英文介系詞片語定位與英文介系詞推薦
Attachment of English Prepositional Phrases and Suggestions of English Prepositions

蔡家琦　　劉昭麟
Chia-Chi Tsai　　Chao-Lin Liu

國立政治大學資訊科學系
National Chengchi University, Taipei, Taiwan
{g9906, chaolin}@cs.nccu.edu.tw

摘要

本研究專注於介系詞相關的二個議題：介系詞片語定位與介系詞推薦。我們將這二個議題抽象化為一個決策問題，並提出一個一般化的解決方法。這二個問題共通的部分在於動詞片語；一個簡單的動詞片語含有最重要的四個中心詞（headword）：動詞、名詞一、介系詞和名詞二。由這四個中心詞做為出發點，透過 WordNet 做階層式的選擇，在大量的案例中尋找語義上共通的部分，再利用機器學習的方法建構一般化的模型。此外，針對介系詞片語定位問題，我們挑選實驗具挑戰性的介系詞做實驗。藉由使用真實生活語料，我們的方法處理介系詞片語定位的問題，可以有不錯的表現；而對於介系詞推薦的問題，我們的方法難有全面比較的對象，但精準度可達到 47.76%。本研究發現，高層次的語義可以使分類器有不錯的分類效果，但透過階層式的語義選擇能使分類效果更佳。這顯示我們確實可以透過語義歸納一套準則，用於二個介系詞的議題。相信成果在未來會對機器翻譯與文本校對的相關研究有所價值。

關鍵字：語義分析、機器翻譯、文本校對

Abstract

This paper focuses on problems of attachment of prepositional phrases (PPs) and problems of prepositional suggestions. We transform the problems of PPs attachment and prepositional suggestions into an abstract model, and apply the same computational procedures to solve these two problems. The common model features four headwords, i.e., the verb, the first noun, the preposition, and the second noun in the prepositional phrases. Our methods consider the semantic features of the headwords in WordNet to train classification models, and apply the learned models for tackling the attachment and suggestion problems. This exploration of PP attachment problems is special in that only those PPs that are almost equally possible to attach to the verb and the first noun were used in the study. The proposed models consider only four headwords to achieve satisfactory performances. This study reconfirms that semantic information is instrument for both PP attachment and prepositional suggestions.

keyword : semantic analysis, machine translation, text proofreading

1　緒論

英文介系詞在句子裡所扮演的角色通常是用來使介系詞片語更精確的補述上下文，英文介系詞的使用對於英文母語的使用者而言是很直覺，即使英文母語的使用者不知道文法結構，仍然可以精確地表達語義。但對於電腦而言卻很難知道語義，因此不容

易判斷正確的修飾對象。對於非英文母語的使用者，自然且正確地表達是有困難的。在現今資訊科技盛行爆炸的時代，我們期望透過大量資料以及資訊技術來輔助人類解決問題，並將我們研究應用於電腦自動化的流程。

介系詞一般出現在動詞片語裡，由動詞片語結構可以衍生出來的兩個有趣問題，也就是**介系詞片語定位**與**介系詞推薦**。

介系詞片語定位的問題是解決介系詞片語修飾對象是動詞或名詞片語，用一個具體的例子做說明，以句 1為例子。在我們主觀的認知中比較容易聯想的語境是：這群小孩用湯匙吃蛋糕。根據剛才想像的語境“with a spoon”這個介系詞片語修飾的應該是“ate”這個動詞。但是其實句 1也可以有另外一種想像的空間是這些小孩吃得是旁邊有放湯匙的蛋糕，這時“with a spoon”修飾的對象就是“the cake”這個名詞片語。

句 1. The children ate the cake with a spoon.[1]

介系詞推薦的問題是當動詞片語缺少介系詞時，應該要推薦何種適當的介系詞。對於非英文母語的使用者來說缺少了可以自然使用介系詞的直覺，只能透過介系詞的功能面決定它的用法。有些介系詞因為在功能面是類似的，例如 in、on、at 在時間的用途上是經常被混淆的。非英文母語的使用者只能透過大略的準則判斷介系詞的使用。

因為介系詞的使用是如此的廣泛，但是讓電腦瞭解語義和非英文母語的使用者來說都是有相當的門檻，所以我們對於介系詞的議題感到有興趣。為了可以更精確地使用介系詞，我們將深入探討這二種介系詞的議題。如果能夠解決這些議題，就可以將此應用做為機器翻譯基石和文本校對用途。

我們的研究嘗試找出上下文無關（context-free）的解決方案。這二個問題共通的部份是動詞片語，其結構是「動詞 -名詞片語一 -介系詞 -名詞片語二」的結構，簡化為四個中心詞「動詞 -名詞一 -介系詞 -名詞二（*V-N1-P-N2*）」。中心詞的定義為詞組中最核心被修飾的詞。我們直接探討動詞片語所抽出的四個主要中心詞並以此做為研究的出發點。再利用 WordNet 階層式的概念將中心詞提升到較抽象的語義層級，也就是找出上位詞，並利用資訊技術從大量的語料中找尋是否有一套準則能定位介系詞片語和推薦正確的介系詞。

在本研究，我們將介系詞片語定位問題與介系詞推薦問題分別做了一些假設與簡化。介系詞片語定位問題，在現實生活中，可能有的答案包含：修飾動詞、修飾名詞、二者皆可或其它。我們簡化為只有修飾動詞與修飾名詞二種可能。介系詞推薦的問題，在只提供動詞、名詞一和名詞二的資訊下，答案可能不只一個介系詞。因此我們將問題簡化成只有一個答案，只處理只有一個答案的案例。另外，只挑選數量較多的介系詞做實驗。

介系詞片語定位的問題依上述的假設是一個二分類的問題，介系詞推薦的問題則是一個多分類問題，所以，顯然地，我們可以看出推薦問題可能比定位問題的難度要高。

另外，針對介系詞片語定位的問題，許多學者大多希望能夠對所有介系詞找出一套一般化的通則。然而我們從 Ratnaparkhi 等人 [12] 所彙整的中心詞語料庫，也就是 RRR 語料庫[2]，統計各個介系詞數量分布的情況，結果如表 1所示，其中 NPP 為修飾名詞的介系詞片語，而 VPP 為修飾動詞的介系詞片語。可以發現每一個介系詞的定位情況都不相同，因此在我們的研究會針對各個介系詞歸納適用的準則。

表 1: RRR 語料庫，NPP 與 VPP 的數量

介系詞	NPP	VPP	總數	介系詞	NPP	VPP	總數	介系詞	NPP	VPP	總數
about	187	86	273	for	1342	1310	2652	of	6553	61	6614
as	123	497	620	from	360	716	1076	on	736	826	1562
at	166	594	760	in	1999	2061	4060	to	566	1486	2052
by	151	326	477	like	30	21	51	with	397	739	1136

研究成果裡，在介系詞片語定的問題中，本研究的效果比同樣考慮四個中心詞的最

[1] 出自 Chris Manning 和 Hinrich Schütze，Foundations of statistical natural language prochessing 書中 8.3 節

[2] https://sites.google.com/site/adwaitratnaparkhi/publications/ppa.tar.gz?attredirects=0&d=1

大熵值法（Max Entropy）好，但與考慮上下文的 Stanford 剖析器結果是差不多。而在介系詞推薦的問題裡，我們的方法比起於目前方法的成果是有一小段差距。

介系詞的相關研究議題，一直是許多學者努力研究的目標，Baldwin 等人 [2] 於 2009 年時，回顧近十年各式各樣介系詞相關的議題，其中包含了本研究有興趣的二個介系詞議題。

從早期開始，有不少學者採用機率統計的方式試圖解決介系詞片語定位問題（如 Hindle 和 Rooth [8]、Liu 等人 [9] 和 Ratnaparkhi 等人）。經常使用得基本特徵資訊包含動詞片語的四個中心詞：動詞、名詞一、介系詞和名詞二。透過四個中心詞，再經由機率統計模型計算介系詞片語可能的定位。然而對假設已知四個中心詞，Atterer 和 Schütze [1] 指出這個假設不是憑空而來。但本研究依舊假設中心詞是已知條件，將中心詞的取得視為前處理的一部分，我們抽取中心詞的研究則是依靠 Stanford 剖析器[3]，而 Stanford 剖析則是建立在 Collins [3] 的研究之上。

在不少文獻中，可以看到每個介系詞均有自己的特色，如 Stetina 和 Nagao [13] 試圖為每一個介系詞製作合適的分類器。且大部分的介系詞更是有慣用方式，可參考表 1，例如介系詞“of”大多數的時候都是定位名詞。因此 Coppola 等人將語料中有“of”的案例去除。

推薦問題與定位問題的歷史相比較，是屬於比較年輕的問題。目前許多研究大多視為是語文學習應用，且視為是「介系詞校正」的問題，較早的相關研究有 De Felice 和 Pulman [5]、Gamon 等人 [7] 和 Tetreault 和 Chodorow [14]。

校正嚴格來說可以分成二階段：第一個階段是偵錯，第二個階段是更正。De Felice 和 Pulman [6]、Gamon 等人和 Helping Our Own 2012 Shared Task（Wu 等人 [16] 和 Quan 等人 [11]）都是二個階段皆著重。De Felice 和 Pulman [5] 是著重於後者。本研究也是著重於後者，廣義來說，我們將推薦視為是一種校正，但因為本研究不包含偵錯，所以我們強調是介系詞推薦。

De Felice 和 Pulman [5] 的研究與本研究的介系詞推薦是較相近，同樣是使用文法正確的語料庫訓練模型，且將實驗限縮在常用的介系詞。

2 語料介紹

2.1 語料庫

本研究使用 Peen Treebank 3（以下簡稱 PTB3）與 RRR 來作為介系詞片語定位問題的語料庫，而用自行蒐集的報導資料來當作介系詞推薦的語料庫。

RRR RRR 語料庫是由 Ratnaparkhi 等人 [12] 由 PTB0.5 匯整而成。其中每一筆資料都紀錄 PTB0.5 動詞片語中的四個中心詞與定位標記，我們將這種紀錄方式稱之為 RRR 格式。

PTB3 PTB3 是一個將自然語言結構化的資料庫，在許多自然語言處理的研究都被視為是黃金標準。本研究使用的版本是現行版本第三版，其中內容包含了三年份華爾街日報共 2499 篇報導，共有 98732 句結構化的句子，並且將這些句子分成 25 節。

華爾街日報與紐約時報 我們從華爾街日報[4]與紐約時報[5]的網站上蒐集了 2011 年部分報導內容，其中包含了華爾街日報的 68983 句和紐約時報的 55358 句。內容屬性上，華爾街日報是屬於財經類報導，而紐約時報是屬於綜合類的報導。這二類報導文章的句型句法都是屬於較現代的用法。

2.2 前處理

前處理的部分包含了句子的斷句與剖析、中心詞抽取、雜訊過濾以及挑選有挑戰性的介系詞等工作。流程圖可參考圖 1，圖中上半部是前處理的流程，下半部表示的是語料庫進入前處理的階段。使用華爾街日報與紐約時報需要從斷句與剖析句子的流程開始處理；使用 PTB3 語料庫，則是從結構樹中抽取中心詞的流程開始處理；使用 RRR 語料庫直接從雜訊過濾開始處理。最後所有語料彙整成 RRR 的資料格式，再統一處理

[3] Stanford Parser 2.0 版（2012 年 2 月 3 日），http://nlp.stanford.edu/software/lex-parser.shtml
[4] http://asia.wsj.com/home-page
[5] http://www.nytimes.com/

雜訊。雜訊過濾是一件重要的工作，雜訊包含了中心詞是定冠詞、代名詞等情況或是碰撞問題等情況。對於介系詞片語定位問題，挑選挑戰性介系詞是找出修飾動詞與修飾名詞機率相近的介系詞。每個語料庫介系詞分布情況大致上差不多，但仍有些許差異，因此我們以 RRR 語料庫為主。對於介系詞推薦的問題，則是找到數量較多或是差不多的介系詞。

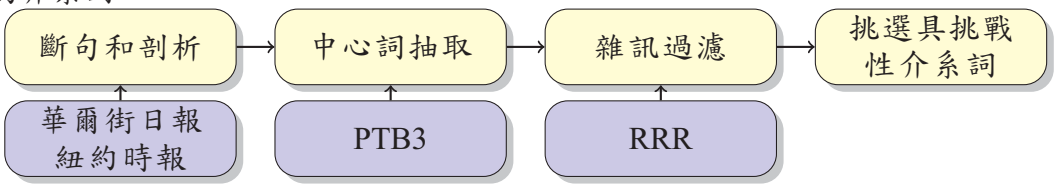

圖 1: 前處理流程圖

2.2.1 句子剖析與斷句

我們先利用 Stanford 剖析器與 Lingpipe[6]將所蒐集的語料斷句，接下來僅留下二者斷句有共識的句子。接著再利用 Stanford 剖析器剖析留下的句子，剖析後可得到結構樹。

2.2.2 中心詞抽取

當語料是結構樹時，才會需要中心詞抽取。我們的目標是從結構樹中比對修飾動詞或名詞的介系詞片語，如圖 2和圖 3分別表示修飾名詞與修飾動詞的介系詞片語結構，我們採用 Penn Treebank 的風格表示。這二個結構最大的不同點在於 PP 這個節點是掛在 NP 或是 VP 之下。二個圖中 VP 下方最左邊的節點表示是不同形態的動詞，如過去式、過去分詞等；IN 表示的是介系詞，“to” 這個介系詞會另外被表示成 TO。我們使用 Stanford Tregrex[7]比對圖 2和圖 3的樣式。

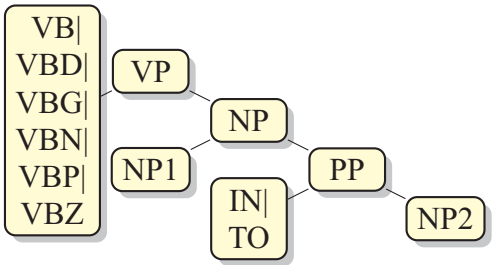

圖 2: 動詞片語: 修飾名詞

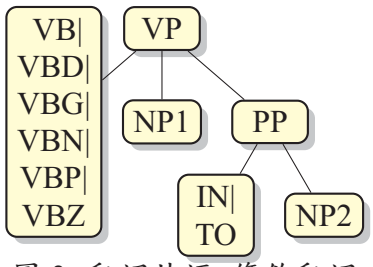

圖 3: 動詞片語: 修飾動詞

以句 2為例子，底線是我們要抽取的目標，它符合圖 3結構。比對出四個詞組如表 2片語一欄所示。最後利用 Stanford 剖析器的 SemanticHeadFinder[8]類別將的四個主要詞組的中心詞找出，得到的結果如表 2中心詞一欄所示。

句 2. ((S (NP-SBJ (DT The) (JJ Venezuelan) (JJ central) (NN bank))
(VP (VBD set) (NP (PP (NP (DT a)(ADJP (CD 30) (NN %))(NN floor))
(IN on)(NP (DT the) (NN bidding)))))(. .)))

表 2: 中心詞抽取

	片語	中心詞
動詞	(VBD set)	set
名詞片語一	(NP (NP (DT a) (ADJP (CD 30) (NN %)) (NN floor)))	floor
介系詞	(IN for)	for
名詞片語二	(NP (DT the) (NN bidding))	bidding

2.2.3 雜訊過濾

在我們的語料庫裡，也有不少句子因為語法上的關係，可能會有一些特殊的符號和數字被當成是中心詞，例如：“%”。然而這些符號和數字在我們的方法中是很難抽象化的，因此我們會事先將這些符號和數字過濾。

[6]http://alias-i.com/lingpipe/
[7]Stanford Tregrex 2.0.1 版（2012 年 1 月 6 日），http://nlp.stanford.edu/software/tregex.shtml
[8]http://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/trees/SemanticHeadFinder.html

除了上述情況之外，我們也發現雖然 RRR 的語料庫經由 Ratnaparkhi 等人整理過，但 Pantel 和 Lin [10] 在 RRR 語料庫裡找到 133 筆名詞一或名詞二為“the”，PTB3 裡也有出現“the”的被當成是名詞的案例。另外在 RRR 與 PTB3 語料庫也均有一些名詞是“a”或“an”的情況。類似的情況，我們亦將之視為雜訊。

此外，我們會先利用 WordNet 做詞幹還原。接著，再給定還原後的詞彙和詞性，如果 WordNet 沒有查詢任何同義詞集（synset），那麼也會被過濾。

Coppola 等人 [4] 曾提到，如果名詞一是代名詞，則介系詞片語有較高的機率是定位於動詞。另一方面，代名詞不被收入於 WordNet 內，因此當名詞是代名詞的情況在我們的二個研究問題中也會過濾。

對於介系詞定位問題，碰撞是指當有二個以上的動詞片語具有四個相同的中心詞但介系詞定位卻不相同的情況。對於介系詞推薦問題，碰撞是指當動詞、名詞一和名詞二相同，但介系詞有二個以上的情況。上述這二類的案例，目前在本研究中暫不處理，因此也將之視為雜訊。

2.2.4 挑選具挑戰性的介系詞

在介系詞片語定位的問題中，我們將為每一個介系詞特製化分類器，並挑選修飾名詞與修飾動詞數量平衡的具挑戰性介系詞。這可以幫助我們去除掉幾乎有習慣用法的介系詞“of”，使我們專注於幾個較難分類的介系詞。挑選具挑戰性介系詞時，因為後續使用機器學習演算法建構模型，所以不希望數量太少。因此我們採用 $Entropy$ 如式 (1) 和頻率這二個指標挑選介系詞，我們的目的是找平衡且數量多的介系詞。

$$Entropy = \sum_{d \in D} -Pr(d) \log_2 Pr(d) \quad (1)$$

式 (1) 中，以“of”為例，介系詞片語定位問題為 D，且只有二個分類因此 $D = \{VPP, NPP\}$，$Pr(d)$ 為修飾名詞與修飾動詞所佔的比例。因此，我們可以知道 $Pr(NPP) = 6553/6614$、$Pr(VPP) = 16/6614$，最後可以計算出 $Entropy$。$Entropy$ 數值越大表示偏好二個類別的數量越平衡越具挑戰性。

在介系詞推薦的問題中，我們會從語料庫挑選數量較多的介系詞。

2.3 目的語料

目的語料是我們經由 2.2節前處理的方法得到的結果。介系詞片語定位問題的語料，我們將以 RRR 所選到的介系詞為主，實際上，在我們的統計中，每個語料庫的介系詞分布幾乎都是差不多的。介系詞推薦問題則是依個別介系詞數量多寡不同而選擇的介系詞。對這二個問題我們設定了一個分布線（Distribution），分布線的意義是亂猜可以達到最佳的精準度。

介系詞片語定位問題，根據的介系詞數量分布的情況，分布線定義，如式 (2)。式 (2) 的 $Pr(VPP)$ 與 $Pr(NPP)$ 表示修飾動詞與修飾名詞在語料庫裡佔得總量的比例。

$$Distribution = Max(Pr(V), Pr(N)) \quad (2)$$

介系詞推薦問題，我們比較著重於分析各個介系詞分類的情況，因此設定以介系詞在語料庫佔得總量計算分布線，如式 (3)，其中 $|x|$ 表示 x 出現的頻率。而總體的分布線，則是以單獨分布線較高者為準。

$$Distribution = |Preposition|/|Total| \quad (3)$$

2.3.1 介系詞片語定位語料

表 3、表 4和表 5是 RRR 語料經由篩選過慮後的結果，我們選出“for”、“on”、“in”、“with”、“from”和“to”，其中前三個介系詞都是數量多且較平衡的情況，而後三者則是數量多但較不平衡的情況，混合表示是將這六個介系詞一起做實驗。訓練語料、驗證語料和測試語料的分布大致上都是差不多。

表 6、表 7和表 8是 PTB3 過濾後的結果，分布的情況與 RRR 大致是相同的。為與 Stanford 剖析器做比較，我們選用 PTB3 的 02 到 21 節做測試語料，22 節做驗證語料，00、01、23 和 24 節做測試語料。

表 3: RRR 前處理結果: 訓練語料

介系詞	v	n	Entropy	分布線
for	829	869	0.9996	51.18%
on	512	485	0.9995	51.35%
in	1392	1314	0.9994	51.44%
with	454	268	0.9516	62.88%
from	451	237	0.9290	65.55%
to	1145	394	0.8207	74.40%
混合	4783	3567	0.9846	57.28%

表 4: RRR 前處理結果: 驗証語料

介系詞	v	n	Entropy	分布線
for	147	169	0.9965	53.48%
on	120	100	0.9940	54.55%
in	272	289	0.9993	51.52%
with	73	47	0.9659	60.83%
from	74	52	0.9779	58.73%
to	190	82	0.8831	69.85%
混合	876	739	0.9948	54.24%

表 5: RRR 前處理結果: 測試語料

介系詞	v	n	Entropy	分布線
for	111	148	0.9852	57.14%
on	66	93	0.9791	58.49%
in	156	200	0.9890	56.18%
with	55	35	0.9641	61.11%
from	60	32	0.9321	65.22%
to	135	76	0.9428	63.98%
混合	583	584	1.0000	50.04%

表 6: PTB3 前處理結果: 訓練語料

介系詞	v	n	Entropy	分布線
for	732	892	0.9930	54.93%
on	512	523	0.9999	50.53%
in	1531	1241	0.9921	55.23%
with	450	269	0.9538	62.59%
from	441	290	0.9690	60.33%
to	1064	335	0.7941	76.05%
混合	4730	3550	0.9853	57.13%

表 7: PTB3 前處理結果: 驗証語料

介系詞	v	n	Entropy	分布線
for	23	39	0.9514	62.90%
on	30	22	0.9829	57.69%
in	72	61	0.9951	54.14%
with	10	5	0.9183	66.67%
from	14	10	0.9799	58.33%
to	34	16	0.9044	68.00%
混合	183	153	0.9942	54.46%

表 8: PTB3 前處理結果: 測試語料

介系詞	v	n	Entropy	分布線
for	115	189	0.9568	62.17%
on	104	101	0.9998	50.73%
in	288	289	0.1000	50.09%
with	74	51	0.9754	59.20%
from	77	46	0.9537	62.60%
to	182	77	0.8780	70.27%
combine	840	753	0.9978	52.73%

表 9是表 8測試語料的原句總句數，我們會將這些原句讓 Stanford 剖析器剖器，再比對介系詞定位是否正確。

表 9: PTB3 前處理結果：測試語料原句句數

for	on	in	with	from	to
296	203	546	122	120	232

2.3.2 介系詞推薦語料

表 10是數量較大的語料庫，我們將處理後的語料切成訓練、測試資料，並從中選出數量較多的 11 個介系詞做實驗。

表 10: 華爾街日報與紐約時報前處理結果

	訓練資料		測試資料			訓練資料		測試資料	
	數量	分布線	數量	分布線		數量	分布線	數量	分布線
of	7341	28.36%	2390	27.71%	from	1300	5.02%	413	4.79%
in	5353	20.68%	1801	20.88%	at	1109	4.28%	359	4.16%
for	2892	11.17%	916	10.62%	as	694	2.68%	239	2.77%
to	2471	9.55%	892	10.34%	by	522	2.02%	162	1.88%
on	2248	8.68%	768	8.91%	about	329	1.27%	112	1.30%
with	1625	6.28%	572	6.63%	總數	25884	28.36%	8624	27.71%

3 研究方法

經過第2節語料處理後，可以得到動詞片語的四個中心詞：動詞、名詞一、介系詞和名詞二。在我們的研究裡，我們將這四個中心詞視為是已知條件，在這樣的條件下，對介系詞片語定位與介系詞推薦問題建構一般化的模型。

3.1 特徵處理

經過 WordNet 查詢而來的同義詞集被我們視為是特徵，然而這樣的特徵只是一個符號，但在現行許多機器學習的演算法，大多都是需要量化後的數值，因此如何將特徵量化是一個重要的議題。

本節特徵的處理包含了特徵量化與特徵加權，特徵加權可視為是廣義特徵量化的過程，因為加權本身也是將特徵數值化的一個過程。而本這節特徵量化特別強調如何表現特徵的存在，我們稱之為狹義特徵量化的定義。特徵加權主要是基於狹義特徵量化再給予不同的詮釋面向。

3.1.1 特徵量化

我們對於這三種量化的方式都有不同的詮釋：在一個透過 WordNet 查詢的詞彙中，從查詢到的第一個同義詞集到根節點所有的同義詞集，二元法考慮的面向是將所有節點都視為是均等的存在；平均法考慮的是每個節點平均負擔的語義；累計法考慮的是每一條至根節路徑中每一個節點被使用的頻率。下面我們將一一介紹每一種量化的方式：

二元法 二元法表示我們的特徵值只有 1 與 0，這代表所有同義詞集都視為是均等的存在。一個詞彙透過 WordNet 可以查詢的同義詞集以及至根節點間所有的同義詞集，凡是用到的同義詞集均以 1 表示，反之沒有用到以 0 表示。

平均法 二元法單純只考慮了同義詞集出現與否，然而一個同義詞集可能會有二個以上的上位詞，這使得一個同義詞集到根節點的路徑不只一條，因此我們認為這些分叉的路徑應該平均分擔這個詞彙的語義，這代表每個同義詞集在該詞彙裡平均負擔的語義。將原本是二元法的特徵值除以路徑數，若分叉的路徑有相同的同義詞集，那麼我們會再將該節點的特徵值合併，最後再將所有的路徑計算平均。藉此衡量同義詞集在一個詞彙中的重要性。

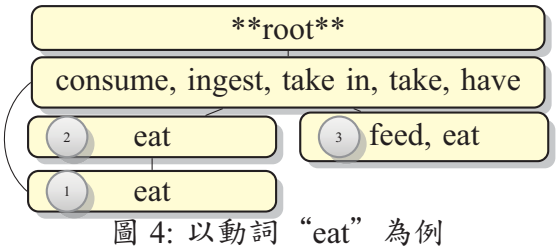

圖 4: 以動詞 “eat” 為例

圖 4表示動詞 “eat” 在 WordNet 的結構，每個節點都是上下位詞的關係，$\{**root**\}$ 是虛擬節點，在圖中的編號表示查詢 WordNet 得到的詞義（sense）編號，圖中僅列 3 個。以詞義編號 1 與詞義編號 2 作例子。路線量化的結果如表 11中間二欄，詞義編號 2 的路徑是 $\{eat\} - \{consume, ingest, takein, take, have\} - \{**root**\}$，因為詞義編號 2 ，只有走過這條路徑此，因此每個被走過的節點量化結果皆為 1。詞義編號 1 的路徑二條分別是 $\{eat\} - \{eat\} - \{consume, ingest, takein, take, have\} - \{**root**\}$ 和 $\{eat\} - \{consume, ingest, takein, take, have\} - \{**root**\}$，這時候我們會把 1 平均分擔於這二條路徑，因此這二條路徑上的每個量化的節點各是 0.5。接著我們合併同一個詞彙中相同的同義詞集，在詞義編號 1 的例子裡，二條路徑有 3 個同義詞集 $\{eat\}$、$\{consume, ingest, takein, take, have\}$ 和 $\{**root**\}$ 是重複的，因此我們把原本分擔於二條路徑上的 0.5 相加，使之成為 1，X 的部分視為 0，結果如表 11右邊二欄所示。最後，再計算每個同義詞集平均被經過的次數。以表 11詞義編號 1 與詞義編號 2 中的二個 $\{eat\}$ 為例，較抽象的 $\{eat\}$ 被經過次數只有一次，因此合併每個詞義量化後的結果再除以 1；較具體的 $\{eat\}$ 被經過的次數有二次，因此量化的結果相加後再除以 2。量化的結果如表 12所示。

表 11: 路線量化

	路線量化			加總合併	
詞義編號	1		2	1	2
路徑	1	2	1	1	1
$\{eat\}$	0.5	0.5	X	1	X
$\{eat\}$	0.5	X	1	0.5	1
$\{consume, ingest, takein, take, have\}$	0.5	0.5	1	1	1
$\{**root**\}$	0.5	0.5	1	1	1

表 12: 合併路徑量化

同義詞集	加總	平均法	累計法
$\{eat\}$	1	1	1
$\{eat\}$	1.5	0.75	1.5

累計法 與平均法相比較，累計法是比較重視上位詞，越是上位的同義詞集越是抽象，也表示被經過的次數會越多次，被我們視為是較具代表性的同義詞集。

同樣以“eat”為例，開始路線的標記同表 11。最後，將所有同義詞集加總的結果做正規化，正規化是將最後的量化的結果轉換到 0 到 1 之間，表 12的累計法未正規化。

3.1.2 特徵加權

我們也對二種不同的特徵加權的方法各有不同詮釋：同樣是考慮透過 WordNet 查詢的詞彙，從查詢到的同義詞集到根節點間所有的同義詞集，詞義頻率考慮的面向是找出較常被使用到的詞義，這可以幫助我們處理一些語義歧義的問題；語義深度考慮到 WordNet 是階層式的架構，在 3.1.1節是以類似詞袋（bag of word）概念量化的方式中，多加入了一些階層式的概念。底下我們分別介紹二種加權方式：

詞義頻率 頻率是取自於 WordNet 所記載的頻率，在 WordNet 所記載的頻率不僅是針對字面的頻率，而是有考慮詞義的頻率。雖然在我們的研究中不做語義歧義處理，但利用這點我們可以稍辨識常用詞義為何。通常頻率越高表示越常被用到。

以 $\{eat\} - \{consume, ingest, takein, take, have\} - \{**root**\}$ 這條路徑為例，透過 WordNet 我們可以查詢到 $\{eat\}$ 這個同義詞集在 WordNet 的頻率是 13，因此我們以 13 代表這條路徑在這個詞彙中的重要性。

語義深度 語義的深度取自於該節點到根節點所經過的節點數量，也就是樹的深度。當到根節點的路徑不只一條時，則會計算平均深度長。當語義的深度越深，則該同義詞集被我們視為越不重要，反之越淺則越重要。因此我們將語義深度以倒數表示，再乘上同義詞集在 3.1.1節中量化的結果。

同樣以 $\{eat\} - \{consume, ingest, takein, take, have\} - \{**root**\}$ 這條路徑為例，節點深度依序為 $3 - 2 - 1$。

3.2 特徵選擇

一個詞彙可能多義，但我們不做語義歧義處理，而是將所有可能的語義及其不同層次的抽象化語義均納入特徵池 (feature pool)，所以特徵池的特徵數量會非常的多，特徵池表示所有候選的同義詞集。由於特徵池非常龐大，而的特徵是從 WordNet 查詢來，所以這些特徵彼此存在著階層式的關係。因此我們設計了一套階層式特徵選擇的方法，而透過這樣階層式的選擇後，便可以找出具代表性的特徵，觀察與介系詞最相關連的語義層次為何，並瞭解我們研究的問題在何種語義層次上是可以被解決的。

3.2.1 階層式選擇

階層式選擇方法可參考演算法 1所示。首先，我們會將所有案例中的三個中心詞動詞、名詞一和名詞二（介系詞片語定位問題是給定已知的介系詞；介系詞推薦問題的介系詞則是答案）透過 WordNet 查詢同義詞集及到根節點間的所有同義詞集，並且將這些同義詞集都放到特徵池裡。首先，先從特徵池選出所有案例的最底層的同義詞集將之視為初始的特徵，利用 3.1節的方法將特徵數值化。再透過 3.2.2節篩選條件過濾不具代表性的同義詞集，被留下的同義詞集會繼續參選下一個世代的階層式選擇；被過濾掉的同義詞集則會被拋棄，在下個世代階層式選擇中，會以上位詞來取代。被保留下的特徵與被新選上的同義詞集也就是被拋棄的同義詞集的上位詞，會再被數值化，

然後重做階層式選擇，如此反覆直到終止條件成立。這裡我們所設定的終止條件是當特徵量過少即會停止。另一方面，由於同義詞集在越高層次特徵量會越少，因此透過這樣階層式的選擇，可達到縮減特徵的目的。

我們將以圖 5為範例解釋階層式選擇流程。圖 5是一個簡化的 WordNet 結構，每一個節點都代表一個同義詞集，其中 S_i 代表同義詞集的編號。表 13是簡化的語料庫，假設我們只有三筆案例，將三個中心詞簡化成一個，每一個中心詞透過 WordNet 查詢後，都至少有一條從該節點到根節點 S_4 的路徑。同義詞集量化的過程，我們以二元法做為範例。

第 0 個世代被視為是初始的世代。首先我們會挑出所有案例最底層的同義詞集作為第 0 世代特徵，選上的同義詞集有 S_1、S_2 和 S_5，再以二元法量化，結果如表 14世代 0。透過 3.2.2節特徵篩選條件過濾後，假設 S_1 與 S_5 是這個世代被選出需要淘汰的特徵，那麼我們就會挑選 S_1 的上位詞 S_3 和 S_5 的上位詞 S_6 補上。此時 S_3 同時也是 S_2 的上位詞，因此對於 VP2 的案例而言，VP2 多出一個新的參選特徵，接著再重新量化新選出的特徵如表 14世代 1。接著在第二代參選中，如果我們淘汰 S_3，就會再以 S_4 補上如表 14世代 2 所示，那對於 VP2 這個案例 S_3 就會被視為是不存在。在第三代的參選中淘汰 S_2 則應補上 S_4，但 S_4 已經存在，所以這個世代特徵只會減少不會新增，如表 14世代 3。

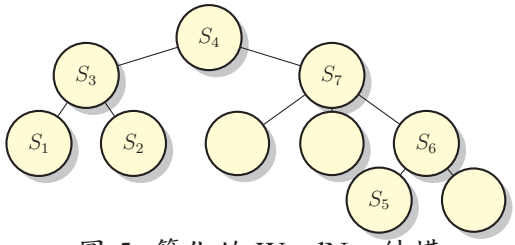

圖 5: 簡化的 WordNet 結構

表 13: 簡化語料庫案例

編號	同義詞集至根節點路徑
VP1	$\{S_1\} - \{S_3\} - \{S_4\}$
VP2	$\{S_2\} - \{S_3\} - \{S_4\}$
VP3	$\{S_5\} - \{S_5\} - \{S_7\} - \{S_4\}$

表 14: 階層式選擇範例: 世代

	世代 0			世代 1			世代 2			世代 3	
代號	S_1	S_2	S_5	S_3	S_2	S_6	S_4	S_2	S_6	S_4	S_6
VP1	1	0	0	1	0	0	1	0	0	1	0
VP2	0	1	0	1	1	0	1	1	0	1	0
VP3	0	0	1	0	0	1	1	0	1	1	1

3.2.2 篩選條件

篩選條件的方式可以分為三種：（一）以計算該特徵使用的頻率並過濾低頻的部分，（二）計算熵比例（gain ratio）並且過濾熵比例等於 0 的特徵，（三）計算共現（collocation）同義詞集。被過濾的同義詞集表示其抽象化的程度不夠因此不具代表性，所以我們將其再度抽象化並以上位詞取代。

詞頻 在我們特徵選擇方法中，假設越抽象化的同義詞集應該是越重要且越常被使用。因此我們統計每個世代的特徵在案例中出現在的次數，將該回合的特徵的頻率依多寡排列並設二個門檻值，當特徵頻率低於 10 或門檻值時就會被過濾。

熵比例 熵比例是我們用來計算該特徵代表性的方法之一，以 $\{entity\}$ 為例子，$\{entity\}$ 這個特徵是名詞的同義詞集最抽象化的概念，所有的名詞的根節點一定是 $\{entity\}$，因此當這個同義詞集被選上做為特徵後，語料庫中的所有案例都會有 $\{entity\}$ 這個特徵，雖然它是詞頻最高的特徵，但因為它高到每個案例都有，因此這個特徵的重要性反而大大降低，所以我們透過熵比例把計算為 0 的特徵過濾。

共現同義詞集 除了單個同義詞集的頻率外，我們也統計了共現同義詞集頻率，也就是在這回合作為同義詞集的特徵中，每二個同義詞集一起出現的頻率。由於我們是將動詞、名詞一與名詞二的同義詞集混合在一起，所以再我們統計共現同義詞集後，我們也定了一些規則，將較不合理的狀況去除。不合理的組合包含：動詞 + 動詞、名詞一 + 名詞一、名詞二 + 名詞二組合。另外，我們大膽假設了一些情況，如果名詞二與

演算法 1 階層式特徵選擇

輸入： 語料庫

輸出： N 個世代具代表性特徵

find_representational_features_for_each_generation(Corpus)

{將語料庫每個案例的動詞、名詞一和名詞二做特徵處理}
for all c ∈ Corpus **do**
fp$_{candidate}$ ← feature_processing(c)
end for
i = 0 { 初始世代 }
for all c ∈ Corpus **do**
g$_i$ ← find_all_leaves_in_candidate_feature_pool(c)
end for
while (terminal_ conditions_ cannot_ be_satisfied) **do**
ft = build_feature_vector(fp$_{candidate}$, g$_i$)
fp$_{keeped}$, fp$_{abandoned}$ = select_feature(ft)
i = i + 1 { 進入下一個世代 }
g$_i$ ← fp$_{keeped}$
for all f ∈ fp$_{abandoned}$ **do**
g$_i$ ← find_hypenyms(f, fp$_{keeped}$, fp$_{abandoned}$)
end for
end while
return g

演算法 2 尋找上位詞

find_hypenyms(f, fp$_{keeped}$, fp$_{abandoned}$)

ch = find_canditante_hynernyms(f)
for all h ∈ ch **do**
if (h ∉ fp$_{keeped}$ ∨ h ∉ fp$_{abandoned}$) **then**
hypernyms ← h
end if
end for
return hypernyms

名詞一或名詞二與動詞的關聯性較強，那麼我們也把動詞 + 名詞一的組合刪除。透過上述的規則，我們希望可以再減少一些不具代性的特徵。

3.3 模型建構

特徵選擇後，下一步是使用機器學習的演算法建構模型，用以決策類別。

3.3.1 基準模型建構

我們針對介系詞片語定位問題設計了一套類似於 Naïve Bayes 的演算法，稱為基準模型，也就是模型決策的結果會被我們視為是基線。在特定的介系詞之下，我們考慮的不再是單一特徵而是動詞、名詞一與名詞二的同義詞集組合。在考慮同義詞集組合的情況下，我們有機會知道怎樣的組合是較有機會可以解決介系詞定位問題。

考慮動詞片語的四個中心詞，若限定在特定介系詞 P 的情況下，則以 $\vec{W} = (V, N1, N2)$ 表示一個動詞片語，其中 V、$N1$、$N2$ 分別表示動詞、名詞一以及名詞二。若我們想要解決的介系詞片語定位問題 D 是定位修飾動詞或修飾名詞一，則 $D = \{VPP, NPP\}$。更進一步，我們可計算修飾動詞的機率 $Pr = (D = VPP|\vec{W})$ 和修飾名詞的機率 $Pr = (D = NPP|\vec{W})$。透過 WordNet 查詢後，動詞的同義詞集表示為 $V = \{s_{v_1}, s_{v_2}, \cdots, s_{v_i}\}$，名詞一表示為 $N1 = \{s_{n1_1}, s_{n1_2}, \cdots, s_{n1_j}\}$，名詞二表示為 $N2 = \{s_{n2_1}, s_{n2_2}, \cdots, s_{n2_k}\}$。

以 $\vec{S} = \{s_{v_i}, s_{n1_j}, s_{n2_k}\}$ 代表 $\vec{W}$ 一個可能的詞義組合特徵，以 $R(\vec{S})$ 表示所有可能的詞義組合特徵。

因此我們可以將模型表示成式 (4)，式 (5) 我們假設 D 與 $\vec{W}$ 在已知 $\vec{S}$ 的情形下是條件獨立，$\vec{S}$ 展開後得到式 (6)，若再假設式 (6) 個別詞彙在語境中所負擔的詞義角色與其它詞彙無關，則最後可以得到式 (7)。

$$Pr(D|\vec{W}) = \sum_{\vec{S}\in R(\vec{S})} Pr(D, \vec{S}|\vec{W}) = \sum_{\vec{S}\in R(\vec{S})} Pr(\vec{S}|\vec{W}) \times Pr(\vec{D}|\vec{W}, \vec{S}) \quad (4)$$

$$= \sum_{\vec{S}\in R(\vec{S})} Pr(\vec{S}|\vec{W}) \times Pr(\vec{D}|\vec{S}) \quad (5)$$

$$= \sum_{\vec{S}\in R(\vec{S})} Pr(s_{v_i}, s_{n1_j}, s_{n2_k}|V, N1, N2) \times Pr(D|s_{v_i}, s_{n1_j}, s_{n2_k}) \quad (6)$$

$$= \sum_{\vec{S}\in R(\vec{S})} Pr(s_{v_i}|V) \times P(s_{n1_i}|N1) \times Pr(s_{n2_i}|N2) \times Pr(D|s_{vi}, s_{n1_j}, s_{n2_k}) \quad (7)$$

根據上述式子，最後推得的結果中 $Pr(s_{v_i}|V)$、$P(s_{n1_j}|N1)$ 和 $Pr(s_{n2_k}|N2)$ 項，以動詞為例，可經由式 (8) 而得，其中 $WN(S)$ 表示一個詞彙的其中一個同義詞集經由 WordNet 查詢得到的頻率。由於某些詞義詞頻可能為 0，因此我們使用 *Laplace estimator* 概念做平滑化（smooth），而式 (8) 中 $|V|$ 表示 V 的個數。

$$Pr(s_{v_i}|V) = (WN(s_{v_i}) + 1)/((\sum_{s_{v_m}\in V} WN(S_{v_m})) + |V|) \quad (8)$$

$Pr(D|s_{v_i}, s_{n1_j}, s_{n2_k})$ 項，則是再訓練時，經由統計而得。

$$Pr(D|s_{v_i}, s_{n1_j}, s_{n2_k}) = |(s_{v_i}, s_{n1_j}, s_{n2_k}, D)|/|(s_{v_i}, s_{n1_j}, s_{n2_k})| \quad (9)$$

在只使用三個中心詞的情況，我們相信語義抽象化程高時，就可以使模型分類出大多數的案例。所以我們挑選代入名詞一與名詞二的同義詞集是從 WordNet 名詞根節點往下數第三層的同義詞集。動詞同義詞集與名詞較不同是它的樹狀結構深度淺，且虛擬根節點下一層的同義詞集非常多，因此我們挑選虛擬根節點下一層同義詞集的類別（lexicographer）代入。將這些選上的同義詞集經統計計算後代入式 (7)。

3.3.2 傳統模型建構

傳統模型表示我們使用的是現在常用的熱門演算法。我們共選了三種 SVM、C4.5 和 Naïve Bayes 演算法，其中 SVM 採用的是 Libsvm-3.11[9]版本，而後二者 C4.5 與 Naïve Bayes 使用的工具是 Weka3-6-6[10]版本。

SVM 我們選用的 kernel method 為 RBF，需要調整參數 γ 和 $cost$ 以使參數最佳化。我們使用 grid search 演算法調整參數，利用的工具是 libsvm 的 grid.py。將 x 軸對應到 $cost$ 參數範圍設為 -5 到 11，步數為 2。而 y 軸對映到 γ 參數範圍設為 -11 到 3，y 軸步數皆設為 2。再將 x 軸與 y 軸值代入涵數 $f(n) = 2^n$，並測試 $f(x)$ 與 $f(y)$ 分類的效果。

C4.5 的演算法需要調整二個參數：每個節點至少包含的案例數和 $confidencefactor$。將前者參數設為 x，範圍設為 5 到 50，步數為 5。後者參數設為 y，範圍設為 0.05 到 0.45，步數為 0.05。最後直接將 x 與 y 值代入演算法測試分類效果。

Naïve Bayes 則是無調整體的參數。

3.3.3 高階模型（meta learner）建構

現在許多成功的分類器背後都不單使用一個分類器，而是採用多個分類器整合而成。這類似於將每個模型都視為是一位專家，讓每位專家都發表自己的看法。

我們同樣使用第 3.3.2節所提到三種演算法：SVM、C4.5 和 Naïve Bayes 來建構高階模型。先利用訓練語料建立傳統模型，再決策驗證語料的答案，將決策的答案當作高

[9]http://www.csie.ntu.edu.tw/ cjlin/libsvm/

[10]http://www.cs.waikato.ac.nz/ml/weka/

階模型的訓練資料，用以訓練高階模型，最後把測試語料當作最終上線的測試資料並利用訓練好的高階模型來做最後決策。

我們設定了不同的條件，挑選品質較佳的傳統模型來建構高階的模型。條件 1，是將所有傳統模型結果都用來訓練高階模型；條件 2，表示會先計算所有模型的精準度，只挑選高於平均的模型來訓練高階模型；條件 3，同條件 2，但是我們只選擇傳統模型是由 SVM 演算法訓練而成的模型，事實上由 SVM 所訓練而成傳統模型表現是比較好的。

4 實驗評量與分析

本節將分析第 3 節建立模型的成效。

4.1 實驗評量

本研究中，評量方法最後會以百分比的方式呈現於實驗分析中。

我們以精準度（$Accuacy$）做為評量模型總體的方式。假設模型可以答對的案例數為 T，而語料庫的案例數為 N，則精準度如式 (10)。

$$Accuracy = T/N \tag{10}$$

另外，我們以準確率（$Precision$）、召回率（$Recall$）和綜合評量（F_1-$measure$）評量各個類別的決策效果。若是介系詞片語定位的問題，那麼決策的類別粗分為答對與答錯。若是評量介系詞校正問題，類別是各個介系詞。

精確率表示模型對於該類別分類的正確性。假設某一模型在語料庫中，決策某一類別數量為 E，而模型可以對該類別正確做決策的數量為 D，則準確率可以描述如式 (11)。

$$Precision = D/E \tag{11}$$

召回率表示的是我們的模型對於該類別的信心程度。假設某一類別有 G 個案例數，那麼召回率定義如式 (12)

$$Recall = D/G \tag{12}$$

實務上，我們希望精確率和召回率都很高。但往往結果是高精確率與低召回率或高召回率低精確率。在這樣的情況下，需要一個綜合評合的指標，因此我們採用綜合評量（F_1-$measure$），如式 (13)。

$$F_1\text{-}measure = (2 \times Precision \times Recall)/(Precision + Recall) \tag{13}$$

4.2 實驗分析：介系詞片語定位

本節將分析整體介系詞片語定位的實驗成果。我們根據 3.1.1節、3.1.2節和 3.2.2節設計 12 種不同的條件組合做實驗。實驗結果顯示不同量化和加權的方法對結果並沒有太大的差別，而篩選特徵的方式對於實驗結果是比較有影響。每個世代的特徵篩選結果，在精準度差不多的情況下，考慮共現義同義詞集相較於考慮詞頻是比較有用處。因為共現同義詞集是屬於較嚴苛的條件，以致在訓練時使用的特徵量是比較少。此外，詞頻與共現詞頻的門檻值不宜太高，若將每個世代使用的同義詞集照頻率高低排序，我們會發現與 Zipf's law 曲線是一樣的，這說明常用的同義詞集是有集中的現象，少用同義詞集數量較多，因此門檻值設太高容易在開始時就過濾掉大多數的同義詞集。另外，階層式選擇大約在 3 到 5 個世代就可以找到有用的特徵，之後的世代精準度就會開始明顯往下遞減。

我們從這些條件組合中，利用驗證語料選出表現最佳的模型，若有二個以上模型表現一樣好，則平均計算精準度。而選出最佳的模型幾乎都是以 SVM 演算法訓練而成。最後測試語料會用以衡量最終精準度。

在介系詞片語定位問題裡，我們將傳統模型再分為單一模型、混合資料模型。單一模型表示對每個介系詞特製化分類器，混合資料模型則是將我們挑選的 6 個介系詞混合訓練。

表 15是 RRR 語料的綜合結果，PTB3 語料庫我們也做了同樣的實驗，二者結果差不多。PTB3 語料庫比較著重於與 Stanford 剖析器做比較。表 15的 Max Enp 表示最大熵值法，最佳高階模型表示挑選每個介系詞最好的高階模型，高階混合資料 (3)SVM 表示混

合資料的高階模型是以 SVM 訓練且利用條件 3 選擇傳統模型。算數平均表示將所有介系詞都視為是一樣重要，因此所有介系詞的權重相同；而加權平均表示考慮到各個介系詞的數量，將數量視為是權重。

比較表 15的單一模型、混合資料模型與基線的精準度，單一模型與混合資料模型的精準度均較基準模型高。這表示透過階層式的特徵選擇比起只固定選擇較高層次的特徵有效。單一模型、混合資料模型精準，二者是差不多。但單一模型可以有效的針對特定介系詞找出具代表性的同義詞集做為特徵，這是混合資料模型無法做到。且混合資料模型的語料量遠較單一模型大，這可能也是無法突顯單一模型效果的原因之一。若能提升單一模型語料量，那麼單一模型應該還有進步的空間。最佳高階單一模型與單一模型，前者略勝一些；而高階混合資料與混合資料相比較，結果差不多，整體而言高階模型改善的幅度有限。最後比較我們的方法與最大熵值法，除表中基線精準度與最大熵值法差不多外，可以觀察我們每一組實驗的結果不僅較分布線佳而且也較最大熵值法好。

表 15: RRR 實驗結果

介系詞	個數	分布線 (%)	精準度 (%)					
			Max Enp	基線	單一	混合資料	最佳高階單一	高階混合資料 (3)SVM
for	259	60.86	62.55	67.57	74.13	73.75	74.13	71.81
on	159	63.90	67.92	67.30	69.81	72.96	73.58	74.84
in	356	64.64	75.56	72.47	79.21	79.78	79.21	80.34
with	90	64.80	60.00	63.33	66.67	67.78	72.22	68.89
from	92	66.67	69.57	75.00	79.35	78.26	82.61	76.09
to	211	66.80	72.51	70.62	85.31	83.89	88.15	83.89
算數平均		60.35	68.02	69.38	75.75	76.07	78.32	75.98
加權平均		59.21	69.41	69.67	76.95	77.21	78.66	77.12
混合資料	1167	50.04						

我們發現 Stanford 剖析器會將原本不是動詞片語的結構誤認為是動詞片語；反之也有可能原是動詞片語的結構，但 Stanford 剖析器無法辨識。我們將 PTB3 測試語料的原句讓 Stanford 剖析，並將答案粗分成二類答對與答錯，結果如表 16所示，P 表示精準率，R 是召回率，F 是綜合評量。如果將二種情況的案例去除，可得表 17。這時候比較我們的方法與 Stanford 剖析器的結果，可以看到二者的精準度差不多。

表 16: SP 答題狀況

介系詞	P(%)	R(%)	F(%)
for	87.26	74.90	80.61
on	89.73	75.72	82.13
in	88.39	79.53	83.73
with	91.01	72.97	81.00
from	88.17	78.10	82.83
to	87.82	79.72	83.57

表 17: PTB3 實驗結果

介系詞	個數	分布線 (%)	精準度 (%)		
			SP	單一	混合資料
for	247	63.97	74.90	77.63	77.63
on	173	50.29	75.72	75.28	79.02
in	469	50.32	79.53	77.47	78.68
with	111	56.76	72.97	67.20	75.20
from	105	62.86	78.10	76.15	79.67
to	217	70.51	79.72	83.59	81.47
算數平均		59.12	76.82	76.22	78.61
加權平均		57.72	77.53	77.25	78.77
混合資料	1322	52.27			

雖然我們的方法與 Stanford 剖析器差不多，但二種方法各有優缺點。在考慮語境資訊上，我們的方法考慮的語境資訊較少；但 Stanford 剖析器考慮的是全句，使用資訊較多。我們的方法需要事先給定四個中心詞；Stanford 剖析器只要有完整的句子便能夠剖析、定位修飾對象。

4.3 實驗分析：介系詞推薦

定位與推薦問題是用同樣的模型建構方式。除了華爾街日報與紐約時報外，我們也以 RRR 語料庫重複第 3節流程的實驗。不同組合的條件結果，影響較大的是考慮共現

同義詞集的條件，不僅可以有效的大幅減少不必要的特徵且效果依舊不錯。三種傳統模型訓練結果仍舊是 SVM 勝過 C4.5 與 Naïve Bayes，後二者又以 C4.5 較佳。

數量大語料庫，在受限硬體環境的情況下，較沒有像辨法 RRR 語料庫將所有條件組合一一跑過一次，藉以挑選最好的模型條件。因此我們從 RRR 語料庫的結果中，挑選表現最佳的條件用於大語料庫上。介系詞推薦的分析主要會著重在華爾街日報與紐約時報組成的大語料庫上。

表 18: 大語料庫實驗結果

介系詞	分布線 (%)	P(%)	R (%)	F (%)
of	27.71	51.40	70.13	59.32
in	20.88	49.91	60.74	54.80
for	10.62	36.66	30.46	33.27
to	10.34	48.39	35.43	40.91
on	8.91	54.74	45.83	49.89
with	6.63	35.24	21.50	26.71
from	4.79	30.18	16.22	21.10
at	4.16	38.65	30.36	34.01
as	2.77	40.46	22.18	28.65
by	1.88	31.88	13.58	19.05
about	1.30	45.16	25.00	32.18
精準度		47.76%		

表 19: 對照組，原文僅表示到小數後第二位

介系詞	個數	分布線 (%)	P(%)	R(%)	F(%)
of	7485	38.18	88	78	83
to	4841	24.69	78	87	82
in	4278	21.82	75	78	77
on	1483	7.56	66	65	65
with	1520	7.75	73	69	70

表 20: 混淆矩陣: 華爾街日報與紐約時報

		決策答案										
		of	in	for	to	on	with	from	at	as	by	about
實際答案	of	**1801**	283	72	66	48	31	28	31	17	2	11
	in	421	**1101**	74	49	58	28	16	31	14	2	7
	for	**355**	188	243	42	26	26	6	21	4	1	4
	to	275	140	65	**283**	42	27	6	34	6	9	5
	on	216	124	36	24	**315**	13	12	14	7	0	7
	with	**267**	95	34	21	20	94	8	16	10	3	4
	from	**125**	117	37	28	20	7	56	17	1	4	1
	at	**111**	58	28	21	18	4	8	100	2	7	2
	as	**123**	30	10	6	7	7	2	7	45	2	0
	by	**71**	31	13	13	7	5	3	4	2	11	2
	about	**49**	14	5	7	3	5	0	0	1	0	28

表 18是實驗的結果。單獨看每個介系詞效果，“on”的精確率最好的，“of”的召回率和綜合評量是最好。與表 20混淆矩陣（confusion matrix）一起觀察，可以看到介系詞幾乎都偏好“of”，第二名是“in”。綜合在 RRR 語料庫所觀察的結果，我們發現模型的偏好可能與介系詞的特性關係較小，而與介系詞在語料庫裡分布的數量多寡比較有關，表中的召回率與分布線幾乎是呈現正相關，因此我們目前推測的模型效果與介系詞各類別語料的數量比較有關係。

在介系詞推薦實驗中，目前所回顧論文提及的語料取得較為困難，這使得我們難與其它方法比較。然而介系詞推薦只要能夠取得文章，就可以利用現有的工具，自動化的處理取得我們所需的部分。所以我們可以很容易大量取得語料完成實驗，因此在足夠大量的語料下，我們相信即使不能完全與其它方法相比較，但仍然可以參考語料庫介系詞的分部，以知道目前研究的成效。若與研究性質較近文獻比較，目前的成果與 De Felice 和 Pulman [5] 的成果是有一段差距，參考表 19。但本研究中所考慮的資訊的僅包三個中心詞，而 De Felice 和 Pulman 的研究則是考慮了上下文的語義而視窗大小

（window size）設為 6，這也顯示我們的研究還有進步的空間。

5 結論

本研究以四個中心詞為出發，透過 WordNet 階層式的語義概念，建構一般化的模型，同時解決二個介系詞相關的議題。我們的實驗結果顯示，一般化的模型對介系詞片語定位問題能有不錯的表現，特別我們專注於幾個較具有挑戰性的介系詞。但對於介系詞推薦問題，與現行較好的成果是有一小段差，而目前的實驗成果只顯示混淆矩陣易偏好語料量較多的類別，那在未來我們希望能透過更多實驗探究可能的原因。我們發現透過 WordNet 發掘不同程度的抽象語義確實有助於改善分類效果，但 WordNet 對詞彙的描述分類過於細，因此希望未來能透過 SUMO[11]再改善實驗成果。應用方面，希望介系詞片語定位與介系詞推薦在未來能對機器翻譯和文本校對有所幫助，並期望二者在未來可以輔助人類解決問題。其它限於篇幅因此不能在本文中全面交代相關細節，相關細節可參考 Tsai [15]。

致謝

本研究承蒙國科會研究計畫 NSC-100-2221-E-004-014-的部分補助，僅此致謝。我們感謝評審對於本文的各項指正與指導。

參考文獻

[1] M. Atterer and H. Schütze, “Prepositional Phrase Attachment without Oracles,” *Computational Linguistics*, vol. 33, no. 4, pp. 469–476, 2007.

[2] T. Baldwin, V. Kordoni, and A. Villavicencio, “Prepositions in Applications: A Survey and Introduction to the Special Issue,” *Computational Linguistics*, vol. 35, no. 2, pp. 119–149, 2009.

[3] M. J. Collins, “Head-driven Statistical Models for Natural Language Parsing,” Ph.D. dissertation, University of Pennsylvania, 1999.

[4] G. F. Coppola, A. Birch, T. Deoskar, and M. Steedman, “Simple Semi-supervised Learning for Prepositional Phrase Attachment,” in *Proceedings of the 12th International Conference on Parsing Technologies*, 2011, pp. 129–139.

[5] R. De Felice and S. G. Pulman, “Automatically Acquiring Models of Preposition Use,” in *Proceedings of the Fourth ACL-SIGSEM Workshop on Prepositions*, 2007, pp. 45–50.

[6] ——, “A Classifier-based Approach to Preposition and Determiner Error Correction in L2 English,” in *Proceedings of the 22nd International Conference on Computational Linguistics*, vol. 1, 2008, pp. 169–176.

[7] M. Gamon, J. Gao, C. Brockett, and R. Klementiev, “Using Contextual Speller Techniques and Language Modeling for ESL Error Correction,” in *Proceedings of Joint Conference on Natural Language Processing 2008*, 2008, pp. 449–456.

[8] D. Hindle and M. Rooth, “Structural Ambiguity and Lexical Relations,” *Computational Linguistics*, vol. 19, no. 1, pp. 103–120, 1993.

[9] C.-L. Liu, J.-S. Chang, and K.-Y. Su, “The Semantic Score Approach to the Disambiguation of PP Attachment Problem,” in *Proceedings of the ROC Computational Linguistics Conference III*, 1990, pp. 253–270.

[10] P. Pantel and D. Lin, “An Unsupervised Approach to Prepositional Phrase Attachment Using Contextually Similar Words,” in *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, 2000, pp. 101–108.

[11] L. Quan, O. Kolomiyets, and M.-F. Moens, “KU Leuven at HOO-2012: A Hybrid Approach to Detection and Correction of Determiner and Preposition Errors in Non-native English Text,” in *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, 2012, pp. 263–271.

[12] A. Ratnaparkhi, J. Reynar, and S. Roukos, “A Maximum Entropy Model for Prepositional Phrase Attachment,” in *Proceedings of the Workshop on Human Language Technology*, 1994, pp. 250–255.

[13] J. Stetina and M. Nagao, “Corpus Based PP Attachment Ambiguity Resolution with a Semantic Dictionary,” in *Proceedings of the Fifth Workshop on Very Large Corpora*, 1997, pp. 66–80.

[14] J. R. Tetreault and M. Chodorow, “The Ups and Downs of Preposition Error Detection in ESL Writing,” in *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, 2008, pp. 865–872.

[15] C.-C. Tsai, “Attachment of English Prepositional Phrases and Suggestions of English Prepositions,” Master’s thesis, National Chengchi University, 2012.

[16] J.-C. Wu, J. Chang, Y.-C. Chen, S.-T. Huang, M.-H. Chen, and J. S. Chang, “Helping Our Own: NTHU NLPLAB System Description,” in *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, 2012, pp. 295–301.

[11] http://www.ontologyportal.org/index.html