

中文文字蘊涵系統之特徵分析

Feature Analysis of Chinese Textual Entailment System

黃文奇 Wan-Chi Huang, 吳世弘 Shih-Hung Wu*

朝陽科技大學資訊工程系

Department of Computer Science and Information Engineering

Chaoyang University of Technology

{s9727603, shwu}@cyut.edu.tw

*Contact author

陳良圃 Liang-Pu Chen, 谷圳 Tsun Ku

資訊工業策進會

Institute for Information Industry

{eit, cujing}@iii.org.tw

摘要

文字蘊涵(Textual Entailment)的定義是判斷兩個句子能否互相推論。推論可分為五種類型：正向、反向、雙向、矛盾、獨立。這五種類型分別代表著不同的蘊涵關係。文字蘊涵辨識(Textual Entailment Recognition)是相當困難的自然語言處理問題。由於中文文字蘊涵的文獻較缺乏，本篇論文將中文文字蘊涵辨識提出了一個流程，提供給之後想要做這個題目的人的作為一個參考。中文的文字處理相較於英文的文字處理有許多不同的難處，在本篇論文中，我們將介紹處理中文的文字處理遇到的難處以及處理的流程。我們的系統使用支援向量機(Support vector machine, SVM)作為區分類型的演算法。使用的特徵分為兩個方向：1.文字特徵 2.語意特徵。

關鍵字：文字蘊涵、tree kernel、支持向量機、語意分析

一、緒論

近幾年來，文字蘊涵受到關注，主要是因為大家瞭解到文字蘊涵將使我們能夠更準確的去推論自然語言的語義關係[1]以及處理一些重要的應用[2]。像是檢索系統經常會檢索出成千上萬筆資料，卻難以判斷哪個句子是與問句最相關的。於是透過蘊涵的推論，從這些成千上萬的資料中挑選出最相關的句子。由於兩個句子中的關係有許多種，例如：蘊涵(entailment)、改寫(paraphrase)以及獨立(independence)等，語意推論的目的就是在於判斷兩個句子之間是屬於哪一種關係。可以將推論分為五種類型：正向、反向、雙向、矛盾、獨立這五種類型。這五種類型也分別代表著不同的蘊涵關係。正向推論為可以從 t1 句子中推論出 t2 的句子，即代表 t1 句子完整的包含著 t2 句子的資訊；而反向推論正好相反；雙向即是 t1 與 t2 兩個句子互相完全包含著彼此的資訊；矛盾即是兩個句子中提到

的資訊是互相矛盾的；獨立則是兩個句子中提到的資訊是完全不相關的。如表一。

表一中雙向蘊涵的例子比較屬於是改寫(paraphrase)，更複雜的文字蘊涵推論就像是 $t_1 \rightarrow s, s \rightarrow t_2$ 。透過 t_1 的句子可以推論出涵義 s ，接著透過涵義 s 可以推論出 t_2 例如： t_1 ：小明殺了小華。 t_2 ：小華死了。從 t_1 我們可以推論出的 s 有很多，如：小明是殺了小華的兇手、小華被殺了、小華死了。這種推論需要有邏輯推論以及許多背景知識才可以達成。基於中文處理的成本以及困難度考量，本篇論文主要針對改寫(paraphrase)去作分析。

表一 各種類型的例句

類型	例句
正向蘊涵 (forward)	t_1 ：日本時間 2011 年 3 日 11 日，日本宮城縣發生芮氏規模 9.0 強震，造死傷失蹤約 3 萬多人
	t_2 ：日本時間 2011 年 3 日 11 日，日本宮城縣發生芮氏規模 9.0 強震
反向蘊涵 (reverse)	t_1 ：美國主權債信評級從最高的 A A A 調降一級到 A A +
	t_2 ：美國主權債信評級從最高的 A A A 調降一級到 A A +，將造成美國每年的借貸成本增加約一千億美元
雙向蘊涵 (bidirection)	t_1 ：賓拉登在巴基斯坦美軍攻擊中死亡
	t_2 ：巴基斯坦美軍攻擊中殺死賓拉登
矛盾蘊涵 (contradiction)	t_1 ：張學友在 1961 年 7 月 10 日，生於香港，祖籍天津
	t_2 ：張學友生於 1960 年
獨立蘊涵 (independence)	t_1 ：黎姿與”殘障富豪”馬廷強結婚。
	t_2 ：馬廷強為香港“東方報業集團”創辦人之一馬惜如之子

在自然語言問答系統(Question Answering system)中。使用者輸入的是一個完整的問句。系統回傳的也是一個個完整的句子。並且連結到相對應的文章中。基本的檢索，通常會有上萬句的句子被系統挑選出來。要從這上萬個句子中，去挑選出哪一個最為相關，我們就可以透過文字蘊涵系統來輔助。例如 ntcir9 提供的範例中[3]，我們輸入 t_1 ：「1997 年香港回歸中國。」，我們可以從檢索回來的句子中，挑選出 t_2 ：「香港的主權和領土是在 1997 由英國政權歸還給中國的。」並且將它排序到較為前面的順序。因為 t_1 與 t_2 的關係為反向，代表著 t_1 為 t_2 蘊涵意義的一部分。所以我們可以認為 t_2 很可能是使用者要檢索的句子。我們也可以利用其他類型，來決定這些句子排序的位置。像是語意為矛盾、獨立的句子我們就將它挑除，因為它幾乎不可能為使用者所想要檢索的句子。正向、反向、雙向我們就將之排序到較前面的位置，因為這些句子較有可能為使用者要檢索的資訊。

相較於英文中文的文字處理難度高出了許多。因為在英文的句子中，每個詞

都以空白分開，且英文的文法也制定的較為清楚，如：時態、詞性...等等，在英文中都有較為明確的規範。中文卻是整個句子都連在一起，所以第一步必須要先斷詞，斷詞在中文處理上面就有一定的難度。因為處理時斷詞的結果好壞都會影響到後續處理的結果。且中文的文法相較於英文，也較為模糊，所以中文的語意判斷難度相較於英文會高出許多。

二、相關研究

其他語言到目前為止，文字蘊涵在中文的領域較缺少相關的文獻，我們只能參考其他語言文字蘊涵處理的方法。在英文處理文字蘊涵的文獻[4]將處理英文文字蘊涵的各個方法做了分析，並且將各種方法整分成下面幾個類別。

(一)、需要透過背景知識達成之方法

1、整合背景知識與邏輯推論

由於人們在平常生活中已經很習慣使用自然語言表達意見，所以自然的有許多背景知識都已經不自覺得變成常識，判斷兩句子是否可以推論都覺得很理所當然，但是對於電腦來說並不是如此。推論是可以從邏輯蘊涵來檢查，像是使用定理證明文句對中的文本蘊涵[5][6][7][8]。有一部份的學者使用含有語意的辭典來擷取出詞彙的邏輯意義，使用 WordNet[9]或者擴展 WordNet[10]。例如於 WordNet 中“暗殺”為“殺”的下義詞（更有具體的意義），像下面所示，x 暗殺 y 可以推論成上義詞 x 殺 y。所以像謀殺、刺殺...等等的這些都可以推論出相同的上義詞。

$$\forall x \forall y \text{ 暗殺}(x, y) \Rightarrow \text{殺}(x, y)$$

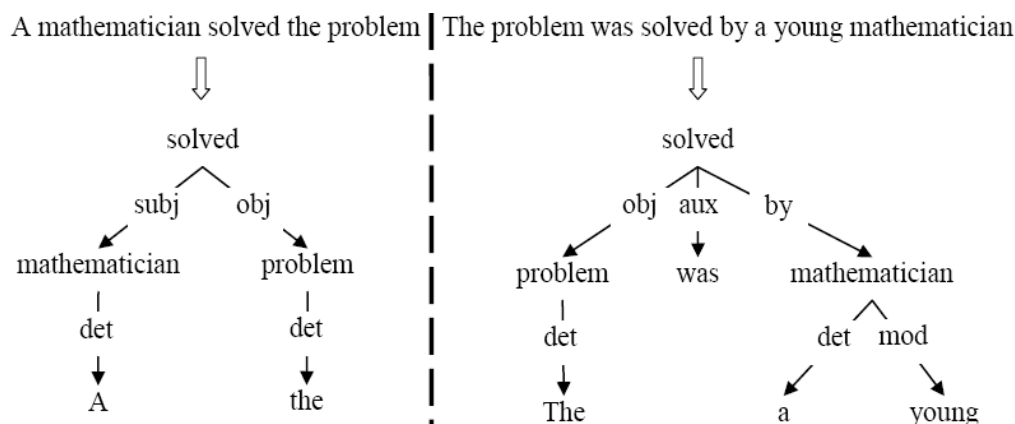
2、整合背景知識與向量空間模型

將每個輸入語言表達的字，對映到一個向量，可以看出用詞的分佈強度，特別是當句子其他的字也都對應到同一個語料庫中，則會明顯看出用字的分佈[11]。例如要求共同出現的字，出現在特別的語法依存關係上[12]。在最簡單的情況下，為每一個表達向量的總和或字詞對應的向量總和，但更複雜的方法也已提出[13]。句子可以通過測量檢測距離向量的兩個輸入表達式來判斷是否為改寫的句子，例如，通過計算其餘弦相似性(cosine similarity)。

(二)、不需要透過背景知識達成之方法

1、透過表面文字

將文字對經過一些加工，如詞性(POS, Part of speech)標記或命名實體識別(NER, Named entity recognition)標記。對輸入兩個字串計算字串編輯距離(edit distance)[14]，計算其共同的字數，或組合幾種字串相似度措施[15]，包括使用機器翻譯評測的方法，如 BLEU(Bilingual Evaluation Understudy) [17][18]都可能有助於文字蘊涵。



圖一 兩句非常相似句子的依存關係樹[1]

2、基於語法相似度

另一種常見的方法是在語法等級。依存語法剖析器(parser)[20][21]普遍用於文本蘊涵研究，一個句子輸出的剖析結果是一個圖(通常是一個樹狀結構)其節點是句子的字或詞性標記，其邊緣對應詞與詞之間的句法依存關係，例如：以依賴於動詞或名詞開頭的名詞片語，或者以名詞開頭的形容詞片語。圖一顯示了兩句話的依存關係樹[1]，分析這兩棵樹可以得知這兩句句子的蘊涵關係。例如計算共同剖析樹的邊[16][19]或使用其他樹的相似性計算方法，例如：樹的編輯距離[22][23][24]。相似性得分可以表示輸入的句子可能是改寫的程度。

3、透過機器學習

許多系統採用結合多個測量相似度的方法，在計算各種程度的相似度(表面字串，句法和語義的表示)合併使用機器學習[26][27]。每一對輸入的文句對(P1, P2)，由特徵值向量代表(f1, ..., fm)，我們用機器學習來判斷他們是否是一個特定的改寫或文字蘊涵。該向量包含多個相似度的特徵。前處理階段將每個輸入對轉換為一個特徵向量[28]。前處理還包括正規化，例如，日期將轉換成一個統一的格式，個人的名稱，組織，地點等使用命名實體識別轉換為正規化表示。代詞及指稱詞語，可能會被替換的成原本的詞[25]，構句的差異也可能標準化(例如，被動句可以轉換為主動句)。特徵向量可以統包使用文字或部分的句法和語義表現[29]。最後這個特徵向量，將當作支持向量機(SVM)的輸入值，去學習及區分各種文字蘊涵的類別。

三、資料處理與特徵分析：

本研究將語意蘊涵識別分為特徵分析及機器學習兩個主要部份。本研究提出的特徵分析流程包括前置處理、背景知識的替換程序、表面文字特徵分析程序、語意句法分析程序四個部分。資料以 NTCIR-9 提供的 421 對文句對為分析依據[3]。

(一)、前置處理

由於人們再撰寫句子的表達時可能會使用到一些替代詞以及範圍性的詞性。用程式擷取特徵，通常會遇到處理上的困難，所以我們都會必須要將資料做一些前置處理，才可以程式進行運算。

1、括號選擇性替代：

一個意思可能會有兩個詞可以代表，像是音譯、中英文、代表涵義、簡寫…等等的。在撰寫文章時，爲了讓讀者可以明確的知道作者想要表達的資訊，作者會使用括號，將讀者可能聯想到的詞也都包含進來，避免造成讀者的誤會。例如：車諾比核事故(切爾諾貝利核事故)、湯姆·克魯斯(Tom Cruise)。所以在此我們將括號中的文字與前面的詞使用陣列儲存。在特徵擷取時我們將兩個詞都同時列入考量。

2、時間正規化：

在文本中時間的表達方式有很多種格式及字型，如：中文、數字全形、數字半形、數字以「-」隔開、範圍型態等，參見表二。在此將以上各種格式，統一轉換成陣列方式，以方便之後進行比對的步驟。轉換成陣列以便後續程式作比較。

3、時間運算：

在有些例子中，需要透過運算之後才能夠知道資訊是否匹配。例如：t1：「蘇哈托政權在一九九八年結束，執政卅二年。」、t2：「蘇哈托一九六六年執政，對印尼進行了卅二年的鐵腕統治。」在例子中 t1 出現時間詞「一九九八年」與「卅二年」。經過運算之後，會與 t2 的「一九六六年」符合，所以經過時間運算結果，t1 的時間將與 t2 的時間匹配。

表二 各種時間表達方式之例句

時間型態	時間表達方式
中文	一九九七年二月廿三日
數字全形	1 9 9 7 年 2 月 2 3 日
數字半型	1997 年 2 月 23 日
數字以「-」隔開	1999-05-07
範圍	1999 年延長至 2001 年

(二)、背景知識的替換

在撰寫文章時，作者會使用一些簡寫或者是替代詞，好讓整篇文章可以更爲通順。由於作者已有相關的背景知識，會覺得這些事情是常識，所以並不會對於那些替代詞多做解釋。事實上往往讀者閱讀一些文章時，可能會因爲背景知識不足而需要去查閱許多資料，才可以讀懂那些文章的意思。這種人們都會遇到的事情，程式也必須處理。

1、年號統一：

在時間的表示詞中，有一些年號，是需要經過統一的，例如：乾隆 56 年等於西元 1791 年，是因爲「乾隆」等於 1735 年。昭和 57 年等於西元 1982 年，是因爲「昭和」等於 1925 年。將年號統一，以方便我們之後的分析。

2、地名正規化：

作者在書寫文件時，有時會爲了方便，而將一些詞簡寫，地名就是其中一個時常被簡寫的對象。例如：「台灣、印度、美國」這些國家的名稱，時常都會被簡寫爲：「台、印、美」。但是當我們在處理文字蘊涵分析的時候，我們必須要先將簡寫恢復成原地名。這樣在之後比對才會匹配。

(三)、表面字串特徵分析

1、時間：

許多文句對中的兩個句子都含有時間元素，當兩邊時間不符合時，可能是一個分析文字蘊涵的依據。從 421 句文件集中挑出來的文句對一共有 146(34.67%) 個文句對包含時間。我們將時間分析細分爲三個匹配程度，如：時間爲完全匹配、部分時間匹配、時間完全不匹配（表三）。如表三中，時間完全批配的例子，再此 t1 中含有的時間爲 2000 年，而 t2 中含有的句子也同樣爲 2000 年，所以我們將此例子視爲時間完全匹配。部分時間匹配(1)的例子，其中 t1 的時間爲「一九七八年十月十六日」，而 t2 的時間爲「1978 年」。由於「1978 年」只匹配到「一九七八年」，「十月十六日」，並沒有完全匹配。部分時間匹配(2)的例子，其中 t1 的時間爲「一九七八年十月十六日」，而 t2 的時間爲「1978 年」。由於「1978 年」只匹配到「一九七八年」，「十月十六日」，並沒有完全匹配。時間完全不匹配的例子，再 t1 的例子中時間爲「1987 年」，t2 的例子中「1988 年」。我們從這兩個時間中，明顯的看出，時間不匹配。

表三 時間批配程度之例子

批配程度	例子
時間爲完全匹配	t1：據他所知，這是查爾斯首度參加雪梨-荷芭特帆船賽，而查爾斯一向是注重安全、非常謹慎的人，他更想參加 2000 年雪梨奧運帆船賽。
	t2：2000 年奧運在雪梨舉辦
部分時間匹配(1)	t1：若望保祿二世一九七八年十月十六日被選爲教宗
	t2：若望保祿二世於 1978 年當上教宗
部分時間匹配(2)	t1：蘇哈托 1921 年 6 月 8 日出生
	t2：蘇哈托（Suharto，民間常用「Soeharto」，1921 年 6 月 8 日－2008 年 1 月 27 日）
時間完全不匹配	t1：張藝謀 1987 年以「紅高粱」拿下柏林影展金熊獎
	t2：柏林電影節應該是張藝謀的福地。1988 年，他執導的《紅高粱》贏得了最佳影片金熊獎，成爲中國電影的首個金熊獎

經過統計，這三種匹配程度所包含的類別數量如表四，可以從表中看出，完全匹配的部分，屬於 B 的機率高了很多。這是因爲當兩個句子的意義可以互相推論時，在時間上必須要完全匹配才有可能意義是一樣的。如果兩個句子都在講

同一件事情，可是日期不一樣的話，這樣就不算是完全匹配了。

表四 批配程度分佈與類別關係

	F	R	I	B	C
完全匹配	9	17	3	30	17
部分匹配	12	8	7	0	0
不匹配	1	1	2	0	7

2、句子長度：

針對句子的長度分析，將 t1 的句子長度與 t2 的句子長度做比較。即將 t1 的句子長度減去 t2 的句子長度，以用來統計各個類別中，句子長度與類別的關係。統計結果如表五所示。

表五 句子長度與類別關係

類別	F	R	I	B	C
數量	12	-19	-4	0	0

由於向前蘊涵(F)的定義為，t1 的涵義中完全的包含了 t2 的句子，所以 t1 的文字長度，應該要比 t2 的文字長度還要長許多。相反地，如果是反向蘊涵(R)則為反之，t2 的長度應該要比 t1 的長很多。而矛盾(C)以及雙向蘊涵(B)的涵義分別為互相矛盾以及互相包涵，在句子中所需提到的內容都差不多。所以句子的長度也都會差不多。獨立(I)，只要兩個句子不是在談同一個內容，他們就算是獨立的句子。所以句子的長度為不一致。

3、Bleu(Bilingual Evaluation Understudy)：

Bleu當初是被設計來測量機器翻譯(machine translation)的品質。一個良好的機器翻譯需要包含適當，準確以及流暢的翻譯[30][31]。Bleu是考量句子的相似度，經過適當修改參考，一定程度詞語的差異在選擇和語序上面。而Bleu主要的概念是使用片語匹配長度平均權重值。

4、否定詞偵測

當文句對只有差別一個否定詞時，從文字層面去計算文句對得相似度會得到很高的分數。但是從語意層面來看此文句對卻完全不相同。如：「今天天氣很好」與「今天天氣很不好」。此文句對中只有差別一個「不」否定詞。然而完全改變了此語句的語意。所以我們提出要偵測文句對中是否有否定詞，作為特徵。

(四)、語意句法分析：

1、同義詞替換：

表達同一個意思的詞彙有許多，例如：大兒子、長男、獲得、得到...等。在

人的眼中這些都是表達為第一個兒子的意思。但是要讓程式擁有這些背景知識，是一件相當困難的事情。因為處理中文必須事先經過斷詞系統處理，但斷詞系統有時會把詞彙斷的太細。如：「長男」會被斷成「長」與「男」兩個字。經由剖析處理後，產生的剖析樹會長得不一樣。導致計算語法相似度時產生出許多雜訊。為解決這個問題，透過事先建立一份同義詞清單，進行同義詞的替換。但為了減少因斷詞錯誤造成的影響，所以在此採用長詞優先來作替換。

2、語法分析(syntax analysis)：

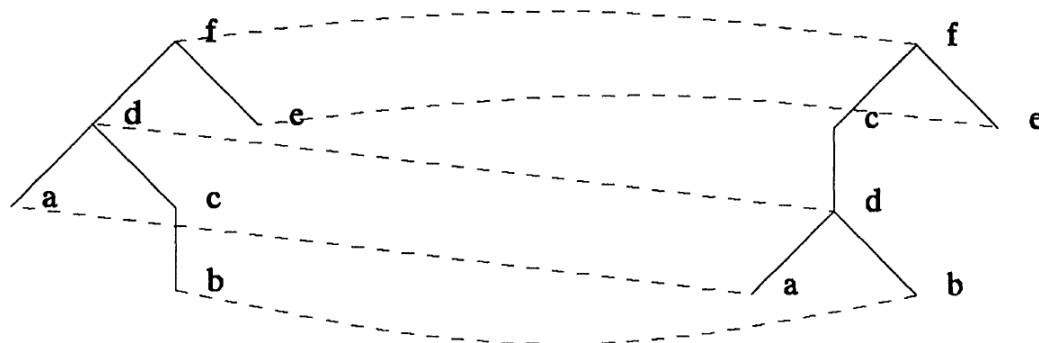
分析句法需要透過剖析器(parser)將整個句子的句法標注，才可以計算整個句子的句法。本篇論文使用的是史丹佛剖析器(Stanford parser) [32]來剖析句子的句法。但由於史丹佛剖析器在剖析繁體中文的標注時常會判斷錯誤，所以使用的時候把資料轉換成簡體中文效果會較好。在此是透過自行開發出來的簡繁轉換系統做轉換[33]。

(1)、計算tree distance

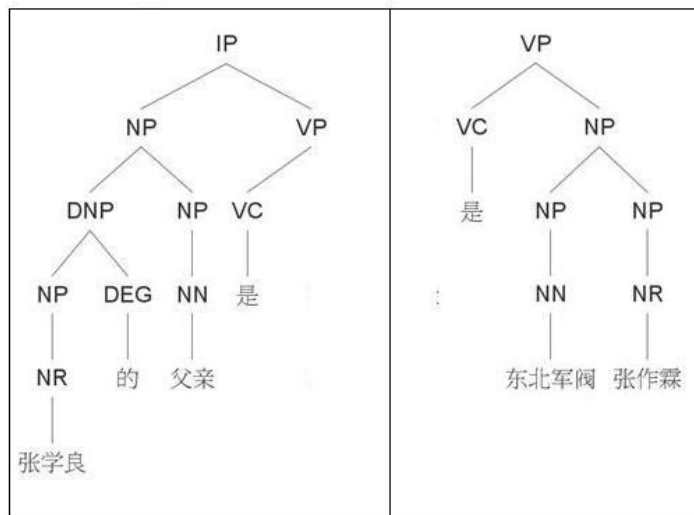
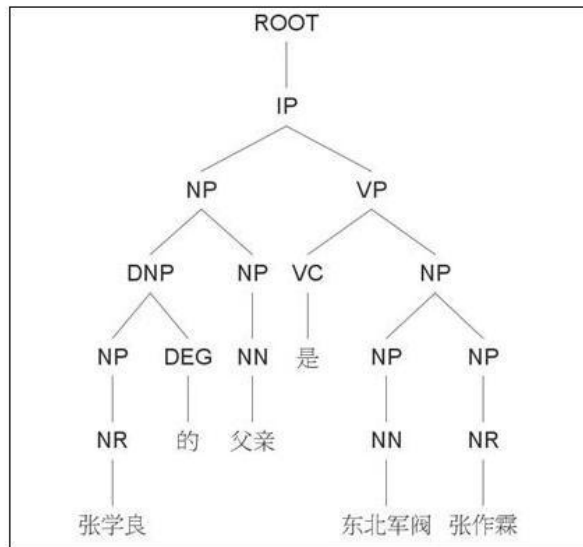
有許多學者使用tree distance去計算兩個剖析樹的相似度[21][22][23]。所謂的tree distance主要的概念就是t1的parser tree需要經過幾次插入(insert)、刪除(delete)、替代(Substitution)才可以等於t2的parse tree。如圖二[24]中，可以看出這兩顆樹中間只有差了一個c節點，在tree distance運算中只需要刪除c節點兩個parse tree就完全mapping在一起。他們的tree distance就是2。

(2)、Fast Tree Kernel (FTK)

FTK為修改Quadratic Tree Kernel (QTK)[5]的演算法，QTK主要的意義為計算兩個Tree的匹配數量。FTK為事先將動詞為中心點，分別跟其他的sub-tree合併成subset tree。如圖三中「張學良的父親是東北軍閥」將會以是這個動詞作為分割點，分割出「張學良的父親是」、「是東北軍閥」這兩個subset tree。並根據每個集合計算subset tree的匹配數量，並且求出參數最大值 $t = \arg \max_{i \in S} [34]$ 。



圖二 parse tree對應圖



圖三 根據動詞所分解出來的sub-set tree

四、實驗

本次針對五種類別進行分析與實驗，並且使用交叉驗證的方式來實做的系統。流程如圖四所示，首先輸入文句對，進行前處理，其中前處理包括：括號選擇性替代，年號統一，以及地名正規化。之後使用 ICTCLAS 系統對句子進行斷詞，接著進行簡繁轉換才將句子使用 Stanford parser 將句子剖析，接著進行特徵擷取的動作，本次擷取的特徵如表七所示，並將擷取的特徵輸入給 SVM 進行訓練以及測試，輸出將得到判斷是否為蘊涵的結果。

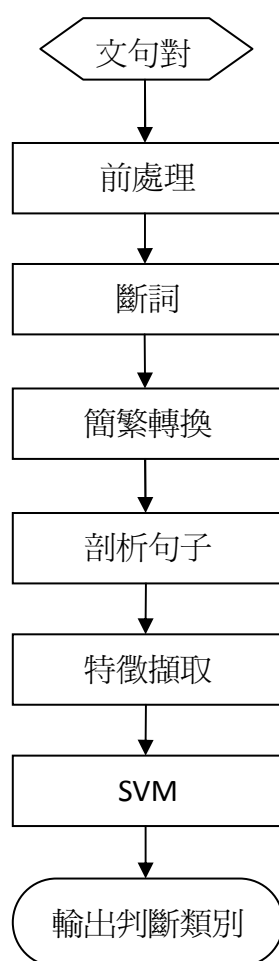
(一)、資料來源：

本次研究我們的資料來源取自日本 NTCIR 第九屆中，RITE(Recognizing Inference in Text)比賽子項目的開發資料(Development Data)。而在此資料中，一共有 421 個文句對。其中向前蘊涵(F, Forward Entailment)一共有 87 個文句對，

反向蘊涵(R, Reverse Entailment)一共有 97 個文句對，雙向蘊涵(B, Bidirectional Entailment)一共有 82 個文句對，矛盾(C, Contradiction)一共有 74 個文句對，獨立(I, Independence)一共有 81 個文句對如表六。

表六 每一個類別的資料數量

標籤(Label)	數量(Number)
向前蘊涵(F, Forward Entailment)	87
反向蘊涵(R, Reverse Entailment)	97
雙向蘊涵(B, Bidirectional Entailment)	82
矛盾(C, Contradiction)	74
獨立(I, Independence)	81



圖四 本次實驗之流程圖

(二)、使用工具：

1. ICTCLAS[35]:由於中文的句子並不像英文的句子，每個詞都以空白分開。所以要處理中文的句子首先第一個步驟需要進行斷詞，將每個詞分開。目前較為普遍的斷詞工具有兩種：由中央研究院所研發的CKIP斷詞系統[36]，與中國科學院

所研發的ICTCLAS。其中CKIP是用來處理繁體中文。ICTCLAS則是繁體與簡體中文都可以處理。所以我們的實驗選擇使用ICTCLAS。

2. Stanford parser[32]：另外一個很重要的工具就剖析器，由於要計算語意需要使用到 parser，所以在我們的實驗中使用的是 Stanford parser。因為 Stanford parser 是依據 Chinese Treebank[37]的標準，所以 Stanford parser 能夠處理英文以及簡體中文。

3. LIBSVM [38]：由於要分類的類別一共有五類，所以採用 LIBSVM 作為分類的分類器。因為此分類器可以一次分多個類別，可以避免掉傳統只能分兩類對應不到類別的問題。

4.簡繁轉換系統[33]：由於 Stanford parser 只能夠處理簡體中文，於是需要透過一個簡繁轉換的系統。這次實驗所使用的系統為自行開發的系統[33]。

(三)、使用特徵：

使用的 feature 部份是參考[16]中的特徵。因為[16]為參考文獻中[4]數據最好的，但由於[16]中的特徵為處理英文時使用，所以有些特徵中文並沒有，所以只採用部份特徵。以及加入一些上面所分析的部份特徵。所使用的特徵如表七中所列。目前實驗是將分為五種類別做四個實驗。

1.baseline 系統

其中 baseline 系統分別使用表七中 1 到 9 的特徵值，因為這些特徵值是較容易做計算的，實驗結果如表八所示，實驗的文句對一共有 421 對其中有 220 對是判斷正確(52.25%)。

2.tree mapping 系統

相較於 baseline 系統，額外加入了第 10 個 tree mapping 的特徵值，此特徵是計算 Subset Tree mapping 的值。所以使用了 1~10 的特徵值。此實驗用來測試看看語法特徵對於文字蘊涵能夠有多大的幫助，實驗結果如表九所示，實驗的文句對一共有 421 對其中有 226 對是判斷正確(53.68%)。

3.time mapping

在我們先前的試驗分析中，可以從表七中看出時間批配是一個蠻有用的特徵，所以單獨將此特徵挑選出來做實驗看效果如何。相較於 baseline 系統中，而外加入第 11 個 time mapping 的特徵，所以使用了 1~9 的並且額外加入了第 11 個特徵，實驗結果如表十所示，實驗的文句對一共有 421 對其中有 223 對是判斷正確(52.96%)。

4. Remove length

我們可以從表五中看出本資料集的長度特徵較為明顯，當然這有可能是本資料集才有的特色，所以在此將第 6 到 9 個關於長度特徵拿掉。用來測試這次實驗是否過度依賴長度特徵。實驗結果如表十一所示，實驗的文句對一共有 421 對其中有 209 對是判斷正確(49.64%)。

表七 實驗所使用之特徵

1. unigram recall
2. unigram precision
3. Bleu precision
4. Bleu recall
5. Bleu F-measure
6. difference in sentence length (character)
7. absolute difference in sentence length (character)
8. difference in sentence length (term)
9. absolute difference in sentence length (term)
10. Subset tree mapping
11. Time mapping

五、結論與未來展望

本篇論文提出了一個處理中文文字蘊涵辨識的流程，可以給之後想要做中文文字蘊涵的人作為一個參考。本次實驗只有使用部份的特徵，實驗的量也並不大，但是我們將這次的系統視為一個 **baseline**。可以從實驗中看出 **tree mapping** 與 **time mapping** 確實可以略為增加判斷文字蘊涵的效果。並且將長度的特徵拿掉之後，雖然有略為的下降，但是幅度不至於影響太多，所以以此驗證了此系統並不是非常的依賴長度特徵。

之後我們將增加各種特徵，並且去分析出有用的特徵以改進中文文字蘊涵的效果。像是年號統一、同義詞替換、地名正規化，這一類需要事先建立出背景知識才能夠使用程式去執行。但是這要建立出這些背景知識需要花費非常多的成本，需要經過很長的時間去累積詞彙的數量。目前我只有針對語料庫中遇到需要處理的建立出來，所以未來我們希望可以將每次遇到要處理的詞，建立出越來越完整的一個可以讓電腦使用的背景知識辭典。

表八 分為五類的 **baseline** 系統結果

Actual	Predicted					Total
	F	R	B	I	C	
F	60	3	9	10	5	87
R	0	68	9	15	5	97
B	5	6	56	1	14	82
I	17	35	8	16	5	81
C	11	12	21	10	20	74
Total	93	124	103	52	49	421

表九 分爲五類的 time mapping 系統結果

Actual	Predicted					Total
	F	R	B	I	C	
F	61	3	8	10	5	87
R	0	69	9	14	5	97
B	6	6	57	1	12	82
I	19	32	8	17	5	81
C	11	12	19	10	22	74
Total	93	122	101	52	49	421

表十 分爲五類的 time mapping 系統結果

Actual	Predicted					Total
	F	R	B	I	C	
F	61	3	8	10	5	87
R	0	69	9	14	5	97
B	6	6	57	1	12	82
I	19	32	8	17	5	81
C	11	12	19	10	22	74
Total	90	129	100	57	45	421

表十一 分爲五類的 time mapping 系統結果

Actual	Predicted					Total
	F	R	B	I	C	
F	54	6	12	12	5	87
R	7	58	10	10	5	97
B	8	2	59	3	10	82
I	16	37	9	16	5	81
C	18	12	18	8	22	74
Total	99	115	108	59	40	421

Acknowledgement

本研究依經濟部補助財團法人資訊工業策進會「100 年度數位匯流服務開放平台技術研發計畫」辦理。感謝國科會贊助部分研究經費，計畫編號 NSC-100-2221-E-324 -025-MY2。

六、參考文獻

- [1] Gennaro Chierchia and Sally McConnell-Ginet, "Meaning and Grammar: An introduction to Semantics", The MIT press, Cambridge, MA, 2000.

- [2] Ido Dagan and Oren Glickman, Probabilistic textual entailment: Generic applied modeling of language variability, In Proceedings of the Workshop on Learning Methods for Text Understanding and Mining, Grenoble, France, 2004.
- [3] NTCIR 9, Recognizing Inference in Text task, http://artigas.lti.cs.cmu.edu/rite/Main_Page.
- [4] Ion Androutsopoulos and Prodromos Malakasiotis, “A survey of paraphrasing and textual entailment methods”, Journal of Artificial Intelligence Research, Volume 38, pages 135-187, 2010.
- [5] Michael Collins and Nigel Duffy, “New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron”, Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, 2002.
- [6] Johan Bos, Katja Markert, “Recognising textual entailment with logical inference”, Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, Vancouver, B.C., Canada, 2005.
- [7] Marta Tatu and Dan Moldovan, “COGEX at RTE 3”, In Proceedings of ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, pages 22–27, Prague, Czech Republic, 2007.
- [8] Marta Tatu, Dan Moldovan, “A semantic approach to recognizing textual entailment”, In Proceedings of HLT/EMNLP 2005, pages 371–378, Vancouver, Canada, 2005
- [9] Christiane Fellbaum, “WordNet: An Electronic Lexical Database”, The MIT Press, 1998.
- [10] Dan I. Moldovan and Vasile Rus, “Logic form transformation of WordNet and its applicability to question answering”, In Proceedings of the 39th Annual Meeting of ACL, pages 402–409, Toulouse, France, 2001.
- [11] Dekang Lin, “An information-theoretic definition of similarity”, In Proceedings of the Fifteenth International Conference on Machine Learning (ICML-98) Madison, Wisconsin, 1998.
- [12] Sebastian Padó and Mirella Lapata, “Dependency-based construction of semantic space models”, Computational Linguistics, Volume 33, No. 2, pages 161–199, 2007.
- [13] Jeff Mitchell and Mirella Lapata, “Vector-based models of semantic composition”, In Proceedings of the 46th Annual Meeting of ACL: HLT, pages 236–244, Columbus, OH., 2008.
- [14] Levenshtein, V, “Binary codes capable of correcting deletions, insertions, and reversals”, Soviet Physice-Doklady, 10, pages 707–710, 1966.
- [15] Prodromos Malakasiotis, Ion Androutsopoulos, “Learning textual entailment

- using SVMs and string similarity measures”, In Proceedings of ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, pages 42–47, Prague, Czech Republic, 2007.
- [16] Wan, S., Dras, M., Dale, R., & Paris, C., “Using dependency-based features to take the “parafarce” out of paraphrase”, In Proceedings of the Australasian Language Technology Workshop, pages 131–138, Sydney, Australia, 2006.
- [17] Kishore Papineni, Salim Roukos, Todd Ward and Wei-Jing Zhu, “BLEU: a method for automatic evaluation of machine translation”, In Proceedings of the 40th Annual Meeting on ACL, pages 311–318, Philadelphia, PA, 2002.
- [18] Liang Zhou, Chin-Yew Lin and Eduard Hovy, “Re-evaluating machine translation results with paraphrase support”, In Proceedings of the Conference on EMNLP, pages 77–84 , Sydney, Australia, 2006.
- [19] Prodromos Malakasiotis, “Paraphrase recognition using machine learning to combine similarity measures”, In Proceedings of the 47th Annual Meeting of ACL and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Suntec, Singapore, 2009.
- [20] Igor Mel'cuk, “Dependency Syntax: Theory and Practice”, State University of New York Press, 1987.
- [21] Sandra Kübler, Ryan McDonald, and Joakim Nivre, “Dependency Parsing”. Synthesis Lectures on Human Language Technologies. Morgan and Claypool Publishers, 2009.
- [22] Selkow, S., “The tree-to-tree editing problem”, The Journal of Information Processing Letters, Volume 6, No. 6, 184–186, 1977.
- [23] Kuo-Chung Tai, “The tree-to-tree correction problem”, The Journal of ACM, Volume 26, No. 3, 422–433, 1979.
- [24] Kaizhong Zhang and Dennis Shasha, “Simple fast algorithms for the editing distance between trees and related problems”, SIAM Journal of Computing, Volume 18, No. 6, pages 1245–1262, 1989.
- [25] oll’a, D., Schwitter, R., Rinaldi, F., Dowdall, J., & Hess, M, Anaphora resolution in EXTRANS. In Proc. of the Int. Symposium on Reference Resolution and Its Applications to Question Answering and Summarization, pp. 23–25, Venice, Italy, 2003.
- [26] Mitchell, T, “Machine Learning”, Mc-Graw Hill. 1997.
- [27] Ethem Alpaydin, “Introduction to Machine Learning”, The MIT Press, 2004.
- [28] Yujie Zhang and Kazuhide Yamamoto, “Paraphrasing spoken Chinese using a paraphrase corpus”, The Journal of Natural Language Engineering, Volume 11, No. 4, pages 417–434, December, 2005.

- [29] Fabio Massimo Zanzotto and Lorenzo Dell'Arciprete, "Efficient kernels for sentence pair classification", In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Volume 1, pages 91–100, Singapore, 2009.
- [30] Eduard Hovy, "Toward finely differentiated evaluation metrics for machine translation", In Proceedings of the Eagles Workshop on Standards and Evaluation, Pisa, Italy, 1999.
- [31] J.S. White and T. O'Connell, "The ARPA MT evaluation methodologies: evolution, lessons, and future approaches", In Proceedings of the First Conference of the Association for Machine Translation in the Americas, pages 193–205, Columbia, Maryland, 1994.
- [32] Stanford parser, <http://nlp.stanford.edu/software/lex-parser.shtml>
- [33] Min-Hsiang Li, Shih-Hung Wu, Ping-che Yang and Tsun Ku, "Chinese Characters Conversion System based on Lookup Table and Language Model", In Proceedings of the 22nd Conference on Computational Linguistics and Speech Processing (ROCLING 2010), pages 113-127, Nantou, Taiwan, September 2010.
- [34] Alessandro Moschitti, "Making tree kernels practical for natural language learning", In Proceedings of 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL2006), pages 113–120, Treviso, Italy, 2006.
- [35] ICTCLAS, <http://ictclas.org/>
- [36] CKIP, <http://ckipsvr.iis.sinica.edu.tw>
- [37] Roger Levy and Christopher Manning, "Is it harder to parse Chinese, or the Chinese Treebank?", In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, Volume 1, pages 439-446, Sapporo Convention Center, Japan, 2003.
- [38] LIBSVM, <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>