應用跳脫語言模型於同義詞取代之研究

Skip N-gram Modeling for Near-Synonym Choice

陳士婷　Shih-Ting Chen
元智大學資訊管理學系
Department of Information Management
Yuan Ze University
s996222@mail.yzu.edu.tw

何維晟　Wei-Cheng He
元智大學資訊管理學系
Department of Information Management
Yuan Ze University
s1006250@mail.yzu.edu.tw

關松堅　Philips Kokoh Prasetyo
Living Analytics Research Centre
Singapore Management University
philipskokoh@gmail.com

禹良治　Liang-Chih Yu
元智大學資訊管理學系
Department of Information Management
Yuan Ze University
lcyu@saturn.yzu.edu.tw

摘要

同義詞(Near-Synonym)不只在自然語言應用中是重要的一環，也是對第二語言學習者很重要的部分。同義詞雖然是一群意思相近的單字集合，但在特定的情況與特殊用法下，選擇錯誤的同義詞會造成句意上的誤解，甚至是整個文法錯誤，因此我們希望能夠藉由上下文的訊息，再利用系統分辨出正確的同義詞，來協助外語學習者做有效率的學習。目前為止已有許多同義詞的相關研究，這些研究的方法包含：點式交互資訊(Pointwise Mutual Information, PMI)與 N 連詞(N-gram)模型都是常用的方法，我們想使用與以往不同的方法來提升正確率，因此我們使用跳脫(Skip N-gram)語言模型的方法針對 SemEval-2007 資料進行實驗，結果顯示我們提出的方法是可行的，正確率也有明顯的提升。

關鍵字：同義詞、點式交互資訊、跳脫語言模型

一、緒論

詞彙語意在許多自然語言應用中扮演很重要的角色。像是”arm”這個英文單字，他就有武器(weapon)和手臂(bodypart)這兩個意思可供系統來作詞義消歧的動作。此外，同義詞在自然語言中的應用非常多。例如:arm 假設他意思等同於 weapon，與他同義的詞就包含 weapon 本身和 arsenal，在這個範例 arm 這個單字就可擴張成 weapon 和 arsenal 兩個單字，藉由這樣的性質應用在資訊檢索的詞彙擴張上，可增進其應用效益[1,2]。另外，也可利用同義詞於電腦輔助語言學習(Computer-Assisted Language Learning, CALL)[3,4]。

最近有許多關於同義詞的研究，表示有些同義詞因為他們的特殊用法與搭配上的限制，所以實際運用上是不可互換的，如以下所示：

(1) ____coffee [5]

Near-Synonym：{strong, powerful}

(2) ghastly____ [6]

Near-Synonym：{error, mistake}

(3) ____ under the bay [7]

Near-Synonym：{bridge, overpass, tunnel}

在上面的(1)和(2)兩個範例都是因上下文搭配限制的範例，範例(1)中兩個同義詞 strong, powerful 都有強大、強壯的意思，但是 strong 還有濃烈的意思，因此此句意思是濃咖啡時，正確答案應該是 strong coffee，而不可使用 powerful；範例(2)中 error, mistake 都有錯誤的意思，在這個範例中英語國家的人通常都是使用 ghastly mistake 因此應選擇 mistake，；範例(3)的同義詞集{bridge, overpass, tunnel}代表一個可以穿越障礙將分離的兩個地方連接起來的物理結構。假設在”under the bay”的上下文中，原本的單字為”tunnel”。”tunnel”這個單字無法被同義詞集裡的其他同義詞取代，因為其他同義詞的語意在這裡是不合乎情理的[7]。從上面三個範例就可知道同義詞雖是意義相近，但因為用法上的限制，所以無法完全取代與互相交換。有時以英文為母語的人都不一定能正確分辨，何況是一個學習第二語言的人在分辨上更是難上加難，所以我們希望可以藉由系統分析判斷同義詞之間的差異，來幫助語言學習者使他們能學習到正確的語言知識。

學習外語者最基本的就是從單字學起，在學習過程中，同義詞是必定會遇到的難題，原因在於同義詞反映出一個詞彙通常不只有一個意思；另一個原因是同義詞意思雖然相近，但是根據習慣以及特殊用法他們會選用特定詞彙，因此我們希望藉由上下文的訊息，讓系統能夠分辨出同義詞之間的細微差異，並選擇出正確的同義詞，以幫助第二語言學習者在學習上的效率。本研究的目的就是想使用新的研究方法來分析一個特定句子，使系統能夠自動選擇出正確的同義詞，避免選擇錯誤的同義詞造成整個句子語意上

的錯誤，並讓語言學習者了解單一詞彙不只有一種意思，而是還含有其他意義，在遇到同義詞問題時可以確切明白他們之間的差異，來增加學習的效率及正確性，並且在寫作文章時也可以靈活運用同義詞，使文章更加豐富、多元。

本研究是使用跳脫語言模型(Skip N-gram)的方法對同義詞作選擇，跳脫語言模型就是將 N 連詞(N-gram)方法與跳脫式(Skip)方法做結合，N 連詞方法是利用目標詞周圍連續 N 個字詞在 Web 1T 5-gram corpus 出現的頻率去計算出 N 連詞的分數。跳脫式方法是以 N 連詞擷取出的詞組為基礎，將 N 連詞詞組中某些字詞可以為任何單字的情況下，他們在 Web 1T 5 gram corpus 中出現的次數加總，跳脫式方法是為了補強 N 連詞在 N 較大時出現頻率時常過低的缺陷，因此在 N 較大時我們使用跳脫式方法，這就是本論文提出的跳脫語言模型方法。

二、文獻探討

(一) Web 1T 5-gram

我們研究方法使用的是 Google Web 1T 5-gram corpus 做為系統消歧的語料庫，此語料庫是 Google 從 2006 年由網路上蒐集的，語料庫是由 1 到 5 連詞，以及這些連詞所出現的頻率組成，在學術上有很多研究也使用 Web 1T 5-gram 語料庫，有些學者使用 Google 語料庫來校正拼錯的英文單字[8]、有些學者利用 Google 語料庫來推斷名詞之間的語意關係[9]。表 1 為 Web 1T 5-gram 的相關資料。

表 1 Web 1T 5-gram

資料大小約 24GB	
Tokens	1,024,908,267,229
Sentences	95,119,665,584
Unigrams	13,588,391
Bigrams	314,843,401
Trigrams	977,069,902
Fourgrams	1,313,818,354
Fivegrams	1,176,470,663

(二) 詞彙選擇驗證

我們利用系統自動選擇出最適合的同義詞後，要如何驗證我們選出的同義詞是否適合上下文也是一個很重要的問題，因此有學者提出 FITB(fill-in-the-blank)任務來驗證，FITB(fill-in-the-blank)是較早的學者所研發的驗證方式，其任務內容是將句子中的目標詞去除，留下空格(gap)，將同義詞集裡的同義詞替換在空格上，然後根據各個研究學

者使用不同研究方法計算出來的分數，選出最適當的同義詞後與原文做驗證，因為此方式是從原本完整的句子將正確的目標詞去除，因此在驗證時，只要將原本的目標詞和學者們選出來的同義詞作比對，就可明顯知道各個學者選取的同義詞是否適當[10,11] ，圖 1 為 FITB 的中文和英文範例。

English Sentence: This will make the _____ message easier to interpret.
Original word: error
Near-synonym set: {error, mistake, oversight}

Chinese Sentence: 這 將 使 這 _____ 訊息 容易 解釋
Original word: 錯誤
Near-synonym set: {錯誤, 錯, 差錯, 失察, 過失}

圖 1 中文與英文的 FITB 驗證範例

(三) 同義詞研究

在自然語言處理的研究領域中，對於同義詞的研究非常多。Inkpen 早期研究是使用 PMI(Pointwise Mutual Information,點式交互資訊)方法[6]，PMI 就是比較兩個詞之間共同出現的機率，不同同義詞計算出不同的分數後，分數越高者就是研究者認為最適合的同義詞；Gardiner 和 Dras 也在同義詞研究上使用 PMI 的方式來判別[11]。另外也有人使用 N 連詞的方式來研究同義詞的詞意問題，Inkpen 也曾使用 N 連詞的方式做同義詞選擇的研究，N 連詞就是藉由目標字周圍連續 N 個字詞出現的頻率，計算出分數的高低，來選擇適當的同義詞[12]。除了 PMI 和 N 連詞方法外，WSD(Word sense disambiguation) 也是時常運用在同義詞選擇的方法，WSD 是藉由目標詞和同義詞是否為同義來判別[13]。Dagan 描述 WSD 是一個間接的方法，因為他需要有中間詞意確認的步驟，從而提出一個意義相配的技術來解決這項任務[14]。

三、研究方法

本章我們將先介紹資料前處理的部分，之後再介紹本論文研究方法所會運用到的觀念，最後在介紹本論文所提出的方法。

(一) 資料前處理

資料前處理的部分包含擷取同義詞、測試句以及測試句替換同義詞，因為我們處理的是英文語料，每個單字之間已有空白符號斷開，所以不用像中文語料必須經由斷詞程式將詞與詞之間斷開，以下介紹資料前處理的部分：

1. 擷取測試句:測試句中的原始資料檔為 XML 檔，並且是完整的句子，我們須將 XML 的標籤去除後，再從完整的句子中擷取出所有包含目標字的 5 連詞(5-gram)詞組，範例如表 2：

表 2 測試句擷取範例(句子來源：EIC)

範例:目標詞為 clean

原始資料:
<instance id="388"> <context>Grace has the money to <head>clean</head> up .</context> </instance>
結果:
has the money to clean the money to clean up money to clean up .

2. 擷取同義詞：同義詞的擷取方式是由同義詞檔中擷取出來，擷取範例如表 3：

表 3 同義詞擷取範例(句子來源：EIC)

原始資料:
clean.v 388 :: win 1;profit greatly 1;clear 1;prosper 1;accumulate 1;make a fortune 1;
結果:
同義詞集:{win, profit greatly, clear , prosper, accumulate, make a fortune}

3. 測試句替換同義詞：我們將擷取出來的測試句與同義詞作替換搭配，產生新測試句，如表 4：

表 4 測試句替換同義詞範例(句子來源：EIC)

原始測試句:(5 連詞擷取出的測試詞組有三組，我們以一組為範例做說明)
has the money to clean
替換結果:
has the money to win has the money to profit greatly has the money to clear has the money to prosper has the money to accumulate has the money to make a fortune

(二) 方法概念簡介

我們提出的方法跳脫語言模型是利用跳脫式方式來彌補 N 連詞的缺點，N 連詞主要概念就是給定一個詞，然後去預測出下一個詞，其參考的依據是利用字詞出現的頻率高低，當字詞頻率較高時，則此字詞的機率越大，但是使用 N 連詞有個缺陷，當 N 越大他的正確性越高，但是出現的頻率往往很低或是 0 的情況，因此我們使用準確性較高的 5 連詞再結合跳脫式方式來取代連詞頻率過低的情況，希望能夠因此提升準確率。本研究跳脫語言模型方法是以 Islam 和 Inkpen 提出的 5 連詞方法為基礎[12]，以下對 N 連詞和跳脫語言模型做詳細介紹。

(三) N 連詞(N-gram)

N 連詞是一種常用的語言模型，主要是將句子裡目標詞周圍 N 個字擷取出來成為詞組，依照需求不同所取的 N 也不同，取出詞組後再利用 Google Web 1T 5-grams 語料庫搜尋擷取出的詞組頻率，將頻率應用機率統計的概念算出分數，藉由得到的分數選取合適的單字，根據 N 的不同，又分為單連詞(Unigram)、2 連詞(Bigram)、3 連詞(Trigram)、4 連詞(Fourgram)、5 連詞(Fivegram)。

1. N 連詞模組建立:

句子以 $s = ...w_{i-4}w_{i-3}w_{i-2}w_{i-1}w_iw_{i+1}w_{i-2}w_{i-3}w_{i+4}...$ 表示，w_i 代表目標詞，也就是同義詞替換的位置。省略不包含目標字的 5 連詞，因為他們的值是相同的，所以只考慮 $P(w_i|w_{i-4}^{i-1})$，$P(w_{i+1}|w_{i-3}^{i})$、$P(w_{i+2}|w_{i-2}^{i+1})$、$P(w_{i+3}|w_{i-1}^{i+2})$ 與 $P(w_{i+4}|w_i^{i+3})$，以上五個項目，根據 5-gram 語言模型和平滑方式將公式定義為:

$$\begin{aligned} P(s) &= \prod_{i=0}^{5} P(w_i|w_{i-n+1}^{i-1}) \\ &= \prod_{i=0}^{5} \frac{C(w_{i-n+1}^{i}) + (1+\alpha_n)M(w_{i-n+1}^{i-1})P(w_i|w_{i-n+2}^{i-1})}{C(w_{i-n+1}^{i-1}) + \alpha_n M(w_{i-n+1}^{i-1})} \end{aligned} \quad (1)$$

其中 $M(w_{i-n+1}^{i-1})$ 為 5 連詞頻率過少的部分以平滑方法取代，公式為

$$M(w_{i-n+1}^{i-1}) = C(w_{i-n+1}^{i-1}) - \sum_{w_i} C(w_{i-n+1}^{i}) \quad (2)$$

這裡 $C(\cdot)$代表 N 連詞從 Web 1T 5-gram 語料庫中搜尋的頻率。假如較高階的 N 連詞的頻率找不到，將會往下尋找較低階的 N 連詞頻率，如果較低階頻率也找不到時，則繼續往更低階 N 連詞尋找，依此類推；相反的較高階的 N 連詞頻率找的到時，則直接採用較高階 N 連詞，就不會往下考慮低階的 N 連詞頻率。

(四) 跳脫語言模型(Skip N-gram)

跳脫式方法是將 5 連詞的詞組中，保留 N 個字詞後其餘字詞設定成可以為任意詞，然後到 Web 1T 5-gram 語料庫中重新搜尋頻率，用以替代 N 連詞頻率過低的情況，依據 N 的不同，可分為 Skip4、Skip3、Skip2，表 6-8 分別為 Skip4、Skip3、Skip2 的範例:

表 6 Skip4 範例

5 連詞詞組: has the money to clean	
Skip4	
* the money to clean	243
has * money to clean	0
has the * to clean	1099
has the money * clean	0
has the money to *	0

表 7 Skip3 範例

5 連詞詞組: has the money to clean	
Skip3	
* * money to clean	1025
* the * to clean	51774
* the money * clean	652
* the money to *	243
has * * to clean	5999
has * money * clean	0

表 8 Skip2 範例

5 連詞詞組: has the money to clean	
Skip2	
has the * * *	1435
has * money * *	0
has * * to *	6071
has * * * clean	21113
* the money * *	652
* the * to *	53074
* the * * clean	311100

表 6-8 中*代表任何單字，我們以 has the money to *為例，他代表 has the money to clean、has the money to afford、 has the money to back 、has the money to cover...等所有五連詞中前四個單詞為 has the money to 的集合，將這些詞組所有頻率相加起來就是 has the money to *的頻率。

我們為了改善 N 連詞在 N 較大時的缺陷，所以利用跳脫式方法來改善，我們將測試句使用 N 連詞方法在 N=2，3，4，5 時的頻率句數統計資料如表 9：

表 9 N 連詞頻率句數統計

	頻率為 0 的句數	正確句數
5-gram	950	370
4-gram	404	596
3-gram	352	591
2-gram	158	329
總句數:1703		

根據表在 N=5、N=4 和 N=3 的情況下，他無法找到頻率的句數較多，N=5 雖然無法找到的句數最多，但是的它的正確率卻是相對最高的，因此我們保留 N=5 的部分，而將 N=4 與 N=3 時改用 Skip4 與 Skip3 來代替。

四、實驗與結果分析

我們的實驗資料是引用 SemEval-2007 所提供的資料，SemEval-2007 是第四屆國際語意評測研討會，他提供許多詞義消歧的任務，主要是為了提升我們對同義詞與一詞多義的現象更加了解。我們參與的是 SemEval-2007 第 10 項任務[15]，他提供任務所需的實驗資料，讓參與任務團隊針對相同的資料進行實驗，並訂定統一的評分方式，以下我們簡單介紹任務提供的實驗相關資料與評分方式，如需要更詳細的資料可參考 McCarthy 和 Navigli 發表的論文[15]。

(一) 實驗資料-資料來源

實驗資料來源是由 Sharoff 的 English Internet Corpus(EIC)所取得的，此語料庫是 Sharoff 撰寫的一支在網路上抓取語料的語料庫系統。實驗資料包含 201 個單字，單字詞性分別有名詞、動詞、形容詞、副詞，而每一個單字再挑選 10 個句子，總共 2010 句。在 2010 句中將其中 1710 句當作實驗的測試句，再扣除 7 句同義詞集為 0 的部分，實際測試資料為 1703 筆。表 10 為實驗資料整理表。

表 10 實驗資料資訊(資料來源:SemEval-2007[15])

PoS	#
Noun	497
Verb	440
Adjective	468
Adverb	298
All	1703

(二) 實驗資料-同義詞集

SemEval-2007 在這項任務[15]找來以英文為母語的 5 個人，針對測試的 1710 筆資料，在不限時間的情況下，填上每個人認為適合的同義詞，每個人不限定只能填一個，只要

認為適合都可填寫，因此可提出三個以上的同義詞，表 11 為同義詞集資訊。範例: If this Government had been doing its **job** they would have total confidence.

表 11 同義詞集資訊(資料來源:SemEval-2007[15])

參與者	1	2	3	4	5
替代詞	duty function	bit	responsibility	duty task	role
job.n 433 :: duty 2;function 1;bit 1;responsibility 1;task 1;role 1;					

(三) 評分方法

此任務[15]的評分方式是將系統依據研究方法提出一個最佳的同義詞後，利用兩個計算方法來評分，一個是召回率(Recall)，一個是最多頻率召回率(Mode Recall)，兩者之間的主要差別在於召回率(Recall)將系統提出的所有同義詞根據 5 位註解人所註解的次數算出分數，而最多頻率召回率(Mode Recall)只考慮擁有最高註解次數的同義詞與系統所選擇的最佳同義詞是否相同來計算分數，以下介紹兩種評分公式，表 12 為召回率(Recall)的變數資料。

$$R = \frac{\sum_{a_i : i \in T} \frac{\sum_{res \in a_i} freq_{res}}{|a_i| \cdot |H_i|}}{|T|} \tag{3}$$

表 12 Recall 變數資料

變數	代表意義
T	至少有兩個同義詞的測試句個數
H_i	同義詞集裡的同義詞次數加總
$freq_{res}$	最佳同義詞於同義詞集裡的次數

$$Mode\ R = \frac{\sum_{bg_i \in T_m} 1 \text{ if } bg_i = m_i}{|T_m|} \tag{4}$$

表 13 Mode Recall 變數資料

變數	代表意義
T_m	同義詞集擁有最多次數的測試句個數
bg_i	最佳的同義詞
m_i	註解次數最多的同義詞

(四) 實驗結果與分析

1. N-gram 與 Skip 組合分析

我們決定 N-gram 與 Skip 的結合方式是先將所有可能的組合全部列出來實作後，將組合的結果與純 N-gram 方法相比選出結果最佳的組合，結果如表 14:

表 14 N-gram+Skip 結果數據

	Recall	Mode Recall
N-gram	30.31	39.84
N5S4N3N2N1	31.55	39.84
N5N4S3N2N1	31.54	39.67
N5N4N3S2N1	30.97	37.8
N5S4S3N2N1	**31.6**	**40.24**
N5N4S3S2N1	30.9	37.48
N5S4N3S2N1	31.08	38.13
N5S4S3S2N1	30.88	36.99

由表 14 可看出在所有 N-gram 與 Skip 組合中結果最好的是 N5S4S3N2N1，N5S4S3N2N1 的意思代表這是 5-gram、Skip4、Skip3、2-gram、ungram 的組合，將 N5S4S3N2N1 組合與純 N-gram 相比，有加入 Skip 的方法比純 N-gram 的結果好，而 Skip 與 N-gram 結合時，Skip 在 N=4 與 N=3 位置計算出來的數據最好，因此我們跳脫語言模型的組合是使用 N5S4S3N2N1 的結合方式。

2. 正確率(Accuracy)

$$Accuracy = \frac{\sum_{bg_i \in T_m} 1 \text{ if } bg_i = \text{original word}}{\text{All}} \quad (6)$$

其中 bg_i 代表我們提出的最佳同義詞，originalword 代表測試句中原本的目標字，All 代表測試句的總句數，根據以上計算方式計算出來的結果如下表 15：

表 15 正確率結果數據

System	Accuracy
N-gram	30.30%
N5S4N3N2N1	34.66%
N5N4S3N2N1	**39.16%**
N5N4N3S2N1	32.11%
N5S4S3N2N1	38.63%
N5N4S3S2N1	34.42%
N5S4N3S2N1	33.71%
N5S4S3S2N1	33.83%

表 15 中以 N5N4S3N2N1 的結果最好 39.16%，我們是和參與 SemEval-2007 任務的團隊比較，因此雖然 N5N4S3N2N1 在正確率這裡是最好的，但我們還是選擇在 SemEval-2007 任務評分標準中最好的結合結果 N5S4S3N2N1 作為我們跳脫語言模型的結合方式。我們將 N-gram 與 Skip 的所有組合與純 N-gram 相比之下，N-gram 結合 Skip 的數據都比純 N-gram 來的好，因此可以證明我們提出的跳脫語言模型確實可以改善 N-gram。

3. 結果範例討論

此節我們將利用結果範例來討論跳脫語言模型是否真能夠改善 N 連詞，表 16 與表 17 為我們自行使用的 N 連詞結果與跳脫語言模型的結果比較:

表 16 N-gram 與 Skip N-gram 比較結果範例一

測試句: I ____ over and made a U turn while Chris got out, ran over and took a picture. 原始目標字: pull				
同義詞	pull		stop	
N-gram	5-gram		5-gram	
	pull over and made a	0	stop over and made a	0
	4-gram		4-gram	
	pull over and made	0	stop over and made	0
	over and made a	0	over and made a	0
Skip	Skip4		Skip4	
	pull * and made a	0	stop * and made a	0
	pull over * made a	0	stop over * made a	0
	pull over and * a	1172	stop over and * a	171
	pull over and made *	0	stop over and made *	0

表 17 N-gram 與 Skip N-gram 比較結果範例二

測試句: Java so that all of the clone( ) methods catch the CloneNotSupportedException rather than ____ it to the caller. 原始目標字: pass				
同義詞	pass		hand	
N-gram	5-gram		5-gram	
	than pass it to the	0	than hand it to the	50
Skip	Skip 4		Skip 4	
	than pass * to the	0	than hand * to the	50
	than pass it * the	157	than hand it * the	50
	than pass it to *	0	than hand it to *	50

表 16 與表 17 因為資料過多，我們無法列出全部數據，所以僅列出代表性的數據，表 16 為 5 連詞與 4 連詞頻率為 0 的情況，範例一中使用 N 連詞方法選出的最佳同義詞為 stop，跳脫語言模型最佳同義詞為 pull，原因在於使用 N-gram 方法時 pull 和 stop 在 5-gram 與 4-gram 頻率都為 0 只能往下找 3-gram、2-gram 與 ungram，在 4-gram 以下的低階連詞 stop 的頻率高於 pull 因此 N-gram 的最佳同義詞就選擇 stop；跳脫語言模型方面，在 Skip4 時就可明顯看出 pull 的頻率高出 stop 許多，因此跳脫語言模型的最佳同義詞為 pull。表 17 為兩個同義詞裡其中一個 5 連詞頻率不為 0 的情況，範例二中 N 連詞方法選出的最佳同義詞為 hand，跳脫語言模型最佳同義詞為 pass，原因在於使用 N-gram 方法時，hand 在 5 連詞的頻率為 50，pass 卻為 0，所以 N 連詞最佳同義詞為 hand，但是在 Skip4 時頻率卻是 pass 多於 hand，所以跳脫語言模型最佳同義詞選擇 pass。由上面兩個例子，我們可以證明跳脫語言模型確實比純 N 連詞的方法好。

五、結論

本篇論文使用跳脫語言模型的方法，並針對對 SemEval-2007 的第 10 項任務[15]進行實驗。我們採用跳脫語言模型的主要原因在於 N 連詞的方法準確性很高，但是缺點就是當 N 越大的時候，往往在語料庫中的出現頻率都是非常低，甚至是 0 的情況，因此我們希望使用跳脫的方法來取代 N 連詞的缺點，我們將 N 較大時，以跳脫方法的頻率來取代 N 連詞的頻率，使正確性能提高，而我們將跳脫語言模型方法參與 SemEval-2007 做實驗的結果以及分析過後，我們提出的方法確實能夠提升同義詞選擇的正確率，和其他團隊以 N 連詞方式進行實驗的結果相比，也明顯提升，因此我們提出的跳脫語言模型確實能夠彌補 N 連詞的缺點。未來工作將進一步找出正確率過低的原因，修改方法的計算方式或是找出新方法，促使正確率能夠再提升，讓同義詞之間的差異可以更加明確。

誌謝

本研究感謝林志誠同學在先期工作上所做的努力。此外，本研究承蒙國科會 NSC 99-2221-E-155 -036 -MY3 費補助特此致謝。

參考文獻

[1] D. Moldovan and R. Mihalcea, "Using WordNet and Lexical Operators to Improve Internet Searches," *IEEE Internet Computing*, pp. 34-43, 2000.

[2] J. Bhogal, A. Macfarlane, and P. Smith, "A Review of Ontology based Query Expansion," *Information Processing & Management*, pp. 866-886, 2007.

[3] C. Cheng, "Word-Focused Extensive Reading with Guidance," In *Proc. of the 13th International Symposium on English Teaching*, pp. 24-32, 2004.

[4] S. Ouyang, H. H. Gao, and S. N. Koh, "Developing a Computer-Facilitated Tool for Acquiring Near-Synonyms in Chinese and English," In *Proc. of IWCS-09*, pp. 316-319, 2009.

[5] D. Pearce, "Synonymy in Collocation Extraction," In *Proc. of the Workshop on WordNet and Other Lexical Resources at NAACL-01*, 2001

[6] D. Inkpen, "A Statistical Model of Near Synonym Choice," *ACM Trans. Speech and Language Processing*, pp. 1-17, 2007.

[7] L. C. Yu, C. H. Wu, R. Y. Chang, C. H. Liu, and E. H. Hovy, "Annotation and verification of sense pools in OntoNotes," *Information Processing & Management 46(4)*, pp.436-447, 2010.

[8] A. Islam, D. Inkpen, "Real-word spelling correction using Google Web 1T 3-gram, " In *Proc. of EMNLP-09*, pp. 1241-1249, 2009.

[9] P. Nulty, F. Costello, "Using lexical patters in google Web 1T corpus to deduce semantic relation between nouns," In *Proc. of the NAACL/HLT Workshop on Semantic Evaluations: Recent Achievements and Future Directions,* pp. 58-63, 2009.

[10] P. Edmonds, "Choosing the Word Most Typical in Context Using a Lexical Co-occurrence Net Network," In *Proc. of ACL-97*, pp. 507-509, 1997.

[11] M. Gardiner and M. Dras, "Exploring Approaches to Discriminating among Near-Synonyms," In *Proc. of the Australasian Technology Workshop*, pp. 31-39*,* 2007.

[12] A. Islam, D. Inkpen, "Near-Synonym Choice using a 5-gram Language Model," In *proc*. *Research in Computing Science*, pp. 41-52, 2010.

[13] D. McCarthy, "Lexical Substitution as a Task for WSD Evaluation," In *Proc. of the SIGLEX/SENSEVAL Workshop on Word Sense Disambiguation at ACL-02*, pp. 109-115, 2002.

[14] I. Dagan, O. Glickman, A. Gliozzo, E. Marmorshtein, and C. Strapparava, "Direct Word Sense Matching for Lexical Substitution," In *Proc. of COLING/ACL-06*, pp. 449-456, 2006.

[15] D. McCarthy, R. Navigli, "The English lexical substitution task," In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, pp. 48-53, 2009.