伸建立文章語意比對演算法，改善傳統在相似度上無法比對出文章的議題，並提出相對應之方法流程。

關鍵詞：本體論架構、文章相似度比對。

一、緒論

近年來由於網際網路的發達，許多資訊透過網路被複製、傳播和儲存，因此造成了資訊量的暴增。資訊使用者需耗費大量的時間，才能從大量的資訊找到需要的資訊，因此資訊檢索就顯得相當重要。為了增加檢索的準確度和效率，文章分類[10]和以文找文[6]成為了近年來熱門的主題。主要的方法是利用文章的詞彙做比對以找出文章的相似度，但比對的方法往往只從字面上去做比對，並沒有在語意層次上做比對，因此無法考慮到同義詞和一詞多義的比對問題。

針對語意問題的處理，最常見的方法是利用 WordNet 做為知識基礎。WordNet 定義同義詞的集合和集合之間的關係[14]，因此可以利用 WordNet 的知識找出同義詞。另一種方法是利用人工對文集做前置處理，如文章分類、語意標記，以避免語意上發生的錯誤，但這些方法都需要大量的人工來建立字典和做前置處理，因此相當耗時耗力。

針對需要大量人工的問題，常見的解決方法是統計的方法。統計方法是由大量文集統計出詞彙出現在文章的頻率，利用此頻率來做比對的依據，處理語意上的問題。但統計方法需大量的文集來增加準確度，因此需要耗費成本來處理大量額外的文集和有可能發生收集不到相關文集的問題。只利用詞彙在文章出現頻率的關係做為比對的依據，只能估計出相關詞[1]的相似度，並不能更深入的找出語意上的差異。

針對上述的問題，本論文提出一個語意層次的比對方法，以解決同義詞和一詞多義比對的問題。同時避免耗費大量人工和時間以及依賴字典與收集並處理額外文章的缺點。為了達到語意上的比對，需要語意相關的知識做為比對的依據，但為了不依賴字典或統計數據，因此取法於本體論定義出了一個語意知識表示模型，以描述文章所包含的語意結構。而且為了不花費太多的人力收集並處理大量文章，因此採用樣式基礎的擷取方法自動從文章中擷取出語意結構。樣式基礎的擷取方法是事先定義出樣式，並利用樣式比對以擷取出文章裡詞彙和詞彙之間的關係[11]。從文章擷取出語意結構後，就能利用語意結構做為語意比對的依據。由於語意結構是取法於本體論，因此利用本體論之間的語意比對方法做為文章詞彙相似度比對的方

法。

二、研究動機

文章間同義詞的比對方法可分為二種，利用 WordNet 為知識基礎的比對方法和潛在語意分析(Latent Semantic Analysis, LSA)[16]為統計基礎的比對方法。文章間一詞多義的比對方法也可以分為二種，利用人工標記文集的監督式語意歧異解析[5]和利用統計方法的監督式語意歧異解析[9]。

一般利用 WordNet 或一般英文字典做為知識基礎的比對方法，最大的缺點是依賴性，當字典中沒有欲比對的詞彙時，就會發生無法比對的問題[3]。而現實生活中的很多人名或專有名詞會隨著時間不斷的被產生，且建立或更新英文字典需花費大量的人力。潛在語意分析為統計基礎的比對方法只利用詞彙在文集出現的頻率做為比對知識，因此只能提供相關詞的比對。

本論文平衡這些方法的優缺點，提供一個有效率的比對方法；此方法不需要額外的知識基礎，可以達到語意上的比對，而不僅是相關詞的比對。

(一) 文章語意結構

本論文參考本體論的定義，使用語意結構來表示文章的語意。本體論包含三個重要的元素：概念、屬性和關係[6][8][15]。概念代表真實環境中的某個類別，屬性是對於概念更詳細的敘述，關係表現出概念之間組成的結構。Keet [7]指出結構可以代表出二個本體論中的概念之間語意上的差異，因此透過結構的比對就能找出二個本體論中的概念之間語意上的差異。但是一般的文章並沒有像本體論一樣的結構來表示出語意，因此要比對文章之間詞彙語意上的相似度就需借由額外的輔助知識，如英文字典。

將文章轉換成類似本體論的結構，就能透過結構來表達詞彙之間的語意，也就不需借由額外的輔助知識，直接從文章的語意結構中來評估詞彙之間的語意相似度。因此可事先定義語意結構來表示文章中的語意。在 WordNet 中定義出了詞彙的本體論，把概念看成是名詞同義詞的集合，概念之間用關係來組合在一起，並透過組合出來的階層式架構來敘述這些概念。本論文延伸 WordNet 的詞彙本體論，利用文章中的動詞和介詞做為文章中的概念之間的關係，對於文章的概念加以敘述，如此一來透過動詞和介詞組織成的結構就能表示出文章裡名詞的語意。

三、文章語意比對演算法

本論文提出的文章語意比對演算法流程如圖 1 所示，流程中包含三大部份：文章前處理、語意結構擷取和語意比對。文章 p 和文章 q 分別代表不同的二篇文章。文章前處理包括下列幾項：詞性標注、片語標注和停字過濾。詞性標注是對文章中的單字做詞性分析並貼上詞性標籤；片語標注是對文章做斷字處理並做片語類型標注；停字過濾是過濾冠詞、副詞等語意結構擷取用不到的單字。語意結構擷取的方法是利用樣式基礎的擷取，透過樣式比對可以分別擷取文章 p 和文章 q 的語意結構 p 和語意結構 q。最後做語意上的比對。

(一) 文章前處理

本論文中所定義的語意結構是由名詞、動詞和介詞所組成，因此利用詞性標注的工具對文章進行這些詞性做標注。本論文是使用工具 CRFTagger 進行詞性標注[17]，此工具是採用條件式隨機域(Conditional Random Fields)的統計模型來處理詞性標注。

名詞可為修飾詞加上名詞後所組成的名詞片語；動詞也可為動詞加上介詞後所組成的動詞片語，因此可對文章進行片語標注的處理。由於已經對文章做詞性標注的處理，並加入了詞性資料，因此可利用詞性資料進行片語標注。本論文利用工具

CRFChunker 來處理片語標注[17]。

此外，在語意結構中利用到文章的名詞、動詞和介詞，因此為了方便處理文章，對文章做停字過濾的處理，如冠詞、副詞等。

(二) 擷取文章的語意結構

經過詞性標注和片語標注等文章前處理後，已把詞性和片語的資料加入文章裡。接下來的問題是如何利用這些資料從文章之中擷取語意結構。Hearst[11]利用詞彙句法樣式基礎的擷取方法，提供簡單有效的方法擷取 WordNet 的詞彙本體論。因為樣式基礎的擷取方法不需要額外的知識庫和複雜的剖析，只需要定義簡單的樣式，並利用樣式比對方法來擷取概念和概念之間的關係，所以不會有太多擷取成本。本論文參考此想法，利用樣式基礎擷取文章的語意結構，並建立詞彙句法樣式以做為擷取的基礎。但這些樣式並不能保證適用於所有文集，因此需針對欲比對之文集作增加或修改樣式的工作。依據 Finkelstein-Landau 和 Morin 的詞彙句法樣式的流程[12]和本體論與本論文語意結構之差異修改後的建立流程如下：

1. 對文集的句子做詞性標注、片語標注和停字過濾等前處理。
2. 針對前處理完的句子，每個句子每次只選擇一對詞彙做為擷取目標。
3. 針對步驟 2 找出的成對詞彙，選定成對詞彙之間欲擷取的關係。
4. 把完成詞彙和關係選定的句子表示成詞彙句法表示式。
5. 利用人工比對步驟 4 所找出的詞彙句法表示式之間的相似度，並利用人工對詞彙句法表示式一般化，以找出詞彙句法樣式。
6. 透過專家對詞彙句法樣式做驗證。
7. 利用驗證過的樣式擷取文章的語意結構。
8. 透過專家對擷取出來的語意結構做驗證，如果發生錯誤，由專家修正錯誤的樣式，如果擷取不完全則重覆步驟 2~8 以新增新的樣式。

透過上述的流程做樣式比對就可以找出詞彙之間的關係。範例如下：

(S) [[NP, Chien-Ming/NNP, Wang/NNP], [VP, begin/VB], [NP, season/NN], [IN, on/IN], [NP, disabled/JJ, list/NN], [after, after/IN], [VP, pulling/VBG], [NP, right/JJ, hamstring/NN], [IN, on/IN], [NP, Friday/NNP]]

(P9) NP1 {[,] NP2 ...[,] [or | and] NPm} IN NPm+1 {[,] NPm+2 ...[,] [or | and] NPm+n} → R(IN, NPi, NPj), 0<i<=m, m<j<=m+n

(P11) NP1 {[,] NP2 ...[,] [or | and] NPm} {'NP | 's NP | and other LIST | and other LIST | especially LIST | especially LIST | IN LIST | W VP LIST } {[,]} VP NPm+1 {[,] NPm+2 ...[,] [or | and] NPm+n}*

→ R(VP, NPi, NPj), 0<i<=m, m<j<=m+n

(P12) NP1 {[,] NP2 ...[,] [or | and] NPm} {'NP | 's NP | and other LIST | and other LIST | especially LIST | especially LIST | IN LIST | W VP LIST| VP LIST }+ [after | before] VP1 NPm+1 {[,] NPm+2 ...[,] [or | and] NPm+n}

→ R(VP1, NPi, NPj), 0<i<=m, m<j<=m+n

句子 S 透過樣式 P9 的比對可以找出 R(on, season, disabled list)和 R(on, right hamstring, Friday)的關係；透過樣式 P11 的比對可以找出 R(begin, Chien-Ming Wang, season)的關係；透過樣式 P12 的比對可以找出 R(pulling, Chien-Ming Wang, right hamstring)的關係。透過上述的方法就能建立出文章的語意結構。

(三) 詞彙語意比對

擷取出文章的語意結構後，接下來就是如何利用二篇文章的語意結構比對文章之間詞彙的語意。Rodríguez 和 Egenhofer[13]利用欲比對概念之相鄰的概念互相比較，計算本體論之間概念語意相似度。由於本論文提出的語意結構是取法於本體論的結構，因此可利用類似的方法來計算語意結構之間詞彙的語意相似度。但由於語意結構並不完全相同於詞彙本體論，因此必需做一些修改。本論文語意結構的知識完整性是與文章的內容相關的，不像本體論一開始就由人工建立好完整的知識，因此在比對時從文章裡擷取出的語意結構，其知識完整性會影響到比對的結果。利用權重值來調整字面、屬性和語意相似度佔整體相似度評估的比例，而權重

值可由使用者依經驗自訂。本論文的語意結構並不包含屬性，因此將屬性相似度的權重值訂為零。另外，為了避免從文章擷取出的語意結構之知識完整性影響到比對的結果，加入相關參數來調整字面和語意相似度的權重值，即知識完整性越高，語意相似度佔整體相似度評估的比例越大。整理後的公式如(1)所示。

$$S(a^p,b^q)=w(a^p,b^q)*S_n(a^p,b^q)+(1-w(a^p,b^q))*S_w(a^p,b^q) \quad (1)$$

$S(a^p,b^q)$代表語意結構 p 裡概念 a 和語意結構 q 裡概念 b 之間的相似度評估，$S_n(a^p,b^q)$代表語意結構 p 裡概念 a 和語意結構 q 裡概念 b 之間語意上的相似度，$S_w(a^p,b^q)$代表語意結構 p 裡概念 a 和語意結構 q 裡概念 b 之間字面上的相似度，由於沒有包含屬性，因此去掉屬性相似度的評估。$w(a^p,b^q)$代表評估函數的權重值，其公式如(2)如示。

$$w(a^p,b^q)=\begin{cases}\dfrac{|A_n^p||B_n^q|}{\alpha} & |A_n^p||B_n^q|\le\alpha,\alpha>0\\ 1 & |A_n^p||B_n^q|>\alpha\end{cases} \quad (2)$$

其中$|A_n^p|$代表本體論 p 裡 a 概念的相鄰概念的數量，$|B_n^q|$代表本體論 q 裡 b 概念的相鄰概念的數量。α是由使用者自定的語意信任門檻值，α值是代表當語意結構的語意知識完整度超過多少時使用者才完全相信語意評估結果，反之則由語意知識完整度和α值的比例做為分配的權重。語意知識完整度是 a 概念的相鄰概念的數量和 b 概念的相鄰概念的數量的乘積，乘積越大表語意知識完整度越高。

評估二個本體論相似度前必需先找到最近的父節點把二個本體論連接在一起，之後利用概念在連接後的本體論裡與父概念之間的距離做為評估的標準。本論文的語意結構並沒共通的根節點，故無法利用相同的方法評估，因此本論文使用另一種評估的方式。文章的名詞片語是修飾詞加上名詞所組成，而修飾詞是對於名詞做更進一步的修飾，即對原名詞做特殊化敘述，如果把名詞視為現實生活中實體所組成的集合，而修飾後的名詞就代表集合中的子集合，如球代表現實生活中所有球的集合，而白色的球就代表球的集合裡白色的球所組成的子集合，因此如果名詞片語的修飾詞越多可以視為越特殊化的敘述，換個角度即名詞片語裡單字的數量越多就越特殊化。整理後的公式如(3)和(4)所示。

$$S_w(a^p,b^q)=\frac{|A_w^p\cap B_w^q|}{|A_w^p\cap B_w^q|+\alpha(a^p,b^q)|A_w^p/B_w^q|+(1-\alpha(a^p,b^q))|B_w^q/A_w^p|} \quad (3)$$

where

$$\alpha(a^p,b^q)=\begin{cases}\dfrac{word\_count(a^p)}{word\_count(a^p)+word\_count(b^q)} & word\_count(a^p)\le word\_count(b^q)\\ 1-\dfrac{word\_count(a^p)}{word\_count(a^p)+word\_count(b^q)} & word\_count(a^p)>word\_count(b^q)\end{cases}$$

$$S_n(a,b)=\frac{|A_n^p\cap B_n^q|}{|A_n^p\cap B_n^q|+\alpha(a,b)|A_n^p/B_n^q|+(1-\alpha(a,b))|B_n^p/A_n^q|} \quad (4)$$

where

$$|A_n^p\cap B_n^q|=\left[\sum_{i\le n}\max_{j\le m}S(a_i^p,b_j^q)\right]$$

$$S(a_i^p,b_j^q)=w(a_i^p,b_j^q)*S_n(a_i^p,b_j^q)+(1-w(a_i^p,b_j^q))*S_w(a_i^p,b_j^q)$$

(四) 詞彙語意比對範例

圖 3 是圖 2 文章的部份語意結構，圖 5 是圖 4 文章的部份語意結構。將語意結構 a 的詞彙(right hander)和語意結構 b 的詞彙(Chien-Ming Wang)做語意比對，二者字面上的相似度利用公式(3)來評估，結果如下：

$$\alpha(right\_hander,Chien\_Ming\_Wang)=\frac{word\_count(right\_hander)}{word\_count(right\_hander)+word\_count(Chien\_Ming\_Wang)}=\frac{2}{2+3}=0.4$$

$$S_w(right\_hander^a,Chien\_Ming\_Wang^b)=\frac{|\{\ \}|}{|\{\ \}|+0.4|\{right,hander\}|+0.6|\{Chien,Ming,Wang\}|}=0$$

二者字面上的相似度利用公式(4)來評估，結果如下：

$$A_n^p = \{\text{conditioning drills, season, right hamstring, friday, disabled list}\}$$

$$B_n^q = \left\{\begin{matrix}\text{month, leading candidate, Opening Day, right hamstring, season,}\\ \text{disabled list, Friday}\end{matrix}\right\}$$

$$S_n(right\_hander, Chien\_Ming\_Wang) = \frac{4}{4+0.4*1+0.6*3} = 0.645$$

最後利用公式(1)把二者整合在一起，在此 α 設為 40，最後比對的結果如下：

$$w(a^p, b^q) = \frac{|A_n^p||B_n^q|}{\alpha} = \frac{5*7}{40} = 0.875$$

$$S(right\_hander, Chien\_Ming\_Wang) = 0.875*0.645 + 0.125*0 = 0.564$$

計算的結果可以發現 right_hander 和 Chien_Ming_Wang 之間有 0.564 的相似度。從計算的過程中可以發現，0.564 是代表，語意相似度佔 87.5%和字面上的相似度佔 12.5%的情況下，二個詞彙的相鄰概念有 64.5%的交集(語意上相似度為 0.645)，而字面上完全不一樣(字面上相似度為 0)。0.564 代表二個詞彙的相近程度。二個詞彙是否能為同義字，通常需使用者根據某個應用領域的經驗設定門檻值。二個詞彙的相近程度大於門檻值代表二個詞彙在應用領域很有可能是同義詞。

(五) 其他方法之比較

與其他方法的比較結果如表 1。從表中可以看出，語意結構基礎語意比對方法相對其它方法，只需較少的建立輔助知識的人力成本，而且不需處理額外的文章。雖然在處理單篇文章需要較多的成本，但整體成本還在可接受的範圍。

四、實驗結果

本章節依據上一章所提出的方法進行文章之間詞彙的語意比對實驗。

(一) 實驗環境

實驗範例由 www.samspublishing.com 的文章 Vacationing in Java 取其中的一些片段組成一篇小短文。利用實驗研究方式，驗證本論文提出的方法。利用工具 CRFTagger 和 CRFChunker 對文章進行標注前處理[17]，之後使用本論文所發展的文章詞彙語意比對工具，對文章進行語意結構擷取和比對的工作。CRFTagger 是利用 WSJ 文集訓來出來的模型，經設計者測試可達到 97%的正確率，每秒可處理 500 個句子[17]。CRFChunker 也是利用 WSJ 文集訓來出來的模型，經設計者測試 F1 的分數可以達到 95.77，每秒可處理 700 個句子[17]。文章詞彙語意比對工具是以上一章方法所設計的工具。

(二) 實驗設計

為了驗證本論文提出的文章詞彙之間的相似度比對的方法，因此使用了三篇文章。文章 1，如圖 6 所示，是從網路擷取並整理後的文章片段。文章 2，如圖 7 所示，是將文章 1 中的大部份詞彙(斜體字加粗體)用同義字取代。文章 3，如圖 8 所示，是將文章 1 中的大部份詞彙保留，但修改一些詞彙(斜體字加粗體)造成文章語意改變。本論文採用了 Jaccard 係數[2]進行驗證，以 $\frac{|A^p \cap B^q|}{|A^p \cup B^q|}$ 表示，其中 A^p 代表文章 p 裡的詞彙集合 A，B^q 代表文章 q 裡的詞彙集合 B。

為了驗證本論文提出的文章中詞彙之間語意比對方法能成功的找出同義字和一字多義的字，一開始利用 Jaccard 係數計算文章 1 與文章 2 之間的相似度與文章 1 與文章 3 之間的相似度，之後加入本論文的方法於 Jaccard 係數的相似度計算，並觀察是否能提高辨識度。

(三) 實驗結果

表 2 是三篇文章的詞彙，表 3 是文章與詞彙的相關矩陣。利用 Jaccard 係數計算文章 1 與文章 2 之間的相似度與文章 1 與文章 3 之間的相似度的結果如表 4 所示。圖 9、圖 10 和圖 11 分別代表文章 1、文章 2 和文章 3 的部份語意結構。利用文章的詞彙之間語意比對方法比對出文章 1 與文章 2 裡詞彙之間的語意相似度和文章 1 與文章 3 裡詞彙之間的語意相似度，如表 5

和表 6 所示。由於三篇文章都是短文，知識完整性不會太高，因此將α值(語意信任門檻值)設為 2，代表知識完整性大於 2 就相信語意相似度評估結果。另加上語意相似度門檻值 0.6，只要語意相似度達到 0.6 就認為二個辭彙是相關的。最後再利用 Jaccard 係數計算經過語意比對後文章 1 與文章 2 之間的相似度和文章 1 與文章 3 之間相似度的結果如表 7 所示。

(四) 實驗結果分析及討論

文章 2 為將文章 1 的詞彙換成同義字，文章 3 則是更換文章 1 少數詞彙的語意，造成文章語意改變。基於上述的設計，文章 1 與文章 2 是語意較相近的文章，文章 1 與文章 3 是語意差異較大的文章。比較表 4 和表 7 可以發現，利用傳統資訊檢索的文章相似度計算方法，由於只利用詞彙字面上的比對，因此文章比對結果文章 1 與文章 2 差異較大，而文章 1 與文章 3 較相近。同時也可看出傳統資訊檢索的文章相似度計算方法並無比對出文章的語意相似度。加入語意比對後的文章比對結果，文章 1 與文章 2 較相近，而文章 1 與文章 3 差異較大。所以證明本論文的方法能比對出文章的詞彙之間語意相似度。

經由檢視表 5 和表 6 可發現，本論文提出的語彙之間語意比對方法，在語意比對上會還是會有誤判情形發生，如表 5 中的 Java vacation 和 java。從圖 9 和圖 10 中可發現，Java vacation 和 Java travel 底下只有一個節點 place，從表 5 中可以找到文章 1 和文章 2 裡 place 的語意相似度值為 0.83，而把α值設為 2，計算的結果為 $(1/2)*0.83+(1-1/2)*0.5=0.66$，文章 1 的 Java vacation 和文章 2 的 Java travel 語意相似度為 0.66。而從圖 9 和圖 10 可以發現，文章 2 的 Java 底下沒有任何節點，所以 Java vacation 和 java 只能從字面上來比對，比對結果為 0.75。因為 0.75>0.66，所以結果為 Java vacation 和 java 反而比 Java vacation 和 Java travel 來得相似。但是如果把α值設為 1，Java vacation 與 Java travel 的語意相似度變成 $(1/1)*0.83+(1-1/1)*0.5=0.83$，則 Java vacation 和 Java travel 就會變得比較相似。

由以上敘述可以發現，從文章擷取到的知識完整度會影響語意相似度的計算結果。知識完整度越高表示有更多的知識幫助語意相似度的計算，知識完整度太低時只能從字面上去比對。而α值(語意信任門檻值)代表知識完整度高於多少時就可完全信任語意相似度的結果，反之則只能比對字面上的相似度。透過α值的調整能幫助修正比對結果。但是α值如果調太低表示太容易信任語意比對，有可能兩個字彙只因為一點點的知識交集就被誤認為是同義字，但實際上它們只有一點點相關。

另一個值得討論的值是語意相似度門檻值，實驗定為 0.6，如果調低到 0.2 或調高到 0.9 都會影響到文章相似度比對結果。而這個值代表是否能正確的辨識二篇文章之間的差異，是屬於文章相似度比對的需要探討的範圍。而本論文主要是針對文章詞彙之間的語意比對，因此在此並不深入探討。

五、結論與未來展望

一般使用知識基礎方法，受限於知識來源需要大量人力來幫助建立。而使用統計基礎方法，又會有只能找出相關詞和需額外收集並處理大量文章的問題。本論文提出的方法，只需要借助有現有的英文句結構樹資料庫(Penn Treebank)訓練出來的條件式隨機域模型(Conditional Random Fields Model)來進行詞性和片語標注。透過 Phan 的實驗[17]，可以知道本論文使用的詞性和片語標注工具平均每秒可以處理 290 句的句子。且本論文使用的詞彙句法樣式(Lexico-syntactic Patterns)只需花費少量人力建立的，之後透過樣式比對擷取出文章詞彙的語意結構。最後透過簡單的語意結構比對就能找出文章裡同義詞和一詞多義的詞彙。

未來將更進一步探討α值(語意信任

門檻值) 和語意相似度門檻值與語意比對結果之間的關係。並且應用於大量文章的相似度比對，更進一步實際驗證方法的效果。另外，也將本論文所提出的方法真正的應用於實際網路上的文章分類和以文找文的問題。

六、參考文獻

[1] 石逸民，「從全球資訊網擷取同義詞」，國立中正大學資訊工程研究所博士論文，2003。

[2] 許中川，陳景揆，「探勘中文新聞文件」，資訊管理學報，第 7 卷，第 2 期，pp. 103-122，2001.

[3] 陳以理，林蘭綺，吳典松，「自然語言處理技術於專利文件分析之應用」，第二屆學生計算語言學研討會，2004。

[4] 黃雲龍，張佑任，「中文全文資訊檢索之效能評量初探」，資訊管理研究，第二期，pp37-60，2002。

[5] 賴育昇，李坤霖，吳宗憲，「網際網路 FAQ 檢索中意圖萃取與語意比對之研究」，Proceedings of ROCLING XIII，Taipei，Taiwan，2000.

[6] A. Suarez, M. Noeda and M. Palomar, “A Method of Restricted Knowledge Acquisition from WordNet,” Proceeding of the third International Conference on Knowledge-Based Intelligent Information Engineering System, IEEE, pp. 38-41, 1999.

[7] B. Chandrasekaran, J. R. Josephson, and V. R Benjamins, “What Are Ontologies, and Why Do We Need Them,” IEEE Intelligent Systems, pp.20-26, Jan.-Feb, 1999.

[8] C. Maria (Marijke) Keet, “Aspects of Ontology Integration,” Literature research & background information for the PhD proposal, School of Computing, Napier University, Scotland, 2004.

[9] C. S. Lee, Y. H. Kuo, C. H. Liao, and Z. W. Jian, “A Chinese Term Clustering Mechanism for Generating Semantic Concepts of a News Ontology,” Journal of Computational Linguistics and Chinese Language Processing, vol. 10, no. 2, pp. 277-302, 2005.

[10] D. Yarowsky, “Unsupervised Word Sense Disambiguation rivaling Supervised Method,” Proceedings of the Thirty-third Annual Meeting of the Association for Computational Linguistics, pp. 189-196, 1995.

[11] M. A. Hearst, “Automated Discovery of WordNet Relations,” To Appear in WordNet: An Electronic Lexical Database and Some of its Applications, Christiane Fellbaum (Ed.), MIT Press, 1998.

[12] M. Finkelstein-Landau and E. Morin, ”Extracting semantic relationships between terms: supervised vs. unsupervised methods,” Workshop on Ontologial Engineering on the Global Info. Infrastructure, 1999.

[13] M.A. Rodriguez and M.J. Egenfoher, “Determining Semantic Similarity among Entity Classes from Different Ontologies,” IEEE Transactions on Knowledge and Data Engineering, vol. 15, no. 2, pp. 442-456, 2003.

[14] N. F. Noy and C. D. Hafner, “The State of the Art in Ontology Design,” AI Magazine, pp. 53-74, Fall 1997.

[15] N. F. Noy and D. L. McGuinness, “Ontology Development 101: A guide to Creating Your First Ontology,” Technical Report KSL-01-05, Stanford Medical Informatics, Stanford University, 2001.

[16] T. K. Landauer, P. W. Foltz, and D. Laham, “An introduction to Latent Semantic Analysis,” Discourse Processes, vol. 25, pp. 259-284, 1998.

[17] Xuan-Hieu Phan, “FlexCRFs: Flexible Conditional Random Fields,” http://flexcrfs.sourceforge.net/, 2005.

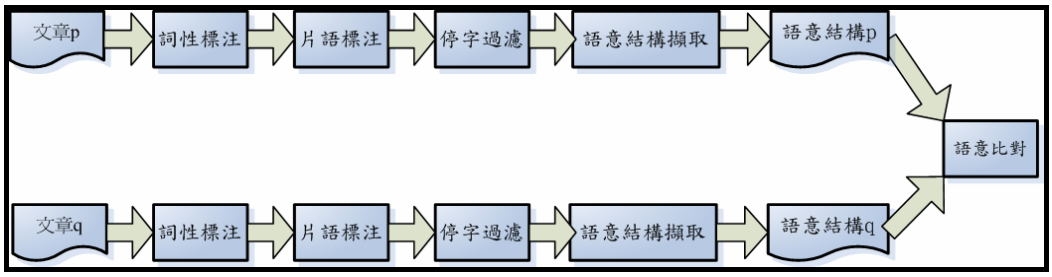

圖 1. 演算法流程圖

TAMPA, Fla. — With the injury to probable Opening Day pitcher Chien-Ming Wang — the right-hander will begin the season on the disabled list after pulling his right hamstring while running conditioning drills on friday — the starter for the Yankees first game against Tampa Bay, April 4, becomes less clear.

圖 2. 範例文章 a

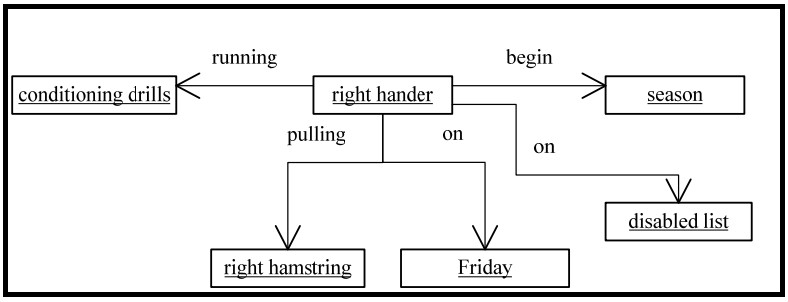

圖 3. 語意結構 a

Chien-Ming Wang will begin the season on the disabled list after pulling his right hamstring on Friday. Wang, who was a leading candidate to start on Opening Day, will likely miss a month of action now. "You're talking late April," GM Brian Cashman said. "You don't want stuff like that to happen, but unfortunately it does happen. When they happen, you handle it." Jeff Karstens will likely act as the Yankees' fifth starter while Wang is out.

圖 4. 範例文章 b

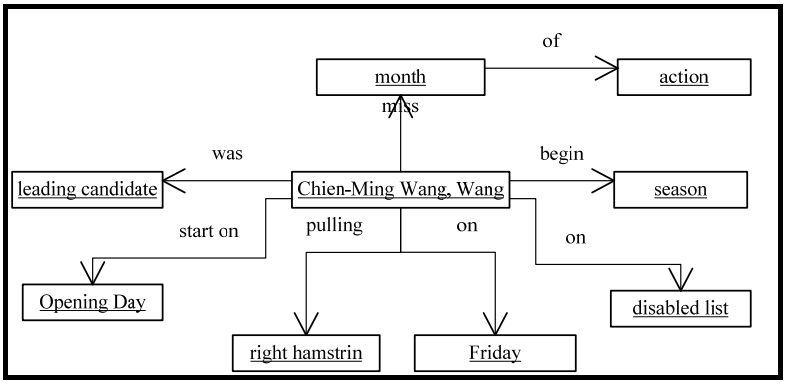

圖 5. 語意結構 b

表 1. 語意比對方法比較表

	建立輔助知識的人力成本	需額外的文章	處理單篇文章的成本	語意比對層級	輔助知識
知識基礎語意比對方法	多	無	(1)片語標注 (2)停字刪除 (3)語意相似度計算	同義詞	(1)英文句結構樹資料庫 (2)條件式隨機域模型 (3)詞彙本體論
統計基礎語意比對方法	少	有	(1)片語標注 (2)停字刪除 (3)統計詞彙出現在文章中的頻率 (4)語意相似度計算	相關詞	(1)英文句結構樹資料庫 (2)條件式隨機域模型 (3)詞彙在文章中出現的頻率
監督式語意歧異解析	多	無	(1)片語標注 (2)停字刪除 (3)人工標記文集 (4)語意歧異解析	一詞多義	(1)英文句結構樹資料庫 (2)條件式隨機域模型 (3)字典 (4)人工標記文集
非監督式語意歧異解析	中	有	(1)片語標注 (2)停字刪除 (3)統計語意出現在詞彙中的頻率 (4)語意歧異解析	一詞多義	(1)英文句結構樹資料庫 (2)條件式隨機域模型 (3)字典 (4)語意出現在相同動詞關係詞彙中的頻率
語意結構基礎語意比對方法	少	無	(1)詞性標注 (2)片語標注 (3)停字刪除 (4) 語意結構擷取 (5)語意相似度計算	同義詞 一詞多義	(1)英文句結構樹資料庫 (2)條件式隨機域模型 (3)詞彙句法樣式 (4)語意結構

The Java vacation begins at a place visited by programmer regularly. The place is the Web site of Sun, the company that developed the Java. To get there, go to http://java.sun.com. The Java division of Sun takes responsibility for the advancement of the Java and related software. Java site of Sun is the place to find the latest released versions of the Software Development Kit. This site also has press releases about Java-related products, full documentation for Java. Sun is the first place to look for each new development kit and addition to the language.

圖 6. 文章 1

The ***Java travel*** begins at a place visited by programmer regularly. The place is the ***Java home page*** of Sun, the ***business*** that developed the Java . To get there, go to http://java.sun.com. The ***Java department*** of Sun takes responsibility for the ***progress*** of the Java and related software. ***Java home page*** of Sun is the place to find the latest released versions of the Software Development Kit. This ***home page*** also has press releases about Java-related products, ***full manual*** for Java. Sun is the first ***business*** to look for each new development kit and addition to the language.

圖 7. 文章 2

The Java vacation begins at a place visited by ***drinker*** regularly. The place is the Web site of ***Starbucks***, the company that developed the Java. To get there, go to http://java. starbucks.com. The Java division of ***Starbucks*** takes responsibility for the advancement of the Java and related coffee. Java site of ***Starbucks*** is the place to find the latest released versions of coffee. This site also has press releases about Java -related products, full documentation for Java. ***Starbucks*** is the first place to look for each new ***coffee*** and addition to the ***coffee***.

圖 8. 文章 3

表 2. 文章的詞彙

	文章中的詞彙
文章 1	Java vacation, place, programmer, Web site, Sun, company, Java, Java division, responsibility, advancement, related software, Java site, latest versions, Software Development Kit, site, press releases, Java-related products, full documentation, first place, new development kit, addition, language
文章 2	Java travel, place, programmer, Java home page, Sun, business, Java, Java department, responsibility, progress, related software, Java home page, latest versions, Software Development Kit, home page, press releases, Java-related products, full manual, first business, new development kit, addition, language
文章 3	Java vacation, place, drinker, Web site, Starbucks, company, Java, Java division, responsibility, advancement, related coffee, Java site, latest versions, coffee, site, press releases, Java-related products, full documentation, first place, new coffee, addition

表 3. 文章與詞彙的相關矩陣

	文章 1	文章 2	文章 3	文章 1∩文章 2	文章 1∩文章 3	文章 1∪文章 2	文章 1∪文章 3
Java vacation	1	0	1	0	1	1	1
place	1	1	1	1	1	1	1
programmer	1	1	0	1	0	1	1
Web site	1	0	1	0	1	1	1
Sun	1	1	0	1	0	1	1
company	1	0	1	0	1	1	1
Java	1	1	0	1	0	1	1
Java division	1	0	1	0	1	1	1
responsibility	1	1	1	1	1	1	1
advancement	1	0	1	0	1	1	1
related software	1	1	1	1	1	1	1
Java site	1	0	1	0	1	1	1
latest versions	1	0	1	0	1	1	1
Software Development Kit	1	1	0	1	0	1	1
site	1	0	1	0	1	1	1
press releases	1	1	1	1	1	1	1
Java-related products	1	1	1	1	1	1	1
full documentation	1	0	1	0	1	1	1
first place	1	0	1	0	1	1	1
New development kit	1	1	0	1	0	1	1
addition	1	1	1	1	1	1	1
language	1	1	0	1	0	1	1
Java travel	0	1	0	0	0	1	0
Java home page	0	1	0	0	0	1	0
business	0	1	0	0	0	1	0
Java department	0	1	0	0	0	1	0
progress	0	1	0	0	0	1	0
home page	0	1	0	0	0	1	0
full manual	0	1	0	0	0	1	0
first business	0	1	0	0	0	1	0
drinker	0	0	1	0	0	0	1
Starbucks	0	0	1	0	0	0	1
related coffee	0	0	1	0	0	0	1
coffee	0	0	1	0	0	0	1
new coffee	0	0	1	0	0	0	1
向量長度				12	16	30	27

表 4. 文章的相似度

	相似度
文章 1 與文章 2	$12/30 = 0.40$
文章 1 與文章 3	$16/27 = 0.59$

圖9. 文章1的部份語意結構

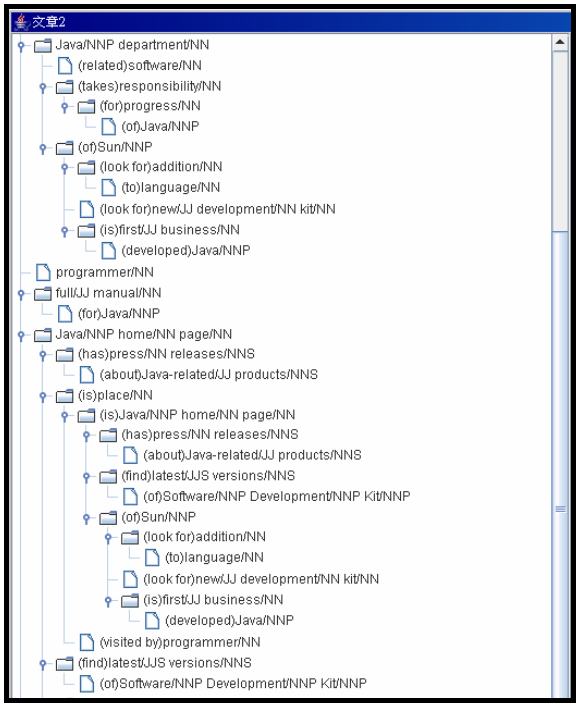

圖10. 文章2的部份語意結構

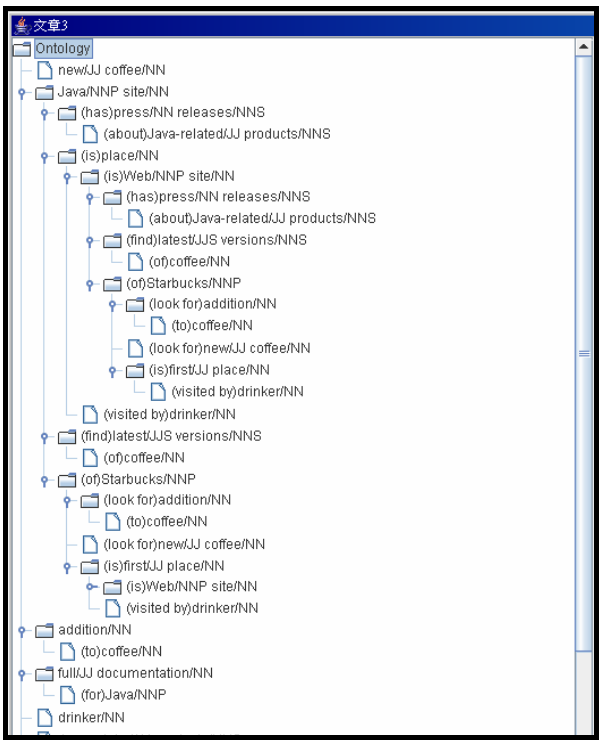

圖11. 文章3的部份語意結構

表5. 文章1,2裡詞彙語意相似度

文章 1 的詞彙	文章 2 的詞彙	語意相似度	文章 1∩文章 2(>0.6)
Java vacation	Java	0.75	1
place	place	0.83	1
programmer	programmer	1	1
Web site	home page	0.87	1
Sun	Sun	0.66	1
company	business	0.5	0
Java	Java	1	1
Java division	Java department	0.80	1
responsibility	responsibility	0.75	1
advancement	progress	0.5	0
related software			0
Java site	Java home page	0.87	1
latest versions	Latest versions	1	1
Software Development Kit	Software Development Kit	1	1
site	home page	0.87	1
press releases	press release	1	1
Java-related products	Java-related products	1	1
full documentation	full manual	0.75	1
first place	place	0.83	1
new development kit	new development kit	1	1
addition	addition	1	1
language	language	1	1
向量長度			19

表6. 文章1,3裡詞彙語意相似度

文章 1 的詞彙	文章 2 的詞彙	語意相似度	文章 1∩文章 2(>0.6)
Java vacation	Java	0.75	1
place	place	0.83	1
programmer	programmer	1	1
Web site	home page	0.87	1
Sun	Sun	0.66	1
company	business	0.5	0
Java	Java	1	1
Java division	Java department	0.80	1
responsibility	responsibility	0.75	1
advancement	progress	0.5	0
related software			0
Java site	Java home page	0.87	1
latest versions	Latest versions	1	1
Software Development Kit	Software Development Kit	1	1
site	home page	0.87	1
press releases	press release	1	1
Java-related products	Java-related products	1	1
full documentation	full manual	0.75	1
first place	place	0.83	1
new development kit	new development kit	1	1
addition	addition	1	1
language	language	1	1
向量長度			19

表 7. 文章的語意相似度

	語意相似度
文章 1 與文章 2	$19/23 = 0.82$
文章 1 與文章 3	$10/33 = 0.30$