行政院國家科學委員會專題研究計畫 成果報告

貝氏網路與分類技術之基礎研究與應用：建構學生學習歷程之模型與語意標記(第 2 年)
研究成果報告(完整版)

計畫類別：個別型
計畫編號：NSC 95-2221-E-004-013-MY2
執行期間：96 年 08 月 01 日至 97 年 10 月 31 日
執行單位：國立政治大學資訊科學系

計畫主持人：劉昭麟

計畫參與人員：碩士班研究生-兼任助理人員：何君豪
碩士班研究生-兼任助理人員：鄭人豪
碩士班研究生-兼任助理人員：呂明欣
碩士班研究生-兼任助理人員：林仁祥
碩士班研究生-兼任助理人員：張智傑
碩士班研究生-兼任助理人員：賴敏華
碩士班研究生-兼任助理人員：藍家樑

報告附件：出席國際會議研究心得報告及發表論文

處理方式：本計畫涉及專利或其他智慧財產權，2 年後可公開查詢

中華民國 98 年 01 月 26 日

行政院國家科學委員會補助專題研究計畫 ■ 成 果 報 告
□期中進度報告

貝氏網路與分類技術之基礎研究與應用：
建構學生學習歷程之模型與語意標記

計畫類別：■ 個別型計畫　　□ 整合型計畫
計畫編號：NSC－95－2221－004－003－MY2
執行期間：95 年 8 月 1 日 至　97 年 10 月 31 日

計畫主持人：劉昭麟
共同主持人：
計畫參與人員：碩士班研究生：何君豪、鄭人豪、呂明欣、林仁祥、
張智傑、賴敏華、藍家樑
博士班研究生：無
成果報告類型(依經費核定清單規定繳交)：□精簡報告　■完整報告

本成果報告包括以下應繳交之附件：
□赴國外出差或研習心得報告一份
□赴大陸地區出差或研習心得報告一份
■出席國際學術會議心得報告及發表之論文各一份
□國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、
列管計畫及下列情形者外，得立即公開查詢
■涉及專利或其他智慧財產權，□一年■二年後可公開查詢

執行單位：國立政治大學資訊科學系

中　華　民　國　　98 年　　1 月　　20 日

中文摘要

本研究案主要著力於學生學習模型、電腦輔助試題翻譯、電腦輔助語文教學和中文訴訟文書分類四個研究主題。在學生學習模型方面，我們以貝氏網路來表示學生學習模型，並且提出了一個方法來學習學生的學習模型的方法。在電腦輔助試題翻譯這一項工作中，我們建構了一個實際的系統，用以輔助專家翻譯 TIMSS 試題。在電腦輔助語文教學這一項工作中，我們建構了一個真實的系統，可以輔助國語科教師編輯試題。中文訴訟文書分類並不是這一次研究案的主角，是我們結束前一國科會研究案的工作。本次研究計畫執行期間，合計發表 17 篇論文（兩篇國際期刊論文、三篇國際學術研討會論文國內學術會議方面，則有六篇 ROCLING 論文、四篇 TAAI 論文、一篇 NCS 和一篇 TANET 論文），總頁數達到 133 頁；其中包含一篇人工智慧與電腦輔助教學跨領域研究的優質期刊論文(IJAIED)和一篇計算語言學優質研討會(ACL)的研討會論文。

關鍵詞：貝氏網路、學生學習歷程、建模技術、資訊檢索、電腦輔助語文學習、機器翻譯

Abstract

In this report, we summarize the results of this research project on several fronts. For student modeling, we proposed a simulation-based approach to learn the structures of Bayesian networks that contain unobservable variables. We have built three functioning systems for practical applications of natural language processing techniques. We built an environment for computer-assisted translation of TIMSS test items, an environment for assisting teachers to compose test items for elementary Chinese, and an environment for searching Chinese indictment documents.

Keywords: Bayesian networks, structure learning, learning processes of composite concepts, information retrieval, computer assisted language learning, machine translation

目錄

報告內容

前言

本研究案雖然不是一個整合型計畫，但是一開始即訂定多項目標，因此實際上很難以一份報告來總結所有子項計畫的研究成果。因此，在這一份報告中，我們為個別子項計畫撰寫簡要的文字資料，然後請有興趣深入研究的讀者繼續研讀已經發表的期刊論文或者學術會議論文。

這一個研究案從事兩大類但相互關連的研究工作：一個是認知歷程的建模技術，另一個則是以自然語言處理為基礎的實際軟體系統的建置。認知歷程的建模技術方面，我們以學生學習綜合觀念的問題作為研究的主題。實際軟體系統方面，我們建立了三個不同的系統：中文訴訟文書檢索系統、電腦輔助 TIMSS 試題翻譯環境和電腦輔助國語科試題出題環境。

在實際的環境中，如果我們想要利用人工智慧技術來讓軟體系統提供使用者最好的服務，瞭解使用者真實的需求是必要的基礎。實務上我們很難經常性地詢問使用者的需求和回饋，因此從間接的資訊來推測使用者的興趣或者意向是重要的基礎技術。所以，以上兩大類的研究工作，以長遠的角度來說是有密切關連的。現階段的工作是一個逐漸打底的工作；我們期待繼續朝綜合人工智慧技術、機器學習技術和自然語言處理技術來建構有用的資訊檢索環境和電腦輔助語文學習的環境。

在研究進度方面，計畫主持人全力從事認知歷程的建模技術，因此這一部分的成果比較能夠掌握。應用自然語言處理技術來建立實際系統的部分，則全部是以碩士班研究生執行，雖然能夠維持一些進度，但是計畫推展的速度並不能令人完全滿意。

在研究成果方面，我們發表了 17 篇學術論文，總頁數達到 133 頁。在研究成果小節中我們將分析所達成的成果。我們把各項主要子項工作比較具有代表性的論文附在本份報告的附錄中。就如前面所說明，這一份報告的本身其實只能是我們所進行的所有工作的大摘要而已，所有工作的真正成果已經反映在所發表的論文之中，因此雖然我們必須把論文放在附錄，但是其實論文本身才應該是這一個研究案的成果的真正主角。

附錄包含了四篇論文：*IJAIED* 的期刊論文一篇（建模技術相關論文，**這是一篇出版商有版權的文章，不宜在網路上公開**），ACL 國際學術研討會論文一篇（電腦輔助國語科試題出題輔助系統）， ROCLING 國內學術研討會論文一篇（電腦輔助 TIMSS 試題翻譯環境）和 TAAI 國內學術研討會論文一篇（中文訴訟文書檢索系統）。

研究目的

我們分四個段落簡述四個不同的子項目的研究目的。詳細資料請參閱相關論文。

在建立使用者模型方面，我們希望能夠找到一個好的辦法，讓我們可以在不能夠直接觀測模型中所有相關變數的狀態的情形之下，仍然能夠以貝氏網路來表示所有相關變數的直接和間接機率關係。在所進行的研究中，學生的答題的反應（目前僅以「對」和「錯」表示）是可以直接觀測的變數，而我們所建立的模型包含了學生對於個別觀念的能力。能力與答題的對錯雖然有密切關係，但是關係卻不是邏輯式的，因為有人會因為運氣好答對

題目，也有人會因為一時疏忽等複雜原因，在有相關能力的情形之下，卻沒有答對題目。簡單地說，本項研究是要以學生的答題的對錯來反推學生的學習模式的貝氏網路。

在中文訴訟文書檢索系統中，我們採用了幾種資訊檢索和人工智慧的分類、分群的技術來輔助專業和非專業法學人士來檢索以中文撰寫的地方法院訴訟文書。對於檢索者而言，我們希望能夠提高相關判例的檢索效率，同時這一系統也希望能夠有助於專業人士檢索相關刑事案件的判刑刑度，藉此希望有助於法院判決的一致性。

電腦輔助TIMSS試題翻譯環境的研究，同樣也是結合人工智慧與自然語言處理的應用研究，目的是協助TIMSS試題的翻譯。TIMSS試題的原文是以英文撰寫的國際標準試題，測驗的目的是要評比參與TIMSS計畫的各個國家的科學數理的教學成效。我國參與TIMSS計畫，因此須要把TIMSS試題翻譯為中文試題，好讓我國四年級和八年級（國中二年級）的學生受測。我們建構了一個環境，希望能協助負責翻譯試題的專家，能夠以較低的時間代價從事符合翻譯準則的翻譯工作。

電腦輔助國語科試題出題環境則是利用自然語言處理技術，協助國語科或者華語教師編輯與華語學習相關的試題，好讓教師能夠透過網路從事測驗。這一個系統同時包含了試題編輯、題庫管理、網路施測和測後分析等功能。試題的類型則包含的漢語語音辨識、改錯字試題、中文克漏詞(cloze)、中文量詞和句子重組五個題型。

文獻探討

由於前述的四大項研究各有自己相關的文獻，因此無法在一篇報告中簡單地整合。除了因為研究方向的重要差別，另外也因為相關文獻的量的關係，請有興趣的讀者與評審參閱個別論文中的相關文獻探討的資料。

研究方法

我們分四個段落簡述四個不同的子項目的研究方法。詳細資料請參閱相關論文。

在建立使用者模型方面，我們首先建立一般適性化教學研究所依賴的模型，利用這樣的模型來產生模擬的學生答題表現。有了答題表現的資料，我們才能進行下一步研究。在研究中，我們比較了以經驗法則(heuristics)、類神經網路(artificial neural networks)和支持向量機(support vector machines)所建構的分類器等技術來猜測先前用以產生模擬的學生資料時所使用的貝氏網路模型。除了利用經驗法則來猜測的方法之外，我們須要利用監督式學習法(supervised learning)來訓練類神經網路模型和支持向量機模型，這時我們假設有專業的猜測，讓我們得以限縮所欲尋找的模型的範圍。實驗中，我們假設了學生的答題反應跟其真實能力，只會呈現機率式的關連性，同時操弄這一關連性的不確定性，來研究經驗法則、類神經網路和支持向量機所建構的分類器，在不同的程度的不確定性關連下所能達成的正確性。

在中文訴訟文書檢索系統中，除了典型的inverted indexing之外，我們利用更多的自然語言處理技術，建構不同的管道來協助查詢者找到有用的資料。這其中跟語意比較相關的是我們採用了詞組(term pairs)為基礎的分群機制，讓我們來評比訴訟文書的相關度直覺上來說。以詞組為檢索機制，比較能夠彰顯詞彙的語意。此外，我們也利用詞彙的同現(collocation)

來導引建議檢索檔案。跟我們以詞組為基礎來做檔案分群的理念相似，以同現的分數高低來建議檢索資料，也可能因為比較能夠捕捉到檢索者的意圖而提高檢索效率。

電腦輔助 TIMSS 試題翻譯環境的建置是一個典型的機器翻譯(machine translation)的研究。對於機器翻譯這個研究議題來說，兩年的計畫時程只能建立基礎而已。我們應用語言模型(language models)、雙語對譯資料(parallel corpora)、範例式學習技術(example-based learning)三個主要技術，結合現在受到學界普遍使用的 Moses 和 Lucene 開放式軟體工具建立了一個翻譯輔助環境。本研究案，受到國立台灣師範大學科學教育中心的張主任的協助，因此得以獲得相關的 TIMSS 中英文試題。

電腦輔助國語科試題出題環境提供五大類型試題的編輯：漢語語音辨識、改錯字試題、中文克漏詞、中文量詞和句子重組。因此我們須要利用到語音、漢字構形、漢語詞彙和漢語語法等數個不同層次的語文資訊。我們利用自然語言處理技術，依照試題編輯者（通常是教師）所要求的試題條件，從所蒐集的語文資料找到相關的語料，並且依照所編輯的試題的特性提出有用的建言。試題編輯者可以利用我們的介面建立基本的題庫，進而建立試卷資料庫，爾後學生也可以透過網路作答。學生作答的結果可以立即得到回饋，教師也可以分析所任課的學生群的測驗結果，檢討其教學策略。

研究成果與討論

我們分別簡述四個不同的子項目的研究成果。詳細資料（特別是個別研究的學術意義）請參閱相關論文中比較詳細的討論。

在建立使用者模型方面，我們在 *International Journal of Artificial Intelligence in Education* (*IJAIED*) 發表了一篇 49 頁的長篇論文[1]，在這之前，我們在全國計算機會議發表了一篇中文論文[12]為國內學者介紹這一個研究的縮影 。*IJAIED* 是一個優質的期刊，是 International AIED Society 的正式期刊，由 University of Edinburgh 的教授擔任主編，一年一般只收錄十餘篇論文，其中部分還是兩年一次的 AIED 學術研討會的最佳論文才能獲得推薦。因此研究成果能夠在 *IJAIED* 刊登，應該算是相當不容易的一項成就。

在中文訴訟文書檢索系統方面，我們在 2007 年和 2008 年的人工智慧學會年會(TAAI)發表了三篇論文[6, 13, 14]。

電腦輔助 TIMSS 試題翻譯環境的建置方面，我們在 *Journal of Advanced Computational Intelligence and Intelligent Informatics* (*JACIII*)發表了一篇簡短的期刊論文[2]，在 RANLP 國際學術研討會中發表了一篇論文[5], 在 2007 年和 2008 年的計算語言學研討會(ROCLING)上各發表了一篇論文[10, 17]。

因為所牽涉的問題，不僅僅是資訊科學的技術，同時還有關於教學的可能成效，因此電腦輔助國語科試題出題環境的研究成果，部分是發表在比較接近教育領域的會議中，去接受第一線的使用者的挑戰。這一方面的部分成果發表於 *JACIII* 期刊論文[2]，2008 年的 ACL 國際學術研討會[3]，2008 年的 CAERDA 學術研討會[4]，2007 年的 RANLP 國際學術研討會[5]，兩篇 2008 年的計算語言學研討會(ROCLING)[8, 9]和一篇 2007 年的網際網路研討會(TANET)[15]。ACL 是國際間計算語言學界最著名的國際學術研討會之一，研究成果能夠獲得 ACL 年會收錄，是一項不錯的成就。

除了本份報告目前所報告的四項研究子項目之外，我們這一個研究計畫還做了一些嘗試性質的研究，這一些嘗試性的研究偶而也有一些零星的論文發表。在研究生方面，這兩年期間，有一為研究生曾經探討利用文件分類的技術來猜測新聞報導與股價漲跌趨勢的可能關係[16]，另有一位研究生探討利用文件內容的分析技術，來為研討會投稿論文找尋合適的論文評審委員[11]，這兩項研究經驗都發表在 ROCLING 研討會。此外，我們也有一位大學部同學利用機器學習技術的觀念，發展出一個可以提供任意形狀棋盤的黑白棋(Reversi)服務的軟體服務[7]，這一向研究成果則發表於 TAAI 研討會。

論文列表

以下是因本項研究案所得以發表的學術論文清單

1. Chao-Lin Liu. A simulation-based experience in learning structures of Bayesian networks to represent how students learn composite concepts, *International Journal of Artificial Intelligence in Education*, **18**(3), 237−285. IOS Press, The Netherlands, September 2008.
2. Ming-Shin Lu(呂明欣), Yu-Chun Wang(王昱鈞), Jen-Hsiang Lin(林仁祥), Chao-Lin Liu, Zhao-Ming Gao(高照明), and Chun-Yen Chang(張俊彥). Supporting the translation and authoring of test items with techniques of natural language processing, *Journal of Advanced Computational Intelligence and Intelligent Informatics*, **12**(3), 234−242. Fuji Technology Press, Japan, May 2008.
3. Chao-Lin Liu and Jen-Hsiang Lin(林仁祥). Using structural information for identifying similar Chinese characters, *Proceedings of the Forty Sixth Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (ACL'08), short paper, 93−96. Columbus, Ohio, USA, 2008.
4. Chao-Lin Liu, Jen-Hsiang Lin(林仁祥), and Chih-Bin Huang(黃志斌). A platform for authoring test items for elementary Chinese with techniques of natural language processing, presented in the 2008 CAERDA International Conference (Chinese American Educational Research and Development Association). New York, New York, USA, 2008.
5. Ming-Shin Lu(呂明欣), Jen-Hsiang Lin(林仁祥), Yu-Chun Wang(王昱鈞), Zhao-Ming Gao(高照明), Chao-Lin Liu, and Chun-Yen Chang(張俊彥). Prototypes of using NLP techniques for assisting translation and authoring of test items, *Proceedings of the Workshop on NLP for Educational Resources*, International Conference on Recent Advances in Natural Language Processing 2007 (RANLP'07), 1−6. Borovets, Bulgaria, 2007.
6. 藍家良、賴敏華、田侃文及劉昭麟。訴訟文書檢索系統，*第十三屆人工智慧與應用研討會論文集* (TAAI'08)，305−312。2008 年。
7. 林正宏及劉昭麟。任意棋盤的 Othello 遊戲，*第十三屆人工智慧與應用研討會論文集* (TAAI'08)，443−449。2008 年。
8. 劉昭麟、黃志斌、翁睿妤及莊怡軒。形音相近的易混淆漢字的搜尋與應用，*第二十屆自然語言與語音處理研討會論文集* (ROCLING XX)，108−122。2008 年。
9. 賴敏華及劉昭麟。電腦輔助中學程度漢英翻譯習作環境之建置，*第二十屆自然語言與語音處理研討會論文集* (ROCLING XX)，293−307。2008 年。

10. 張智傑及劉昭麟。以範例為基礎之英漢 TIMSS 試題輔助翻譯，*第二十屆自然語言與語音處理研討會論文集* (ROCLING XX)，308−322。2008 年。
11. 陳禹勳及劉昭麟。電腦輔助推薦學術會議論文評審委員之初探，*第二十屆自然語言與語音處理研討會論文集* (ROCLING XX)，323−337。2008 年。
12. 劉昭麟。利用試題反應建立學生學習歷程模型的一些經驗，*中華民國九十六年全國計算機會議論文集* (NCS’07)，第一冊(下)，359−366。2007 年。
13. 何君豪、鄭人豪及劉昭麟。階層式分群法在民事裁判要旨分群上之應用，*第十二屆人工智慧與應用研討會論文集* (TAAI’07)，794−800。2007 年。
14. 鄭人豪及劉昭麟。中文詞彙集的來源與權重對中文裁判書分類成效的影響，*第十二屆人工智慧與應用研討會論文集* (TAAI’07)，801−808。2007 年。
15. 林仁祥及劉昭麟。國小國語科測驗卷出題輔助系統，*2007 臺灣網際網路研討會論文集* (TANET’07)，論文光碟。2007 年。
16. 陳俊達、王台平及劉昭麟。以文件分類技術預測股價趨勢，*第十九屆自然語言與語音處理研討會論文集* (ROCLING XIX)，347−361。2007 年。
17. 呂明欣、高照明、劉昭麟及張俊彥。針對數學與科學教育領域之電腦輔助英中試題翻譯系統，*第十九屆自然語言與語音處理研討會論文集* (ROCLING XIX)，407−421。2007 年。

計畫成果自評

這一項研究計畫歷時兩年，原本的研究目標包含兩大方向，一個是模型建立技術的研究，另一個則是與自然語言處理相關的研究。在這兩年之中，我們合計發表兩篇期刊論文，三篇國際學術研討會論文和 12 篇國內學術會議論文。

在建立模型技術的研究方面，我們覺得有很值得自豪的成就，能夠在 *International Journal of Artificial Intelligence in Education* (*IJAIED*) 發表長篇論文。*IJAIED* 是 AIED 學會的代表期刊，而 AIED 的學術研討會和 ITS 學術研討會則是電腦輔助教學兩大旗艦級的國際學術研討會。部分的 *IJAIED* 論文還是從 AIED 兩年一次的國際學術會議中精選而得的 (ITS 也是兩年一次的國際學術會議)。因此，我們主觀地相信以兩年多的努力來換取一篇 *IJAIED* 的論文是一項值得的投資。

相對之下，自然語言處理相關的研究的學術成果則顯得較為薄弱，由於研究計畫的規模和過去兩年的兼任研究助理都還是只有由碩士班研究生來擔任，因此只能建立一些基礎的經驗，僅僅在發表論文的數量和研究廣度上做努力。我們在電腦輔助法學資訊檢索，電腦輔助機器翻譯和電腦輔助國語科試題編輯三個方面，都建置了真實可以在網路上使用的軟體，除了為實驗室建立一些可用的軟體工具，為更深層研究建立基礎之外，最明顯可見的成果可能是在於訓練可以進入職場的資訊科技人才。

附錄

本附錄依序包含下列四篇論文。

1. Chao-Lin Liu. A simulation-based experience in learning structures of Bayesian networks to represent how students learn composite concepts, *International Journal of Artificial Intelligence in Education*, **18**(3), 237−285. IOS Press, The Netherlands, September 2008.
2. Chao-Lin Liu and Jen-Hsiang Lin(林仁祥). Using structural information for identifying similar Chinese characters, *Proceedings of the Forty Sixth Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (ACL'08), short paper, 93−96. Columbus, Ohio, USA, 2008.
3. 張智傑及劉昭麟。以範例為基礎之英漢 TIMSS 試題輔助翻譯，*第二十屆自然語言與語音處理研討會論文集* (ROCLING XX)，308−322。2008 年。
4. 藍家良、賴敏華、田侃文及劉昭麟。訴訟文書檢索系統，*第十三屆人工智慧與應用研討會論文集* (TAAI'08)，305−312。2008 年。

A Simulation-Based Experience in Learning Structures of Bayesian Networks to Represent How Students Learn Composite Concepts

Chao-Lin Liu, *Department of Computer Science, National Chengchi University, Taiwan*
chaolin@nccu.edu.tw

Abstract. Composite concepts result from the integration of multiple basic concepts by students to form high-level knowledge, so information about how students learn composite concepts can be used by instructors to facilitate students' learning, and the ways in which computational techniques can assist the study of the integration process are therefore intriguing for learning, cognition, and computer scientists. We provide an exploration of this problem using heuristic methods, search methods, and machine-learning techniques, while employing Bayesian networks as the language for representing the student models. Given experts' expectation about students and simulated students' responses to test items that were designed for the concepts, we try to find the Bayesian-network structure that best represents how students learn the composite concept of interest. The experiments were conducted with only simulated students. The accuracy achieved by the proposed classification methods spread over a wide range, depending on the quality of collected input evidence. We discuss the experimental procedures, compare the experimental results observed in certain experiments, provide two ways to analyse the influences of *Q-matrices* on the experimental results, and we hope that this simulation-based experience may contribute to the endeavours in mapping the human learning process.

Keywords. Student Modelling; Learning Patterns; Bayesian Networks; Computer-Assisted Cognitive Modelling; Computer-Assisted Learning; Machine Learning

INTRODUCTION

Obtaining good student models is crucial to the success of computer-assisted learning. Relying on student models, computerised adaptive testing systems (CATs) may assess students' competence levels more efficiently than traditional pen-and-paper tests by adaptively selecting and administering appropriate test items for individual students (van der Linden & Glas, 2000). If, in addition, a model captures how students learn, then we may apply the model for computer assisted instruction and testing (Nichols et al., 1995; Leighton & Gierl, 2007). For instance, by introducing prerequisite relationships in a refined model, Carmona et al. (2005) showed that there is room for boosting the efficiency of CATs. In this paper, we adopt Bayesian networks (Pearl, 1988; Jensen & Nielsen, 2007) as the language to represent student models, and discuss a simulation-based experience in which we attempted to learn student models with machine-learning techniques based on students' responses to test items. The simulation-based results indicate how and when we can learn students' learning patterns from their item responses, and shed light on some difficulties that we may encounter in similar studies that use the item responses of real students.

Measuring students' competence levels with their responses to test items is a typical problem of uncertain reasoning in CATs. The *slip* and *guess* cases are two frequently mentioned sources of uncertainty, e.g., (VanLehn et al., 1994; Millán & Pérez-de-la-Cruz, 2002). Students may accidentally fail to respond to test items correctly (the *slip* case), or they may just be lucky enough to guess the correct answers to the test items (the *guess* case). Students may also make mistakes intentionally (Reye, 2004). Due to such an uncertain correspondence between students' mastery levels and item responses, researchers and practitioners have applied probability-based methods for student assessment (Mislevy & Gitomer, 1996). Vos (2000) and Vomlel (2004), for instance, showed that probability-based procedures offer chances for teachers to correctly identify students' mastery levels with a fewer total number of test items in tests of variable length.

In recent years, Bayesian networks have offered a convenient computational tool for implementing the probability-based testing procedures and also for cognitive and developmental psychology (Glymour, 2003). Martin and VanLehn (1995) and Mislevy and Gitomer (1996) studied the applications of Bayesian networks for student assessment. Mayo and Mitrovic (2001) conducted a survey of this trend and applied decision theories to optimise their systems for intelligent tutoring. Conati et al. (2002) applied Bayesian networks to both assessing students' competence and recognising students' intention. The research on applications of Bayesian networks in CATs also led to real world performing systems, e.g., SIETTE (Conejo et al., 2004; Guzmán et al., 2007b).

To apply Bayesian networks in an inference task, we need the network structure and the conditional probability tables (CPTs) that implicitly specify the joint probability distribution of all of the variables of interest. Just as we have to learn model parameters when we apply the Item Response Theory (van der Linden & Hambleton, 1997) in CATs, we have to learn the CPTs for Bayesian networks (Mislevy et al., 1999) from students' records, while experts often provide specifications of the network structures. The network structure essentially portrays the structure of the knowledge of the students in the study, and has an influence on the ways in which the decision mechanisms in CATs make inferences about students' mastery levels.

Not surprisingly, researchers have explored different network structures in which the nodes for the variables were organised in different styles. For instance, Millán and Pérez-de-la-Cruz (2002) categorised nodes in their multi-layer Bayesian networks into four types: *subjects*, *topics*, *concepts*, and *questions*. Reye (2004) employed nodes that represented students' competence as the backbone of the network, and associated a uniform substructure with each node on the backbone to assist the process of making inferences about students' competence. Despite the differences in the network structures, both studies emphasised the importance of modelling the prerequisite relationships among the learning targets. Carmona et al. (2005) reported that adding prerequisite relationships in Bayesian networks helped reduce test lengths in CATs. In addition to utilising different categories of variables, researchers may choose to let the nodes for concepts be parent nodes of nodes for test items, or the other way around. Mislevy and Gitomer (1996) and Millán and Pérez-de-la-Cruz (2002) discussed the implications of the different choices which can be made in the directions of the links.

Although the majority of the CAT research community rely on experts to provide network structures, it is conceivable that we may learn the network structures from students' records using the machine learning techniques for Bayesian networks (Heckerman, 1999; Jordan, 1999; Neapolitan, 2004). Vomlel (2004) attempted to apply a variant of the PC-algorithm (Spirtes et al., 2000) that was implemented in Hugin (http://www.hugin.dk) to learn network structures, and augmented the networks with hidden variables based on experts' knowledge. Recently, Desmarais et al. (2006) learned item-to-item knowledge structures from students' records, and compared the learned structures with those reported in (Vomlel, 2004). The item-to-item knowledge structures are special in that the states of all of the nodes in the networks are directly observable, making the learning of the network structures a relatively practical matter. The experience indicates that it is an interesting but challenging task to learn the network structures from scratch in the cases that there are many hidden variables, due in part to the large number of candidate network structures.

We approach the structure learning problem from a different perspective. Instead of trying to learn student models from scratch, we propose methods for helping experts select models that differ in subtle ways. This can be helpful for constructing student models for how students learn *composite concepts*. Assume that it requires knowledge of four *basic concepts*, say *cA*, *cB*, *cC*, and *cD*, to learn a composite concept *dABCD*. In this case, will we be able to tell whether students manage to learn *dABCD* by directly integrating *cA*, *cB*, *cC*, and *cD* or whether they first integrate *cA*, *cB*, and *cC* into an intermediate product and then integrate this intermediate product with *cD*? To what extent can the use of machine learning techniques help us to identify the direct prerequisites necessary for the production of the composite concept?

We explore methods to answer this question by expressing the problem with Bayesian networks and by learning the network structures based on students' responses to test items. Although there are various methods for learning Bayesian networks (Heckerman, 1999; Neapolitan, 2004), our learning problem is distinct. We face a problem of learning the structure of hidden variables because we cannot directly observe students' competence levels of the concepts. The students' item response patterns that we can observe and collect have only an indirect and uncertain relationship with students' actual competence patterns, which is a challenge that has long been discussed in the literature on CATs, e.g., (Martin & VanLehn, 1995; Mislevy & Gitomer, 1996). Although the states of the hidden nodes for the competence levels can only be inferred indirectly, we are sure of the existence of the hidden nodes, so our focus is to learn the structure that relates the hidden variables. Finally, for any practical problems that involve three or more basic concepts, there are at least four hidden variables in question, making the target problem nontrivial.

In order to explore the effectiveness of different computational techniques for the target problems, we employ the device of simulated students which has been used in many studies on methodologies for intelligent tutoring systems, e.g., (VanLehn et al., 1994; Vos, 2000; Mayo & Mitrovic, 2001; Millán & Pérez-de-la-Cruz, 2002; Liu, 2005; Desmarais et al., 2006; Matsuda et al., 2007). We generated the item responses of the students that were simulated with a specific Bayesian network whose structure encoded beliefs about how students learned composite concepts. We could control the degree of uncertainty in the relationship between the item responses and the mastery levels by adjusting the simulation parameters. Hiding the original Bayesian network, we applied mutual information (MI) (Cover & Thomas, 2006), search-based methods, artificial neural networks (ANNs) (Bishop, 1995), and support vector machines (SVMs) (Cortes & Vapnik, 1995) to analyse students' item responses to determine the structure of the original network.

We report experimental results and discuss observations that are potentially useful for further studies. The quality of the predictions that are made by our classifiers depends on many factors, e.g., the algorithms that we used to guess the network structures, the degree of uncertainty in the relationships between the students' competence levels and the item responses, and the quality of the training data for the machine-learning algorithms. On average, using SVMs as the underlying classification mechanism offers the best performance and efficiency, when training data of good quality is available. Experimental experience provides hints on the principles that are useful for guiding the designs of further studies. More specifically, we identify some methods for determining the quality of training data, provide two analytical methods for comparing the influences of *Q*-matrices on the experimental results, and report situations when different classification methods may offer better performance. Specific details will be discussed in appropriate sections.

We define the target problems and provide background information in Preliminaries[†], discuss the applications of mutual information, search-based methods, artificial neural networks, and support vector machines to the problems in Methods for Model Selection, and present the design of experiments in Design of the Experiments. In Idealistic Evaluations, we evaluate and compare the effects of the proposed methods under different combinations of *slip*, *guess*, and *Q*-matrices, when the quality of training data is good. In More Realistic Evaluations, we investigate the results of experiments under different combinations of *slip*, *guess*, and *Q*-matrices, when the quality of training data is relatively poor. Finally, we summarise the implications of the simulation results and review more relevant literature in Summary and Discussion.

PRELIMINARIES

We outline the nature of the problems that we would like to solve in the first subsection, and explain how we formulate the target problems with Bayesian networks in the second subsection. Using Bayesian networks as the representation language, we provide a more precise definition of the target problem in the third subsection, show how we simulate students' item responses in the fourth subsection, look into the issue about computational complexity in the fifth subsection, and illustrate the difficulty of solving the target problems with existing software in the last subsection.

The Simulated World

We consider a set of concepts $\breve{C}$ and an item bank $\Im$ that contains test items for $\breve{C}$. Some concepts in $\breve{C}$ are **basic** and others are **composite**. Learning a composite concept requires the students to integrate their knowledge about certain basic concepts. A composite concept, say *dABC*, is the result of integrating knowledge about basic concepts *cA*, *cB*, and *cC*. Let $\breve{C}$ contain *n* concepts, i.e., $\breve{C} = \{C_1, C_2, \cdots, C_n\}$. For each concept $C_j \in \breve{C}$, we have a subset $\Im_j = \{I_{j,1}, I_{j,2}, \cdots, I_{j,m_j}\}$ in $\Im$ for testing students' competence in C_j. For easier reference, we call C_j the **parent concept** of the items in $\Im_j$. The concepts that students *directly* integrate to form a composite concept C_k are also referred as the parent concepts of C_k. Based on this definition, a prerequisite concept is not necessarily a parent concept of a composite concept. More specifically, *cA* and *cB* are *not* parent concepts of *dABC* when, for instance, students learn *dABC* by integrating *dAB* and *cC*, although *cA* and *cB* must be prerequisites of *dABC*. We refer to a student's competence in the concepts being studied as a **competence pattern**, and assume that students demonstrate special patterns in their competence. Students that share the same competence patterns form a **subgroup**.

We employ the convention of the *Q-matrix*, originally proposed to represent the relationships between concepts and test items (Tatsuoka, 1983), for the encoding of the competence of a subgroup in the basic concepts and also in being able to integrate the parent concepts into composite ones. In Table 1, there are two *Q*-matrices that are separated by the double bars, and the "SID" column shows the identification of the subgroups. We will use these *Q*-matrices in the experiments reported in Idealistic Evaluations and More Realistic Evaluations. Let $q_{j,k}$ denote the cell at the j^{th} row and the k^{th} column in a *Q*-matrix. If C_k is a basic concept, we set $q_{j,k}$ to 1 when students of the j^{th} subgroup has the competence in C_k; if C_k is a composite concept, we set $q_{j,k}$ to 1 when students of the j^{th} subgroup has the ability to integrate all of the parent concepts of C_k. Hence, if the k^{th} concept is composite, the j^{th} subgroup is competent in the concept only if $q_{j,k} = 1$ *and* the j^{th} subgroup is competent in all of the parent concepts of the k^{th} concept. Based on this definition, $q_{j,k}$ is related to both the *rule nodes* and the *rule application nodes* that are defined by Martin and VanLehn (1995).

Table 1. Competence patterns in two *Q*-matrices

SID	Competence in (integrating) concepts													
	cA	*cB*	*cC*	*dAB*	*dBC*	*dAC*	*dABC*	*cA*	*cB*	*cC*	*dAB*	*dBC*	*dAC*	*dABC*
g_1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
g_2	1	1	1	0	1	1	1	1	1	1	1	1	0	1
g_3	1	1	1	1	1	0	1	1	1	1	1	0	1	1
g_4	1	1	1	1	0	1	1	1	1	1	1	0	0	1
g_5	0	1	1	0	1	0	1	1	1	1	0	1	1	1
g_6	1	1	0	1	0	0	1	1	1	1	0	1	0	1
g_7	1	0	1	0	0	1	1	1	1	1	0	0	1	1
g_8	1	1	1	0	0	0	1	1	1	1	0	0	0	1

The competence patterns, which are used in our simulations, are not as deterministic as they appear. In the simulations, we intentionally introduce some degrees of uncertainty to reflect the possibility that teachers may not categorise the subgroups precisely. This is similar to the concept of *residual ability* discussed in (DiBello et al., 1995, page 362). We will go further into this issue when we present our simulator in Generating Student Records.

As discussed in (DiBello et al., 1995, pages 365 and 370), we can apply *Q*-matrices in different ways, depending on the interpretation of the rows and columns. In addition, the contents of the matrices can differ in a wide variety of ways, and, consequently, researchers can report results of experiments using a selected number of *Q*-matrices typically. Different choices of the *Q*-matrices certainly influence the results of our experiments, and we will discuss this issue shortly.

Example 1. In the *Q*-matrices shown in Table 1, we assume that students form only eight subgroups, although there could be 2^7 subgroups in a problem that includes seven concepts. The competence pattern for the subgroup g_8 in the left *Q*-matrix is {1, 1, 1, 0, 0, 0, 1}. By adopting the left *Q*-matrix, we assume that a typical student in g_8 should be competent in all basic concepts, should be able to integrate the parent concepts for *dABC*, but cannot integrate the parent concepts for *dAB*, *dBC*, and *dAC* at the time of the experiments. ■

A Formulation with Bayesian Networks

We choose to use Bayesian networks to represent student models, because Bayesian networks are a popular choice for researchers to capture the uncertain relationship between students' performance and their competence in many research projects, e.g., (VanLehn et al., 1998; Mislevy et al., 1999, Millán & Pérez-de-la-Cruz, 2002; Reye, 2004; Vomlel, 2004; Carmona et al., 2005; Chang et al., 2006; Almond, 2008). We employ nodes in Bayesian networks to represent students' competence in concepts and the correctness of their responses to test items. For easier recognition, we use the names of the concepts as the names of the nodes that represent the concepts. The names of the nodes that represent the correctness of the item responses are in the form of $iX\alpha$, where *i* denotes *item*, *X* is the name of the parent concept, and α is the identification number of the test item. When there is no risk of confusion, we refer to the nodes that represent concepts simply as the concepts and the nodes that represent test items simply as test items. Hence, in Figure 1, we have seven different concepts—three basic ones (*cA*, *cB*, and *cC*) and four composite ones (*dAB*, *dBC*, *dAC*, and *dABC*). As a simplifying assumption, each simulated student will respond to three test items designed for every concept. For instance, $\Im_{cA} = \{iA1, iA2, iA3\}$, and *iA1*, *iA2*, and *iA3* are test items for *cA*.

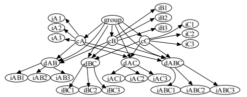


Fig. 1. A complete Bayesian network

All nodes are dichotomous in our simulation, except for the *group* node. In all simulations, *group* will be used as a special node that represents the student subgroups, and it can have such values as g_1, g_2, …, and g_γ, where γ depends on the design of the simulations. Nodes representing competence levels may have either **competent** or **incompetent** as their values, and nodes representing item responses may have either **correct** or **incorrect** as their values.

The links in a Bayesian network signify direct relationships between the connected nodes, and the nodes that are not directly connected are conditionally independent (Pearl, 1988; Jensen & Nielsen, 2007). There are no strict rules governing the directions of the links in Bayesian networks, except that a valid Bayesian network must not contain any directed cycles and that it is recommended that we follow the causal directions in model construction (Russell & Norvig, 2002). The literature has discussed the implications of different choices of the directions of the links for CATs, e.g., (Mislevy & Gitomer, 1996; Millán & Pérez-de-la-Cruz, 2002; Glymour, 2003; Liu, 2006d). We employ the most common choices, and discuss relevant issues in Impacts of Latent Variables and Summary and Discussion. As a result, links point from the parent concepts to the integrated concepts and from the parent concepts to their test items.

In Figure 1, the values of *group* come from the set of possible student subgroups. If we use either of the two *Q*-matrices in Figure 1, *group* will have eight possible values, each denoting a possible student group. Since the subgroup identity of a student affects the competence pattern, there are direct links from *group* to all concept nodes.

We defer the discussion of how we set the contents of the conditional probability tables to Generating Student Records.

[†] We use the font of **Helvetica** for section headings to avoid the need to use numbered section headings.

The Target Question and Assumptions

Our target problem is to learn how students learn composite concepts by observing students' *fuzzy* (Birenbaum et al., 1994) item-response patterns that have only an indirect relationship with their competence patterns. Students' item responses are fuzzy because they do not necessarily indicate students' actual competence.

A composite concept is a concept that requires the knowledge of two or more basic concepts. For instance, Mislevy and Gitomer (1996) used "Mechanical Knowledge", "Hydraulics Knowledge", "Canopy Knowledge", and "Serial Elimination" as the prerequisites for "Canopy Scenario Requisites--No Split Possible", and Vomlel (2004) included "Subtraction", "Cancelling Out", and "Multiplication" as the basic capabilities that are necessary for finding the solution for $(\frac{3}{4}\times\frac{5}{6})-\frac{1}{8}$.

Although it is convenient to use the nodes for all the prerequisites as the parent nodes of the node for the composite concept, we anticipate that constructing a more precise model that reflects the process of the learning of the composite concept may improve the performance of CATs and other computer-assisted learning tasks. This anticipation is related to the study of cognitive diagnostic assessment (Nichols et al., 1995; Leighton & Gierl, 2007). Indeed, Carmona et al. (2005) report that introducing prerequisite relationships into their multi-layered Bayesian student models enables their CAT system to diagnose students with a fewer number of test items. Furthermore, if teachers know how students normally learn a composite concept, the teachers will have more information as to how to provide appropriate and specific help for students who fail to demonstrate competency in the concept (Naveh-Benjamin et al., 1995). For instance, if students normally learn *dABC* by integrating *cA* and *dBC* and if a student shows a lack of competence in *dABC*, a teacher may have to consider the student's ability in learning *dBC* from *cB* and *cC* in addition to providing the student with information about the three basic concepts. Using Vomlel's arithmetic problem as an example, we are wondering how computational techniques can help us compare the merit of the (partial) Bayesian networks shown in Figure 2.

Therefore, we consider the problem of how the use of computational techniques can help us identify students' learning patterns. To facilitate the discussion about the ways in which a composite concept may be learned, we define the notation that we will use to represent how students learn a composite concept. Let τ denote the composite concept which we would like to know how students learn. Assume that there are α basic concepts included in τ. Based on our **non-overlapping assumption** that we present below, τ can have at most α parent concepts. If some of τ's parent concepts are composite, τ will have less than α parent concepts. We denote a way of learning τ by a *computational form* of τ . A computational form of τ may have one or more parts, the parts are connected by underscores, and each part of the computational form represents a parent concept of τ.

Definition 1. Assume that learning κ requires the knowledge of α basic concepts. Let $\{\pi_1,\pi_2,\ldots,\pi_\mu\}$ denote the set of parent concepts of κ, where $\mu\leq\alpha$. The **computational form** of the way to learn κ is $\pi_1\_\pi_2\_\ldots\_\pi_\mu$. Each computational form of a composite concept represents a **learning pattern** for students to learn the composite concept.

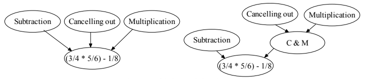


Fig. 2. Which model is better?

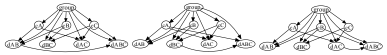


Fig. 3. Three other ways to learn dABC (from left to right): AB_C, BC_A, AC_B

Definition 2. (**The non-overlapping assumption**) We assume that any two parent concepts defined in Definition 1 do not have common basic concepts.

The non-overlapping assumption presumes that students must learn composite concepts from non-overlapping components. Specifically, the parent concepts of the composite concepts do not include common basic concepts. Hence, there are only four possible ways to learn *dABC*: (1) integrating *cA*, *cB*, and *cC* directly (denoted by A_B_C); (2) integrating *dAB* and *cC* (denoted by AB_C); (3) integrating *dBC* and *cA* (denoted by BC_A); and (4) integrating *dAC* and *cB* (denoted by AC_B). The structure shown in Figure 1 is A_B_C. Figure 3 shows three other ways to learn *dABC*, and, from the left to right, they are AB_C, BC_A, and AC_B (Nodes for test items are not included for readability of the networks in Figure 3 and other Bayesian networks that we will discuss later in this paper.)

The non-overlapping assumption simplifies the space of the possible answers. Without excluding the possibility of overlapping ingredient concepts, we would have to consider AB_BC, AB_AC, and BC_AC if we minimise the number of overlapping basic concepts. We would also have to consider cases like AB_BC_A and even AB_BC_AC_A if we do not minimise the number of overlapping basic concepts. It is certainly possible that a student can learn *dABC* with these alternative methods. However, we leave these more challenging possibilities for future studies.

As we present more details about the designs of our experiments, it will become clear that the methods we propose do not require the application of the non-overlapping assumption. However, making use of the assumption simplifies the space of the possible solutions, while the proposed methods can still be applied without the assumptions.

We do not assume further limitations on the ways that students might integrate the candidate parent concepts. For instance, under some circumstances, one might believe that a student cannot integrate *cA* and *cB* unless *cA* is already a part of another relevant concept, say *cAC*. In this case, one might learn *dABC* from *dAC* and *cB* but not from *dAB* and *cC*. We did not consider such special constraints in our study.

Definition 3. (**The common assumption**) All students learn a composite concept with the same learning pattern.

The common assumption presumes that all students use the same strategy to learn a composite concept. The purpose of using this assumption is just to simplify the presentation of our discussion. It is understood that there is no clear support for this rather controversial assumption. However, the current goal of our methods is to select exactly one best candidate from the many possible ways of learning the composite concept. It will become clear, as we present our methods in the rest of this paper, that we can easily modify our methods to select the top *k* candidate solutions for human experts to make the final judgment about how students may learn the composite concept. We simply have to present the *k* highest-scored candidate structures to the experts to relax the common assumption. Therefore, we hope this assumption is not as provocative as it might appear.

In summary, we would like to find ways to tell which of the candidate networks, e.g., those in Figure 3, was used to generate the simulated students records.

Generating Student Records

The contents of the conditional probability tables (CPTs) of the Bayesian networks were generated based on a *Q*-matrix (e.g., those contained in Table 1), a given network structure (e.g., those shown in Figures 1 and 3), and simulation parameters according to the methods described in (Liu, 2005). When generating the CPTs, we considered not only the chances of *slip* and *guess* but also the chances of students' abnormal behaviours that deviated from the typical competence patterns of the subgroups to which they belonged. To capture the uncertainty of this latter type, we inherited the concepts of *group guess* and *group slip* discussed in (Liu, 2005), but set both *group guess* and *group slip* to ***groupInfluence***. More precisely, when $q_{j,k}=1$, we assigned a high probability for the j^{th} subgroup being competent in the k^{th} concept (if C_k is basic), and this probability is sampled uniformly from [1-*groupInfluence*, 1], where *groupInfluence* is a simulation parameter selected for individual experiments. Hence, even if $q_{j,k}=1$, $\Pr(C_k=\text{competent}\mid group=g_j)$ might not be equal to 1, and students of the j^{th} subgroup might not be competent in the k^{th} concept. Similarly, when $q_{j,k}=0$, we assigned a low probability for the j^{th} subgroup being competent in the k^{th} concept (if C_k is basic), and this probability is sampled uniformly from [0, *groupInfluence*]. Hence, even if $q_{j,k}=0$, students of the j^{th} subgroup might be competent in the k^{th} concept.

The conditional probabilities of correctly responding to test items given different competence levels were specified with a standard procedure that has been commonly employed in the literature, e.g., (Martin & VanLehn, 1995; Mayo & Mitrovic, 2001; Conati et al., 2002; Millán & Pérez-de-la-Cruz, 2002). Instead of using two simulation parameters for *slip* and *guess*, we set these two parameters to the same value and called it ***fuzziness***. Hence the probabilities $\Pr(I_{j,k}=\text{correct}\mid C_j=\text{competent})$ and $\Pr(I_{j,k}=\text{correct}\mid C_j=\text{incompetent})$ were, respectively, sampled uniformly from [1- *fuzziness*, 1] and [0, *fuzziness*]. Notice, again, that the value of *fuzziness* functioned as the bounds of the actual values of *guess* and *slip* but not their values.

Similar to what has been reported in the literature, e.g., (DiBello et al., 1995; Mayo & Mitrovic, 2001; Conati et al., 2002), we employed the concept of *noisy-and* (Pearl, 1988) for setting the conditional probabilities for the composite concepts which have multiple parent nodes. Noisy-and nodes reflect a probabilistic version of the "AND" relationship in traditional logics. The degree of noise is controlled by the simulation parameter *groupInfluence*. Readers are referred to (Liu, 2005) for more details.

We controlled the percentages of the subgroups in the entire simulated student population by manipulating the prior distribution over the node *group*. We could use any prior distribution for *group* in the simulator. In the reported experiments, the node *group* took the uniform distribution as its prior distribution. Hence, if we were simulating a population of 10000 students that consisted of eight subgroups, each subgroup might have approximately 1250 students.

Table 2. A sample of simulated students' item responses for the Bayesian network shown in Figure 1 and the left *Q*-matrix in Table 1 (1 and 0 denoting correct and incorrect, respectively)

group	Test Items													
	iA1	*iA2*	*iA3*	*iB1*	*iB2*	…	*iAB1*	*iAB2*	*iAB3*	*iBC1*	…	*iABC1*	*iABC2*	*iABC3*
g_1	1	1	1	1	1	…	1	1	1	1	…	1	1	1
g_1	1	1	0	1	1	…	1	1	1	1	…	1	1	0
g_1	1	1	1	1	1	…	1	0	1	1	…	1	1	1
…	…													
g_2	1	1	0	1	1	…	0	1	0	1	…	1	1	0
g_2	1	1	1	1	0	…	1	0	0	1	…	1	1	1
…	…													
g_5	0	0	0	1	1	…	0	0	0	1	…	0	0	1
g_5	0	1	0	1	1	…	1	0	0	1	…	0	1	0
…	…													
g_8	1	1	1	0	1	…	0	0	0	0	…	1	1	1
g_8	1	1	1	1	1	…	0	1	0	0	…	0	1	1
…	…													

In summary, we created Bayesian networks with the procedure reported in (Liu, 2005), and we controlled the degree of uncertainty by two parameters, i.e., *groupInfluence* and *fuzziness*. Given the network structure and the CPTs, we had a functioning Bayesian network, and could apply this network to simulate item responses of different types of students. We employed a uniform random number generator in simulating students' behaviours with a typical Monte Carlo simulation procedure. For instance, we randomly sampled a number, δ, from a uniform distribution [0, 1]. If the conditional probability of correctly responding to *iA2* was 0.3 for a particular subgroup of students and if δ>0.3, we would assume that this student responded to *iA2* incorrectly. Students of the same subgroup may have different item responses to the same item because we independently drew a random number for each test item and each simulated student.

Example 2. Table 2 shows the data for certain students that we generated with the Bayesian network shown in Figure 1 and the left *Q*-matrix shown in Table 1, when setting *groupInfluence* and *fuzziness* to 0.05 and 0.10, respectively. Each row in Table 2 contains a record for a simulated student, e.g., the first simulated student correctly responds to all of the test items while the second simulated student fails *iA3* and *iABC3*. Although we always simulate item responses for students of all of the subgroups, we cannot show all of the data here. Notice that, due to the degree of uncertainty which was simulated and which was controlled by *groupInfluence* and *fuzziness*, a student who should be competent in a concept might not respond correctly to a test item for that concept. For instance, the second student of g_1 fails to respond correctly to *iA3*, although all the members of g_1 are supposed to be competent in *cA* as indicated by the *Q*-matrix.■

Computational Complexity

Assume that there are β basic concepts in $\bar{C}$. The computational complexity of our target problem comes from both the number of different ways that students can learn the composite concept which, directly or indirectly, integrates all β basic concepts and the number of different *Q*-matrices.

Table 3. Results of computing Formula (2) grow exponentially with β

β	3	4	5	6	7	8	9	10
$\bar{S}(\beta)$	4	14	51	202	876	4139	21146	115974

Given the non-overlapping assumption and the common assumption, the number of different ways that students can learn the composite concept which integrates all β basic concepts is related to the Stirling number of the second kind (Knuth, 1973). Formula (1) shows the number of ways to partition *t* different objects in exactly *i* nonempty sets.

$$S(t,i)=\frac{1}{i!}\sum_{j=0}^{i-1}(-1)^j\binom{i}{j}(i-j)^t \tag{1}$$

Formula (2) shows the number of ways to partition β different objects in more than two nonempty sets, and Table 3 illustrates how the number of possible learning patterns grows with β. $\bar{S}(\beta)$ is the number of possible ways to learn a composite concept from β basic concepts.

$$\bar{S}(\beta)=\sum_{i=2}^{\beta}S(\beta,i)=\sum_{i=2}^{\beta}\left(\frac{1}{i!}\sum_{j=0}^{i-1}(-1)^j\binom{i}{j}(i-j)^\beta\right) \tag{2}$$

The choice of the *Q*-matrix influences the prior distribution for the students being simulated, and is an important issue for studies that employ simulated students (VanLehn et al., 1998). There can be a myriad number of different *Q*-matrices, cf. (DiBello et al., 1995), and clearly the chosen *Q*-matrix affects the difficulty of identifying the learning pattern of interest. When there are β basic concepts in $\bar{C}$, there can be as many as $n=2^\beta-1$ different concepts in $\bar{C}$, and there can be as many as 2^n different competence patterns, as we have explained in Example 1. In principle, a student can belong to any of these 2^n patterns. Because each of these 2^n patterns can be either included or not included in the *Q*-matrix, there are 2^{2^n} different *Q*-matrices. Note that such quantities occur only in the worst-case scenario as not all of these 2^n patterns and not all of the 2^β-1 concepts are practical.

We can choose to include all possible competence patterns in a *Q*-matrix, or, alternatively, we can make the *Q*-matrix include only those patterns that appear to be helpful for identifying the learning patterns. In the former case, there is only one possible *Q*-matrix, but the size of this *Q*-matrix will be quite large. For β=3 and β=4, the *Q*-matrices will include, respectively, 128 and 32768 competence patterns. In the latter case, the selection of *Q*-matrices is equivalent to choosing a certain population of students to participate in our studies in order that we can achieve our goals. For instance, all of the values in the *dABC* columns of the *Q*-matrices in Table 1 are set to 1. As explained in Generating Student Records, such a setting makes the simulated students very likely to be able to integrate the parent concepts of *dABC* to learn *dABC*, and, if we want to learn how students learn *dABC*, it should be reasonable to recruit students who appear to be competent in *dABC* in our studies. Hence the choice for the settings of the *dABC* columns of the *Q*-matrices in Table 1 is not groundless. We will discuss the influence of *Q*-matrices in more detail in Influences of the *Q*-Matrices and More Realistic Evaluations when we present the experimental results.

Example 3. Based on this discussion, we choose to report results for interesting *Q*-matrices in which there are only three or four basic concepts. For the $\bar{C}$ used in Table 1, β=3 and *n*=7. There are four different ways to learn the composite concept *dABC*, 128(=2^7) different competence patterns, and 2^{128} possible *Q*-matrices, so there are 2^{130} (=4×2^{128}) problem instances. For the problem in which we consider four basic concepts (i.e., β=4), there will be 14 different ways to learn *dABCD* based on Formula (2). A complete enumeration of the subsets of {*A*, *B*, *C*, *D*}, without considering the empty subset, includes 15 configurations, which makes *n* =15 in $\bar{C}$ (cf. Table 7). Hence, for this case, we have 32768(=2^{15}) competence patterns and 14×2^{32768} different problem instances. ■

Impacts of Latent Variables

In addition to the large search space that was discussed in Computational Complexity, another major difficulty in learning the learning patterns comes from the fact we cannot directly observe the levels of competence of the students. What we have at hand are students' responses to test items that are indirectly and probabilistically related to the actual competence levels. The literature, e.g., (Heckerman, 1999), has addressed common issues in learning network structures with hidden variables, and some, e.g., (Desmarais et al., 2006), have discussed issues that are specific to learning network structures for educational applications. In this subsection, we look into problems that are directly related to our target problems.

If we could directly observe the states of competence levels of concepts, we would be able to apply theoretical inference tools. Let *CI*(*X*, *Y*, *Z*) denote the situation that variables in *X* and *Z* become independent when we obtain information about the variables in *Y*. For simplicity, we say *X* and *Z* are *c*onditionally *i*ndependent given *Y* when *CI*(*X*, *Y*, *Z*) holds. (Note that *X*, *Y*, and *Z* may contain one or more variables.) Take the case for learning *dABC* as an example. If we can directly observe the states of *group*, *cA*, *cB*, *cC*, *dAC*, and *dABC*, we will find that *CI*(*dAC*, {*group*, *cA*, *cB*, *cC*}, *dABC*) if the actual structure is the network shown in Figure 1. We can tell whether the conditional independence holds based on the criteria for judging whether *d-separation* (Pearl, 1988) holds in Bayesian networks, and the data generated with this network are expected to reflect the independent relationship. Hence, we would be able to tell the learning pattern for *dABC* by checking whether *CI*(*dAC*, {*group*, *cA*, *cB*, *cC*}, *dABC*) and other relevant conditional independence relationships hold.

In reality, we *cannot* directly observe the states of *group*, *cA*, *cB*, *cC*, *dAC*, and *dABC*, and can observe only the states of the test items for *cA*, *cB*, *cC*, *dAC*, and *dABC*, i.e., the states of *iAj*, *iBj*, *iCj*, *iACj*, and *iABCj*, where *j*=1, 2, 3. This information is helpful but does not allow us to determine the answer to the problem for sure, because *CI*({*iACj*|*j*=1,2,3},{*iAj*, *iBj*, *iCj*|*j*=1,2,3}, {*iABCj*|*j*=1,2,3}) fails to hold for any structure shown in Figures 1 and 3 now. In Figure 1, even if we further assume the availability of information about *group* either because of students' records or because of the help of student assessment software, nodes *iACj* and nodes *iABCj*, *j*=1,2,3, remain probabilistically dependent. In this network, only direct information about the competence levels, i.e., *cA and cC*, or either of *dAC* and *dABC*, can d-separate nodes *iACj* and nodes *iABCj*, *j*=1,2,3. As a consequence, if we can observe only the states of the nodes for test items, we cannot tell the difference among different ways of learning *dABC* based on the concept of d-separation.

The research into learning Bayesian networks from data has made significant progress in recent years (Heckerman, 1999; Neapolitan, 2004). Yet, the problem of learning Bayesian networks with hidden variables is relatively more difficult. Based on our limited knowledge, existing algorithms can tackle problem instances that consider a limited number of hidden variables but such algorithms do not explicitly attempt to learn the relationships among a set of hidden variables, which is the focus of this paper. In addition to the consideration of hidden variables, a further major technical challenge in learning Bayesian networks is missing values in some of the training data. We disregard this consideration at this moment, though it is possible for a real student not to answer all the questions in a test. We as-

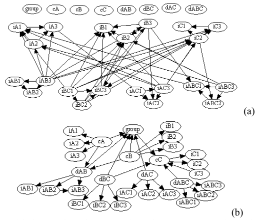


Fig. 4. Learning with the PC algorithm: (a) Only with item responses (b) With complete data

sume that students will be motivated to respond to all of the test items, though it may be quite difficult to ensure that students will make their best efforts.

In order to show the applicability and limitation of the existing algorithms, we tried our problem with the PC-algorithm (Spirtes et al., 2000) implemented in Hugin. Hugin was implemented and is being supported by the research team that originally invented the junction-tree algorithm (Jensen & Nielsen, 2007), so we believe that it is a reliable software tool. We generated records for 10000 students with the procedure described in Generating Student Records. (More details about our simulation and experiments are provided in Generating Datasets) In that simulation, we used the network shown in Figure 1 and the left *Q*-matrix shown in Table 1, and we set *groupInfluence* and *fuzziness* to 0.05. (It will become clear that setting both *groupInfluence* and *fuzziness* to 0.05 is the simplest case in our experiments.) After recording test records for 10000 simulated students, we removed the data for *group* and all of the nodes for concepts to achieve a table like Table 2. We informed the PC-algorithm that the values for these nodes were missing in training instances, and achieved the network shown in Figure 4(a) when we set "Level of Significance" to 0.05 (which is the default value in Hugin). This recommended choice of the Level of Significance has also been adopted in other research work, e.g., (Vomlel, 2004). With the absence of all of the data for *group* and concept nodes, the PC-algorithm isolated these nodes from the rest of the network. If we ignore the isolated nodes in Figure 4(a), the resulting network appears as an item-to-item knowledge structure that relates nodes representing test items (Desmarais et al., 2006). Note that, without an appropriate introducing of the hidden nodes into the learned structure, the nodes for the test items become probabilistically related, resulting in a very complicated network in Figure 4(a) when compared with the network in Figure 4(b). Interested readers may refer to (Desmarais et al., 2006) for the techniques for learning item-to-item knowledge structures.

We obtained the network shown in Figure 4(b) from the PC-algorithm in Hugin by using the original simulation data, while not removing the data for *group* and the concept nodes. We manually arranged the nodes in Figure 4(b) to put them in positions that were similar to their counterparts in Figure 1. A simple comparison of these two networks shows that the directions of the links between the concept nodes in the learned network (Figure 4(b)) are quite different from those in the original network (Figure 1). In addition, the node *dABC* has only one parent node in Figure 4(b), which is obviously incorrect. These two networks look quite similar except for these differences.

The differences between the networks shown in Figure 4 show how *direct* observations about the nodes in the networks help the PC-algorithm to build better networks. The network in Figure 4(b) does not have isolated nodes, and looks more similar to that in Figure 1 from which the simulation data were created. Qualitatively, the network in part (b) reflects the relationships among the variables more concisely and faithfully than the network in part (a).

The implication of the differences in directions of the links in Figure 1 and Figure 4(b) is a complex issue, and we cannot jump immediately to the conclusion that Figure 1 is a superior option, as might be the case had we actually learned a network from real data. Although applying causal relationships in determining the directions of links in Bayesian networks generally helps us build more concise networks (Russell & Norvig, 2002), links in Bayesian networks do not necessarily reflect causal relationships (Pearl, 1988). Indeed, we can apply Shachter's arc reversal operations (Shachter, 1988) to reverse the directions of the links in Bayesian networks *and* preserve the joint probability distributions. If the applications ultimately rely only on the joint probability distributions implicitly represented by the Bayesian networks, the structure of the learned Bayesian network will not seriously affect the application of the learned network. A structure that is unnecessarily complex will make the inference algorithm run less efficiently, but that will not affect the correctness of an inference procedure. Hence, if we learn Bayesian networks to build better CAT systems, the structures of the learned Bayesian networks may not play a crucial role, unless the learned networks can encode the joint probability distributions of important variables more precisely. For instance, Carmona et al. (2005) report that adding links for prerequisite relationships enables their assessment system to actually shorten the test lengths for variable-length tests.

From our perspective, the difference in directions of the links in Figure 1 and Figure 4(b) indicates that learning student models from scratch does not help much for identifying the structure of the network based on which students' item responses were generated. The aim of our work is to identify this unobservable Bayesian network based on students' external performance, when students, either consciously or unconsciously, utilise a common strategy to learn a composite concept and if this strategy can be represented by Bayesian networks. Hence, we propose that we use computer software as an aid in the selection of the best model from a set of candidate models that experts provide. We hope that this is a more viable approach for some problems, and we present our methods in the following sections.

METHODS FOR MODEL SELECTION

The main goals of our experiments are to evaluate the effectiveness of the proposed methods. The hidden structures of the Bayesian networks embody the abstract learning patterns, so our algorithms aim at guessing the hidden structures that were used to create the simulated test records, and we call our programs the **classifiers**, henceforth. (Depending on the context of the discussion, we may say that we want to learn the learning patterns, or we may say that we want to learn the hidden structures of the Bayesian networks.) We discuss three different ways to build the classifiers in three subsections.

Mutual Information-Based Methods

Consider the problem of learning the learning pattern for *dABC*. When there is only one actual structure, we can consider the networks shown in Figure 3 as competing structures, and we can try to define scores for the competing structures to compare their fitness to the data.

Although students' item responses provide only *indirect* evidence about the values of the concept nodes, they are still useful for estimating the states of the concept nodes. Given the estimated states, mutual information-based measures will become useful. Intuitively, the nodes that represent the parent concepts of a composite concept should contain a greater amount of information with the node that represents the composite concept. Let *MI*(*X*;*Y*) denote the mutual information (Cover & Thomas, 2006) between two sets of random variables *X* and *Y*. Formula (3) shows the definition of *MI*(*X*;*Y*); where *d*(*X*) and *d*(*Y*) are, respectively, the domains of *X* and *Y*, and *x* and *y* are, respectively, the values of *X* and *Y*.

$$MI(X;Y)=\sum_{x\in d(X)}\sum_{y\in d(Y)}\Pr(X=x,Y=y)\log\frac{\Pr(X=x,Y=y)}{\Pr(X=x)\Pr(Y=y)} \quad (3)$$

Let *H*(*X*) denote the entropy of *X*, *H*(*X*|*Y*) the conditional entropy of *X* given *Y* (Cover & Thomas, 2006), and *R*, *S*, and *T* three sets of random variables. We can show that *MI*(*R*;*T*)>*MI*(*S*;*T*) implies *H*(*T*|*R*)<*H*(*T*|*S*).

$$MI(R;T)>MI(S;T)\Rightarrow H(T)-H(T\mid R)>H(T)-H(T\mid S)\Rightarrow H(T\mid R)<H(T\mid S)$$

Since entropy is a measure for gauging the uncertainty about random variables, this derived inequality suggests that *R* may be more related to *T* than *S* is to *T* (because the information about *R* makes *T* less uncertain than the information about *S* does). Experience has shown that mutual information is useful for studying student classification (Liu, 2005; Weissman, 2007). For the current study, we prefer the set of candidate concepts that contain a larger amount of mutual information about the target composite concept, when trying to find the parent concepts of a composite concept.

Based on this heuristic interpretation, if the actual structure is the leftmost one in Figure 3, then *MI*(*dAB*, *cC*; *dABC*) should be larger than *MI*(*dAC*, *cB*; *dABC*). Analogously, if the actual structure is the rightmost one in Figure 3, then the inequality should be reversed.

In order to apply this heuristic principle, we use the observed item responses to estimate the obscure competence levels. We have assumed that students will respond to three test items for each concept in Generating Student Records, so students may give correct answers to 0%, 33%, 67%, or 100% of the test items for each concept. We can use this percentage as the estimation for the state of a concept node, and, similarly, we can estimate the joint distributions of multiple concept nodes. For instance, Pr(*dAB*=33%, *cC*=67%) is set to the percentage of students who correctly answered one item and two items, respectively, for *dAB* and *cC*. In estimating the joint probabilities, we smooth the probability distributions to avoid zero probabilities because some configurations of variables may not appear in the samples by chance, cf. (Witten & Frank, 2005). We add 0.001 to every different configuration of the variables. By adding this small amount to the count of each configuration of the variables, we will not distort the actual probability distribution reflected by the students' records and also, at the same time, completely avoid the problem of zero probability. With this procedure, we have a way to estimate the mutual information measures. Hence, we can try the following heuristic for learning how students learn composite concepts.

Table 4. A sample of statistics for responses to test items designed for *dAB* and *cC*

		dAB				row total
		0%	33%	67%	100%	
cC	0%	765	901	573	867	3106
	33%	971	453	432	431	2287
	67%	567	648	865	358	2438
	100%	643	729	199	598	2169
column total		2946	2731	2069	2254	10000

Heuristics 1. Let $\Omega=\{\omega_1,\omega_2,\cdots,\omega_\sigma\}$ be the set of computational forms for all possible ways to learn a composite concept *τ*. Let Π_j be the set of parent concepts represented by ω_j, where *j*=1,2,…,σ. We choose Π^* as the parent concepts of *τ* if Π^* is the set of parent concepts represented by the ω^* specified in the following formula.

$$\omega^*=\arg\max_{\omega_j\in\Omega} MI(\Pi_j;\tau). \quad ■$$

Example 4. Using some simulated data similar to those shown in Table 2, our classifier constructs a table like Table 4. Table 4 contains counts for 10000 simulated students who responded correctly to 0%, 33%, 67%, and 100% of test items designed for *dAB* and *cC*. We do not consider the smoothing operations at this point as we wish to focus on the function of this numerical example. The "row total" and "column total", respectively, show the counts of students who correctly responded to items for *cC* and *dAB*. Individual cells in the table show the counts of students who correctly responded to the test items with the percentages specified on the row *and* in the column. There were 10000 simulated students, so the estimated values for Pr(*dAB*=33%), Pr(*cC*=67%) and Pr(*dAB* =33%,*cC*=67%) are, respectively, 0.2731, 0.2438, and 0.0648. Hence the classifier can estimate the individual probability distributions for *dAB* and *cC*. It can also estimate the joint distribution for *dAB* and *cC*.

When using a larger table containing data for *dAB*, *dAC*, *cB*, *cC*, and *dABC*, the classifier can compute the mutual information *MI*(*dAB*, *cC*; *dABC*) and *MI*(*dAC*, *cB*; *dABC*) with Formula (3), and can apply Heuristic 1 accordingly. For instance, if *MI*(*cA*, *cB*, *cC*; *dABC*) is the largest among the estimated values of *MI*(*dAB*, *cC*; *dABC*), *MI*(*dAC*, *cB*; *dABC*), *MI*(dBC, *cA*; *dABC*), and *MI*(*cA*, *cB*, *cC*; *dABC*), then the structure is A_B_C. If *MI*(*dAC*, *cB*; *dABC*) is the largest among the estimated values of *MI*(*dAB*, *cC*; *dABC*), *MI*(*dAC*, *cB*; *dABC*), *MI*(*dBC*, *cA*; *dABC*), and *MI*(*cA*, *cB*, *cC*; *dABC*), then the structure is AC_B. ■

We will examine the effectiveness of this heuristic method in experiments.

Search-Based Methods

An obvious drawback of applying Heuristic 1 is that we will have to compute the estimated mutual information for each possible way of learning the composite concepts. We have seen how the number of candidate structures can grow with the number of basic concepts in Table 3. Instead of computing the MI measures for all competing structures, it is possible to do the comparison incrementally using a search-based procedure. We present and explain the search procedure, provide a simple running example, and analyse the computational complexity of the proposed algorithm in this subsection.

Algorithm. **`Search4Pattern`**

Input. Students' item responses (e.g., the data listed in Table 2) and the target composite concepts (e.g., *dABCD*)

Output. The most likely way to learn the target composite concept

Procedure.

```
1. If the target composite concept involves only two basic concepts, return these basic concepts.
2. Let $\kappa$=2, $\rho$=∞, and $\sigma$ be an empty set. Denote the target composite concept by $\tau$, and let $\beta$ be
   the number of basic concepts included in $\tau$. Set $\omega_1^*$ to $\tau$'s computational form that is the
   concatenation of all symbols for the basic concepts included in $\tau$.
3. Find all legal ways to split $\omega_{\kappa-1}^*$ into $\kappa$ parts. Let $\Omega_\kappa=\{\omega_1,\omega_2,\cdots,\omega_{size(\kappa)}\}$ denote the set of
   legal splits of $\omega_{\kappa-1}^*$, where size($\kappa$) denotes the number of elements in $\Omega_\kappa$.
4. Let $\{\pi_{j,1},\pi_{j,2},\cdots,\pi_{j,\kappa}\}$ be the set of candidate parent concepts that we concentrate to form an
   $\omega_j\in\Omega_\kappa$. Compute the score for each $\omega_j\in\Omega_\kappa$, $j$ = 1, 2, …, size($\kappa$).
       $score(\omega_j)\equiv MI(\pi_{j,1},\pi_{j,2},\cdots,\pi_{j,\kappa};\tau)$
5. Find $\omega_\kappa^*$ such that $\omega_\kappa^*=\arg\max_{\omega_j\in\Omega_\kappa} score(\omega_j)$.
6. If $score(\omega_\kappa^*)\le\rho$ and $\sigma$ is not an empty set, return $\sigma$. Otherwise, set $\rho$ to $score(\omega_\kappa^*)$, set $\sigma$ to
   the set of candidate parent concepts represented by $\omega_\kappa^*$, and increase $\kappa$ by 1.
7. If $\kappa>\beta$, return $\sigma$. Otherwise, return to step 3.   ■
```

We include step 1 in the algorithm just to make the algorithm methodologically complete. We do not expect a normal condition when we have to run our algorithms to find the learning pattern for a composite concept that consists of only two basic concepts.

At step 2, we conduct initialization operations for the algorithm. We set ω_1^* to the unique computational form of *τ* that is simply the sequence of symbols that represents the basic concepts required for learning *τ*. For instance, ω_1^* will be ABCD if *τ* is *dABCD*.

Step 3 is the key step by which we search for the solution hierarchically. This step requires the definition for *legal* ways of splitting $\omega_{\kappa-1}^*$. A computational form for a learning pattern of *τ* contains one or more symbols, and a legal splitting of the computational form converts *exactly* one of these parts into two smaller parts. A legal split of $\omega_{\kappa-1}^*$ is called a **successor** of $\omega_{\kappa-1}^*$. For instance, {ABC_D, ABD_C, ACD_B, BCD_A, AB_CD, AC_BD, AD_BC} is the set of successors of ABCD, and A_B_CD and AB_C_D are the only successors of AB_CD. Two or more computational forms can share a successor. For instance, A_B_CD is a successor to both AB_CD and BCD_A. We cannot split A_B_C_D further because it does not have any parts that include two or more symbols. (Note that *size*(*κ*) is equal to *S*(*β*,*κ*) as defined in Formula (1).)

Step 4 computes the scores for each $\omega_j\in\Omega_\kappa$. The scores are defined as the estimated mutual information as discussed in Formula (3). Recall that a computational form, as defined in Definition 1, contains names of parent concepts, i.e., $\{\pi_{j,1},\pi_{j,2},\cdots,\pi_{j,\kappa}\}$, of a composite concept. A $\pi_{j,t}$ represents a corresponding concept of the *t*th part of ω_j. For instance, if ω_j is AB_C_D, we have $\pi_{j,1}=dAB$, $\pi_{j,2}=cC$, and $\pi_{j,3}=cD$.

Step 5 finds the ω_κ^* that has the largest score among all $\omega_j\in\Omega_\kappa$.

At step 6, if the largest score of the successors is smaller than or equal to the score of the current candidate, then the current candidate becomes the answer. Otherwise, the successor that has that largest score becomes the current candidate. Notice that this search procedure prefers simpler structures by using "≤" rather than "<". This design choice should bring to mind the principle of *Occam's razor*, which prefers simpler models against complex ones, and this principle is commonly embraced in the machine learning literature (Witten & Frank, 2005). Evidently, **`Search4Pattern`** can be applied to solve the problem for any value of *β*, and the algorithm must stop when *κ* becomes larger than *β* at step 7.

Example 5. We illustrate the search procedure for learning how students learn *dABCD* in Figure 5. In Figure 5, arrows connect computational forms and their successors, and successors include exactly one more component than the original computational forms. Part (a) shows the complete search space, and part (b) shows a particular search example. The search procedure begins by setting ω_1^* to ABCD, and the search goes from the left to the right. We compute the scores for the competing structures in which *dABCD* has only two parent concepts at steps 2, 3, and 4. The structure that has the largest score becomes the *current candidate* at steps 5 and 6. (Assume that ABD_C is the current candidate in Figure 5(b).) At step 7, we return to step 3 to compute the scores of the successors of the current candidate. In the second iteration of the algorithm, we repeat steps 3 and 4, and compute the scores for the computational forms that contain three components, namely, AB_C_D, AD_B_C, and A_BD_C in Figure 5(b). We call the computational form, in $\Omega_{3,}$ that has the largest score at step 5 the *new candidate* (say A_BD_C). At step 6, if the score for the current candidate (ABD_C) is higher than that for the new candidate (A_BD_C), we return the current candidate as the answer. Otherwise, we replace the current candidate with the new candidate and carry out step 7. In the latter case, we will have to compute a score for A_B_C_D, which must be the only successor to the new candidate in Figure 5. If the score of A_B_C_D is larger than that of the new candidate, then A_B_C_D is the answer, otherwise the new candidate is the answer. ■

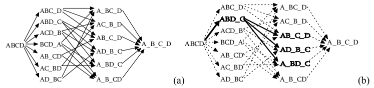


Fig. 5. Learning the models through a search procedure: (a) The complete search space for learning *dABCD* (b) An example

Table 5. Percentage of avoided computation by using **Search4Pattern** grows with β

number of basic concepts in τ (i.e., β)	3	4	5	6	7	8	9	10
total number of candidate structures	4	14	51	202	876	4139	21146	115974
an upper bound of checked structures	4	12	26	50	92	168	310	582
a lower bound of saving in percentage	0.00	14.3	49.0	75.2	89.5	95.9	98.5	99.5

Computationally, using **Search4Pattern** is more efficient than directly computing the scores for all candidate structures, which is illustrated by the data in Table 5. We duplicate the first row and the second row of Table 5 from Table 3. Except for the trivial case when β is 2 at step 1, we must run at least the iteration for κ=2. It is easy to verify that when κ=2, there will be $2^{\beta-1}$-1 elements in Ω_κ. This is the number of different ways to split β different objects into two nonempty sets, and is equal to $S(\beta,2)$ as defined in Formula (1). Ω_κ can have at most β elements for the following iterations in which κ=3, κ=4, …, $\kappa=\beta$-1. There are only β basic concepts in τ, so we can split any Ω_j, where j=2, 3,…, β-2, in at most β different ways. Hence, during these intermediate search steps, **Search4Pattern** will compute at most $\beta\times(\beta$-3) scores. In the worst case, **Search4Pattern** must run the iteration for $\kappa=\beta$, and will stop when $\kappa>\beta$ at step 7. In this very last iteration, Ω_κ can have only one element, which represents the situation when students learn the target composite concept directly from β basic concepts.

Hence, in the worst case, **Search4Pattern** computes at most ($2^{\beta-1}$-1+ $\beta\times(\beta$-3)+1) scores. The third row of Table 5 shows this quantity for different values of β. Note that the numbers are pessimistic estimates of the number of times that **Search4Pattern** has to compute scores. For instance, when β is 4, **Search4Pattern** computes at most 11 scores rather 12 scores as discussed in Example 5. The difference between the actual times of computing the scores and their pessimistic estimates comes from two sources. First, we do not necessarily reach the case when $\kappa>\beta$ for all different ways of learning the target composite concept. In addition, Ω_κ must have fewer than β successors in $\Omega_{\kappa+1}$ when κ is between 3 and β-1. The fourth row of Table 5 shows a lower bound of the avoided computation in percentage. To obtain the percentage in each column, we subtract the quantity in the third row from the quantity in the second row, and divide the difference by the quantity in the second row.

Model-Based Methods: ANNs and SVMs

In addition to using the heuristic method and the search-based method, we build classifiers by employing the data about mutual information measures to train artificial neural networks (ANNs) (Bishop, 1995) and support vector machines (SVMs) (Cortes & Vapnik, 1995) for better performance. We experiment using two specific classes of ANNs: probabilistic neural networks (PNNs) (Wasserman, 1993), which are a variant of radial basis networks, and feed forward back-propagation networks (BPNs) that are implemented in MATLAB (http://www.mathworks.com). Support vector machines are a relatively new tool that can be applied to the task of classifications, and we try the C-SVC SVMs that are implemented in the LIBSVM package (Chang & Lin, 2001). We can train ANNs and SVMs with training patterns that are associated with known class labels, and the trained ANNs and SVMs can be used to classify the classes of test patterns.

We must determine what features the ANNs and SVMs will use to do classification. In addition to the estimated mutual information that we have to compute to apply Heuristic 1, we introduce more features that are computed from these original features. Based on the evidence that we gathered in experiments (Liu, 2006b), we found that the classifiers performed relatively poorly when the estimated values of the largest mutual information and the second largest mutual information were close, so we chose to add the ratios between the estimated mutual information as features of the training instances. We divide each of the raw (estimated) mutual information by the largest mutual information to create new features. We also divided the largest mutual information by the second largest, and divided the largest mutual information by the average mutual information.

Table 6. A sample instance for training ANNs and SVMs

class label: AB_C					
original features		derived features			
MI(dAB,cC;dABC)	0.17	*MI(dAB,cC;dABC)/max*	1.00	*max/runnerUP*	2.12
MI(dBC,cA;dABC)	0.04	*MI(dBC,cA;dABC)/max*	0.23	*max/avg*	1.94
MI(dAC,cB;dABC)	0.06	*MI(dAC,cB;dABC)/max*	0.33		
MI(cA,cB,cC;dABC)	0.08	*MI(cA,cB,cC;dABC)/max*	0.47		

Example 6. Table 6 shows a training instance for learning *dABC* by integrating *dAB* and *cC*, which is indicated by the class label AB_C. Let *max* denote the largest mutual information among the original features, *runnerUp* the second largest, and *avg* the average of all original features. We need to compute the scores for four competing structures that are shown in Figures 1 and 3, and they are shown in the leftmost column of the table. A simple comparison and calculation show that *max*=0.17, *runnerUp*=0.08, and *avg*=0.0875 in this example.

We also compute new features that are defined based on the original features. For instance, *MI*(*dBC*,*cA*;*dABC*)/*max*=0.04/0.17=0.23 and *max*/*avg*=0.17/0.0875=1.94. Among these new features, we observed in experiments that *max*/*runnerUp* is quite indicative of the danger that a wrong decision can be made. When this ratio is small, it is generally dangerous to apply Heuristic 1. In this particular case, the fact that *max*/*runnerUp* is 2.1 indicates that it is quite safe for us to choose AB_C as the way students that learn *dABC*. The chance of choosing a wrong solution by applying our heuristics increased when this ratio fell below 1.2 in many of our pilot experiments. ■

We can compute the number of features for this procedure of preparing the training instances. When there are β basic concepts included in the composite concept, there will be $\bar{S}(\beta)$ original features and $\bar{S}(\beta)+2$ derived features. As we have shown in Table 3, the total number of features can grow explosively. Trying to examine the possibility and effects of reducing the computational load, we will reduce the number of features using the principle component analysis (PCA) (Jolliffe, 2002) in Effects of Methods.

There are further details that we should provide about how we applied the ANNs and SVMs. For instance, we had to choose different parameters in applying both the ANNs and SVMs, and we scaled all feature values into the range [-1, 1] to improve the performance of the resulting ANNs and SVMs. These details are important but it is more appropriate to discuss them along with the experiments, so we defer such discussion until then.

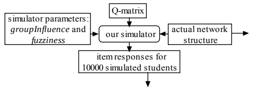

Q-matrix
simulator parameters: groupInfluence and fuzziness
our simulator
actual network structure
item responses for 10000 simulated students

Fig. 6. Creating records of item responses of simulated students

DESIGN OF THE EXPERIMENTS

We explain the generation of student data, the major steps for an individual experiment, the evaluation of the classification results, and the major categories of experiments in four subsections.

Generating Datasets

Figure 6 shows the main flow of how we create the test records for the simulated students. The simulator requires three different types of input. They include the skeleton of a Bayesian network that encodes the learning patterns of the simulated students, the *Q*-matrix that specifies the competence patterns of the simulated students, and simulation parameters *groupInfluence* and *fuzziness* that control the degrees of uncertainty in the students' item responses. The Bayesian networks can be provided by domain experts who have good reasons to employ the competing structures, and each of these structures represents a possible learning pattern of students. The *Q*-matrices are related to students' competence and how the students apply their knowledge (Martin & VanLehn, 1995). Recall that we explained, in Generating Student Records, that the provided *Q*-matrix, *groupInfluence*, and *fuzziness* will influence the underlying joint distribution that we randomly create for the provided skeletal Bayesian network. We discussed a sample output in Example 2. Both the network structures and the simulated data will be used in further experiments.

Although we have been using examples in which students need the knowledge about three basic concepts to learn the composite concept *dABC*, we will also present results of the experiments in which students need the knowledge about four basic concepts to learn the composite concept *dABCD*. We have shown the networks for cases for three basic concepts in Figures 1 and 3, and have applied their computational forms to refer to these network skeletons in The Target Question and Assumptions. With our *non-overlapping* assumption (stated in Definition 2), there can be only four ways to learn *dABC*: A_B_C, AB_C, AC_B, and BC_A.

Figure 7 shows the networks for cases when four basic concepts are included in the target composite concept, *dABCD*. In Figure 7(a), we do not show the nodes for the test items for readability. There would be 45 (=3×15) extra nodes otherwise. Note that, except for *dABCD*, the parent nodes of all nodes for composite concepts are nodes for basic concepts. This is not a necessary assumption, and the composite concepts that require the knowledge of three basic concepts can be learned by any conceivable way. In drawing the networks shown in Figure 7(b), we only draw the node for *dABCD* and its parent nodes. All the other parts are exactly the same as their counterparts as already drawn in Figure 7(a). For instance, the parent concepts of *dABD* in the network that used the leftmost sub-network in the top row of Figure 7(b) are also *cA*, *cB*, and *cD*. For convenience, we refer to these skeletal networks by their computational forms. Namely, from top to the bottom row and from left to right in both rows, we have A_B_CD, AB_C_D, ACD_B, ABD_C, and A_B_C_D in Figure 7(b).

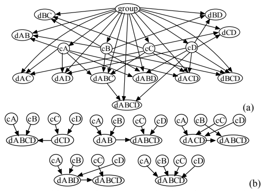


Fig. 7. (a) One possible way to learn *dABCD* (b) Five other ways to learn *dABCD*

We will use **3baiscs** and **4basics** as defined below to specify the setups of the experiments. Based on the *non-overlapping* assumption and data shown in Table 3, **3baiscs** includes all four different ways to learn *dABC*. There are 14 possible ways to learn *dABCD*, and we arbitrarily choose two cases that contain two parent concepts, two cases that contain three parent concepts, and one case that contains four parent concepts.

Definition 4. When we try to learn the learning pattern for *dABC*, we provide A_B_C, AB_C, AC_B, and BC_A to the simulator, and call this set **3baiscs**.

Definition 5. When we try to learn the learning pattern for *dABCD*, we provide A_B_CD, AB_C_D, ACD_B, ABD_C, A_B_C_D to the simulator, and call this set **4basics**.

We employed the *Q*-matrices in Table 1 for the learning problems of *dABC*, when experimenting with **3baiscs**. Table 7 shows a *Q*-matrix that we used in many of our experiments when we used **4basics** for the learning problems of *dABCD*. The contents of the *Q*-matrix in Table 7 are special in that we chose to set all the columns for the basic concepts and the target composite concepts to 1. This is equivalent to assuming the nature of the types of the students we recruit for a study of learning how they learn. If we are interested in learning how students learn *dABCD*, it should be reasonable to suppose that we will recruit students who appear to be competent in all required basic concepts and the target composite concept. In addition to using this *Q*-matrix, we may also change the contents for different purposes in other experiments. For instance, in the experiments reported in Alternative *Q*-Matrices, we set some numbers in the basic concepts and the target composite concept to 0.

In the experimental evaluation, we set *groupInfluence* and *fuzziness* to different values in {0.05, 0.10, 0.15, 0.20, 0.25, 0.30}. Hence, there can be 36 combinations of *groupInfluence* and *fuzziness* in our experiments. We did not try values larger than 0.3 because they were beyond the considerations normally discussed in the literature (e.g., VanLehn et al., 1998; Junker, 2006; Pardos et al., 2007). Some researchers have reported observations of larger values for these parameters for special reasons; for instance, Beck and Sison (2004) observed a large value for the case of *guess*, and linked the observation with the speech recognition technology.

Table 7. Competence patterns in a *Q*-matrix

SID	competence in (integrating) concepts														
	cA	*cB*	*cC*	*cD*	*dAB*	*dAC*	*dAD*	*dBC*	*dBD*	*dCD*	*dABC*	*dABD*	*dACD*	*dBCD*	*dABCD*
g_1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
g_2	1	1	1	1	1	0	0	0	0	1	1	1	1	0	1
g_3	1	1	1	1	0	1	0	1	0	1	1	1	0	1	1
g_4	1	1	1	1	1	0	0	0	0	1	1	1	0	0	1
g_5	1	1	1	1	1	0	1	0	1	0	1	0	1	1	1
g_6	1	1	1	1	1	0	0	0	0	1	1	0	1	0	1
g_7	1	1	1	1	1	1	1	0	0	0	1	0	0	1	1
g_8	1	1	1	1	1	0	0	0	0	1	1	0	0	0	1
g_9	1	1	1	1	0	0	0	1	1	1	0	1	1	1	1
g_{10}	1	1	1	1	1	0	0	0	0	1	0	1	1	0	1
g_{11}	1	1	1	1	1	1	0	0	1	1	0	1	0	1	1
g_{12}	1	1	1	1	1	0	0	0	0	1	0	1	0	0	1
g_{13}	1	1	1	1	0	0	1	1	0	0	0	0	1	1	1
g_{14}	1	1	1	1	1	0	0	0	0	1	0	0	1	0	1
g_{15}	1	1	1	1	0	0	0	0	1	1	0	0	0	1	1
g_{16}	1	1	1	1	1	0	0	0	0	1	0	0	0	0	1

For a network structure, a *Q*-matrix, and a particular combination of *groupInfluence* and *fuzziness*, we typically created test records for 10000 simulated students. A **test record** contains the correctness of a student's item responses to all test items. Table 2 shows some sample test records. The **setting** for an experiment is constituted by a particular combination of *groupInfluence* and *fuzziness*, a structure for the Bayesian network that represents the candidate learning pattern, and a given *Q*-matrix. For convenience, we use the term **a subset** of an experiment to refer to a group of the settings in which we considered a specific combination of *groupInfluence* and *fuzziness*, the structures in **4basics** (or **3baiscs**), and a given *Q*-matrix. In an **experiment**, we used many different subsets of experiments to compare the effects of the influential factors.

Recall that we discussed the creation of the joint probability distribution for a Bayesian network with the help of random numbers in Generating Student Records. Hence, the generated Bayesian networks and the simulated test records varied with the seed for the random number generator. In order to obtain information about the average performance of our classifiers, we created 600 network instances for each setting in an experiment, and simulated 10000 students from each of these network instances. For convenience, we will refer to data that are created from a set of such 10000 simulated students as an **instance**.

Example 7. An experiment for studying the learning patterns for *dABCD* may employ 1.08 billion simulated students, if we consider all 36 combinations of *groupInfluence* and *fuzziness*. In this case, each subset demands 30 million students. We obtained 30 million by multiplying three factors: five candidate networks in *groupInfluence* and *fuzziness*, 600 network instances per candidate network, and 10000 students per network instance. The total of 1.08 billion is the result of multiplying 30 million by 36. ■

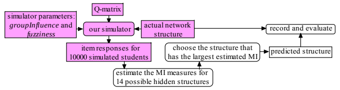

Q-matrix
simulator parameters: groupInfluence and fuzziness
our simulator
actual network structure
record and evaluate
item responses for 10000 simulated students
choose the structure that has the largest estimated MI
predicted structure
estimate the MI measures for 14 possible hidden structures

Fig. 8. Flow for evaluating Heuristics 1

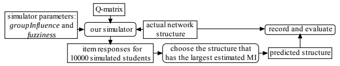

Q-matrix
simulator parameters: groupInfluence and fuzziness
our simulator
actual network structure
record and evaluate
item responses for 10000 simulated students
choose the structure that has the largest estimated MI
predicted structure

Fig. 9. Flow for evaluating **Search4Pattern**

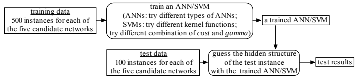

training data
500 instances for each of the five candidate networks
train an ANN/SVM
(ANNs: try different types of ANNs;
SVMs: try different kernel functions;
try different combination of cost and gamma)
a trained ANN/SVM
test data
100 instances for each of the five candidate networks
guess the hidden structure of the test instance with the trained ANN/SVM
test results

Fig. 10. Training and evaluating ANNs and SVMs

Steps of the Experiments

Due to the different nature of the heuristic methods, the search-based methods, and the machine learning-based methods, we evaluated the classifiers in slightly different ways.

Figure 8 shows the main steps for evaluating the heuristic principle. We duplicate the shadowy part from Figure 6 to show how the simulator worked for our classifiers. When we worked on the learning problems of *dABCD*, $\bar{S}(4)$ is 14 as shown in Table 3. To guess the hidden structure of each of the 3000 (=5×600) network instances that we had generated for a subset of an experiment in which we considered the structures in **4basics**, our classifier estimated the 14 mutual information measures based on the test records of 10000 simulated students, and guessed the hidden structure based on Heuristic 1. We conducted the experiments for the learning problems of *dABC* analogously.

Figure 9 shows that we evaluated **Search4Pattern** with almost the same method that we had used to evaluate Heuristic 1. The major difference was that we computed the scores for candidate structures hierarchically as explained in Search-Based Methods. Due to this hierarchical search procedure, we may save costs in computing scores for all the candidate solutions as analysed in Search-Based Methods.

Figure 10 summarises the main steps that we took to apply ANNs and SVMs in our work. In a subset of an experiment that was designed for the learning problems of *dABCD*, we created 600 network instances for each of the candidate networks shown in Figure 7(b). We split the network in-

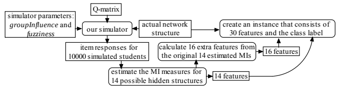

Q-matrix
simulator parameters: groupInfluence and fuzziness
our simulator
actual network structure
create an instance that consists of 30 features and the class label
item responses for 10000 simulated students
calculate 16 extra features from the original 14 estimated MIs
16 features
estimate the MI measures for 14 possible hidden structures
14 features

Fig. 11. Flow for creating data for training and testing ANNs and SVMs

stances into training and test sets, as in all supervised learning (Witten & Frank, 2005). The training set included 500 training instances that we generated from the students' records obtained from 500 network instances for a candidate network. The test set included 100 test instances generated from the students' records obtained from the remaining 100 network instances. Because we mixed the instances created for each of the networks shown in Figure 7(b), we obtained a total of 2500 (=5×500) training instances and 500 (=5×100) test instances in a subset of an experiment.

Figure 11 provides more details about how we prepared the training and test instances for experimentation with ANNs and SVMs. In addition to the original 14 estimated MI measures, we obtained 14 more features by computing and using the ratios between the estimated MI measures as features. The process is similar to that we outlined for Example 6. We divided the original 14 estimated MI measures by the largest estimated MI measure in each training instance. We then obtained two more features from the following procedure. We divided the largest estimated MI measure by the second largest estimated MI measure, and divided the largest estimated MI measure by the average of all estimated MI measures. Hence, we used 30 features for each of the 500 training instances for each of the five candidate networks in **4basics**. The actual answers (also called class labels in Example 6) were attached to the instances for both training and testing. An example of a training instance created for the learning problems of *dABC* was presented in Table 6.

In summary, we created a training instance with records of 10000 simulated students, and there were 2500 training instances, each with 30 attributes and a class label. When testing the trained ANNs and SVMs, we produced the 16 extra features from the original 14 estimated MI measures for each of the test instances as well. The actual class label was attached to the test instance so that we could compare the actual and predicted classes, but the trained ANNs and SVMs did not peek at the actual answers.

Note that, although we created students' data only from the networks shown in Figure 7 for the learning problems of *dABCD*, our classifiers did not necessarily take advantage of this information. Specifically, our classifiers, which employed Heuristic 1 and **Search4Pattern**, did not "know" this restriction, so they were free to guess any of the possible answers. In contrast, the classifiers that employed ANNs and SVMs "expected" this constraint and confined their answers to within the five possible answers, because they are supervised-learning techniques (Witten & Frank, 2005). The experiments for the **3basics** cases were conducted analogously.

Measurement of Quality

We report the accuracy for the measurement of the quality of our classifiers, although we also employed *confusion matrices* (Witten & Frank, 2005) to analyse some of the internal data. The **accuracy** for an experiment is the percentage of correct prediction of testing network instances that we used to create the simulated data. We also used the F measure that weighed *recall* and *precision* equally (Witten & Frank, 2005) to measure the performance. We provide experimental results in terms of accuracy and F measure in Table 8. However, we observed that the F measures and the accuracy we collected were similar to each other in the experiments, so we chose to report results in terms of their accuracy.

The total number of network instances for training and testing differed among the subsets of the experiments as explained in the previous subsection, so the basis for calculating accuracy is not the same for different experiments. For evaluating the performances of the heuristics and **Search4Pattern** with **4basics**, we used 3000 (=5×600) test instances. For evaluating the performances of the ANNs and SVMs with **4basics**, there were, respectively, 2500 (=5×500) and 500 (=5×100) training and test instances for each different subset in an experiment. When working with **3basics** for the learning problems of *dABC*, we had 2400 (=4×600) test instances for evaluating the heuristics and **Search4Pattern**, and had, respectively, 2000 (=4×500) and 400 (=4×100) training and test instances for evaluating the ANNs and SVMs in a subset of an experiment.

Major Categories of Experiments

In Idealistic Evaluations and More Realistic Evaluations, we examine how influential factors may affect the final accuracy. Figure 12 summarises the relationships between the experiments that we discuss in these sections. The experiments were designed so that the experience may be useful for further real world studies.

In all experiments discussed in Idealistic Evaluations, we assume that we are able to obtain correct values for *groupInfluence*, *fuzziness*, and *Q*-matrices. The main focus of Effects of Methods and Parameters is the comparison among the effectiveness of different computational methods and the influence of the simulation parameters *groupInfluence* and *fuzziness*, and the details in this section are used as a foundation for all the experiments that follow. In Alternative *Q*-Matrices, we examine the influence of the number of learning patterns with simple examples, and look into the influence of the contents of the *Q*-matrices both experimentally and analytically. In Influences of the *Q*-Matrices, we analyse the influence of the *Q*-matrix with an alternative analytical method.

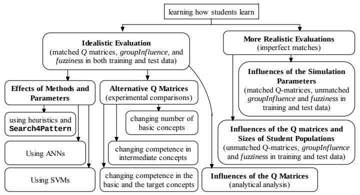

learning how students learn
Idealistic Evaluation
(matched Q matrices, groupInfluence, and fuzziness in both training and test data)
More Realistic Evaluations
(imperfect matches)
Effects of Methods and Parameters
Alternative Q Matrices
(experimental comparisons)
Influences of the Simulation Parameters
(matched Q-matrices, unmatched groupInfluence and fuzziness in training and test data)
using heuristics and Search4Pattern
changing number of basic concepts
Using ANNs
changing competence in intermediate concepts
Influences of the Q matrices and Sizes of Student Populations
(unmatched Q-matrices, groupInfluence and fuzziness in training and test data)
Using SVMs
changing competence in the basic and the target concepts
Influences of the Q Matrices
(analytical analysis)

Fig. 12. Organisation of sections Idealistic Evaluations and More Realistic Evaluations

The main purpose of More Realistic Evaluations is to explore what might happen when the reported methods are applied in certain challenging scenarios. Using different choices of *groupInfluence*, *fuzziness*, and *Q*-matrices in creating training and test data may show us how the proposed methods will perform when we have incorrect expectations of the students about whose learning patterns we have an interest. In the first subsection, we see the effects of using different combinations of *groupInfluence* and *fuzziness* in creating the training and test data. In the second subsection, we discuss the effects of using different *Q*-matrices in creating the training and test data.

IDEALISTIC EVALUATIONS

We applied the procedures that we presented in Design of the Experiments to evaluate the influences of different approaches and simulation parameters on the performance of our classifiers. In Effects of Methods and Parameters, we first compare the effectiveness of different approaches with the *Q*-matrix shown in Table 7, and, in Alternative *Q*-Matrices, we study the effects of changing the contents of the *Q*-matrix in Table 7. In Influences of the *Q*-Matrices, we discuss a viewpoint for quantitatively analysing the influence of the *Q*-matrices.

Effects of Methods and Parameters

In this subsection, we compare the effectiveness of applying the approaches that we discussed in Methods for Model Selection. We report the experimental results of guessing of the learning patterns using the heuristic and the search methods first, with the ANNs next, and then with the SVMs. We used the structures in **4basics**, the *Q*-matrix in Table 7, and 36 different combinations of *groupInfluence* and *fuzziness* in each of the experiments discussed in this subsection.

Using heuristics and Search4Pattern

We tested Heuristic 1 and **Search4Pattern** with the procedures shown in Figures 8 and 9, respectively. We can compare the accuracies achieved by these procedures using the charts shown in Figure 13. In both charts, the horizontal axes show the decimal parts of the values of *fuzziness*. The legend for a curve shows both the origin of the data and the decimal parts of *groupInfluence*. For instance, "s05" indicates that the search method was used when *groupInfluence* was 0.05, and "h15" indicates that Heuristic 1 was used when *groupInfluence* was 0.15. The vertical axis shows the accuracy as explained in Measurement of Quality. Here, each point in the charts shows the accuracy of a subset of the experiment, and the accuracy is the percentage of the correct prediction of the hidden structures of the 3000 (=5×600) different networks for a specific combination of *groupInfluence* and *fuzziness*.

Experimental results indicated that both the heuristics-based and search-based methods can predict the correct structure better than 90% of the time when both *groupInfluence* and *fuzziness* does not exceed 0.15. Neither methods performed very well beyond this range, but the search-based method offered similar or better prediction than the heuristics-based one.

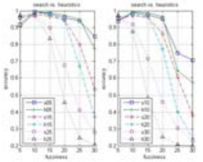


Fig. 13. Using Heuristic 1 and **Search4Pattern** for prediction

Using **Search4Pattern** offered better classification accuracies than using the heuristic, in general. When we applied Heuristic 1, we compared the scores of learning patterns that contained different numbers of parent concepts, e.g., *MI*(*dABC*, *cD*; *dABCD*) and *MI*(*dAB* ,*cC*, *cD*; *dABCD*). Given that we had only estimated values of mutual information, the inequality in the granularity underlying the estimations further infected the performance of our classifiers that used Heuristic 1. In contrast, when we applied **Search4Pattern**, we compared the scores of learning patterns that contained the same numbers of parent concepts, e.g., those in the same columns in Figure 5. The comparisons were relatively more meaningful, and we achieved a better performance in the experiments. Although the formal algorithm looks complex, running **Search4Pattern** took just a few milliseconds, when β was just 4, to obtain a data point on any curves, because the algorithm just compared a few numbers only and we implemented the algorithm in C. (The running time was measured on a Windows XP machine with Pentium IV 2.8G CPU and 1.24G RAM.)

In the most challenging case when both *groupInfluence* and *fuzziness* were set to 0.30, the accuracy for the heuristics-based method was only 20%, a figure that would be obtained for a random guess among five alternatives. This is an interesting observation, because we allowed our classifier to guess any of the 14 possible structures in Figure 5. If we had made random guesses among the 14 possible answers, we could have seen as low as 7% in accuracy. Hence, 20% in accuracy was *not* the result of a random guess.

This phenomenon is related to two factors. The first factor is that we used basic concepts as the parent concepts of all of the composite concepts, except *dABCD*, as discussed in Generating Datasets. The second one is that the basic concepts must be prerequisites of *dABCD*, although they might not be the parent concepts of *dABCD*. As a consequence, computing the MI measure as defined in Mutual Information-Based Methods allowed a special favour to the structure A_B_C_D, so the accuracy happened to be equal to the results of random guessing when the possible answers included A_B_C_D (cf. Figure 7). If we had excluded the cases of A_B_C_D from **4basics**, the accuracy would fall below 25%, which would be the result of random guessing if there were only four possible answers.

It is also interesting to find that, when the degree of *fuzziness* reduced, the accuracy did not improve in every case (the upper left corner of the charts). After examining the confusion matrices, we found that our programs had misclassified many A_B_CD and AB_C_D structures as AB_CD, which was not included in **4basics**. Intuitively, this type of error was understandable because the structure AB_CD was very close to those of the actual answers. In addition, another reason was revealed by an inspection of the contents of the *Q*-matrix in Table 7. Many student groups were competent in *dAB* and *dCD* in the *Q*-matrix, so it would have been easy for our classifiers to make incorrect classifications.

Using artificial neural networks

We conducted three sets of experiments with the Neural Network Toolbox in MATLAB. As stated in Steps of the Experiments (cf. Figure 11), we created 3000 instances for each subset of an experiment. Hence, after adding class labels to the instances, we could use 2500 instances as the training data. Training and test data were stratified (Witten & Frank, 2005), so the training data included 500 instances of each of the five competing structures in **4basics**. The remaining 500 instances were used for testing, and we calculated the percentages of correct classifications for the classifiers.

We ran experiments that used probabilistic neural networks (PNNs) and backpropagation networks (BPNs) without doing principal component analysis (PCA) (Jolliffe, 2002). We also ran experiments that used BPNs after doing PCA. When we ran PCA over the features, we eliminated principal components that contributed less than 0.5% to the total variation in the training data.

Our BPNs had three layers. There was an output unit for each possible learning pattern, and an input unit for each feature in the training instances. Let ν be the number of feature (input) units, we used $5+\lfloor \nu/2 \rfloor$ hidden units in the BPNs. We used the *tansig* transfer function for the hidden and output units, and ran the *traingdx* training function for 1000 epochs when the prediction errors on the training data levelled off and remained stable for a large number of epochs. We ran the training processes multiple times and recorded the performance of the best performing models. This is a *random restart* strategy, cf. (Russell & Norvig, 2002), for avoiding local minima, that could be induced by poor initial settings of link weights in training ANNs. In the test stage, we chose the competing structure whose corresponding output unit had the largest output value.

When experimenting with PNNs, we used the default settings in MATLAB. In these experiments, we used the default radial basis function for our PNNs in MATLAB:

$$radbas(x_i, x_j) = \exp(-(\|x_i - x_j\| \times bias)^2),$$

where x_i and x_j represent two instances. The *bias* (or sometimes called the *spread* in the MATLAB manual) for our PNNs was the default value, 0.1. Training PNNs was much faster than training BPNs in MATLAB, as the PNNs would choose the most probable class as the hidden structure.

The charts shown in Figure 14 depict the performance we achieved with different ANNs. The titles of these charts indicate how we conducted the experiments. The data in the leftmost chart came from the classifier that employed probabilistic neural networks (pnn) and for which we did not preprocess the attributes using principal component analysis (nopca). The data in the middle chart came from the classifier that employed back-propagation networks (bpn). The data in the rightmost chart came from the same classifier that we used to create the middle chart, but we pre-processed the training and test instances using the PCA and ignored components that contributed less than 0.5% to the total variation of the training data. The horizontal and vertical axes are the same as those in the charts shown in Figure 13. The legends show the decimal parts of the *groupInfluence* used in the experiments that produced the data.

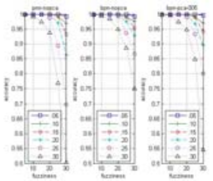


Fig. 14. Using BPNs and PNNs for prediction

All three charts in Figures 14 show the general trend that the accuracy degraded with increasing *groupInfluence* and *fuzziness*. When these parameters were small, it was possible to achieve high accuracy. Clearly, using BPNs without doing PCA offered the best performance. In the most challenging case when both *groupInfluence* and *fuzziness* were set to 0.3, we achieved 75% in accuracy. Carrying out PCA before training BPNs saved a significant portion of training time, as did using PNNs. It took, respectively, approximately 49 and 37 seconds to finish the experiments when *groupInfluence* and *fuzziness* were both set to 0.30 in the middle (nopca) and the right (pca) chart in Figures 14. The execution time was measured on a Windows XP machine with MATLAB 2007a, Pentium IV 2.8G CPU, and 1.24G RAM. Although we reduce the running time of our classifiers by simplifying the data instances with PCA, the resulting sacrifice in accuracy can be undesirable in educational applications.

Comparing the charts in Figures 13 and 14 provides a clue for the net effects of the prior information for training ANNs. All the curves in Figure 14 will lie above their corresponding curves in Figure 13 if we overlap the charts. The difference reached 55% (=75%-20%) when both *groupInfluence* and *fuzziness* were 0.3. The increase in accuracy justifies the extra effort that is necessary for collecting the prior information about the set of hidden structures and the *Q*-matrix. In addition, note that curves were also smoothed near the upper left corner. When the domain experts provide a correct set of the possible learning patterns, our algorithm reduces the chance of making unnecessary errors.

Using supported vector machines

We conducted our experiments with functions provided in LIBSVM (Chang & Lin, 2001). We used the c-SVC type of SVMs in all experiments, and tried three different kernel functions, including polynomial (c-svm-poly), radial basis (c-svm-rb), and sigmoid (c-svm-sm) kernels as they are defined in LIBSVM. (Note that we used the symbol γ in a different context in A Formulation with Bayesian Networks. The γ in the SVM kernel functions denotes a free variable.)

$$\text{polynomial function:} \quad K(x_i, x_j) = (\gamma x_i^T x_j)^3 \tag{4}$$

$$\text{radial basis function:} \quad K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \tag{5}$$

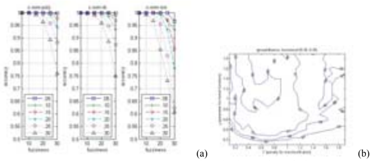


Fig. 15. Using SVMs for prediction: (a) experimental results (b) search for the best parameters

$$\text{sigmoid function:} \quad K(x_i, x_j) = \tanh(\gamma x_i^T x_j) \qquad (6)$$

In fact, although we had adopted some default settings in LIBSVM, we still had to search for the best parameters for SVMs at the time of training the SVMs. In particular, we ran experiments that used different values for the *cost*, *C*, which is the penalty parameter for misclassification, and γ, which appears in the kernel functions listed in Formulae (4), (5), and (6). Different combinations of *C* and γ led to different accuracy in guessing the hidden structures for the test data. We show a contour graph of the accuracy for a subset of the experiment in Figure 15(b), which we created for different combinations of *C* and γ (when *groupInfluence* and *fuzziness* were 0.30 and 0.25, respectively). The numbers on the curves indicate the accuracies in percentage. In our experiments we tried combinations of *C* and γ from values in {0.1, 0.2, …, 1.9}, and used the best accuracy for the test data in these 361 (=19×19) cases when we prepared the charts in Figure 15(a).

The charts shown in Figure 15(a) show the experimental results. The vertical axis, the horizontal axis, and the legend carry the same meanings as those for the charts in Figure 14. The titles of the charts indicate the types of SVMs that were used in the experiments. Similar to the charts in Figure 14, all three charts in Figure 15 show the general trend that the accuracy degraded with increasing *groupInfluence* and *fuzziness*. When these parameters were small, it was possible to achieve high accuracy.

The effort in collecting information about the possible set of hidden structures and the *Q*-matrix proved rewarding again. Comparing the curves for the related experiments in Figures 13 and 15(a) shows us that significant improvements were achieved by using the SVMs. In the middle chart in Figure 15(a), the accuracy stays above 0.75 even when *groupInfluence* and *fuzziness* are 0.3. The heuristics-based method achieved only 0.2 in accuracy under the same situation. The problem that occurred in the upper left corners of the charts in Figure 13 was also absent. The charts shown in Figure 15(a) indicate that using polynomial and radial basis kernels gave almost the same accuracy, and both performed better than the sigmoid kernel. However, it took a longer time for us to train an SVM when we used the polynomial kernel. For instance, it took, respectively, 118 and 18 minutes to try 361 different combinations of *C* and γ when *groupInfluence* and *fuzziness* were both 0.3 in the left chart and in the middle chart in Figure 15(a).

Table 8. Accuracy versus F measures (shown in the form of accuracy/F)

		groupInfluence					
		0.05	0.10	0.15	0.20	0.25	0.30
fuzziness	0.05	1.0000/1.0000	1.0000/1.0000	1.0000/1.0000	1.0000/1.0000	1.0000/1.0000	1.0000/1.0000
	0.10	1.0000/1.0000	1.0000/1.0000	1.0000/1.0000	1.0000/1.0000	1.0000/1.0000	0.9960/0.9960
	0.15	1.0000/1.0000	1.0000/1.0000	1.0000/1.0000	1.0000/1.0000	1.0000/1.0000	0.9900/0.9901
	0.20	1.0000/1.0000	1.0000/1.0000	1.0000/1.0000	0.9980/0.9980	0.9940/0.9941	0.9600/0.9610
	0.25	1.0000/1.0000	0.9960/0.9960	0.9980/0.9980	0.9840/0.9844	0.9480/0.9516	0.8860/0.8952
	0.30	0.9980/0.9980	0.9860/0.9865	0.9680/0.9697	0.9340/0.9374	0.8740/0.8854	0.7500/0.7719

The best performing ANN and SVM models seemed to have achieved the same accuracy. Comparing the charts in Figures 14 and 15(a), we find that different ways of using the ANN and SVM techniques may offer different qualities in prediction. However, the middle charts in Figures 14 and 15(a) suggest that the best-performing ANNs and SVMs offered almost the same performance.

Table 8 shows the actual values of the data that we used to plot the middle chart in Figure 15(a) as well as their corresponding F measures (Witten & Frank, 2005). The precision rates and the recall rates were calculated for each of the five classes in **4basics** first, and the F measure for each of these five classes was set to the average of the precision rate and the recall rate for that class. The reported F measure in the table is the average of the F measures for the five classes. The observed accuracy and F measures were close, as we noted in Measurement of Quality.

Depending on the values of *groupInfluence* and *fuzziness*, it took different lengths of time to run each of the experiments, even when we were using the same setting for SVMs. For instance, it took 206 seconds to compare the effects of 361 combinations of *C* and γ when *groupInfluence* and *fuzziness* were both 0.05, and it took us 1083 seconds when *groupInfluence* and *fuzziness* were both 0.3. On average, we spent nearly 3 seconds for trying out the effects of a combination of *C* and γ when *groupInfluence* and *fuzziness* were 0.3. (We measured the execution time on a Windows XP machine with LIBSVM 2.84 in C, Pentium IV 2.8G CPU, and 1.24G RAM.)

Alternative *Q*-Matrices

In this subsection, we investigate the effects of using different *Q*-matrices in the experiments. Since the experimental results discussed in Effects of Methods and Parameters suggested that using ANNs and SVMs could provide a similar performance, we used only the best performing SVMs, i.e., c-svm-rb in Figure 15(a) in this subsection. (We made this choice partially because LIBSVM is a freeware that we can run on many machines. In contrast, we have only one license for using MATLAB.) We will change (1) the number of basic concepts that are included in the target composite concepts in the first sub-subsection, (2) the way in which we set the values for other intermediate concepts in the second and (3) the competence patterns for the basic and the target composite concepts last.

Effects of number of basic concepts

We ran experiments with **3basics**, the right *Q*-matrix in Table 1, and 36 combinations of *groupInfluence* and *fuzziness*. Notice that we must use different *Q*-matrices for the structures in **3basics** and **4basics**. Hence the differences in the accuracy of the resulting classification cannot be attributed exclusively to the number of basic concepts.

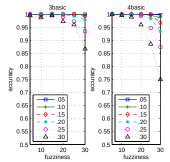


Fig. 16. Using the *Q*-matrices contained in Table 7 and in the right part of Table 1

Figure 16 shows the differences in the accuracy of classification when we reduced the number of basic concepts in the experiments. The chart titled 4basic is a duplicate of the c-svm-rb chart in Figure 15(a), and the chart titled 3basic was produced from the new experiment. When *groupInfluence* and *fuzziness* are large, learning the way students learn *dABC* is easier than learning how they learn *dABCD* by a margin of nearly 10% in this pair of experiments. When *groupInfluence* and *fuzziness* are small, reducing the number of basic concepts did not yield obvious differences.

All else being equal, a problem that considers three basic concepts is not as complex as one that considers four basic concepts in nature. Hence, what we have observed should not be surprising. However, experimental results are affected by many factors including those that we will discuss in the remainder of this paper, so we cannot claim that problems that consider only three basic concepts must be easier than those that consider four basic concepts.

Effects of competence patterns for the intermediate concepts

We refer to the composite concepts that can serve as the parent concepts of the target composite concept as the *intermediate* concepts. When we study the learning problems of *dABCD*, there can be 10 intermediate concepts, including those composite concepts that involve two or three basic concepts, i.e., *dAB*, *dAC*, …, and *dBCD* in Table 7.

Recall that, in the *Q*-matrix in Table 7, we assumed that all of the recruited students were competent in the basic concepts and were able to integrate the parent concepts of *dABCD*. Based on such a setting, the problem of changing the competence patterns for the intermediate concepts can be redefined as one of choosing the number of student groups in the *Q*-matrix. Hence, we selected some student groups that we used in Table 7 as the *Q*-matrices that we used in the new experiments.

We conducted experiments in which the *Q*-matrices contained one, two, four, eight, and sixteen student groups, and we called the *Q*-matrices used in these experiments Q_1, Q_2, Q_4, Q_8, and $Q_{16,}$ respectively. Hoping to do a more meaningful comparison between results of different experiments, we made Q_i a subset of Q_j when i<j. Namely, a student group must belong to Q_j if that student group belongs to Q_i, for any i<j. The first group, g_1, was the obvious choice for Q_1 because it represented the group of perfect students. For easier reference and comparison, Q_{16} is a complete duplicate of the *Q*-matrix in Table 7. This was how we determined Q_1 and Q_{16} in Table 9.

We anticipated that student subgroups that had stronger contrasting competence patterns would help our classifiers make correct decisions, so we chose a student group that was most different from g_1 to be included in Q_2. We computed the *distance* between all pairs of student groups based on the Euclidean distance between the competence patterns of two student groups. Equation (7) shows the definition for the current experiment, where 15 is the number of different concepts in the *Q*-matrix in Table 7. Based on this notion of distance, g_{16} was chosen to be the second student group in Q_2.

$$distance(g_i, g_j) = \sqrt{\sum_{k=1}^{k=15} (q_{g_i,k} - q_{g_j,k})^2} \qquad (7)$$

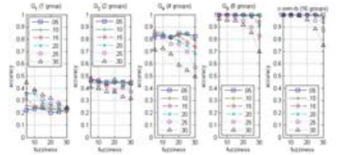


Fig. 17. Effects of changing the competence patterns for the intermediate concepts

Table 9. Competence patterns in four sub-matrices of the *Q*-matrix in Table 7

	SID	competence in (integrating) concepts														
		cA	*cB*	*cC*	*cD*	*dAB*	*dAC*	*dAD*	*dBC*	*dBD*	*dCD*	*dABC*	*dABD*	*dACD*	*dBCD*	*dABCD*
Q_1	g_1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q_2	g_1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	g_{16}	1	1	1	1	1	0	0	0	0	1	0	0	0	0	1
Q_4	g_1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	g_7	1	1	1	1	1	1	1	0	0	0	1	0	0	1	1
	g_9	1	1	1	1	0	0	0	1	1	1	0	1	1	1	1
	g_{16}	1	1	1	1	1	0	0	0	0	1	0	0	0	0	1
Q_8	g_1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	g_3	1	1	1	1	0	1	0	1	0	1	1	1	0	1	1
	g_4	1	1	1	1	1	0	0	0	0	1	1	1	0	0	1
	g_5	1	1	1	1	1	0	1	0	1	0	1	0	1	1	1
	g_7	1	1	1	1	1	1	1	0	0	0	1	0	0	1	1
	g_9	1	1	1	1	0	0	0	1	1	1	0	1	1	1	1
	g_{13}	1	1	1	1	0	0	1	1	0	0	0	0	1	1	1
	g_{16}	1	1	1	1	1	0	0	0	0	1	0	0	0	0	1
Q_{16}	…	duplicate the contents in Table 7														

We then calculated the distances between all pairs of student groups except g_1 and g_{16} that were in Q_2. We put the pair, g_7 and g_9, that had the largest distance between them into Q_4 and the pairs that had the second and the third largest distance into Q_8. Table 9 shows the resulting Q_1, Q_2, Q_4, Q_8, and Q_{16}.

We used the structures in **4basics**, the new *Q*-matrices, and 36 combinations of *groupInfluence* and *fuzziness* in the new experiments. The charts in Figure 17 depict the results of this sequence of experiments. From the left to the right, the results came from the experiments in which we used Q_1, Q_2, Q_4, and Q_8 to create the training and test data. The rightmost chart is a duplicate of the c-svm-rb chart in Figure 15(a).

The results show some interesting trends. Although we used perfect students in Q_1, it was very difficult to learn how students learn when we collected data exclusively from perfect students. This phenomenon became less surprising when we came to believe that it is hard to tell how students learn if they are competent in all relevant concepts. Hence, we consider this simulated result interesting because the simulated results taught us something that we had not expected.

As we added more and more contrasting pairs of student groups into the *Q*-matrices, the average accuracy improved from the leftmost to the rightmost chart. The curves in the individual chart move upward gradually. In addition, we see that the curves for cases that used smaller *groupInfluence* and *fuzziness* do not necessarily fall below the curves for cases that used larger *groupInfluence* and *fuzziness* in an individual chart. This is particularly so in the charts on the left side of Figure 17. This observation shows that the intermediate patterns are as influential as *groupInfluence* and *fuzziness* on the experimental results.

Effects of competence in the basic and the target composite concepts

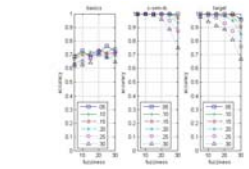


Fig. 18. Effects of changing the competence patterns for the basic and the target composite concepts

In conductin vised part of
the *Q*-matrix isic concepts
when we exp target com-
posite conce et composite
concept. Tab ng the influ-
ence of the b ose of the *Q*-
matrix listed 10. We set
the values fo most of the
contents of (shown in the
right part of bitrarily. We
employed, re and the tar-
get composit 36 combina-
tions of *grou*

The bas s that we ob-
tained when rt from Fig-
ure 15(a) for

The resu beginning of
Design of the Experiments. The differences between the basics and the c-svm-rb charts suggest that it will not be very fruitful if we recruit students who are not competent in the basic concepts in order to study how they might learn the target composite concept. The differences between the target and the c-svm-rb charts are not as salient as those between the basics and the c-svm-rb charts, but the trends still support that we should recruit students who appear to be competent in the target composite concept.

Table 10. Competence patterns in the *Q*-matrices for testing the influence of the basic and the target composite concepts

SID	Q_b				Q_t	SID	Q_b				Q_t
	cA	*cB*	*cC*	*cD*	*dABCD*		*cA*	*cB*	*cC*	*cD*	*dABCD*
g_1	1	1	1	1	1	g_9	0	1	1	1	1
g_2	1	1	1	0	0	g_{10}	0	1	1	0	0
g_3	1	1	0	1	1	g_{11}	0	1	0	1	1
g_4	1	1	0	0	0	g_{12}	0	1	0	0	0
g_5	1	0	1	1	1	g_{13}	0	0	1	1	1
g_6	1	0	1	0	0	g_{14}	0	0	1	0	0
g_7	1	0	0	1	1	g_{15}	0	0	0	1	1
g_8	1	0	0	0	0	g_{16}	0	0	0	0	0

Influen

Table 11. Feasibility of learning patterns changed with *Q*-matrices in Table 9

	SID	Whether the student group can actually use the learning pattern				
		ACD_B	ABD_C	AB_C_D	A_B_CD	A_B_C_D
Q_1	g_1	1	1	1	1	1
Q_2	g_1	1	1	1	1	1
	g_{16}	0	0	1	1	1
Q_4	g_1	1	**1**	1	1	1
	g_7	0	**0**	1	0	1
	g_9	1	**1**	0	1	1
	g_{16}	0	**0**	1	1	1
Q_8	g_1	1	1	1	1	1
	g_3	0	1	0	1	1
	g_4	0	1	1	1	1
	g_5	1	0	1	0	1
	g_7	0	0	1	0	1
	g_9	1	1	0	1	1
	g_{13}	1	0	0	0	1
	g_{16}	0	0	1	1	1
Q_{16}	…	duplicate the contents for corresponding columns in Table 7				

In Alte s based
on the ıps that
differec esulting
Q-matri
Ta le 9. To
make tl student
groups, ıembers
of a pa earning
dABCD s that it
is possi ıly only
AB_C_ d *fuzzi-*
ness ter
Th For in-
stance, 11, be-
cause, a to inte-
grate th Q_4 and
ABD_C is concepts of
dABD, acc exhibit the
learning pa

Table 12. Distances between learning pattern for the data for Q_8 in Table 11

	ACD_B	ABD_C	AB_C_D	A_B_CD	A_B_C_D
ACD_B	0.00	2.00	2.24	2.24	2.00
ABD_C	2.00	0.00	2.24	1.00	2.00
AB_C_D	2.24	2.24	0.00	2.00	1.73
A_B_CD	2.24	1.00	2.00	0.00	1.73
A_B_C_D	2.00	2.00	1.73	1.73	0.00

Table e perform-
ances of o *x* column
in Table 1 ement, i.e.,
V_{ABD_C,Q_4} *dABCD* by
combining nd *dABCD*
columns for g_9 are equal to 1 in Q_4 in Table 9.)

With this notion, we can define the distance between any two learning patterns, p_j and p_k, in a *Q*-matrix by Formula (8) where $\lambda(Q_i)$ denotes the number of student groups in the *Q*-matrix, Q_i.

$$distance_{Q_i}(p_j, p_k) = \sqrt{\sum_{l=1}^{l=\lambda(Q_i)} \left(V_{p_j,Q_i}(l) - V_{p_k,Q_i}(l)\right)^2} \quad (8)$$

Table 12 shows the distances between the five learning patterns based on the data for Q_8 in Table 11. (We computed a table in the same format for Q_1, Q_2, Q_4, and Q_{16}, but do not show them here.) Let $\hat{P}$ denote the set of learning patterns provided by the domain experts for the learning problem. The total distance between learning patterns in $\hat{P}$ in a particular *Q*-matrix is defined in Equation (9).

$$total\_distance(Q, \hat{P}) = \sum_{p_j \in \hat{P}, p_k \in \hat{P}, p_j \neq p_k} distance_Q(p_j, p_k) \quad (9)$$

The function of Equation (9) is very simple. Applying the equation for Q_8 in Table 11, we simply compute the sum of the numbers in Table 12, and the result is 38.36. Since there must be at most 20 non-zero terms in the particular example the average distance for Q_8 is 39.36÷20=1.92. We can easily verify that the average distance for learning patterns in Q_1, Q_2, Q_4, Q_8, and Q_{16} in Table 11 are, respectively, 0.00, 0.60, 1.17, 1.92, and 2.53. (We could have chosen to divide the total distance by 25 because there are 25 terms in Table 12. This choice would not affect the ordering, since every total distance was divided by the same quantity.)

Interestingly, we can verify that the average accuracies depicted in the charts shown in Figure 17 increased in line with the average distances of the learning patterns in Q_1, Q_2, Q_4, Q_8, and Q_{16}. Hence, if we have information about the recruited students and if we can control the recruitment of the students, increasing the average distance between the competing learning patterns may improve our chances to find the actual learning pattern.

Notice that the average distance between the competing patterns is not the only factor that affects the achieved accuracy. The average distance between competing patterns in Q_b and Q_t in Table 10 are 0.00 and 1.92, respectively. Again, we achieved higher accuracies when we used Q_t. However, our classifiers performed differently when we used Q_2 in Table 9 and Q_b in Table 10, even though the average distances between the competing patterns in these *Q*-matrices are both 0.00. Other reasons that make the competing patterns in a *Q*-matrix differentiable will also affect the experimental results.

Moreover, we must be reminded that the zeros and ones in the *Q*-matrices do not deterministically influence the simulated students' behaviours, although the distances computed with Formula (9) remain related to the differences between the learning patterns. We should take into consideration the magnitude of *groupInfluence*, because it affects the relationships between competence patterns and group members, as discussed in Generating Student Records. Furthermore, we have assigned *group slip* and *group guess* (Liu, 2005) to the same value, i.e., *groupInfluence*, in the experiments that we have discussed so far. If we set these two parameters to different values, the Mahalanobis distance (Duda et al., 2001) would be more appropriate to use in place of the Euclidean distance.

MORE REALISTIC EVALUATIONS

In the previous section, we assumed that we were able to provide perfect information about the contents of the *Q*-matrices for the recruited students. The purpose of the experiments was to compare the effectiveness of different classification techniques, of the influences of the simulation parameters, and of different *Q*-matrices.

In this section, we investigate the effects of two types of deviations from the perfect conditions. In the first subsection, we assume that the *groupInfluence* and *fuzziness* used by the simulator are different from those exhibited by the real students (i.e., in the test data), while the experts provide perfect *Q*-matrices. In the second subsection, we relax the assumption of the need to acquire perfect *Q*-matrices, and assume that the *Q*-matrices that we conjecture do not necessarily contain the actual competence patterns of real students.

Influences of the Simulation Parameters

We conducted experiments to examine the influence of incorrect guesses of *groupInfluence* and *fuzziness* on the prediction of the learning patterns. To this end, we continued to use the networks shown in Figure 7(b) and the *Q*-matrix contained in Table 7 as discussed at the beginning of Design of the Experiments when we created simulated data with the steps outlined in Figure 11. In all of the experiments that we discussed in Idealistic Evaluations, we used the same combination of *groupInfluence* and *fuzziness* to generate both the training and test data. In the experiments discussed in this subsection, we used different combinations of *groupInfluence* and *fuzziness* when we created training and test data.

Recall that there can be 36 combinations of *groupInfluence* and *fuzziness* when we set these variables to values in {0.05, 0.10, 0.15, 0.20, 0.25, 0.30}. When we intentionally chose different combinations of *groupInfluence* and *fuzziness* in generating training and test data, we could have 1296 (=36×36) different experiments. Hence, we must choose only certain of these possible experiments. Due to this constraint, we set both *groupInfluence* and *fuzziness* to 0.10 and 0.25 when we created training data in two different experiments, and continued to set *groupInfluence* and *fuzziness* to all 36 different combinations when we created test data. Hence, we will see the experimental results of 72 cases.

The statistics in Table 13 show the performance of our classifiers under these relatively unfavourable circumstances. The data in the left half of Table 13 came from the experiment when we set both *groupInfluence* and *fuzziness* to 0.10 when we created data for training the SVMs. We created 36 sets of test data, setting *groupInfluence* and *fuzziness* to 0.05, 0.10, 0.15, 0.20, 0.25, and 0.30. Each experiment used a different combination of *groupInfluence* and *fuzziness*, and included 2500 (=500×5) training instances and 500 (=100×5) test instances. The data in Table 13 shows the accuracies of the trained SVMs. For instance, reading the data from the left half of Table 13, we see that our classifiers achieved 75% accuracy when *groupInfluence* and *fuzziness* were both 0.10 for the training data and when *groupInfluence* and *fuzziness* were, respectively, 0.15 and 0.20 for the test data. We obtained the data shown in the right half of Table 13 with the same procedure, but we trained the SVMs with the data that we created by setting both *groupInfluence* and *fuzziness* to 0.25.

Table 13. Influence of (unmatched) simulation parameters *groupInfluence* and *fuzziness*

groupInfluence for test data	*fuzziness* for test data											
	0.05	0.10	0.15	0.20	0.25	0.30	0.05	0.10	0.15	0.20	0.25	0.30
0.05	1.00	1.00	1.00	1.00	0.89	0.54	0.20	0.22	0.29	0.35	0.65	0.87
0.10	1.00	1.00	1.00	0.97	0.41	0.33	0.20	0.22	0.31	0.57	0.98	0.97
0.15	1.00	1.00	0.99	**0.75**	0.37	0.23	0.24	0.27	0.43	0.86	0.96	0.97
0.20	1.00	1.00	0.84	0.39	0.23	0.21	0.24	0.33	0.79	0.93	0.98	0.90
0.25	0.99	0.85	0.41	0.23	0.20	0.20	0.47	0.63	0.86	0.96	0.96	0.76
0.30	0.81	0.39	0.23	0.20	0.20	0.20	0.44	0.78	0.89	0.94	0.82	0.55
	(*groupInfluence*, *fuzziness*) for training data											
	(0.10, 0.10)						(0.25, 0.25)					

Recall that we assumed that there are sources from which we can acquire the information about the *Q*-matrix, the candidate structures, and the values for *groupInfluence* and *fuzziness*. In contrast, the *groupInfluence* and *fuzziness* that we used to create the test data were assumed to represent the characteristics of real students. It is important to also recall that the values of *groupInfluence* and *fuzziness* confine the ranges of the data in the generated conditional probability tables. For instance, as we explained in Generating Student Records, the probability of making an unintentional mistake (i.e., the *slip* cases) will be between 0 and 0.10 when *fuzziness* is 0.10. We *did not* set the chance of slip to 0.10 when *fuzziness* was 0.10.

Statistics in Table 13 show the importance of acquiring a correct combination of *groupInfluence* and *fuzziness*. Neither the classification accuracies in the left or in the right part of the table can compete with the experimental results that we observed when we assumed the availability of correct *groupInfluence* and *fuzziness*, e.g., those depicted in Figure 15.

The data in Table 13 also indicate that we achieved better results when the guessed *groupInfluence* and *fuzziness* are closer to the actual *groupInfluence* and *fuzziness*. It is interesting to note that the proposed method showed limited robustness. We could achieve good prediction accuracy even when the *groupInfluence* and *fuzziness* that we used to generate the training and the test data were not the same, although the classification accuracy deteriorated with the increasing divergence between the guessed and the actual values of *groupInfluence* and *fuzziness*.

Influences of the *Q*-Matrices and Sizes of Student Populations

So far, we have assumed that we can use perfect *Q*-matrices for generating training data. What might occur if this assumption does not hold? In order to make the design of experiments more complete, we conducted experiments under such special situations, and we discuss what we observed in this subsection.

When we use different *Q*-matrices to generate training and test data, we are simulating the situation in which we have imprecise expectations about students' competence patterns. Such imprecision will have adverse effects on the performance of machine-learning based methods.

It is not easy to find a pair of *Q*-matrices that are of general interest, however. As discussed in Computational Complexity, we can have 2^{32768} different *Q*-matrices when we consider problems that include only 4 basic concepts. Selecting which two different *Q*-matrices from this enormous amount of different choices for experiments can become a problem itself. Other researchers have faced this kind of selection problem as well, e.g., DiBello et al. (1995) (pages 365 and 370) discussed issues related to the choice of *Q*-matrices for cognitive measurement problems.

In this subsection, we discuss the experimental results that we obtained when we used the *Q*-matrix shown in Table 14 to create test data, while the data for training SVMs were created with other *Q*-matrices. The contents of Table 14 were chosen such that every learning pattern shown in Figure 7(b) can be exercised by at least one student group in the table (cf. the discussion in Influences of the *Q*-Matrices). More specifically, g_2 and g_3, respectively, support ACD_B and ABD_C; g_4 and g_5, respectively, support AB_C_D and A_B_CD; g_6 supports A_B_C_D; g_1 supports all these learning patterns (i.e., g_1 supports all patterns in **4basics**); and g_7 represents a group of students who are not competent in *dABCD*.

We needed two *Q*-matrices to compare and show the effects of *Q*-matrices on the classification accuracy. Since we have used the *Q*-matrix in Table 7 in many of our experiments, it was natural to continue to use this *Q*-matrix to create the training data in this section. Notice that the *Q*-matrix shown in Table 14 shares only one competence pattern with the *Q*-matrix shown in Table 7. This setup is meant to simulate the situation under which we can guess only one of the competence patterns of the students. The other *Q*-matrix was chosen so that it included the entirety of the contents of the *Q*-matrix that was used to create the test data. This was achieved by combining the *Q*-matrices shown in Table 7 and Table 14. Based on whether the *Q*-matrix used for creating the training data also included the *Q*-matrix used for creating the test data, we call the first kind of experiments `NotIncluded` (i.e., only Table 7) and the second kind of experiments `Included` (i.e., the union of Table 7 and Table 14).

Recall that the selection of *groupInfluence* and *fuzziness* influences the classification accuracy. When we created training and test data with different *Q*-matrices in the `Included` and `NotIncluded` experiments, the selected *groupInfluence* and *fuzziness* affected the experimental results as well.

With the chosen *Q*-matrices, *groupInfluence*, and *fuzziness*, we created and conducted the experiments. Experimental results indicated that we obtained higher classification accuracy in the `Included` experiments than in the `NotIncluded` experiments. When both *groupInfluence* and *fuzziness* were 0.1 in creating both the training and test data, there were larger differences in classification accuracy in these two kinds of experiments; when *groupInfluence* and *fuzziness* were 0.25, the differences reduced. As we reported in previous sections, larger *groupInfluence* and *fuzziness* made the classification more difficult, and could have contributed to the shrunk differences in accuracy when we used different *Q*-matrices in the experiments.

We were also curious about the influences of the size of student population on the classification accuracy. Hence we created data sets with different sizes of student populations in the experiments. More specifically, we conducted and compared the experimental results that were obtained when there were about 625 and 375 (=0.6×625) simulated students for each student group in two sets of experiments. (Due to the randomness that we reported in Generating Student Records, we cannot control the exact number of students in a student group.) We repeated both the `Included` and `NotIncluded` experiments, and observed that using a greater number of simulated students in training the SVM classifiers helped us achieve better classification accuracy.

Table 14. Competence patterns in the *Q*-matrix for the new experiments

SID	Competence in (integrating) concepts														
	cA	*cB*	*cC*	*cD*	*dAB*	*dAC*	*dAD*	*dBC*	*dBD*	*dCD*	*dABC*	*dABD*	*dACD*	*dBCD*	*dABCD*
g_1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
g_2	1	1	1	1	0	0	0	0	0	0	0	0	1	0	1
g_3	1	1	1	1	0	0	0	0	0	0	0	1	0	0	1
g_4	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1
g_5	1	1	1	1	0	0	0	0	0	1	0	0	0	0	1
g_6	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1
g_7	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0

It is perhaps not very surprising to have observed that the results in the `Included` experiments were better than those in the `NotIncluded` experiments and that using more simulated students led to better classification results. However, it was useful to know that the simulated results agreed with the intuition.

More interestingly, we found that, when the *Q*-matrix, *groupInfluence*, and *fuzziness* that were used in creating the training and test data did not match very well, SVMs did not necessarily provide a better performance than **`Search4Pattern`**. (In Effects of Methods and Parameters, SVMs provided a better performance when these influential factors were the same for creating both the training data and the test data.) The setup for creating the test data influenced the performance of both the SVMs and the **`Search4Pattern`** algorithm, but the setup for creating the training data affected only the SVMs because the performance of **`Search4Pattern`** does not rely on the training procedure at all. When the assumptions adopted for creating training data differed very much from the assumptions adopted for creating test data, **`Search4Pattern`** may offer a better performance.

SUMMARY AND DISCUSSION

We wrap up this paper by summarising our findings and referring to additional related work.

A Lesson for Learning about Learning

Experimental results indicate that learning the learning patterns with students' item response patterns is not easy but it is possible. In Idealistic Evaluations, we assumed that we obtained exact information about the *Q*-matrices, *groupInfluence*, and *fuzziness*. We observed that it was quite simple to apply Heuristic 1 and **`Search4Pattern`**, but that they did not perform as well as the model-based methods when we can train SVMs and ANNs with exact information about *Q*-matrices, *groupInfluence*, and *fuzziness*. The best performing SVMs and ANNs offered similar prediction accuracies in our experiments. We found that a good selection of the student groups, i.e., the contents of the *Q*-matrix, will affect how well we can learn about students' learning patterns. Hence, we discussed two different ways to analyse the quality of the *Q*-matrices.

We employed both *groupInfluence* and *fuzziness* in the simulation to summarise the influence of other factors in the study. This is similar to the *residual ability* discussed in (DiBello et al., 1995, page 362). In addition, the actual value of *fuzziness* is related to the *positivity* in the Unified Model (DiBello et al., 1995, page 369). It is easy to prove that the positivity of an item increases with the value of *fuzziness* when *fuzziness* is between 0 and 0.5, which is the case in all our simulations. Hence, in general, it becomes increasingly difficult to identify students' learning patterns correctly as we increase the degree of *fuzziness*.

In More Realistic Evaluations, we relaxed the assumptions for the obtaining of exact information about the *Q*-matrices, *groupInfluence*, and *fuzziness*, and discussed the results of experiments that were conducted under more realistic conditions. Since we have explained, in Computational Complexity, that there can be a myriad of different real world situations, we discussed only some of the possible ones in this paper. We found that whether we used exact information about *Q*-matrices, *groupInfluence*, and *fuzziness* in creating the data for training SVMs significantly influenced the resulting accuracies in experiments. It was interesting to observe that the SVMs did not necessarily outperform **`Search4Pattern`** when we had only imperfect information about *Q*-matrices, *groupInfluence*, and *fuzziness*. Hence, it would definitely be rewarding to seek more exact information about these influential factors.

Although we spent the greater part of our time in this present work in learning the learning patterns for a composite concept that involves four basic concepts, due to computational costs also discussed in (DiBello et al., 1995, page 364), the proposed approach can be applied to learning the learning patterns of more complex composite concepts. What is required is that we should explore the problem space incrementally. Namely, building the structures for simpler composite concepts before trying to learn how students learn more complex composite concepts, where the "simple vs. complex" notion is based on the number of basic concepts included in the composite concepts. With appropriate basic building blocks (sometimes called "objects" in computer science), we will be able to build models for more complex composite concepts.

The use of simulated students in the experiments can appear as a weakness in this study. Under no circumstances can simulated students replace real students for decisive answers. In practice, student modelling for CATs must choose some levels of abstraction for the students in the models, and this practical imperfectness also exists in systems that aim at mental simulation (Weng & Huang, 2006). Nevertheless, we have considered many important factors, including *groupInfluence*, *fuzziness*, competence patterns in the *Q*-matrices, and imperfect guesses in the experiments. Hence, we hope that the scale of the experiments and the reported observations justify the plan of using the simulated results to identify important issues that we may encounter when we use data for real students in future studies.

Obviously, we have not completed all paths of the exploration for this problem in this already lengthy paper. For instance, we mentioned that the search-based method and SVMs complemented each other in the more realistic experiments in More Realistic Evaluations. This observation suggests that one may seek to combine the predictions made by these two methods to achieve better results, which is the so-called *stacking* method as used in the machine learning community (Witten & Frank, 2005). However, we would prefer to explore this opportunity with real students when possible.

More on Related Work

What we have discussed so far involves the issues of (1) the definition of "causal relationships," (2) representing the causal relationships with Bayesian networks, and (3) learning the causal models for variables of interest from indirect evidences. Using the most intuitive interpretation of the word "causal," we believe that being competent in a parent concept, say *dAB*, is a fundamental basis for a student to be able to learn a more complex concept, say *dABC*, under the normal conditions. Hence, we believe that the first issue is not a major concern in this paper.

It cannot be denied that our work is related to the modelling of causal relationships among random variables with the use of only indirect evidence. Inferring the causal relationships among variables of interest can have a wide range of applications. Hence, it should not be surprising that researchers of many disciplines have studied this topic in the literature, e.g., (Rost & Langeheine, 1997; Glymour & Cooper, 1999; Chockler & Halpern, 2004; Halpern & Pearl, 2005). In fact, the learning of graphical structures to represent causal relationships among factors of interest is a common interest in science, and is not limited to the learning of Bayesian networks; for instance, Desjardins (2001) attempts to learn causal structures of chemical reactions with unobservable variables.

Bayesian networks themselves do not necessarily represent causal relationships (Pearl, 1988), but it is possible to represent causal relationships with Bayesian networks (Cooper, 1999; Glymour, 2003). Not all applications of Bayesian networks to student assessment aim at building causal models, and may choose whatever structures that will fulfil the needs of probabilistic reasoning (Millán & Pérez-de-la-Cruz, 2002). For instance, when considering a capability that has multiple prerequisites, all the nodes that represent the prerequisites may be used as the parent nodes of the node that represents the integrated capability, very similar to the approach taken by people who use Concept Maps (Novak, 1990). Some researchers also reverse the arc directions between nodes for the prerequisites and the integrated capability (Millán & Pérez-de-la-Cruz, 2002). Nevertheless, using the nodes that represent the prerequisites as the parent nodes is a more common and intuitive choice (Martin & VanLehn,1995; Millán & Pérez-de-la-Cruz 2002).

Among the research works that adopt Bayesian networks for student modelling, the way we build Bayesian networks is related to Millán and Pérez-de-la-Cruz's (2002) categorising nodes for representing *subjects*, *topics*, *concepts*, and *questions*. In their continuing work, Carmona et al. (2005) showed that adding appropriate links for encoding prerequisite relationships in Bayesian networks can improve the efficiency in adaptive student assessment. Yet another related work considering the prerequisite relationships in Bayesian networks is by Reye (2004), but the structures proposed by Reye are quite different from what we see in this paper and Millán's models.

Our work is also related to the research of multilevel models based on the Item Response Theory (IRT) (Fox, 2005). If we take the relationships between the test items and the basic concepts as the first-level IRT model, and the relationships between the basic concepts and the composite concepts as the higher levels, our models, e.g., the one shown in Figure 7(a), are related to multilevel IRT models. From this viewpoint, our work is an instance of studying how computers can help experts determine the structures of their multilevel IRT models. However, to make our models be more qualified as IRT models, we have to strengthen our models by adding more parameters to quantify the relationships between item responses and competence in concepts.

Given that we chose to represent the prerequisite relationships with Bayesian networks, our problems become instances of learning the hidden structures among the related concepts (Heckerman, 1999; Neapolitan, 2004). Learning the structures directly from data is not an easy task, particularly when the values of many of the random variables are completely missing. The domain knowledge provided by domain experts is believed to help us learn models of higher qualities (AUAI, 2006). Although we cannot explore all the problem instances that one can imagine due to the number possible combinations as discussed in Computational Complexity, we explored some interesting settings in the experiments, and the results show the importance of the quality of source information.

It is possible to learn the prerequisite relationships from some related work, e.g., theory about knowledge structure (Falmagne et al., 2003) and item-to-item knowledge structure (Desmarais et al., 2006). Learning item-to-item knowledge structure requires certain special techniques. Figure 4(a) as discussed in Impacts of Latent Variables is an item-to-item structure that we learned with the PC algorithm in Hugin. Clearly there are places in the structure where we can improve, e.g., the directions of some arcs should be reversed, and interested readers can refer to (Desmarais et al., 2006). Certain recent research results, e.g., (Albert et al., 2007; Guzmán et al., 2007a) report the applications of hierarchical structures are also related to our work.

Concluding Remarks

We have achieved a wide range of classification accuracies in our experiments, depending on the quality of our preparation of the training data and the students' responses. Experimental results suggest that, when we can acquire sufficiently good advice on a problem, machine-learning techniques (both the best performing ANNs and SVMs) may help us identify the hidden learning processes nearly 90% of the time in favourable conditions. When we cannot acquire advises of higher quality, search-based methods, i.e., **`Search4Pattern,`** can become a good alternative. When we do not have adequate information about the students and when the relationship between students' item responses and their competence levels are very uncertain, it becomes very difficult to infer how students learn based on their item response patterns.

We have identified a method, that we discussed along with Formulae (8) and (9), to predict the influences of different *Q*-matrices. This analytical viewpoint helps us choose student subgroups that can help us achieve higher accuracies in learning student models. The selection of *Q*-matrices in experiments is an important issue in realistic studies (DiBello et al., 1995, pp. 370–371). All else being equal, increasing the *total_distance*, which is defined in Formula (9), increased the chances of identifying the correct learning patterns.

Although the use of simulators must result in some degree of distance or abstraction from the real situations and cannot mimic all the characteristics of real students perfectly, we believe that results observed in our simulation-based experiments have shed some light on the nature of this learning problem about learning.

Do we really need to know and include the prerequisite relationship among concepts in student models? Mislevy and Gitomer (1996) state and we agree that "The nature and the grain-size of a student model in an intelligent tutoring system ought therefore to be targeted to the instructional options available." If we cannot take advantage of the detailed models, there is perhaps no incentive for endeavouring to find comprehensive models. Carmona et al. (2005) have shown that student models that consider prerequisite relationships make their adaptive student assessment more efficient. We also hope that more instructional options will become available with the advent of detailed student models, thereby forming a synergistic relation between the two.

The work reported in this paper is related to cognitive diagnostic assessment for education. Cognitively informed models have the potential to help computers assist human's learning activities in a more effective and efficient way (Nichols et al., 1995; Conati, 2002; Alkhalifa, 2006; Leighton & Gierl, 2007). More specifically, in a recently edited book by Leighton and Gierl (2007), Huff and Goodman (2007) elaborate several issues that are related to employing cognitive diagnostic assessment for providing instructionally relevant information that serves the needs for education in addition to scoring. Gierl et al. (2007) discuss four possible structures for describing the relationships between attributes in test development. We hope that the proposed methods and the experimental results presented here may contribute to the efforts in mapping the human learning process and cognitive diagnostic assessment.

ACKNOWLEDGEMENTS

We are grateful to several anonymous researchers for their comments on previous versions of this paper (Liu, 2006a,b,c). The comments led us to conduct more experiments which helped us enrich the breadth and depth of this paper. Special thanks go to the anonymous reviewers of this journal for their directions to help this paper improve on the sections on interdisciplinary discussion. Miss Yu-Ting Wang participated in this project when we had just begun the exploration (Liu & Wang, 2006), and Miss Moira Breen helped the proofreading of the manuscript. Work reported in this paper was supported in part by the grants NSC-93-2213-E-004-004, NSC-94-2213-E-004-008 and NSC-95-2221-E-004-013-MY2 from the National Science Council and in part by the project 96H061 in National Chengchi University under the ATU plan of the Ministry of Education of Taiwan.

REFERENCES

Albert, D., Hockemeyer, C., Mayer, B., & Steiner, C. M. (2007). Cognitive structural modelling of skills for technology-enhanced learning. *Proceedings of the Seventh IEEE International Conference on Advanced Learning Technologies*, 322–324.

Alkhalifa, E. M. (Ed.). (2006). *Cognitively Informed Systems: Utilizing Practical Approaches to Enrich Information Presentation and Transfer*. PA, USA: Idea Group.

Almond, R. G, Mislevy, R. J., Williamson, D. M., & Yan, D. (2008). Bayesian Networks in Education Assessment, a pre-conference training session presented in the 2008 Annual Meeting of the National Council Measurement in Education, http://www.ncme.org.

AUAI. (2006). Association for Uncertainty in Artificial Intelligence: discussion about definition of causality and learning structural models between Joseph Mitola III, Joseph Y. Halpern, and Lotfi A. Zadeh, 14-26 July 2006 on the mailing list of the AUAI, http://www.auai.org.

Beck, J. E., & Sison, J. (2004). Using Knowledge Tracing to Measure Student Reading Proficiencies, Lecture Notes in Computer Science 3220: *Proceedings of the Seventh International Conference on Intelligent Tutoring Systems*, 624–634.

Birenbaum, M., Kelly, A. E., Tatsuoka, K. K., & Gutvirtz, Y. (1994). Attribute-mastery patterns from rule space as the basis for student models in algebra. *International Journal of Human-Computer Studies*, **40**(3), 497–508.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford, UK: Oxford University Press.

Carmona, C., Millán, E., Pérez-de-la-Cruz, J. L., Trella, M., & Conejo, R. (2005). Introducing prerequisite relations in a multi-layered Bayesian student model. *Proceedings of the Tenth International Conference on User Modeling*, 347–356.

Chang, C.-C., & Lin, C.-J. (2001). LIBSVM: A library for support vector machines, http://www.csie.ntu.edu.tw/~cjlin/libsvm/. (last visited on 20 February, 2008)

Chang, K.-M., Beck, J., Mostow, J., & Corbett, A. (2006). A Bayes net toolkit for student modeling in intelligent tutoring systems. *Proceedings of the Eighth International Conference on Intelligent Tutoring Systems*, 104–113.

Chockler, H., & Halpern, J. Y. (2004). Responsibility and blame: A structural-model approach. *Journal of Artificial Intelligence Research*, **22**, 95–115.

Conati, C. (2002). Probabilistic assessment of user's emotions in educational games. *Applied Artificial Intelligence*, **16**(7–8), 555–575.

Conati, C., Gertner, A., & VanLehn, K. (2002). Using Bayesian networks to manage uncertainty in student modeling. *User Modeling and User-Adapted Interaction*, **12**(4), 371–417.

Conejo, R., Guzmán, E., Millán, E., Trella, M., Pérez-de-la-Cruz, J. L., & Rios, A. (2004). SIETTE: A Web-based tool for adaptive testing. *International Journal of Artificial Intelligence in Education*, **14**, 29–61.

Cooper, G. F. (1999). An overview of the representation and discovery of causal relationships using Bayesian networks. In C. Glymour & G. F. Cooper (Eds.), *Computation, Causation, and Discovery* (pp. 3–62). MA, USA: AAAI Press/The MIT Press.

Cortes, C., & Vapnik, V. (1995). Support-vector network. *Machine Learning*, **20**(3), 273–297.

Cover, T. M., & Thomas, J. A. (2006). *Elements of Information Theory* (second ed.). NY, USA: John Wiley & Sons.

Desjardins, B. (2001). Inference of causal structure using the unobservable. *Journal of Experimental and Theoretical Artificial Intelligence*, **13**(3), 291–305.

Desmarais, M. C., Meshkinfam, P., & Gagnon, M. (2006). Learned student models with item to item knowledge structures. *User Modeling and User-Adapted Interaction*, **16**(5), 403–434.

DiBello, L. V., Stout, W. F., & Roussos, L. A. (1995). Unified cognitive/psychometric diagnostic assessment likelihood-based classification techniques. In P. D. Nichols, S. F. Chipman, & R. L. Brennan (Eds.), *Cognitively Diagnostic Assessment* (pp. 361–389). Hove, UK: Lawrence Erlbaum Associates.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern Classification*. NY, USA: John Wiley & Sons.

Falmagne, J.-C., Doignon, J.-P., Cosyn, E., & Thiery, N. (2003). The assessment of knowledge in theory and in practice (Paper No. 26). Institute for Mathematical Behavioral Sciences, University of California, Irvine, California, USA.

Fox, J.-P. (2005). Multilevel IRT using dichotomous and polytomous response data. *British Journal of Mathematical and Statistical Psychology*, **58**(1), 145–172.

Gierl, M. J., Leighton, J. P., & Hunka, S. M. (2007). Using the attribute hierarchy method to make diagnostic inferences about examinees' cognitive skills. In J. P. Leighton and M. J. Gierl (Eds.), *Cognitive Diagnostic Assessment for Education* (pp. 242–274). NY, USA: Cambridge University Press.

Glymour, C. (2003). Learning, prediction and causal Bayes nets. *Trends in Cognitive Sciences*, **7**(1), 43–46.

Glymour, C., & Cooper, G. F. (Eds.). (1999). *Computation, Causation, and Discovery*. CA/MA, USA: AAAI Press/The MIT Press.

Guzmán, E., Conejo, R., & Pérez-de-la-Cruz, J. L. (2007a). Adaptive testing for hierarchical student models. *User Modeling and User-Adapted Interaction*, **17**(1), 119–157.

Guzmán, E., Conejo, R., & Pérez-de-la-Cruz, J. L. (2007b). Improving student performance using self-assessment tests. *IEEE Intelligent Systems*, **22**(4), 46–54.

Halpern, J. Y., & Pearl, J. (2005). Causes and explanations: A structural-model approach. Part I: Causes. *British Journal for the Philosophy of Science*, **56**(4), 843–887.

Heckerman, D. (1999). A tutorial on learning with Bayesian networks. In M. I. Jordan (Ed.), *Learning in Graphical Models* (pp. 301–354). MA, USA: The MIT Press.

Huff, K., & Goodman D. P. (2007). The demand for cognitive diagnostic assessment. In J. P. Leighton and M. J. Gierl (Eds.), *Cognitive Diagnostic Assessment for Education* (pp. 19–60). NY, USA: Cambridge University Press.

Jensen, F. V., & Nielsen, T. D. (2007). *Bayesian Networks and Decision Graphs* (second ed.). NY, USA: Springer-Verlag.

Jolliffe, I. T. (2002). *Principal Component Analysis*. NY, USA: Springer-Verlag.

Jordan, M. I. (Ed.). (1999). *Learning in Graphical Models*. MA, USA: The MIT Press.

Junker, B. W. (2006). Using on-line tutoring records to predict end-of-year exam scores: Experience with the ASSISTments project and MCAS 8th grade mathematics. In Lissitz, R. W. (Ed.), *Assessing and Modeling Cognitive Development in School: Intellectual Growth and Standard Setting*. Maple Grove, MN: JAM Press.

Knuth, D. E. (1973). *The Art of Computer Programming: Fundamental Algorithms*. MA, USA: Addison-Wesley. (p. 73)

Leighton, J. P., & Gierl M. J. (2007). *Cognitive Diagnostic Assessment for Education*. NY, USA: Cambridge University Press.

Liu, C.-L. (2005). Using mutual information for adaptive item comparison and student assessment. *Journal of Educational Technology & Society*, **8**(4), 100–119.

Liu, C.-L. (2006a). Learning how students learn with Bayes nets. Lecture Notes in Computer Science 4053: *Proceedings of the Eighth International Conference on Intelligent Tutoring Systems*, 772–774.

Liu, C.-L. (2006b). Learning students' learning patterns with neural computing. *Proceedings of the 2006 IEEE International Conference on Systems, Man, and Cybernetics*, 2434–2439.

Liu, C.-L. (2006c). Learning students' learning patterns with support vector machines. Lecture Notes in Computer Science 4203: *Proceedings of the Sixteenth International Symposium on Methodologies for Intelligent Systems*, 601–611.

Liu, C.-L. (2006d). Using Bayesian networks for student modeling. In E. M. Alkhalifa (Ed.), *Cognitively Informed Systems: Utilizing Practical Approaches to Enrich Information Presentation and Transfer* (pp. 282–309). PA, USA: Idea Group.

Liu, C.-L., & Wang, Y.-T. (2006). An experience in learning about learning composite concepts. *Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies*, 187–189.

Martin, J., & VanLehn, K. (1995). Student assessment using Bayesian nets. *International Journal of Human-Computer Studies*, **42**(6), 575–591.

Matsuda, N., Cohen, W. W., Sewall, J., Lacerda, G., & Koedinger, K. R. (2007). Predicting students' performance with SimStudent learning cognitive skills from observation. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Education*, 467–476.

Mayo, M., & Mitrovic, A. (2001). Optimising ITS behaviour with Bayesian networks and decision theory. *International Journal of Artificial Intelligence in Education*, **12**, 124–153.

Millán, E., & Pérez-de-la-Cruz, J. L. (2002). A Bayesian diagnostic algorithm for student modeling and its evaluation. *User Modeling and User-Adapted Interaction*, **12**(2-3), 281–330.

Mislevy, R. J., Almond, R. G., Yan, D., & Steinberg, L. S. (1999). Bayes nets in educational assessment: Where do the numbers come from? *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, 437–446.

Mislevy, R. J., & Gitomer, D. H. (1996). The role of probability-based inference in an intelligent tutoring system. *User Modeling and User-Adapted Interaction*, **5**(4), 253–282.

Naveh-Benjamin, M., Lin, Y.-G., & McKeachie, W. J. (1995). Inferring students' cognitive structures and their development using the "fill-in-the-structure" (FITS) technique. In P. D. Nichols, S. F. Chip-man, & R. L. Brennan (Eds.), *Cognitively Diagnostic Assessment* (pp. 279–304). Hove, UK: Lawrence Erlbaum Associates.

Neapolitan, R. E. (2004). *Learning Bayesian Networks*. NJ, USA: Prentice Hall.

Nichols, P. D., Chipman, S. F., & Brennan, R. L. (Eds.). (1995). *Cognitively Diagnostic Assessment*. Hove, UK: Lawrence Erlbaum Associates.

Novak, J. D. (1990). Concept maps and Vee diagrams: Two metacognitive tools to facilitate meaningful learning. *Instructional Science*, **19**(1), 29–52.

Pardos, Z., Feng, M., Heffernan, N. T., & Heffernan-Lindquist, C. (2007). Analyzing fine-grained skill models using Bayesian and mixed effect methods. *Proceedings of the Thirteenth Conference on Artificial Intelligence in Education*.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. CA, USA: Morgan Kaufmann.

Reye, J. (2004). Student modelling based on belief networks. *International Journal of Artificial Intelligence in Education*, **14**, 63–96.

Rost, J., & Langeheine, R. (Eds.). (1997). *Applications of Latent Trait and Latent Class Models in the Social Sciences*. Münster, Germany: Waxmann.

Russell, S. J., & Norvig, P. (2002). *Artificial Intelligence: A Modern Approach*. NJ, USA: Prentice Hall.

Shachter, R. D. (1988). Probabilistic inference and influence diagrams. *Operation Research*, **36**(4), 589–604.

Spirtes, P., Glymour, C., & Scheines, R. (2000). *Causation, Prediction, and Search* (second ed.). MA, USA: The MIT Press.

Tatsuoka, K. K. (1983). Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement*, **20**, 345–354.

van der Linden, W. J., & Glas, C. A. (Eds.). (2000). *Computerized Adaptive Testing: Theory and Practice*. Dordrecht, The Netherlands: Kluwer.

van der Linden, W. J., & Hambleton, R. K. (Eds.). (1997). *Handbook of Modern Item Response Theory*. NY, USA: Springer.

VanLehn, K., Niu, Z., Siler, S., & Gertner, A. (1998). Student modeling from conventional test data: A Bayesian approach without priors. *Proceedings of the Fourth International Conference on Intelligent Tutoring Systems*, 434–443.

VanLehn, K., Ohlsson, S., & Nason, R. (1994). Applications of simulated students: An exploration. *International Journal of Artificial Intelligence in Education*, **5**(2), 135–175.

Vomlel, J. (2004). Bayesian networks in educational testing. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, **12**(Supplement 1), 83–100.

Vos, H. J. (2000). A Bayesian procedure in the context of sequential mastery testing. *Psicológica*, **21**, 191–211.

Wasserman, P. D. (1993). *Advanced Methods in Neural Computing*. NY, USA: Van Nostrand Reinhold. (pp. 35-55)

Weissman, A. (2007). Mutual information item selection in adaptive classification testing. *Educational and Psychological Measurement*, **67**(1), 41–58.

Weng, J., & Hwang, W.-S. (2006). From neural networks to the brain: Autonomous mental development. *IEEE Computational Intelligence Magazine*, **1**(3), 15–31.

Witten, I. H., & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. CA, USA: Morgan Kaufmann.

Using Structural Information for Identifying Similar Chinese Characters

Chao-Lin Liu **Jen-Hsiang Lin**
Department of Computer Science, National Chengchi University, Taipei 11605, Taiwan
{chaolin, g9429}@cs.nccu.edu.tw

Abstract

Chinese characters that are similar in their pronunciations or in their internal structures are useful for computer-assisted language learning and for psycholinguistic studies. Although it is possible for us to employ image-based methods to identify visually similar characters, the resulting computational costs can be very high. We propose methods for identifying visually similar Chinese characters by adopting and extending the basic concepts of a proven Chinese input method--Cangjie. We present the methods, illustrate how they work, and discuss their weakness in this paper.

1 Introduction

A Chinese sentence consists of a sequence of characters that are not separated by spaces. The function of a Chinese character is not exactly the same as the function of an English word. Normally, two or more Chinese characters form a Chinese word to carry a meaning, although there are Chinese words that contain only one Chinese character. For instance, a translation for “conference” is “研討會” and a translation for “go” is “去”. Here “研討會” is a word formed by three characters, and “去” is a word with only one character.

Just like that there are English words that are spelled similarly, there are Chinese characters that are pronounced or written alike. For instance, in English, the sentence “John plays an important roll in this event.” contains an incorrect word. We should replace “roll” with “role”. In Chinese, the sentence “今天上午我們來**試場**買菜” contains an incorrect word. We should replace “試場” (a place for taking examinations) with “市場” (a market). These two words have the same pronunciation, shi(4) chang(3)[†], and both represent locations. The sentence “經理要我**構買**一部計算機” also contains an error, and we need to replace “構買” with “購買”. “構買” is considered an incorrect word, but can be confused with “購買” because the first characters in these words look similar.

Characters that are similar in their appearances or in their pronunciations are useful for computer-assisted language learning (cf. Burstein & Leacock, 2005). When preparing test items for testing students’ knowledge about correct words in a computer-assisted environment, a teacher provides a sentence which contains the character that will be replaced by an incorrect character. The teacher needs to specify the answer character, and the software will provide two types of incorrect characters which the teachers will use as distracters in the test items. The first type includes characters that look similar to the answer character, and the second includes characters that have the same or similar pronunciations with the answer character.

Similar characters are also useful for studies in Psycholinguistics. Yeh and Li (2002) studied how similar characters influenced the judgments made by skilled readers of Chinese. Taft, Zhu, and Peng (1999) investigated the effects of positions of radicals on subjects’ lexical decisions and naming responses. Computer programs that can automatically provide similar characters are thus potentially helpful for designing related experiments.

2 Identifying Similar Characters with Information about the Internal Structures

We present some similar Chinese characters in the first subsection, illustrate how we encode Chinese characters in the second subsection, elaborate how we improve the current encoding method to facilitate the identification of similar characters in the third subsection, and discuss the weakness of our current approach in the last subsection.

2.1 Examples of Similar Chinese Characters

We show three categories of confusing Chinese characters in Figures 1, 2, and 3. Groups of similar characters are separated by spaces in these figures. In Figure 1, characters in each group differ at the stroke level. Similar characters in every group in the first row in Figure 2 share a common part, but the shared part is not the radical of these characters. Similar characters in every group in the second row in Figure 2 share a common part, which is the radical of these characters. Similar characters in every group in Figure 2 have different pronunciations. We show six groups of homophones that also share a component in Figure 3. Characters that are similar in both pronunciations and internal structures are most confusing to new learners.

[†] We use Arabic digits to denote the four tones in Mandarin.

Proceedings of ACL-08: HLT, Short Papers (Companion Volume), pages 93–96,
Columbus, Ohio, USA, June 2008. ©2008 Association for Computational Linguistics

士土工干千 戌戍成 田由甲申
毋毌 勿匆 人入 未末 采釆 凹凸

Figure 1. Some similar Chinese characters

頸勁 搆溝 陪倍 硯現 裸棵 搞篙
列刑 盆盎盂盅 因困囚 間閒閃開

Figure 2. Some similar Chinese characters that have different pronunciations

形刑型 踵種腫 購構搆 紀記計
園圓員 脛逕徑痙勁

Figure 3. Homophones with a shared component

It is not difficult to list all of those characters that have the same or similar pronunciations, e.g., “試場” and “市場”, if we have a machine readable lexicon that provides information about pronunciations of characters and when we ignore special patterns for tone sandhi in Chinese (Chen, 2000).

In contrast, it is relatively difficult to find characters that are written in similar ways, e.g., “構” with “購”, in an efficient way. It is intriguing to resort to image processing methods to find such structurally similar words, but the computational costs can be very high, considering that there can be tens of thousands of Chinese characters. There are more than 22000 different characters in large corpus of Chinese documents (Juang et al., 2005), so directly computing the similarity between images of these characters demands a lot of computation. There can be more than 4.9 billion combinations of character pairs. The Ministry of Education in Taiwan suggests that about 5000 characters are needed for ordinary usage. In this case, there are about 25 million pairs.

The quantity of combinations is just one of the bottlenecks. We may have to shift the positions of the characters “appropriately” to find the common part of a character pair. The appropriateness for shifting characters is not easy to define, making the image-based method less directly useful; for instance, the common part of the characters in the right group in the second row in Figure 3 appears in different places in the characters.

Lexicographers employ radicals of Chinese characters to organize Chinese characters into sections in dictionaries. Hence, the information should be useful. The groups in the second row in Figure 2 show some examples. The shared components in these groups are radicals of the characters, so we can find the characters of the same group in the same section in a Chinese dictionary. However, information about radicals as they are defined by the lexicographers is not sufficient. The groups of characters shown in the first row in Figure 2 have shared components. Nevertheless, the shared components are not considered as radicals, so the characters, e.g., “頸”and “勁”, are listed in different sections in the dictionary.

2.2 Encoding the Chinese Characters

The Cangjie[‡] method is one of the most popular methods for people to enter Chinese into computers. The designer of the Cangjie method, Mr. Bong-Foo Chu, selected a set of 24 basic elements in Chinese characters, and proposed a set of rules to decompose Chinese characters into elements that belong to this set of building blocks (Chu, 2008). Hence, it is possible to define the similarity between two Chinese characters based on the similarity between their Cangjie codes.

Table 1, not counting the first row, has three sections, each showing the Cangjie codes for some characters in Figures 1, 2, and 3. Every Chinese character is decomposed into an ordered sequence of *elements*. (We will find that a subsequence of these elements comes from a major *component* of a character, shortly.) Evidently, computing the number of shared elements provides a viable way to determine “visually similar” characters for characters that appeared in Figure 2 and Figure 3. For instance, we can tell that “搞” and “篙” are similar because their Cangjie codes share “卜口月”, which in fact represent “高”.

	Cangjie Codes		Cangjie Codes
士	十一	土	土
工	一中一	干	一十
勿	心竹竹	匆	竹田心
未	十木	末	木十
頸	一一一月金	勁	一一大尸
硯	一口月山山	現	一土月山山
搞	手卜口月	篙	竹卜口月
列	一弓中弓	刑	一廿中弓
因	田大	困	田木
間	日弓日	閒	日弓月
踵	口一竹十土	種	竹木竹十土
腫	月竹十土	紀	女火尸山
購	月金廿廿月	構	木廿廿月
記	卜口尸山	計	卜口十
圓	田口月金	員	口月山金
脛	月一女一	逕	卜一女一
徑	竹人一女一	痙	大一女一

Table 1. Cangjie codes for some characters

[‡] http://en.wikipedia.org/wiki/Cangjie_method

Unfortunately, the Cangjie codes do not appear to be as helpful for identifying the similarities between characters that differ subtly at the stroke level, e.g., “士土工干” and other characters listed in Figure 1. There are special rules for decomposing these relatively basic characters in the Cangjie method, and these special encodings make the resulting codes less useful for our tasks.

The Cangjie codes for characters that contain multiple components were intentionally simplified to allow users to input Chinese characters more efficiently. The longest Cangjie code for any Chinese character contains no more than five elements. In the Cangjie codes for “脛” and “徑”, we see “一女一” for the component “巠”, but this component is represented only by “一一” in the Cangjie codes for “頸” and “勁”. The simplification makes it relatively harder to identify visually similar characters by comparing the actual Cangjie codes.

2.3 Engineering the Original Cangjie Codes

Although useful for the sake of designing input method, the simplification of Cangjie codes causes difficulties when we use the codes to find similar characters. Hence, we choose to use the complete codes for the components in our database. For instance, in our database, the codes for “巠”, “脛”, “徑”, “頸”, and “勁” are, respectively, “一女女一”, “月一女女一”, “竹人一女女一”, “一女女一一月山金”, and “一女女一大尸”.

The knowledge about the graphical structures of the Chinese characters (cf. Juang et al., 2005; Lee, 2008) can be instrumental as well. Consider the examples in Figure 2. Some characters can be decomposed vertically; e.g., “盅” can be split into two smaller components, i.e., “中” and “皿”. Some characters can be decomposed horizontally; e.g., “現” is consisted of “王” and “見”. Some have enclosing components; e.g., “人” is enclosed in “口” in “囚”. Hence, we can consider the locations of the components as well as the number of shared components in determining the similarity between characters.

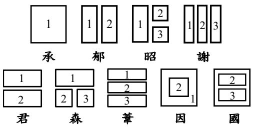


Figure 4. Arrangements of components in Chinese

Figure 4 illustrates possible layouts of the components in Chinese characters that were adopted by the Cangjie method (cf. Lee, 2008). A sample character is placed below each of these layouts. A box in a layout indicates a component in a character, and there can be at most three components in a character. We use digits to indicate the ordering the components. Notice that, in the second row, there are two boxes in the second to the rightmost layout. A larger box contains a smaller one. There are three boxes in the rightmost layout, and two smaller boxes are inside the outer box. Due to space limits, we do not show “1” for this outer box.

After recovering the simplified Cangjie code for a character, we can associate the character with a tag that indicates the overall layout of its components, and separate the code sequence of the character according to the layout of its components. Hence, the information about a character includes the tag for its layout and between one to three sequences of code elements. Table 2 shows the annotated and expanded codes of the sample characters in Figure 4 and the codes for some characters that we will discuss. The layouts are numbered from left to right and from top to bottom in Figure 4. Elements that do not belong to the original Canjie codes of the characters are shown in smaller font.

	Layout	Part 1	Part 2	Part 3
承	1	弓弓手人		
郁	2	大月	弓中	
昭	3	日	尸竹	口
謝	4	卜一一口	竹難竹	木戈
君	5	尸大	口	
森	6	木	木	木
葦	7	廿	木一	手
因	8	田	大	
國	9	田	戈	口一
頸	2	一女女一	一月山金	
徑	2	竹人	一女女一	
員	5	口	月山金	
圓	9	田	口	月山金
相	2	木	月山	
想	5	木月山	心	
箱	6	竹	木	月山

Table 2. Annotated and expanded code

Recovering the elements that were dropped out by the Cangjie method and organizing the subsequences of elements into parts facilitate the identification of similar characters. It is now easier to find that the character (頸) that is represented by “一女女一” and “一月山金” looks similar to the character (徑) that is represented by “竹人” and “一女女一” in our database than using their original Cangjie codes in Table 1. Checking the codes for “員” and “圓” in Table 1 and Table 2 will offer an additional support for our design decisions.

In the worst case, we have to compare nine pairs of code sequences for two characters that both have three components. Since we do not simplify codes for components and all components have no more than five elements, conducting the comparisons operations are simple.

2.4 Drawbacks of Using the Cangjie Codes

Using the Cangjie codes as the basis for comparing the similarity between characters introduces some potential problems.

It appears that the Cangjie codes for some characters, particular those simple ones, were not assigned without ambiguous principles. Relying on Cangjie codes to compute the similarity between such characters can be difficult. For instance, “分” uses the fifth layout, but “兌” uses the first layout in Figure 4. The first section in Table 1 shows the Cangjie codes for some character pairs that are difficult to compare.

Due to the design of the Cangjie codes, there can be at most one component at the left hand side and at most one component at the top in the layouts. The last three entries in Table 2 provide an example for these constraints. As a standalone character, “相” uses the second layout. Like the standalone “相”, the “相” in “箱” was divided into two parts. However, in “想”, “相” is treated as an individual component because it is on top of “想”. Similar problems may occur elsewhere, e.g., “森焚” and “恩因”. There are also some exceptional cases; e.g., “品” uses the sixth layout, but “闆” uses the fifth layout.

3 Concluding Remarks

We adopt the Cangjie alphabet to encode Chinese characters, but choose not to simplify the code sequences, and annotate the characters with the layout information of their components. The resulting method is not perfect, but allows us to find visually similar characters more efficient than employing the image-based methods.

Trying to find conceptually similar but contextually inappropriate characters should be a natural step after being able to find characters that have similar pronunciations and that are visually similar.

Acknowledgments

Work reported in this paper was supported in part by the plan NSC-95-2221-E-004-013-MY2 from the National Science Council and in part by the plan ATU-NCCU-96H061 from the Ministry of Education of Taiwan.

References

Jill Burstein and Claudia Leacock. editors. 2005. *Proceedings of the Second Workshop on Building Educational Applications Using NLP*, ACL.

Matthew Y. Chen. 2000. *Tone Sandhi: Patterns across Chinese Dialects*. (Cambridge. Studies in Linguistics 92.) Cambridge: Cambridge University Press.

Bong-Foo Chu. 2008. *Handbook of the Fifth Generation of the Cangjie Input Method*, web version, available at http://www.cbflabs.com/book/ocj5/ocj5/index.html. Last visited on 14 Mar. 2008.

Hsiang Lee. 2008. *Cangjie Input Methods in 30 Days*, http://input.foruto.com/cjdict/Search_1.php, Foruto Company, Hong Kong. Last visited on 14 Mar. 2008.

Derming Juang, Jenq-Haur Wang, Chen-Yu Lai, Ching-Chun Hsieh, Lee-Feng Chien, and Jan-Ming Ho. 2005. Resolving the unencoded character problem for Chinese digital libraries. *Proceedings of the Fifth ACM/IEEE Joint Conference on Digital Libraries*, 311–319.

Marcus Taft, Xiaoping Zhu, and Danling Peng. 1999. Positional specificity of radicals in Chinese character recognition, *Journal of Memory and Language*, **40**, 498–519.

Su-Ling Yeh and Jing-Ling Li. 2002. Role of structure and component in judgments of visual similarity of Chinese characters, *Journal of Experimental Psychology: Human Perception and Performance*, **28**(4), 933–947.

以範例為基礎之英漢 TIMSS 試題輔助翻譯

張智傑　　　劉昭麟
國立政治大學 資訊科學系
{ g9512 ,chaolin }@cs.nccu.edu.tw

摘要

本論文應用以範例為基礎的機器翻譯技術，應用英漢雙語對應的結構輔助英漢單句語料的翻譯。翻譯範例是運用一種特殊的結構，此結構包含來源句的剖析樹、目標句的字串、以及目標句和來源句詞彙對應關係。將翻譯範例建立資料庫，以提供來源句作詞序交換的依據，最後透過字典翻譯，以及利用統計式中英詞彙對列和語言模型來選詞，產生建議的翻譯。我們是以 2003 年國際數學與科學教育成就趨勢調查測驗試題為主要翻譯的對象，以期提升翻譯的一致性和效率。以 NIST 和 BLEU 的評比方式，來評估和比較線上翻譯系統和本系統所達成的翻譯品質。

關鍵詞：自然語言處理，試題翻譯，機器翻譯，TIMSS

1. 緒論

國際教育學習成就調查委員會(The International Association for the Evaluation of Education Achievement, 以下簡稱 IEA)[20]主要目的在於了解各國學生數學及科學(含物理、化學、生物、及地球科學)方面學習成就、教育環境等，影響學生的因素，找出關聯性，並在國際間相互作比較。自 1970 年起開始第一次國際數學與科學教育成就調查後，世界各國逐漸對國際數學與科學教育成就研究感到興趣，IEA 便在 1995 年開始每四年辦理國際數學與科學教育成就研究一次，稱為國際數學與科學教育成就趨勢調查(Trends in International Mathematics and Science Study，以下簡稱 TIMSS)，至今已辦理過 1995、1999、2003 和 2007 共四屆，共有 38 個國家參加。

我國於 1999 年開始加入 TIMSS 後，由國科會委託國立台灣師範大學科學教育中心(以下簡稱師大科教中心)負責試題翻譯及測驗工作。1999 年的調查對象只有國中二年級學生， 2003 年的調查對象包括四年級及八年級學生。翻譯試題主要的流程包含：從 IEA 取得試題內容，由師大科教中心決議進行翻譯工作分配、中文試題交換審稿校正及翻譯問題討論，最後將中文翻譯試題定稿。至目前為止，師大科教中心已將 1999 和 2003 年試題內容和評量結果，公布於台灣 TIMSS 官方網站[21]，以提供研究之參考。在 TIMSS 的試題內容上，主要的題型種類有選擇題和問答題，試題句型大多為直述句和問句結構所組成，選擇題則多了誘答選項。

以往使用人工翻譯雖然可以達到很高的翻譯品質，但是需要耗費相當多的人力資源和時間，而且在翻譯過程中不同的翻譯者會有不同的翻譯標準(例如：相同的句子，翻譯後的結果不同)；相同的翻譯者也可能在文章前後翻譯方式不一致而產生語意上的混淆。因此間接影響試題難易程度。若直接將英文詞彙透過英漢字典翻譯成相對的中文詞彙，翻譯的結果可能會不符合一般人的用詞順序。另外中文的自由度較高，很容易造成翻譯上用詞順序的不同。例如：“下圖顯示某一個國家所種穀物的分布圖”，也可翻譯為“某一個國家所種穀物的分布圖，如下圖顯示”。可能會影響到受測者的思緒，使作答時粗心的情形會增加。因此，若能利用機器翻譯(machine translation)的技術來輔助翻譯以及調整詞序，以期提高翻譯的品質和效率。

在人工智慧領域，機器翻譯是一個很困難的問題。機器翻譯是指將一種自然語言經過電腦運算翻譯成另一種語言，困難程度也跟來源句和目標句有關，像是英文和葡萄牙文語言的特性較相近，較容易翻譯。而中文跟英文詞序差異很大，且中文比較沒有特定的語法，寫法較自由，對翻譯來說較為困難。機器翻譯發展至今已經超過 50 年。Dorr 等學者[9]將現在機器翻譯依據系統處理的方式來分類，分成以語言學為基翻譯(linguistic-based paradigms)，例如基於知識(knowledge-based)和基於規則(rule-based)等；以及非語言學為基翻譯(non-linguistic-based paradigms) ，例如基於統計(statistical-based)和基於範例(example-based)等。

以知識為基礎的機器翻譯(knowledge-based machine translation)系統是運用字典、文法規則或是語言學家的知識來幫助翻譯。Knight 等學者[11]結合 Longman 字典、WordNet 和 Collins 雙語字典建立一個知識庫，運用在西班牙文翻譯成英文。這種利用字典來幫助翻譯的系統，會有一字多義的情形發生，一個詞彙在字典中通常有一個以上的翻譯。以英翻中為例“current”這個字在字典裡就有十多種不同的翻譯，即使專家也無法找出一個統一的規則，在何種情況下要用何種翻譯，所以在翻譯的品質和正確性上很難滿足使用者。因此，翻譯系統通常都會限定領域來減少一字多義，例如 current 在電子電機類的文章中出現，最常被翻譯為電流，在文學類的文章中，最常被翻譯為現代。

統計式機器翻譯(statistical machine translation，以下簡稱 SMT)是將語料在翻譯之前就已經過計算轉換成統計數據，不需要在翻譯過程中作龐大的數學運算，能有較高的效能。Brown 等學者[6]於 1990 年以英文及法文的雙語語料為來源，提出統計式雙語翻譯架構。假設目標語言為 T 及來源語言為 S，P(T)為目標語言 T 在語料庫中出現的機率，稱為語言模型(language model)，P(S|T)為目標語言 T 翻譯成來源語言 S 的機率，稱為翻譯模型(translation model)。SMT 系統需要大量的語料庫輔助，大多都需要具備雙語對應的語料庫(parallel corpora 或稱 bilingual corpora)，再透過機率公式計算出機率模型。其中 SMT 困難的地方在於需要收集大量可用的雙語語料，當語料越多建立模型所花費的時間越多。Oct 等學者[16]提出單字式(word-based)翻譯模型運用在詞彙對準(word alignment)，並且發展出 GIZA++這套系統。Koehn 等學者[12]進一步將單字式轉變成片語式(phrase-based)翻譯模型，運用片語式翻譯模型翻譯的結果會比單字式翻譯的結果要正確。

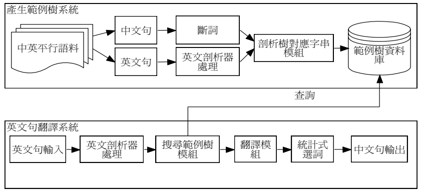
以範例為基礎的機器翻譯(example-based machine translation，以下簡稱為 EBMT)的相關研究已有相當多年歷史，在 1990 年日本學者 Sato 和 Nagao[19]所提出的 EBMT 是將翻譯過程分為分解(decomposition)、轉換(transfer)和合成(composition)三步驟。分解階段是將來源句到範例庫中搜尋，並將所搜尋到 word-dependency tree 當作來源句的 word-dependency tree，並且形成來源句的表示式；轉換階段將來源句的表示式轉換成目標句的表示式；合成階段將目標句的表示式展開為目標句的 word-dependency tree，並且輸出翻譯結果。Al-Adhaileh 等學者[5]將 structured string tree correspondence(SSTC) [7] 運用在英文翻譯成馬來西亞文上，SSTC 是一種能將英文對應馬來西亞文的結構，但此結構並沒有解決詞序交換的問題。目前較完整的 EBMT 系統有 Liu 等學者所提出 tree-string correspondence (TSC)結構和統計式模型所組成的 EBMT 系統[13]，在比對 TSC 結構的機制是計算來源句剖析樹和 TSC 比對的分數，產生翻譯的是由來源詞彙翻譯成目標詞彙的機率和目標句的語言模型所組成。

黃輝等學者所提出的 translation corresponding tree (TCT) [24]，TCT 是針對英文翻譯成葡萄牙文的系統，在 TCT 結構上可以記錄來源句詞彙和目標句詞彙對應的關係、來源句詞彙和目標句詞彙對應的翻譯結果和詞序，但是 TCT 是二元的剖析樹，也就是每個節點都只有兩顆子樹，在 TCT 上詞序只用布林值(boolean value)來記錄，所以 TCT 只能運用在二元剖析樹上。但是有些剖析器所產生剖析樹是多元樹，因此我們提出雙語樹對應字串的結構(bilingual structured string tree correspondence，簡稱為 BSSTC)可以運用在多元剖析樹上，並且 BSSTC 可在翻譯過程中當作詞序交換的參考，根據我們實驗結果，我們能有效的調動詞序，以提升翻譯的品質。完成詞序交換後，再透過字典翻譯成中文，最後運用統計式選詞模型，產生了初步翻譯結果，但本系統尚屬於半自動翻譯系統，故需要人工加以修飾編輯。

除了本節簡單介紹本研究以外，我們將在第二節描述整個系統的架構，第三節說明本篇論文所運用的技術，第四節則呈現出我們的實驗結果，第五節則是結論。

2. 系統架構

由於我們的目的在於利用中英互為翻譯的句子找出詞序關係，並且將英文句和中文句詞序的資訊儲存在電腦中，儲存的格式是將中英文句的詞序關係記錄在英文剖析樹的結構中，此結構將成為之後英文句的結構調整為適合中文的結構的參考。最後再將英文詞彙翻譯成中文詞彙，並利用統計式選詞選出最有可能翻譯成的中文詞彙，讓翻譯的結果更符合一般人的用詞和順序。



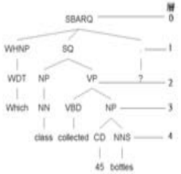
圖一、系統架構圖

本系統的架構如圖一所示。我們針對範例樹產生系統和英文句翻譯系統這兩部份分別簡介如下。

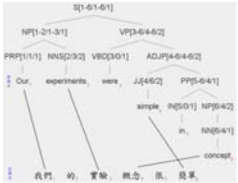
- **範例樹產生系統：** 這個系統利用中英平行語料，這裡的中英平行語料必需要一句英文句對應一句中文句，且每一組中英文句都要是互為翻譯的句子。中文句經過斷詞處理後，被斷成數個中文詞彙，以空白隔開；英文句經過英文剖析器建成英文剖析樹。將斷詞後的結果和英文剖析樹經過剖析樹對應字串模組處理，建成英文剖析樹對應字串的結構樹，此結構樹稱為範例樹。再將每個範例樹取出子樹，並且判斷是否有詞序交換，將需要詞序交換的範例樹全部存入範例樹資料庫中方便搜尋。
- **英文句翻譯系統：**當輸入英文句後，先將句子透過英文剖析器，建成英文剖析樹。有了英文剖析樹就可以透過搜尋範例樹模組，標記英文剖析樹上需要調動詞序的結構，並依照所標記的詞序作調整。詞序調整完成後再將英文結構樹中的英文單字或片語透過翻譯模組做翻譯。其中翻譯模組包含了大小寫轉換、斷詞處理、stop word filtering及stemming，之後將處理過的詞彙透過字典檔做翻譯[3]。每個英文單字或片語都可能有一個以上的中文翻譯，因此需要選詞的機制來產生初步翻譯結果，此翻譯結果尚需要人工作後續的編修。

3. 系統相關技術

根據上一節系統架構的描述分為範例樹產生系統和英文句翻譯系統兩大系統。範例產生樹系統的執行流程為先將中文句斷詞和剖析英文句，再將斷詞和剖析後的結果輸入至剖析樹對應字串模組，並將處理後的範例樹存入資料庫中。英文句翻譯系統的執行流程區分為三大部分，第一部分是由搜尋範例樹模組，將英文剖析樹跟範例樹資料庫作比對，並且將未比對到的子樹做修剪；第二部分將修剪後的剖析樹輸入到翻譯模組翻成中文；第三部分以中英詞彙對列工具及 bi-gram 語言模型，計算出中英詞彙間最有可能之翻譯組合。



圖二、英文剖析樹



圖三、BSSTC 結構的表示法

3.1 雙語樹對應字串的結構(BSSTC)

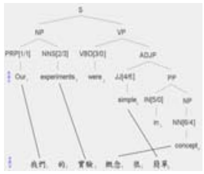
在建立 BSSTC 結構之前，我們必須將中英平行語料中的中英文句先作前處理，我們將英文句透過 StanfordLexParser-1.6[17]建成剖析樹，剖析樹的每個葉子節點為一個英文單字，並以英文單字為單位由 1 開始標號。這裡我們將樹根定義為第 0 層，樹根的子樹是第 1 層，越往下層數越大，故葉子節點必定是英文單字，且不屬於任何一層，如圖二所示。而中文句是使用中研院 CKIP 斷詞系統[1]作斷詞，並以斷詞後的單位由 1 開始標號。這裡的中文句代表來源句；英文句代表目標句。本結構是假設在中英文對應都是在詞彙的對應或連續字串的對應。

假設剖析樹的節點集合 $N=\{N_1, N_2, \ldots, N_m\}$，$m$ 為剖析樹上節點個數，對任一節點 $n \in N$，n 有三個參數分別是 n[STREE//]、n[/STC/]和 n[//ORDER]；我們以 n[STREE/STC/ORDER]來表示。為了方便說明，若節點 n 只有 n[STREE//]和 n[/STC/]，則以 n[STREE/STC/]表示。再假設 $n_{C(n)}$為節點 n 有 1 到 C(n)個子節點。n[STREE//]為節點 n 所涵蓋來源句的範圍，層數最大節點的 n[STREE//]必定對應到一個來源句單字，此參數的功用為當作每個節點的鍵值(primary key)，故在同一棵剖析樹中 n[STREE//]不會重複。n[/STC/]表示以 n 為樹根的子樹，所涵蓋來源句字串的範圍對應到目標句字串的範圍；n[/STC/]也可以是一個數字，表示此子樹包含的目標句字串為目標句字串中的一個字；n[/STC/]也可能是 0，代表來源句無法對應到目標句。n[//ORDER]是由 n[/STC/]計算出來，n[//ORDER]是用來表示來源句跟目標句詞序對應的關係，若來源句跟目標句有詞序不同的情形，就可由 n 與所有兄弟節點的 n[//ORDER]來判斷。ORDER 的範圍由 1 到 C(n)，當 ORDER 越小，代表 n 所對應目標句範圍，比其他兄弟節點的目標句範圍更靠近句子的前段。

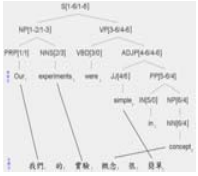
圖三是一個 BSSTC 結構的例子，來源句為英文：“Our experiments were simple in concept”；目標句為中文：“我們的實驗概念很簡單”。首先英文句必須先建成剖析樹，每個葉子節點為一個英文單字，並以英文單字為單位做標號，例如：“Our(1)”, “experiments(2)”, “were(3)”, “simple(4)”, “in(5)”,“concept(6)”。另外中文句經過斷詞的處理後，以斷詞後的單位做標號，例如：“我們(1)”, “的(2)”, “實驗(3)”, “概念(4)”, “很(5)”, “簡單(6)”。中英對應句都標號後，以標號為單位開始做詞彙對準(word alignment)，並標記在剖析樹的節點上。剖析樹是用文法結構來分層，不同層節點能對應到不同的範圍的目標句字串。n[STREE/STC/]若為 VP[3-6/4-6/]，則 STREE 代表節點 VP 對應來源句第三到第六個字 “were simple in concept“；STC 代表“were simple in concept”對應目標句的第四到第六個字“概念很簡單”。$n_{C(n)}$[STREE/STC/ORDER]的兄弟節點(sibling node)若為 JJ[4/6/2]和 PP[5-6/4/1]，我們可以觀察到 JJ 的 ORDER 大於 PP 的 ORDER，故 PP[5-6/4/1]的中文對應「概念」在 JJ[4/6/2] 的中文對應「簡單」之前。

3.2 建立 BSSTC 結構和產生範例樹

建立 BSSTC 結構必需要有英文跟中文互為翻譯的句子，建構的順序是從最底層也就是層數最大的開始標記，再一層一層往上建置到第 0 層為止，標記參數順序是先將所有節點的 n[STREE//]和 n[/STC/]標記完後，再標記 n[//ORDER]。首先，標記最底層 n[STREE//]的方法，是將最底層的節點 n 所對應葉子節點的編號標記在 n[STREE//]。如圖三節點 NNS 所對應來源句的“experiments”的編號為 2，故 NNS[STREE//]中的 STREE 標記為 2。接著標記最底層 n[/STC/]的方法是尋找中英對應句中互為翻譯的中文詞彙和英文詞彙，也就是詞彙對準。詞彙對準若採用人工方式，則相當耗時費力，其本身也是一項困難的研究。因此，我們在此用一個簡單的方法，首先先將中文句經過斷詞處理，這裡我們使用中研院 CKIP 斷詞系統[1]；將英文句每個英文字查尋字典檔，查尋後可能會有超過一個的中文翻譯，將這些中文翻譯跟斷詞後的中文詞彙一個一個作比對，如有比對到則認定互為翻譯，並且標記 n[/STC/]在剖析樹上。如圖三來源句的“experiments”在字典中的翻譯有“實驗”、“經驗”和“試驗”，將這三個中文翻譯到目標句去比對，此例子將會比對到目標句第三個詞彙“實驗”，接著將目標句“實驗”的編號標記在 NNS[2/STC/]中的 STC 上。最後將比對到的個數除以英文句單字的個數，稱為對應率。最佳情況下是每個英文單字都有相對應的中文翻譯，對應率為 1；最差的情況下每個英文單字都沒有相對應的中文翻譯，對應率為 0，所以對應率會落在 0 到 1 之間，值越大代表對應率越高。我們需要夠大的對應率，才能認定為範例樹。因此，需要定一個門檻值來篩選，根據實驗結果當門檻值越高留下來的範例樹越少，而門檻值越低會使翻譯的品質下降。



圖四、僅標記最底層



圖五、僅標記 STREE 及 STC

目前範例樹只將最底層的 n[STREE/STC/]標記完成，如圖四，現在要逐層將未標記 n[STREE/STC/]的節點標記上去。n[STREE//]標記的方式，是將 n_1 到 $n_{C(n)}$的 STREE 都加入 ES 中。ES 為用來儲存 $n_{C(n)}$ [STREE//]中 STREE 的集合。當層數越小，則 n[STREE//]將會涵蓋 1 個以上的來源句的詞彙。若 $n_{C(n)}$ [STREE//]為一個範圍，則將此範圍最大和最小的值加入 ES，最後 ES 內可能為一個數字或兩個以上的數字這兩種情況，如只有一個數字則 n[STREE//]只標記該數字，如有兩個以上的數字則 ES 中最小和最大的數值標記在 n[STREE//]上，格式為 n[最小-最大//] ；n[/STC/]標記的方式，是將 n_1 到 $n_{C(n)}$的 STC 都加入 CS 中。CS 為用來儲存 $n_{C(n)}$ [/STC/]中 STC 的集合。當層數越小，則 n[/STC/]將會涵蓋 1 個以上目標句的詞彙。如 $n_{C(n)}$ [/STC/]為一個範圍，則將此範圍最大和最小的值加入 CS。若 $n_{C(n)}$ [/STC/]出現 0 則不加入 CS。最後 CS 可能為空、一個數字或兩個以上的數字這三種情況，如為空則將 n[/STC/]標記為 0，若只有一個數字則 n[/STC/]只

標記該數字，假如有兩個以上的數字將 CS 中最小和最大的 STC 標記在 *n*[/STC/]上，格式爲 *n*[/最小-最大/]。

假如我們現在要標記圖五第一層的節點 VP，則必需將節點 VP 的子節點 VBD 和 ADJP 的 VBD[3//]及 ADJP[4-6//]中的 STREE 加入 ES 中，因此 ES 包含了 3、4 和 6 三個數字，所以 VP[STREE//]中的 STREE 標記爲 3-6。接著標記 STC，將節點 VP 的子節點 VBD 和 ADJP 的 VBD[3/0/]及 ADJP[4-6/4-6/]中的 STC 加入 CS 中，因爲 0 不會被加入 CS 中，因此 CS 只有 4 和 6 兩個數字，所以 VP[3-6/STC/]中的 STC 標記爲 4-6。

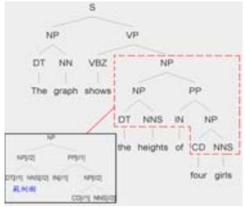
最後，整棵剖析樹的 STREE 跟 STC 都已經標記完成，如圖五，只剩下 ORDER 還沒標記上。ORDER 處理方式分爲兩部分，第一部份：STC 爲 0 之兄弟節點按照由左至右的順序編號；第二部分：比較 *n* 與 STC 非 0 之兄弟節點的大小，並接在第一部份的編號後，由小到大繼續標記編號。例如圖五若要標記 JJ[4/6/ORDER]和 PP[5-6/4/ORDER]的 ORDER，則將 JJ[4/STC/]中的 STC=6 和 PP[5-6/STC/]中的 STC=4 由小排到大，所以 PP[5-6/4/ORDER]中的 ORDER 標記爲 1，JJ[4/6/ ORDER] 中的 ORDER 標記爲 2。

利用上述的方法得到範例樹，如圖三。如直接用整個句子的範例樹到資料庫中作搜尋，將很難搜尋到相同的範例樹，因爲句子越長句子的結構會越複雜，所以相同結構的句子重複出現的可能很低。因此，我們將範例樹的所有子樹分別取出來，每一個子樹所包含的範圍的都是英文句的子句，在不同的句子裡可能會有相同結構的子句，不但可以增加比對到的機率，也能增加範例樹的數量。最後記錄在範例樹資料庫的內容，只有範例樹和 ORDER 參數。STREE 和 STC 不需記錄的原因是每一個句子的每個詞彙都在不同的位置上，則在資料庫中不需要記錄 STREE 和 STC。

範例樹的結構有可能相同，而詞序不同。例如“NP(NP(NN fork))(PP(IN of)(NP(DT the)(NN road)))”，中文翻譯爲“岔路”，而“NP(NP(NN leader))(PP(IN of)(NP(DT a)(NN company)))”， 中文翻譯爲“一間公司的領導者”。很明顯後者中英文用詞順序不同。這裡我們採用多數決，將出現過相同範例樹結構的每種詞序作統計，在範例樹資料庫中記錄出現最多次詞序的結構。如出現最多次的次數相同，則以隨機方式選擇一種記錄在範例樹資料庫中。最後再將範例樹資料庫中沒有詞序交換的範例樹刪除，只保留有詞序交換的範例樹，可以減少搜尋相同範例樹的時間。

3.3 搜尋相同範例樹

範例樹資料庫裡，每一筆資料都包含範例樹和範例樹的 ORDER，而範例樹就是用來當作調整詞序的參考。將輸入的英文句，先透過 StanfordLexParser-1.6[17]建立剖析樹，再將剖析樹中去掉葉子節點的結構，到範例樹資料庫去搜尋是否有相同結構的範例樹，這裡我們將所搜尋到相同的範例樹稱爲匹配子樹。如圖六所示，紅色虛線框是一棵子樹其結構爲“(NP(NP(DT)(NNS))(PP(IN)(NP(CD)(NNS))))”，方形框爲範例樹資料庫中其中一棵範例樹結構爲“(NP(NP[//2](DT[//1]) (NNS[//2])) (PP[//1](IN[//1]) (NP[//2](CD[//1]) (NNS[//2]))))”，我們可以發現範例樹去除 ORDER 後的結構，會跟子樹的結構完全相同，故將此範例樹認定爲匹配子樹。



圖六、剖析樹與範例樹的對應關係

根據搜尋範例樹演算法的流程，如圖七。首先將來源句的剖析樹加到佇列(queue)裡，從佇列裡面取出一棵剖析樹到範例樹資料庫中，搜尋是否有相同結構的範例樹；如爲否，則將此棵樹的下一層的子樹加入佇列，加入佇列的順序爲左子樹到右子樹；如爲是，則將該樹的 ORDER 標記在來源句的剖析樹上，繼續取出佇列內的剖析樹，直到佇列裡沒有剖析樹爲止。所以來源句的剖析樹是由一個以上的匹配子樹所組成。

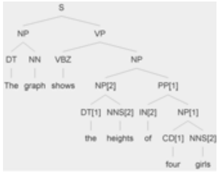
```
輸入：來源句剖析樹 S
      範例樹資料庫 D={$D_1$, $D_2$, ..., $D_m$}，$D_i$∈D
      $D_i$包含 $T_i$與 $O_i$ ，$i$ 爲 1 到 $m$
      $T_i$是第 $i$ 棵範例樹，$O_i$是 $T_i$所標記的詞序
開始
      設佇列 Q 用來儲存剖析樹，初始爲 NULL
      S 加入 Q
      當 Q≠NULL
          從 Q 中 pop 一棵範例樹
          如果在 D 中搜尋到相同的範例樹 $T_i$
            則將 $O_i$標記在 S 上
          否則將下一層子樹加入 Q
結束
輸出：標記好 ORDER 的剖析樹
```

圖七、搜尋範例樹演算法

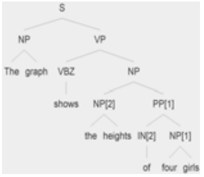
圖六爲剖析樹搜尋範例樹的情形。來源句：“The graph shows the heights of four girls”，剖析樹爲“(S(NP(DT The)(NN graph))(VP(VBZ shows)(NP(NP(DT the)(NNS heights))(PP(IN of)(NP(CD four)(NNS girls)))))”。透過搜尋範例樹演算法找出匹配子樹，首先以節點 S 爲樹根的剖析樹到資料庫作搜尋，搜尋時不包含葉子節點，此例子沒搜尋到匹配子樹，則將節點 S 的子樹 NP 和 VP 加入佇列中。接下來將從佇列中取出的子樹爲 NP，到範例樹資料庫搜尋匹配子樹，但資料庫中沒有相同的範例樹，此時 NP 的子樹皆爲葉子節點，所以並無子樹在加入佇列中。依照先進先出的原則下一個從佇列取出的是 S 的右子樹 VP，在範例樹資料庫中還是搜尋不到，因此要將 VP 的子樹 VBZ 和 NP 加入佇列中，但 VBZ 爲葉子節點，故只有 NP 加入佇列中。接下來是子樹 NP 從佇列中被取出來，子樹 NP 在資料庫中搜尋到相同的範例樹，如圖六的範例樹就是所搜尋到的匹配子樹，因此將範例樹的 ORDER 標記上去，標記後的剖析樹將如圖八所示。此時佇列中已經爲空，搜尋範例樹的流程到此爲止。

標記完 ORDER 之後，將沒有標記的子樹作修剪，也就是將不用作詞序交換的子樹修剪到最小層樹。如圖八節點 S 的右子樹、NP[2]和 NP[1]的子樹皆不需要作詞序交換，因此修剪的結果爲“(S(NP The graph)(VP(VBZ shows)(NP(NP[2] the heights)(PP[1](IN[2] of)(NP[1] four girls))))) ”，如圖九所示。最後從層數最大的每個兄弟節點開始逐層往上依照優先權順序調整剖析樹的結構；調整後的結果將會輸入到翻譯模組產生翻譯。若我們直接取來源句剖析樹的葉子節點作翻譯，將會成爲單字式的翻譯，我們將無法對詞組或片語作翻譯。翻譯的部分會在下一節會作詳細說明。

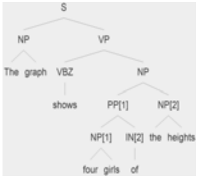
圖九的剖析樹有四層，首先將第四層的兄弟節點“(IN[2] of)(NP[1] four girls)”，依照 ORDER 的順序調整後的順序爲“(NP[1] four girls) (IN[2] of) ”，接下來第三層的兄弟節點“(NP [2] the heights)(PP[1] (NP[1] four girls)(IN[2] of))”交換後的順序爲“(PP[1] (NP[1] four girls)(IN[2] of)) (NP [2] the heights)，此例子接下來詞序沒有再調動，如圖十所示；最後輸入翻譯模組的順序爲“The graph”、“shows”、“four girls”、“of”、“the heights”，由此順序分別作翻譯處理。



圖八、完成 ORDER 標記



圖九、剖析樹修剪後的結果



圖十、調整詞序後的結果

3.4 翻譯處理

經過上一節處理最後得到修剪樹，修剪樹的葉子節點可能爲英文單字(word)、詞組(term)。詞組即爲數個單字結合的字串，不一定爲完整的句子，如“would be left on the floor”或片語(phrase，如名詞片語、動詞片語、形容詞片語等) ，如“in order to”。在翻譯處理上會遇到英文單字或詞組，在英文單字的部分，直接查尋字典檔作翻譯；詞組的部分利用規則詞典檔的片語，和詞組進行字串比對，以找出符合的片語及中文翻譯。以下爲字典檔及規則詞典檔分項說明。

字典檔：字典檔部分我們使用 Concise Oxford English Dictionary[8](牛津現代英漢雙解詞典，收錄 39429 個詞彙)，將前處理過後的英文單字或片語做翻譯對等字搜尋的動作，找出所有和該英文單字的中文詞組，作爲翻譯的候選名單。如無法在字典檔中搜尋到對應的中文翻譯。如姓名和專有名詞，則直接輸出該英文字。

規則詞典檔：爲常用的名詞片語、動詞片語、形容詞片語等詞組，以及試題翻譯小組所決議之統一翻譯詞組以人工的方式建立的中英翻譯對照檔，如 in order to(爲了)。

分成單字和詞組翻譯是因爲若在規則詞典檔比對不到，則用空白來做一般字和字之間的斷詞，也就變成單字的翻譯，因爲詞組較能完整表現出動作或敘述。如只用單字作翻譯，會造成翻譯上的錯誤。惟須注意的是比對的句型若有相似結構但不同長度的字串樣式，則取長度最長的爲結果。如一英文句子爲“…as shown in diagram…”，同時滿足規則詞典檔內的“as shown in diagram”和“in diagram”片語句型，則我們會選擇長度較長的“as shown in diagram”而不是選擇“in diagram”加上“as show”作爲斷詞的結果。

在英文翻譯成中文的過程中，有些英文單字不需要翻譯或是無意義的情形，所以我們將這些單字過濾不翻譯，這些單字稱爲 stop word。例如：冠詞 the 直接去除。介系詞 for、to、of 等，若前一單字爲 what、how、who、when、why 等疑問詞，則允以刪除，另外，to 出現在句首直接刪除。助動詞 do、does 等，判斷方式與介系詞相同。

在翻譯過程中還可能出現詞幹變化(如~ing、~ed 等)和詞性變化(如動詞 break，其過去式爲 broke，被動式爲 broken，以及名詞單複數型態)。詞幹變化的部份，我們利用 Porter[22]演算法還原各詞性(名詞、動詞、形容詞、副詞)；詞性變化的部分，有些是不規則的變化，較難用演算法處理。因此，我們透過 MXPOST[14]詞性標記工具將單字加入標記，再利用 WordNet[23]依照詞性做字典檔搜尋找到原始的型態。

3.5 統計式模組選詞

本系統將英文詞彙利用上一節介紹的翻譯方式，查詢詞典找出所有可能適合英文詞彙的翻譯結果，再利用統計式模組找出最有可能的中文詞彙，此部分已經有呂明欣等學者從事這一項研究工作[3]。以下爲我們修改後的機率模型。

$$\underset{C_{1,n}}{\operatorname{argmax}} \Pr\left(C_{1,n} \middle| E_{1,n}\right) = \underset{C_{1,n}}{\operatorname{argmax}} \prod_{i=1}^{n} [\Pr(E_i|C_i)\Pr(C_i|C_{i-1})] \tag{1}$$

公式(1)中定義 C 爲中文翻譯詞彙，E 爲英文詞彙， $E_{1,n}$爲英文句有 1 到 n 個英文詞彙，中文翻譯詞彙也會有 1 到 n 個，即 $C_{1,n}$。從公式中可發現中文詞彙翻譯成英文詞彙的機率，稱爲中英詞彙對列，即$\Pr(E_i|C_i)$；以及利用前一個中文翻譯選詞的結果 C_{i-1}，找出目前中文翻譯詞彙 C_i共同出現的機率，稱爲 bi-gram 語言模型，即$\Pr(C_i|C_{i-1})$，將兩者相乘取計算後最大的機率值，以近似$\Pr(C_{1,n}|E_{1,n})$的機率值，作爲所選擇的中文翻譯詞彙。在選詞的過程中，$\Pr(E_i|C_i)$與$\Pr(C_i|C_{i-1})$的機率值皆有可能爲 0，我們將乘 0 換成乘上一個極小數(我們預設爲 10^{-6}) ，爲了避免機率值爲 0 的情形，會影響選詞的結果。以下將針對中英詞彙對列和 bi-gram 模型詳細介紹。

中英詞彙對列：將中英語料雙語語料，經過人工的中英語句對列(sentence alignment)技術，接著將中文語料利用中研院 CKIP 斷詞系統[1]加以斷詞；英文語料則是經過大小寫轉換及利用字和字之間空白斷詞，最後輸入至 GIZA++[16]及 mkcls[15]等工具，產生中英詞彙對列結果以及中英詞彙對照機率表。

bi-gram 語言模型：將中文語料統計各中文詞彙和下一個中文詞彙出現的次數，計算其出現機率。我們是利用 SRI Speech Technology and Research Laboratory 所開發的自然語言工具 SRILM[18]來建立 bi-gram 語言模型。

4. 系統翻譯效果評估

本節主要介紹利用本系統翻譯國際數學與科學教育成就趨勢調查 2003 年考題，簡稱 TIMSS2003，並將試題依照年齡別和科目別，分別比較翻譯的品質。最後將與線上翻譯以及呂明欣等學者研發的翻譯系統作比較。評估方式爲利用 BLEU 及 NIST 指標。

4.1 實驗來源

我們主要用來翻譯的來源爲 TIMSS2003 試題，區分數學及科學類別，並且以四年級及八年級爲考試對象，共有四種試題分別爲四年級數學領域 31 題；四年級科學領域 70 題；八年級數學領域 41 題；八年級科學領域 38 題。所有試題都有英文原文試題和師大科教中心所翻譯的中文試題。

所有實驗語料句對數、中英詞彙數、中英總詞彙個數及平均句長，皆如表一所示。用來建立範例樹的來源有教育部委託宜蘭縣建置語文學習領域國中教科書補充資料題庫[4] (以下簡稱國中補充資料題庫)及科學人雜誌。國中補充資料題庫以人工方式完成中英語句對列(sentence alignment)，再經過範例樹的篩選門檻值爲 0.6 的情況下有 565 句。

用來訓練選詞機率模型的來源有自由時報中英對照讀新聞及科學人雜誌。自由時報中英對照讀新聞從 2005 年 2 月 14 日至 2007 年 10 月 31 日，而自由時報中英對照讀新聞本身就已經作好中英語句對列。科學人雜誌是從 2002 年 3 月創刊號至 2006 年 12 月共 110 篇爲語料來源。

表一、實驗語料來源統計

語料	語言	句對數	辭彙數	總詞彙個數(tokens)	平均句長
國中補充資料題庫	中文	2059 句	2333	12460	6.1
	英文		2887	13170	6.4
科學人	中文	4247 句	9279	70411	16.6
	英文		10504	68434	16.1
自由時報中英對照讀新聞	中文	4248 句	19188	145336	34.2
	英文		25782	133123	31.3

4.2 實驗設計

首先，將 TIMSS2003 試題問句以逗號、問號或驚嘆號做爲斷句的單位，每個誘答選項做爲斷句的單位，若一道題目爲一句試題問句及四項誘答選項所組成，則一道題目可斷出五句。經過人工斷句處理 TIMSS2003 試題，四年級數學領域有 165 句；四年級科學領域有 262 句；八年級數學領域有 439 句；八年級科學領域有 236 句，並整理爲文字檔。翻譯時中文試題所運用的中文斷詞爲中研院 CKIP 斷詞系統[1]，英文試題所運用的剖析器爲 StanfordLexParser-1.6[17]，建立範例樹資料庫所使用的語料爲國中補充資料題庫，訓練機率模型所使用的語料自由時報中英對照讀新聞加上科學人雜誌，其中訓練語言模型得到的 bi-gram 共有 134435 個；GIZA++產生中英詞彙對列結果有 128551 組。

表二、TIMSS 試題實驗組別表[†]

八年級 2003 M 組	八年級 2003 S 組	四年級 2003 M 組	四年級 2003 S 組	八年級 2003 MS 組	四年級 2003 MS 組
TIMSS2003 國中數學領域試題	TIMSS2003 國中科學領域試題	TIMSS2003 國小數學領域試題	TIMSS2003 國小科學領域試題	TIMSS2003 國中數學及科學領域試題	TIMSS200 國小數學及科學領域試題

我們評估所使用的工具爲依照 BLEU 及 NIST 標準的 mteval-10，並且我們將參考的中文標準翻譯和系統建議翻譯，每個中文字跟中文字之間用空白作分隔，計算出各別 n-gram 及累加各個 n-gram 的 BLEU 及 NIST 值。主要評估的對象有 Google 線上翻譯、Yahoo!線上翻譯、呂明欣學者的系統(Lu)及本系統互相做比較，並且評估翻譯系統在不同年級的試題內容上，翻譯品質是否會按照越低年級其翻譯品質越好的趨勢。因此，我們將實驗組別分爲八年級和四年級；數學領域以 M 爲代號；科學領域以 S 爲代號，當

[†] 本篇論文 TIMSS 試題實驗組，僅包含 2003 年試題，與呂明欣學者的實驗組並不相同。

作實驗組別的名稱。可以 TIMSS2003 分爲八年級 2003 M 組、八年級 2003 S 組、四年級 2003 M 組及以四年級 2003 S 組四組；在加上 TIMSS 2003 數學及科學領域之八年級試題，和 TIMSS 2003 數學及科學領域之四年級試題，分別爲八年級 2003 MS 組及四年級 2003 MS 組，總共六組，如表二所示。

4.3 實驗結果

依照上一節的實驗設計，我們針對 TIMSS2003 試題驗證本系統、Lu 系統及線上翻譯系統在 BLEU 和 NIST 比較數據。從表三是以 cumulative n-gram scoring 之 4-gram 爲平均值，整理之各組 NIST 及 BLEU 值之比較表。NIST 跟 BLEU 最大的不同在於，NIST 將各 n-gram 詞彙中共現（co-occurrence）的次數比的累加值，當作各 n-gram 平均資訊量的大小，而 BLEU 針對各 n-gram 匹配正確率及相似度進行計分。由此可知當參考翻譯句子和系統翻譯句子用的詞彙相同時，NIST 分數會比較高；當參考翻譯句子和系統翻譯句子用的詞彙順序較相近時，BLEU 分數會比較高。

表三、本系統、Lu 系統及線上翻譯系統之 NIST 及 BLEU 值比較表

組別	八年級 2003 M 組		八年級 2003 S 組		四年級 2003 M 組	
指標	NIST	BLEU	NIST	BLEU	NIST	BLEU
本系統	4.7002	0.1440	4.4089	0.1254	3.9819	0.1304
Lu	3.6185	0.1007	3.5831	0.0890	3.3319	0.0983
Google	4.5268	0.1467	4.8587	0.1848	3.7573	0.1016
Yahoo!	4.8793	0.1455	4.6136	0.1396	4.0457	0.1419
組別	四年級 2003 S 組		八年級 2003 MS 組		四年級 2003 MS 組	
指標	NIST	BLEU	NIST	BLEU	NIST	BLEU
本系統	4.2228	0.1018	4.8613	0.1309	4.4400	0.1138
Lu	3.2495	0.0682	3.8031	0.0966	3.4970	0.0803
Google	4.4445	0.1527	4.9343	0.1611	4.4720	0.1344
Yahoo!	4.4361	0.1442	5.0755	0.1435	4.6070	0.1436

從表三可觀察到，八年級 2003 M 組 NIST 分數以 Yahoo!最高分，但 BLEU 分數與本系統差不多，可知 Yahoo!對八年級 2003 M 組所翻譯的詞彙跟參考翻譯較相同，但 Yahoo!和本系統翻譯後詞序的正確性是差不多的。四年級 2003 M 組試題中有較多特殊符號，例如○和●等，Yahoo!及 Google 線上翻譯系統會將這些特殊符號處理成亂碼，但本系統可以將特殊符號保留下來，故四年級和八年級 2003 M 組與最高分系統的差距較小。先前我們假設翻譯品質是否會按照越低年級其翻譯品質越好的趨勢，觀察八年級 2003MS 組及小四 MS 組，可發現與假設相反，各系統在八年級 2003 MS 組的表現都比四年級 2003 MS 組要好。可推測出本系統其中一種語料爲國中補充資料題庫較符合 TIMSS 八年級 2003 的試題。

我們將八年級 2003M 組和八年級 2003S 組作比較，四年級 2003 M 組和四年級 2003 S 組作比較，可以發現各系統除了 Google 之外，在 M 組上表現都比 S 組好，因爲 M 組的試題內容包含較多的數字，對於翻譯系統較容易處理，而 S 組則包含較多專有名詞，對於翻譯系統較爲困難。接著將本系統與 Lu 系統作比較，Lu 系統和本系統的差別爲沒有作詞序的交換。經過詞序交換後，得到正確的中文詞序，因此選詞的正確性相對會提升，所以本系統在各組的表現都比 Lu 系統要好，顯示詞序交換後會得到品質較好的中文翻譯。

5. 結論

本論文提出 BSSTC 結構，此結構能夠記錄來源句詞彙的位置、目標句詞彙的位置及來源句與目標句詞彙對應的關係；並且將 BSSTC 結構運用在我們實作的翻譯系統上。本系統是利用 BSSTC 結構建立範例樹，將來源句經過搜尋範例樹演算法，來達到修正詞序的目的。最後，在依據修正後的詞序進行翻譯，翻譯時再利用中英詞彙對列工具及 bi-gram 語言模型，選出最適合的中文翻譯，產生建議的翻譯，此翻譯還需要人工編修。

TIMSS 的試題爲數學及科學類，應該要用大量數學及科學類的語料，但實際上我們並無法找到夠多的數學及科學類語料，尤其以中英對應的語料最少，所以我們選用新聞及國中補充資料題庫來擬補語料的不足。不過訓練量還算是不足夠，在選詞上會有許多機率爲 0 的情況，造成選詞錯誤。未來將盡量找尋相關領域的語料，來建立範例樹和訓練語言模型，就能針對不同領域的來客製化翻譯，使翻譯的結果更爲精確。

訓練語料中的斷詞是使用中研院 CKIP 系統，而我們翻譯使用的字典爲牛津字典，兩者所使用的字典並不相同，會使斷詞後的詞彙可能無法在牛津字典中找到，造成選詞錯誤。未來可將翻譯後的詞彙，找出同義詞來擴充詞彙數，便能增加被找到的可能性。

英文的語言特性上並沒有量詞，而中文句中運用了很多的量詞，如缺少量詞也會使中文的流暢度下將。本系統的翻譯結果也缺少中文的量詞。未來若能將翻譯結果塡補上缺少的量詞，便可達到更好的品質。

致謝

本研究承蒙國科會研究計畫 NSC-95-2221-E-004-013-MY2 的部分補助謹此致謝。我們感謝匿名評審對於本文初稿的各項指正與指導，雖然我們已經在從事相關的部分研究議題，不過限於篇幅因此不能在本文中全面交代相關細節。

參考文獻

[1] 中研院中文剖析器檢索系統, http://parser.iis.sinica.edu.tw/ [Accessed: Jun. 30, 2008].

[2] 自由時報中英對照讀新聞, http://www.libertytimes.com.tw/2008/new/jan/15/english.htm [Accessed: Jun. 30, 2008].

[3] 呂明欣，*電腦輔助試題翻譯：以國際數學與科學教育成就趨勢調查爲例*，國立政治大學資訊科學所，碩士論文， 2007。

[4] 教育部委託宜蘭縣發展九年一貫課程件至語文學習領域（英語）國中教科書補充資料暨題庫建置計畫, http://140.111.66.37/english/ [Accessed: Jun. 30, 2008].

[5] M. H. Al-Adhaileh, T. E. Kong and Y. Zaharin, “A synchronization structure of SSTC and its applications in machine translation”, *Proceedings of the International Conference on Computational Linguistics -2002 Post-Conference Workshop on Machine Translation in Asia*, 1–8, 2002.

[6] P. F. Brown, J. Cocke, S. A. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer and P. S. Roossin, “A Statistical Approach to Machine Translation”, *Computational Linguistics*, 79-85, 1990.

[7] C. Boitet and Y. Zaharin, “Representation trees and string-tree correspondences”, *Proceedings of the Twelfth International Conference on Computational Linguistics*, 59–64, 1998.

[8] Concise Oxford English Dictionary, http://stardict.sourceforge.net/Dictionaries_zh_TW.php [Accessed: Jun. 30, 2008].

[9] B. J. Dorr, P. W. Jordan and J. W. Benoit, “A Survey of Current Paradigms in Machine Translation” *Advances in Computers*, London: Academic Press, 1-68, 1999.

[10] Google Translate http://www.google.com/translate_t [Accessed: Jun. 30, 2008].

[11] K. Knight and S. K. Luk, “Building a large-scale knowledge base for machine translation”, *Proceedings of the Twelfth National Conference on Artificial intelligence*, 773-778, 1994.

[12] P. Koehn, F. J. Och and D. Marcu, “Statistical phrase-based translation”, *Proceedings of the Human Language Technology Conference,* 127–133, 2003.

[13] Z. Liu, H. Wang and H. Wu, “Example-based Machine Translation Based on TSC and Statistical Generation”, *Proceedings of the Tenth Machine Translation Summit*, 25–32, 2005.

[14] MXPOST, http://www.inf.ed.ac.uk/resources/nlp/local_doc/MXPOST.html [Accessed: Jun. 30, 2008].

[15] F. J. Och, “An Efficient Method for Determining Bilingual Word Classes”, *Proceedings of European Chapter of the Association for Computational Linguistics*, 71–76, 1999.

[16] F. J. Och and H. Ney, “Improved Statistical Alignment Models”, *Proceedings of the Thirty-eighth Annual Meeting of the Association for Computational Linguistics*, 440–447, 2000.

[17] The Stanford Parser: A statistical parser, http://nlp.stanford.edu/software/ [Accessed: Jun. 30, 2008].

[18] A. Stolcke. SRILM – an extensible language modeling toolkit. *Proceedings of the intelligence Conference on Spoken Language Processing*, 901–904, 2002. http://www.speech.sri.com/projects/srilm/ [Accessed: Jun. 30, 2008].

[19] S. Sato and M. Nagao, Toward Memory-Based Translation”, *Proceedings of International Conference on Computational Linguistics*, 247–252, 1990.

[20] The International Association for the Evaluation of Education Achievement, http://www.iea.nl/ [Accessed: Jun. 30, 2008].

[21] TIMSS 中文版官方網頁, http://timss.sec.ntnu.edu.tw/timss2007/news.asp [Accessed: Jun. 30, 2008].

[22] The Porter Stemming Algorithm, http://www.tartarus.org/martin/PorterStemmer/ [Accessed: Jun. 30, 2008].

[23] WordNet API, http://nlp.stanford.edu/nlp/javadoc/wn/ [Accessed: Jun. 30, 2008].

[24] F. Wong, M. Dong and D. Hu, Machine Translation Based on Translation Corresponding Tree Structure, *Tsinghua Science & Technology*, 25–31, 2006.

[25] YAHOO! 雅虎線上翻譯, http://tw.search.yahoo.com/language/ [Accessed: Jun. 30, 2008].

訴訟文書檢索系統

藍家良　　賴敏華　　田侃文　　劉昭麟
國立政治大學資訊科學系
{ g9542, g9523, g9627, chaolin }@cs.nccu.edu.tw

摘要

在訴訟文書檢索需求上，對於法官而言，須要找到相似的案例輔助判決；對於學習或研究法學的人來說，藉由檢索大量案例，用以分析探討相關議題；而一般民眾，則可藉由實際案例，來吸收法律上的基本知識，用以保障自身權益。

訴訟案件與日俱增，欲閱讀完所有案例顯然不容易，此時便需要一套較完善的檢索系統來輔助使用者。我們利用現有自然語言處理與文字探勘等技術，設計一套分類式檢索系統，依檢索條件搜尋相關案例，並將結果分類輸出，方便使用者對各類別進行查詢，以期減少使用者閱讀文件上的負擔，同時獲得較完整資訊。另設計文件標記與註解功能，供使用者建立個人化資料庫，便於日後檢索或藉由此資訊修正自動分類機制。

關鍵詞：
法學資訊系統、人工智慧與法律、階層式分群

1　緒論

近幾年電腦的軟硬體技術迅速成長，網際網路的高度普及，許多資訊都經由數位化，得以迅速發佈傳播，且使得檢索變得十分便利，在法學相關資訊上亦是如此。我國的司法院法學資料檢索系統[2]，就提供了中央與地方的法規查詢，以及司法解釋和法院的判例，開放給大眾使用。

依司法院統計處[3]資料顯示，台灣在 2007 年地方法院刑事案件終結件數達 41 萬件，民事案件更多達 268 萬件。每天約有一萬件的案件被終結，以台灣約 2300 萬人口來估計，平均約每年每七人就有一人須上法庭，如果去掉未成年和老人等較無犯罪能力的人，比例則更高。這樣的情形並不代表社會環境的惡劣，而是隨著教育水準的提高，資訊取得容易，人們對於法治的概念日益成熟，愈來愈多人懂得利用司法來保障個人的權益。

然而這樣造成了法官與律師等人員，工作量上的極大負擔。此外，即便他們熟讀各項法律條文，但在面對實際案件時，仍有各種不同的情形須要面對，須要參考過去的判決來決定。雖然本節一開始提到數位化可使資訊檢索變得容易，但面對檢索結果仍包含的大量資訊，則仍是一件耗時的事。

基於上述的理由，本研究欲設計一套分類式檢索系統，協助使用者在取得大量結果時，能經由系統自動分類機制，以分類為出發點檢索訴訟文件，對於較不相關的分類，經由少數幾篇的檢查來跳過此分類，能減少檢索的時間與增進檢索效率。

在第 2 節中，我們將介紹相關研究議題；接著在第 3 節介紹本系統的設計以及操作介面；在第 4 節是本系統所使用到的相關技術部分；第 5 節作一些初步的評估；最後第 6 節作結語。

2　相關研究

Hearst[14]提到搜尋資料的人希望能有介面能幫他們把結果作分類，使資料更有意義且更方便其作檢索，快速過濾不符合需求的資訊。作者討論了分群 (clustering) 與建立各個面向類別 (faceted categorization) 兩種方法的優缺點。分群法優點是整個過程可自動化，可找到一些有趣的類別特徵；缺點則是根據統計結果所得出的類別名稱很可能不具代表性，甚至造成類別的凌亂，使得使用者無法透過類別名稱來協助其檢索資訊。而另一種方法則是以人工方式手動建立類別，再依據類別的特徵比對搜尋結果，架構良好的類別可使分類較清楚，是相對於分群法較好的部分，然而這樣可能無法包含所有類別，其類別的決定也必須是搜尋資料的人知道的，這樣對其才有意義。

何君豪[10]將階層式分群法應用在民事裁判要旨分群上，法官可以藉由分群後的結果來一一檢視群集，當檢視完群集中幾篇裁判要旨，發現不符合他的需求，便可以忽略此群集的其他裁判要旨，藉以減少法官耗費在民事裁判要旨閱讀時間。作者利用聚合法[13]的方式來分群，不斷合併相似的案例至一個門檻值。這裡藉由群數或相似度門檻來設定合併終止條件，由使用者個別檢驗，對於無法以簡單描述給予類別定義的問題，是較直接的作法。

Schweighofer 等學者[18]提到將法律文件以向量維度的方式展現是很好的作法，包括在計算相似度、分類或是內容的描述上，也指出單純使用 tf–idf (term frequency – inverse document frequency) [15][16]來計算向量以表示文件，即使利用完善的計算相似度公式仍顯不足，作者使用法律領域較簡單的本體架構 (ontology) 來改進這樣的缺失，另外也針對特定的法律名詞作加重權重的動作。但這些還是會回歸一個基本問題，就是須要有一個公信力的方式設計本體架構，用以建置詞典或是規則。

謝淳達[12]研究相似訴訟文書的檢索，利用詞組為基礎，將文章轉換為向量，以 k 最近鄰居法(k nearest neighbors methods，簡稱 kNN)概念設計分類演算法，作訴訟文書的分類工作。

鄭人豪[11]討論詞彙來源與權重對中文裁判書分類成效影響，比較從 HowNet[5]擷取出詞彙產生之詞典與 TermSpotter 演算法[12]取出訴訟文件特定詞彙輔以人工修正，所建立之詞典，對於分類效果之影響。觀察案例的相似度分佈，找到適當參數，提升分類效果。利用 kNN 作為系統分類機制分析分類效果。另依自省式學習法精神，建立權重調整機制，分析權重調整對分類效果的影響。

3　系統設計

現有的司法院法學資料檢索系統[2]中的裁判書查詢，提供了基本的檢索裁判書功能，可設定法院、裁判類別、案由、時間及關鍵字等條件，取得符合的結果，在介面上顯示至多 100 筆的資訊。這樣的檢索介面可以協助法官和律師，甚至一般民眾取得所需的資訊。然而依照關鍵字搜尋出來的結果，很可能包含大量的資訊，我們仍要透過逐一檢視來過濾，或重新斟酌檢索條件的設定。這樣的問題是在檢索大量裁判案件時容易發生的情形。

本研究欲設計一套訴訟文書檢索系統，提供不同於上述的檢索功能。對於檢索結果而言，如果所獲得符合條件的結果很多，又不希望設定更複雜的檢索條件，那此時須要改進的便是檢索結果的呈現方式。若能使得檢索結果依照裁判書內容的不同進行分類，則可協助使用者在檢索結果時，依照類別作檢索。其中不符合其需求的類別，可以在較短時間內決定略過，方便其檢索的效率以及組織資訊。對於檢索條件而言，加上簡單的檢索條件，協助本系統依此條件作分類，為使用者整理結果，可方便其閱讀。

3.1 前處理

在裁判書的來源上，我們利用司法院法學資料檢索系統中的裁判書查詢[2]功能取得刑事案件的裁判書，使用網路爬梳器 (crawler) 擷取裁判書，並過濾其中有問題的文件，如亂碼等問題的文件。目前本研究所使用裁判書範圍是從民國 88 年至民國 95 年之間各地方法院，裁判案由之中屬於竊盜、搶奪、強盜、贓物、傷害、恐嚇和賭博這七大類的裁判書，以一般文字檔案格式儲存，目前總數為 9296 篇。許多的裁判書中並非只有單一案由，在本研究中，我們將裁判書上的裁判案由欄位視為主要案由，也依此進行統計，如表 1 所示。

表 1　裁判書各別案由數量統計

案由	竊盜	搶奪	強盜	贓物	傷害	恐嚇	賭博
總數目	312	386	1267	2083	989	385	3874

為了解決大量文件的查詢效率，抓取下來的裁判書須經過處理，建立索引等資訊。大量文件檢索為另一個研究議題，本研究中不探討此部分，而是使用現有的工具 Lucene，替裁判書製作反向索引 (inverted indexes) 且提供查詢功能，以增進檢索效率。Lucene 內部負責製作索引之前斷詞功能的程式叫作 Analyzer，其中有多種不同實作方式，像是 Lucene 內建的 StandardAnalyzer、CJKAnalyzer 和 WhitespaceAnalyzer 等。其對中文字的斷詞支援只限於一個字一個字切割或是兩兩成一詞去分割，於是我們選擇國內研究普遍採用中研院的中文斷詞系統[1]，預先進行斷詞，再一一將斷詞後的內容用空白隔開，最後再利用 WhitespaceAnalyzer 依據空白斷詞來建立索引。

在本系統裁判書的分類中，須使用到斷詞詞典來取出文件中詞彙，用以進行相似度的比對。若使用一個很完整的一般常用詞典，易造成向量維度過大 (因為詞彙的組合很多種) ，進而影響系統的效率，且可能因大部分不相關的詞彙而影響結果。對於像法律這樣專業的領域有許多專有名詞，而有些詞彙則是制式或習慣用法，如果能將其建成專業詞典，便能大幅減低一般詞典所造成向量維度過大的問題。本系統使用第 2 節提到 TermSpotter 演算法所擷取出的詞彙作為詞典，將裁判書斷詞轉換為向量，來進行依相似度的分群方法。

3.2 系統功能及操作介面

圖 1 為使用者在操作本系統進行的流程。以下分別簡介各階段內容，詳細作法我們會在後面作介紹。

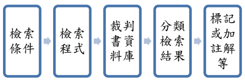


圖 1　系統操作流程

- 一開始使用者可選擇輸入欲檢索的詞彙關鍵字或是一段犯罪事實陳述，以及選擇結果的呈現方式，由不同的檢索程式處理。
- 檢索程式依照分類數目、刑度選擇以及事實相關程度查詢等不同需求條件，由各個程式從資料庫中擷取相關資料進行分類。
- 資料庫為了提供檢索程式大量資料，除了原始的資料 (裁判書) ，另預先建立索引和轉換資料，提供不同檢索程式進行快速存取。
- 最後輸出分類提供使用者選取，檢索內容。另外我們也提供使用者不同類別中出現的相關詞彙資訊，使其能更快掌握相關資訊。
- 在閱讀文件時，可對文件進行標記，如：選擇有興趣之關鍵字，作為日後檢索時特別標示出來之文字，也可進行統計或用以改進分類結果。而加上註解的功能，提供使用者對一段文字作註解，且可自行定義類別，方便日後查詢。

在實作方面我們採用 Java 語言來開發，且使用由 Java 所建構之工具，如 Lucene[7]用來進行全文檢索；HSQLDB[6]資料庫用來儲存標記和註解等資訊，便於系統之整合。

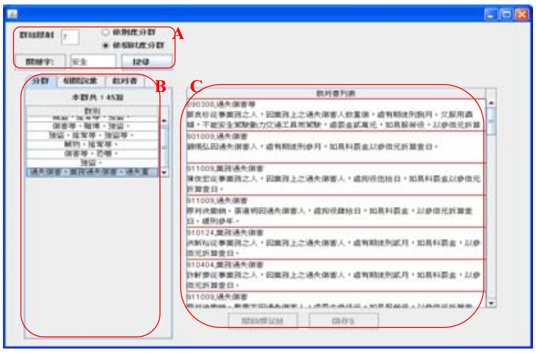
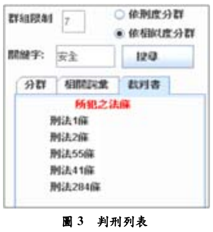


圖 2　操作介面

圖 2 為目前設計的操作介面，主要有 A、B 和 C 三部分。在 A 輸入檢索的條件；搜尋之後，在 B 中顯示分群結果和相關資訊供使用者檢索；C 則依照 B 中的操作顯示分群裁判書列表或裁判書內容。礙於篇幅限制，在之後僅附上部分截圖。

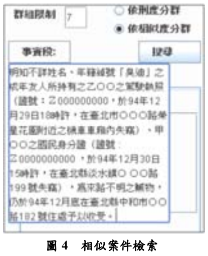
依相似度進行裁判書分群檢索的操作介面，如圖 2 的 A 部分為檢索條件的設定，包含了檢索關鍵字、分群數限制、分群依據等。圖 2 的 B 部分則是在設定好檢索條件，按下「搜尋」後的分群結果，會顯示各分群所包含裁判書之案由統計的前三名。點選任意一群後，顯示此群集的所有裁判書列表，如圖 2 的 C 部分所示。其中每一個選項提供了三個資訊：日期、案由和主文。任點選一篇後便會顯示裁判書內容，同時操作介面切換到「裁判書」標籤，如圖 3 所示，上面會列出此裁判書最後所裁定之刑法法條。點選法條後會自動連結到全國法規資料庫[4]去檢索此法條內容。

前述功能是針對提供資訊較少且較不明確的關鍵字，所進行的分群檢索。接著我們要介紹**相似案件檢索**功能，是可以在檢索條件中輸入一段犯罪事實，透過將其轉換成向量來計算與資料庫中的裁判書相似度，輸出相似的裁判書，提供使用者作裁判的參考依據或作相關案件的研究。輸入介面如圖 4 所示，當我們在圖 2 的 A 部分中按下「關鍵字」的方格，可切換至犯罪事實的輸入方塊，按下搜尋之後，會顯示相似分群。我們提供最相似的兩個案由分群，將其相似案件依相似度排序，並將各案由之案件分為兩半，總共輸出四個分群，操作介面及功能同之前的分群檢索。



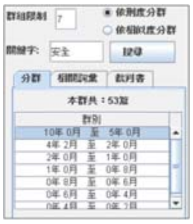
所犯之法條
刑法1條
刑法2條
刑法55條
刑法41條
刑法284條

圖 3　判刑列表



明知不詳姓名、年籍綽號「臭迪」之成年友人所持有之乙○○之駕駛執照（證號：Z000000000，於94年12月29日18時許，在臺北市○○○路華星花園附近之機車車廂內失竊），甲○○之國民身分證（證號：Z000000000，於94年12月30日15時許，在臺北縣淡水鎮○ ○○路199號失竊），為來路不明之贓物，仍於94年12月底在臺北縣中和市○○路182號住處予以收受。

圖 4　相似案件檢索



本群共：53篇

群別
10年 0月 至 5年 0月
4年 2月 至 2年 0月
2年 0月 至 1年 0月
1年 0月 至 0年 8月
0年 8月 至 0年 6月
0年 6月 至 0年 4月
0年 4月 至 0年 2月

圖 5　依刑度分群

接下來介紹**量刑輔助檢索**功能，此功能可延續之前對於未知案件分群檢索後，為了協助使用者檢索裁判書時，能得知過去這些案件被判決的刑罰輕重，以不同的區間分別列出，方便分析檢索。我們利用正規表示法 (regular expression) 從裁判書主文區段中擷取出刑罰量刑的部分，如有期徒刑 (參) 年 (貳) 月。經過統計及排序，依照區間將裁判書分類顯示。另外我們也可以提供法官、律師對特定刑罰的區間作檢索，輔助其考慮案件判決輕重。介面如圖 5 所示，大致上與依相似度進行裁判書分群檢索介面相同，系統將檢索結果依刑度來排序，圖中的例子將結果平均分成七群，未來將再加入圖形化的介面來顯示某一群相關案件的判刑分佈。目前只考慮有期徒刑的刑期，排序後再將不同的區間個別分成一群，未來可加入其他量刑評估機制，如拘役、罰金和易服勞役等。

現在已經介紹分群的功能及介面操作，接著要介紹我們所提供的其他資訊，**相關詞彙查詢**功能。資訊檢索上常用到的基本技術，就是利用詞彙與詞彙之間的共現 (collocation) ，可從前後文來搜尋出時常同時出現的字串，此字串很有可能是具有意義的，代表著這些詞彙可能有實際上的關連性。我們欲提供使用者透過輸入詞彙，本系統找出常出現在其前後的詞。對於一般使用者來說，可以提供與關鍵字時常共現的詞彙組合，點選其中的詞彙，繼續檢索相關裁判書。利用此方式來對搜尋結果作分類，以詞頻高低作為結果排序。此功能可使一般民眾或學習法學的使用者了解相關詞彙的出現情形，也有機會藉此找出某些犯案的關連性。

介面如圖 6 所示，我們在查詢時在關鍵字欄位鍵入「安全」這個詞彙，搜尋之後，點選「相關詞彙」的標籤，便會出現圖中 E 部分第一層的相關詞彙列表，點選「威脅」詞彙便會出現裁判書列表如本節圖 2 的 C 部分形式，同時圖 6 中 F 部分也會出現與檢索詞彙以及所點選的相關詞彙皆共現的詞彙列表，點選後同樣會出現相關裁判書列表。另外為避免第一層相關詞彙列表過大使得顯示速度緩慢，加上圖中間的捲軸來切換頁面，每頁僅顯示 20 項，依照共現頻率由大至小排序。

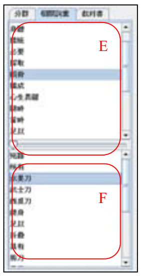


圖 6　相關詞彙列表

使用者在閱讀裁判書時，我們提供**文字標記**的功能。全球資訊網進入 Web 2.0 的時代，O'Reilly[17]提出與過去 Web 1.0 不同的其中一項特徵是從分類學 (taxonomy) 到現在的大眾分類法 (folksonomy) ，由使用者來決定類別，雖說多數暴力(tyranny of the majority)是個須要深入研究的議題，但這樣的分類方式更貼近使用者，如能用在個人化上，也能協助使用者更精確迅速地找到所需的資訊。因此我們加上案件標記的功能，協助使用者對裁判書內容作標記及註解。

目前我們的標記方式分為兩種，一種是選擇某段文字利用更改背景顏色作標記，加上意見註解、設定意見的類別，或是增加新的類別。而文字背景由類別來決定，不同類別可以設定不同顏色。操作介面如圖 7 所示，在選取「致生危害安全」按右鍵

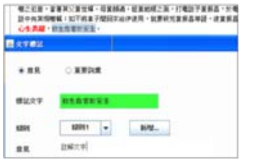
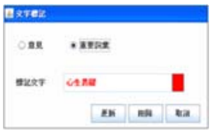


圖 7　標記註解文字

選擇標記之後，會跳出文字標記的視窗，在「標記文字」欄位可以預覽標記成類別 1 後的顏色設定，另外也可以按下「新增」，新增標記類別以及設定標記的顏色，「意見」欄位則可輸入要註記的文字。標記後的資訊會存至資料庫中，以後開啟此文件會將這些部分作標記，點選標記部分之後會再重新顯示註記資訊。

圖 8 為另一種標記方式，重要詞彙標記是選擇一段文字將其前景(即文字)顏色作變更，日後在閱讀裁判書時，本系統會自動將所有包含的重要詞彙依相對應的顏色作標記。圖 7 案件中的「心生畏懼」四個字便是圖 8 標記後的結果。



標記文字 心生畏懼

圖 8　重要詞彙標記

文字加上註解意見並分類的目的，是希望使用者日後可以針對自己所分類的資訊作檢索，可能是特殊情形的判決、文字敘述所代表的含意或是有爭議的陳述等不同的註解類別。而重要詞彙的標記則是可以讓使用者透過自訂的詞彙以不同的顏色呈現而能快速檢視裁判書的內容。進一步由法官所標記的一些特定詞彙也可以作為我們在對裁判書斷詞或分類時，所參考的依據。

對於標記文字或註解紀錄的儲存，我們有兩種方式可以考量，一種是利用標記語言 (Markup Language) 如網頁的形式，使用不同的標籤，來標記註解片段；另一種方式是採用資料庫來儲存與檢索，而不修改原始文件。為了方便管理標記註解的資訊，我們選用後者的方式來記錄相關資訊。考量到目前以個人化為主，在單機上執行，我們考慮的資料庫，屬於較精簡、小量資料存取且可當作一般處理程序而不作為系統服務的嵌入式資料庫（embedded SQL database），如 SQLite[7]和 HSQLDB[6]等開放原始碼的軟體。我們目前採用 HSQLDB，因為此資料庫系統完全由 Java 開發，我們可以將其整合至我們的系統，將資料庫存取視為一般檔案存取的處理程序。另外此資料庫也採用標準的 SQL 語法，若日後本系統要改為線上多人作業，也能較容易作轉換。

4　相關技術

本節介紹所使用到相關技術及討論，以下各小節將分別敘述。

4.1 階層式分群演算法

利用階層式分群演算法，將符合條件的結果進行分群，可依聚合法將文件依相似度不斷合併至適當的群數。本系統採用此演算法，實作裁判書分群檢索功能。在分群的數目上，可由使用者自行決定。

在分群演算法中的相似度計算，何君豪[10]的研究中比較了最小值 (min) 、配合係數 (matching coefficient) 、Jaccard 係數 (Jaccard coefficient) 和餘弦函數 (cosine) 四種方法來計算相似度，結果顯示採用餘弦函數來作計算在整體表現上能得到較好的分群效果。此公式作法是以兩篇文章同時出現詞彙的向量乘積總和除以兩篇文章個別長度的乘積，考慮到了文章長度會影響共同詞彙出現的機率，將兩篇文章個別的長度列入考量，以緩和文章長度對於近似值的影響。基於前述原因，本系統也採用餘弦公式 (公式(1)) 來計算兩篇文章所轉換之詞彙向量 $\vec{X}=\{x_1,x_2,\dots,x_i,\ \dots,x_n\}$， $\vec{Y}=\{y_1,y_2,\dots,y_i,\dots,y_n\}$ 之相似度，其中 x_i 和 y_i 代表 n 個詞彙中詞彙 i 的權重，而 n 則是代表詞典的詞彙數，也就是文章所轉換成向量的維度。依據餘弦公式可得知其計算結果範圍介於 0 到 1，所計算出的值愈大代表其相似度愈高。另外權重的設定是依照 tf–idf 來計算向量以表示文件，如公式(2)所示，對於一份文件中詞彙 i，tf_i 代表這個詞彙在此篇文章中出現的次數，N 代表總文件個數，n_i 則代表此詞彙在幾篇文中出現過。

$$\boldsymbol{sim}(\vec{\boldsymbol{X}},\vec{\boldsymbol{Y}})=\frac{\sum_{i=1}^{n}x_iy_i}{\sqrt{\sum_{i=1}^{n}x_i^2\times\sum_{i=1}^{n}y_i^2}}\qquad\textbf{(1)}$$

$$\boldsymbol{x_i=tf_i\times\log(^{N}/_{n_i})}\qquad\textbf{(2)}$$

演算法如圖 9 所示，其中步驟三在計算相似度時為了減少計算量，我們不計算重複的組合，如 (a, b) 與 (b, a) ，其中 a, b 為兩案例的詞彙向量，所以第一次計算所有 m 個案例間相似度的計算量為 m(m-1)/2，分群數目則由使用者決定。另外在兩個群集的相似度計算上，我們採用完整連結聚合 (complete-linkage agglomerative) ，也就是在群集間相似度計算是取其中兩個點相似度最小的值，作為兩群集間的集合相似度。

```
輸入：裁判書資料庫集合 $db$，檢索條件 $f$，分
      群數 $n$。
輸出：案例分群集合 $TT$。
步驟一：依據 $f$ 從 $db$ 中取出符合條件，各個
        案例之事實段，形成集合 $T$，其中
        包括 $m$ 篇事實段詞彙向量，$T_1... T_m$。
步驟二：建立 $m$ 個詞彙向量集合，形成集合
        $TT$，其中 $TT_i$ 含有詞彙向量 $T_i$，
        $i$ = 1 to $m$。
步驟三：令 $CT$ 用來儲存兩兩集合之相似度
        資訊。
        for( $i$ = 1 to $m$-1){
          for($j$ = $i$+1 to $m$){
            計算 $TT_i$,與 $TT_j$ 之相似度儲存
            至集合 $CT$
        }}//計算各集合間相似度。
步驟四：自 $CT$ 取出相似度最大之集合對，
        進行合併集合，而 $TT$ 轉變為新的
        集合 $TT'$，$TT = TT'$，$m = m$-1。
步驟五：若 $m$ 達到我們目標分群數 ($m = n$)，
        則中止運算，回傳 $TT$
        否則回到步驟三。
```

圖 9　分群演算法

4.2 相似案件分群

向量轉換及相似度計算上我們同樣採用 4.1 節中的方式處理，以下是相似案件檢索的處理流程。

a. 將所有裁判書依照與輸入之犯罪事實相似度由高至低作排序，只留下超過相似度門檻值的裁判書，目前門檻值定為 0.2。
b. 統計各個案由的裁判書數量，取最高兩位作為輸出之結果。案由種類如 3.1 節表 1 所示之七種案由。
c. 上一步所得到之結果，分別有兩個案由之兩類裁判書，而且各自依照相似度由大至小排列，我們再將兩類各自對半分成兩類。之所已將同一類別再分成兩半是讓使用者便於觀看，以及加速大量案件在顯示時的速度。
d. 最後兩個案由共分成四份，將會依照相似度由高至低依序輸出。

我們在作法上延伸 kNN 的概念，利用相似度來進行投票，統計相近的分類，但在輸出的結果上不決定唯一的類別，而是將最有可能的前兩名案由予以輸出，且提供其投票的裁判書列表。這樣對於使用者來說，更有機會取得較多也較正確的相關裁判資訊。

4.3 建立相關詞彙

在這小節我們要介紹共現詞彙的索引建立方式，事先建立共現詞彙，以增進系統的運作效率。

作共現詞彙的相關統計之前，還須要先決定一件事，就是如何將一段文字作斷詞，作為統計的單位。由於英文較無斷詞方面問題，可以用空格作為斷詞的依據，而中文查詢系統須要決定是將詞典所含詞彙從文章擷取出來作為查詢單位，或是以原始文字經斷詞程式斷詞來當作查詢單位。若以詞典斷詞作為單位，很可能錯失一些特殊或隨著不同的資料出現的詞彙，而以原始文字作延伸，很可能找出無意義的組合，這是必須考慮之處。斷詞的問題顯然也會發生在對原始文章的統計及搜尋上，若使用現有的斷詞系統，如中研院的中文斷詞系統[1]，可以完整統計所有詞彙與計算共現詞彙，但同時也會出現許多非使用者所期望的相關詞彙資訊，如：連接詞、冠詞或甚至錯誤的斷詞等，要再加以過濾這些組合，以符合需求。若是單純使用詞典，無論一般或是此領域的專業詞典，會漏掉許多未收錄在字典中的資訊。

在上述情形之下，我們採用中研院的中文斷詞系統將所蒐集的裁判書作斷詞，在斷詞之後，可同時得到詞彙的詞性。另外依詞性過濾一些詞彙，以保持文件中資訊完整性以及儘量減少一些較“沒有幫助”的詞彙，也減少統計量上的負擔。目前過濾的詞彙之詞性有專有名詞，如人名等資訊、連接詞，如「和」及「或」，以及數詞定詞，如「三百五十」，對於目前我們所提供的相關詞彙之查詢目的是無關聯的。而地方詞，如地點「南港」或是門牌號碼「225 號」，我們認為前者應該保留，而後者較不具意義，目前對於地方詞皆暫時予以保留。

在過濾完一些詞彙之後，將詞彙去掉詞性並以空白隔開。我們對相關詞彙的處理分為兩步驟，第一個步驟是建立相關詞彙的索引，第二步驟則是依據相關詞彙取得含有詞彙共現之文件。由於本系統透過 Lucene 建立索引，在前處理上先對所有詞彙建立索引的機制，以便在檢索時能快速建立目標清單。但 Lucene 設計的目的不在於提供相關詞彙查找功能，而是快速且多樣化進行文件檢索，故我們需要額外再建立共現詞彙列表。

我們在第一步驟時只建立各個詞彙的共現詞彙列表，而不儲存共現詞彙所出現的檔案資訊。在第二步驟則是利用 Lucene 的搜尋程式進行檢索，此時才將出現共現詞彙的檔案列出。這樣的方式除了能快速的搜尋之外，同時我們將共現詞彙列表獨立出來，日後也可以進一步讓使用者針對共現詞彙列表作編輯，像是想要去除某些使用者認為沒有必要列出的共現詞彙，如，「檢察署」和「檢察官」。

5　初步評估

首先我們將 3.1 節提到 9296 篇已取得的裁判書經過中研院斷詞系統斷詞，再經過處理，過濾掉 4.3 節所提到的詞彙，得到每份文件中一個一個以空白斷開詞彙的檔案。依此建立 Lucene 的索引檔，這部分耗費約 50 秒的時間，且其中包含約 33000 個的不重複詞彙。

接下來我們測試相關詞彙建置時間，我們目前在建立相關詞彙共現所設定的詞彙間隔為 2，也就是說我們將每個詞彙的前三個與後三個詞彙視為共現詞彙，建立兩兩共現詞彙索引的時間約為 50 秒，建立三個詞彙共現索引時間約為 14 分鐘，未來我們會將較低頻的詞彙過濾，以增快處理速度。至於相關詞彙的檢索速度上，目前都能在一秒內予以回應，沒有明顯的延宕時間。

另外階層式分群檢索的分群時間上，須要考慮到取得的裁判文件個數和欲分群數來決定分群速度，以分成七群為例，若回傳的文件數量為 1100 篇，需要 12 秒左右時間得到結果，若回傳文件數量 950 篇則需要 7 秒左右，700 篇則僅需 2 秒時間。結果顯示，階層式分群在回傳的文件數量增加之下會造成分群時間大幅上升，這是需要再考慮的。評估是否足夠滿足使用者，若不足，能否透過最佳化程式來降低分群時間，或是尋求其他較有效率的分群技術，來完成我們的系統。

相似案件分群的效果部分，從鄭人豪[11]所作不同門檻值間對於正確率影響的實驗部分發現，使用 kNN 所得分類正確率約莫可達到七至八成，目前我們沒有再對門檻值的決定作分析以及修正。

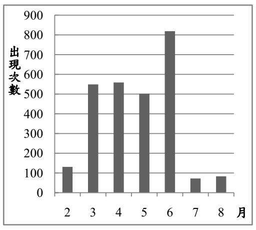


圖 10　賭博案判刑分布圖 1

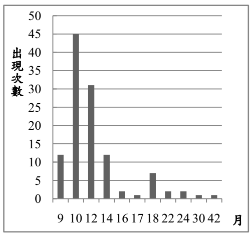


圖 11　賭博案判刑分布圖 2

在判決量刑輔助上，我們試著以裁判書案由出發，統計判刑的分佈，討論此功能對於使用者的幫助。我們以現有 3.1 節表 1 所列的裁判書資料利用正規表示法將裁判書中主文段所記錄的判決刑期部份擷取出，統計了各個判決有期徒刑的刑期，以及刑期的出現頻率。

圖 10 和圖 11 我們統計的是賭博案的部分。為了清楚看見分佈的差異性，我們以兩張圖來展示，其中橫軸座標代表獨立月份，如橫軸中「2」所統計的出現次數代表判刑為兩個月的出現次數。顯然在我們的資料庫中，賭博案刑期大多集中於圖 10，也就是判刑八個月以下，圖 11 是判刑九個月以上，數量上則大幅降低。我們依判決刑度不同觀察部分裁判書，發現賭博常因「意圖營利」、「以賭博為常業」和「聚眾賭博」等理由而論罪。判一年以上較重的罪，多有連續意圖營利、犯罪時間長甚至累犯的情形發生；五個月以下較短刑期會出現非開設賭博場所，而為受雇者或犯罪時間短，犯後態度良好等情形。

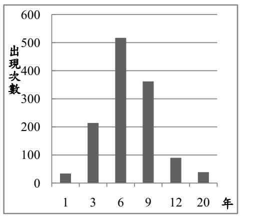


圖 12　強盜案判刑分布圖

圖 12 為強盜案件的判刑分佈，因為分佈區間較廣，我們以「年」作區隔，所以橫軸座標代表某個區間的統計次數，如橫軸中「3」所統計的出現次數代表判刑超過一年且在三年以下的出現次數。相較於賭博案件，顯然強盜案件的判刑刑期偏高 (未統計無期徒刑) 。強盜案常伴隨著恐嚇、傷害、竊盜等事件發生，因此會因為攜帶的兇器、對被害者的傷害程度、是否結夥和盜取的物品等因素而決定判刑，所以刑期分佈相當廣。可以藉由判刑分佈來觀察部分關聯性。

上述我們以一般民眾的角度去觀察，看到一些不同考量點所造成刑罰輕重不同。對於法官來說，可以藉由本系統對於刑期排序，來輔助其判決相似案件；也可依判刑結果出發，研究判刑的適當性；或是針對特殊判決的案件 (如判刑特別重的案件) 作瞭解。

在刑法明文規定上，都有針對各種犯罪行為設定刑罰的區間，但依各個案件的不同情形，刑罰仍應有不同的調整。自由時報電子報(96 年 12 月 17 日)報導中提到，「法官量刑有公式可循，可依被告的犯案次數、危害程度、危險性等變數，算出被告刑期，以增加判決透明度」，「有公式可以參考，可減少困擾，民間司改會律師高涌誠也表示不反對此制度，將來法院可以將『有罪？無罪？』及『量刑』分開辯論，可更保障當事人權利」。在目前沒有量刑公式的情形下，法官心中仍有一把尺，只是寬鬆程度不一，所以常會有判刑過輕過重的情形發生。在量刑公式尚未產生前，我們可以提供法官較清楚的參考資料，為其作好分佈統計，讓法官更客觀地去參考過去的判決考量。日後若有實際的量刑機制產生時，本系統可加入其量刑計算公式，提供更詳盡的分類，以便法官進行檢索以作為判決參考。

本系統整體功能取向異於司法院法學資料檢索系統[2]所著重部分，觀察他們系統可以發現，為了找到特定裁判書提供了判決字號、判決日期以及關鍵字等欄位作為檢索條件，而閱讀個別案件。我們則以內容為出發點，對於裁判書作不同機制分群，以期減少資訊亂度，讓使用者更迅速找到需要的資訊，甚至對現有裁判書作進一步分析探討。

6　結語

本系統在基本設計上，如同一般檢索介面，先提供設定檢索條件，選擇需要檢索的資訊，包含選擇檢索結果的分類形式。接著透過檢索程式，將資料庫中符合的文件取出並進行分類。最後顯示資訊時在不同類別上有簡要資訊，使用者可以選擇類別，接著閱讀其中的裁判書。

我們在提供資訊上，由簡至繁由上而下，透過最精簡的類別資訊，以及相關詞彙資訊，到裁判書的摘要資訊，最後裁判書內容和標記註解等資訊。盡量使用裁判書本身所提供的資訊去分類，除了讓使用者藉由分類獲得更多資訊，也同時協助他們過濾其他較不相關，或不感興趣的資訊，增進在閱讀大量裁判書時的效率。

另外提供裁判書進行加上標記、註解等動作，儲存其紀錄，建立個人化資料庫，以便於日後之查詢。未來我們將加強個人化的部分，對於系統的回饋，例如修正詞典、增刪相關詞彙的查詢等功能，以期系統更符合使用者之需求。此功能對於學習法學的人、法官或是律師等使用者，可協助其建立一套個人化的知識庫，記錄過去對於不同判決、不同犯罪敘述所下的註解，或是標記出個人感興趣的敘述。將前述資訊記錄下來，可提供日後進行檢索，或對其感興趣的內容作反白標記，方便閱讀，以及尋找使用者過去對於相關事件的處理模式。在標記上可加上自定類別，方便日後檢索及閱讀。

目前完成基本的操作介面及系統建置，且實作分群功能，由使用者決定分類群數，將檢索結果分類；以犯罪事實相似度的案由分類，將相似裁判書依案由分類輸出；另外建置詞彙共現的索引資料，在使用者查詢詞彙時，能提供其相關出現的詞彙。

未來將繼續完成系統功能的建置，修正及整合整個系統，讓使用者能更有效率地進行檢索。另外須要設計一個維護系統的介面，對於詞典檔的編輯修改、裁判書資料的增刪及個人標記資料的維護分享等，增加系統的彈性及適用性。

致謝

本研究承蒙編號 NSC-95-2221-E-004-013-MY2 與 NSC-97-2221-E-004-007-MY2 之國科會研究計畫的部分補助，謹此致謝。我們也感謝論文評審的指正，使得我們能夠有機會改進本篇論文與補強研究工作的內容。

參考文獻

[1] 中研院中文斷詞系統，http://ckipsvr.iis.sinica.edu.tw/，最後造訪日期 2008/10/21。
[2] 司法院法學資料檢索系統，http://jirs.judicial.gov.tw/，最後造訪日期 2008/10/21。
[3] 司法院統計處，http://www.judicial.gov.tw/Juds/，最後造訪日期 2008/10/21。
[4] 全國法規資料庫，http://law.moj.gov.tw/，最後造訪日期 2008/10/21。
[5] HowNet 電子詞典，http://www.keenage.com/，最後造訪日期 2008/10/21。
[6] HSQLDB 資料庫，http://hsqldb.sourceforge.net/，最後造訪日期 2008/10/21。
[7] Lucene 全文檢索引擎，http://lucene.apache.org/，最後造訪日期 2008/10/21。
[8] SQLite 資料庫，http://www.sqlite.org/，最後造訪日期 2008/10/21。
[9] 呂明欣和王加元，建構一個簡單的字彙網，自然語言處理學期報告，2007。
[10] 何君豪，*階層式分群法在民事裁判要旨分群上之應用*，碩士論文，國立政治大學資訊科學系，台灣，台北，2007。
[11] 鄭人豪，*中文詞彙集的來源與權重對中文裁判書分類成效的影響*，碩士論文，國立政治大學資訊科學系，台灣，台北，2007。
[12] 謝淳達，*利用詞組檢索中文訴訟文書之研究*，碩士論文，國立政治大學資訊科學系，台灣，台北，2005。
[13] J. Han and M. Kamber, *Data Mining: Concept and Techniques*, Morgan Kaufmann, 2001.
[14] M. A. Hearst, Clustering Versus Faceted Categories for Information Exploration, *Communications of the ACM*, volume 49, issue 4, 59–61, 2006.
[15] K. S. Jones, A Statistical Interpretation of Term Specificity and Its Application in Retrieval, *Journal of Documentation*, volume 28, 11–21, 1972.
[16] H. P. Luhn, A Statistical Approach to Mechanized Encoding and Searching of Literary Information, *IBM Journal of Research and Development*, volume 1, no. 2, 309–317, 1957.
[17] T. O'Reilly, What is Web 2.0–Design Patterns and Business Models for the Next Generation of Software, *Web 2.0 Report*, O'Reilly, 2005.
[18] E. Schweighofer, G. Haneder, A. Rauber and M. Dittenbach, Improvement of Vector Representations of Legal Documents with Legal Ontologies, *Proceedings of the Fifth International Conference on Business Information Systems*, 2002.

國際學術會議出席報告

國立政治大學資訊科學學系劉昭麟
chaolin@nccu.edu.tw

摘要

劉昭麟（以下自稱為報告人）於二零零八年六月中赴美國俄亥俄州哥倫布市(Columbus, Ohio, USA)，參與了計算語言學會(Association for Computational Linguistics，簡稱 ACL)的年會，並且在會議中報告論文。這是這一次出席國際學術會議的報告。本報告首先列出出席會議的時間、地點、所參與的會議的基本資料和相關網址；然後報告參與會議所體驗的觀察和心得；最後提出簡短的結論。

1 出訪地點、時間、參與會議

1.1 基本資料

出訪地點：美國俄亥俄州哥倫布市(Columbus, Ohio, USA)

會議時間：二零零八年六月十五日至六月二十日

參與會議：ACL 2008: The Forty Sixth Annual Meeting of the Association for Computational Linguistics: Human Language Technologies

經費來源：國科會研究經費與政治大學資科系部份補助

發表論文：Using Structural Information for Identifying Similar Chinese Characters（附件五）

相關網址：

ACL: http://www.aclweb.org
ACL 2008: http://www.ling.ohio-state.edu/acl08/

1.2 參與過程

ACL 的年會是歷史悠久的計算語言學學術會議，會議的時間從六月十五日到二十日，其中十五日是主會議前的教學課程(tutorials)，十九日和二十日是主會議之後的工作坊(workshops)。由於距離與時差的問題，報告人必須在台北時間十三日就從台北啟程，於美國當地時間十五日參與了 Building Practical Spoken Dialog Systems 的教學課程，於十六日報告論文，並且於十九日參加了 The Third Workshop on Innovative Use of NLP for Building Educational Applications，最後於美國當地時間二十日離開哥倫布市返國。

參與本次會議的台灣學者明顯偏少，只有遇到前清華大學電機系的蘇克毅教授。我們不能確定這一個低出席率是因為研究經費的限制或者是因為哥倫布市的交通明顯地不是非常方便，須要在美國其他主要都市轉機過來。儘管這些可能的原因，本次會議仍然有許多來自香港、新加坡等亞洲的學者。

2 具體觀察與心得

由於 ACL 在計算語言學界的地位，這一個會議的參與人數非常地多，付費註冊的人數接近 700 人。除了三天的主會議議程之外，有六個會議前的教學課程（參見附件一）和十個會議後的工作坊（參見附件二）。在論文投稿量方面，合計有 470 篇長篇論文的稿件和 275 篇短篇論文的投稿，最後會議接受了 119 篇的長篇論文和 64 篇短篇論文。不管是長篇或者是短篇論文的接受率都僅止於 25%左右。報告人的論文屬於短篇論文。被接受的長篇論文中，數量最多依序是機器翻譯(machine translation)、語意(semantics)、語法(syntax)、問答系統(questions & answering)、統計與機器學習(statistical machine learning)、資訊檢索(information retrieval) 和資訊擷取(information extraction)；這七個領域的論文，合計占了所有長篇論文的 59.66%。

在議程的安排方面，ACL 的設計與其他學術領域的主要會議相似。除了教學課程和工作坊之外，還有為博士班研究生設計的討論議程，請相關領域的專家為現在進行中的博士論文研究提供建言和相互交流的機會。教學課程則是讓主會議的與會者有機會分享一些相對比較成熟的技術，以報告人所參與的 Building Practical Spoken Dialog Systems 來說，就是由 Carnegie Mellon University 的教授與研究生介紹他們所建立的語音辨識系統，並且介紹如何包裝該系統作為應用系統的核心功能。透過這樣的介紹課程，學習者可以獲得起步所需的知識，以比較低的代價瞭解一個相當複雜的系統。工作坊的主要功能則是提供學者有機會討論一些正在發展中的研究議題，以報告人所參與的 The Third Workshop on Innovative Use of NLP for Building Educational Applications 來說，與會者來自許多不同國家，分享他們如何利用計算語言學的相關技術，建構與各國母語和英語相關的語文教學系統。

機器翻譯的相關研究雖然在國內不屬於主流研究重點，不過卻仍然是今年 ACL 主會議的重點項目。機器翻譯的相關論文是所有領域中數量最多的，佔有長篇論文的 23%和短篇論文的 24%，此外還有兩個相關的工作坊(Third Workshop on Statistical Machine Translation 和 Workshop on Parsing German)。Workshop on Parsing German 這一個工作坊相當有趣，未來我們或許可以主辦一些專注於處理亞洲語系語言的工作坊。

如果要看人氣指標的話，資訊檢索和資訊擷取仍然是最容易吸引人的研究議題。比起像機器翻譯、語法研究和語意研究這一些比較基礎的研究，資訊檢索和擷取離應用實務比較接近，因此更容易吸引到人們的注意。

在專題演講(invited talks)方面，我們看到純粹語言學和計算語言學所沒有能夠全心注意的一些語文認知歷程問題。Marc Swerts 強調語言的溝通除了文字和聲音之外，透過視覺管道所發出和接收到訊息，也是人們處理語言的重要依據之一。我們的肢體語言和臉部表情是在語音和用字之外的另一種語言；如果只專注於語音訊號處理或者文字所攜帶的訊息，則常常不能妥善溝通過程互動各方所試圖傳遞的訊息。

六月十八日的專題演講則是一個與資訊檢索相關的演講。不管是以關鍵詞彙，或者是以搜尋範例（例如以文找文）來搜尋資訊的方式，都比較是屬於一次性的搜尋工作。然而，由於人機溝通的效果通常不是完美的，因此以一個程序逐漸地協助查詢者找到真

正想要的資訊，可能是比較務實的目標。Susan Dumais 介紹了許多往這一方向發展的相關的軟體設計理念和實際系統。

今年的 ACL 學術貢獻講(lifetime achievement award)頒給 University of Sheffield 的 Yorick Wilks。Wilks 的演講介紹了他在自然語言處理與人工智慧研究等多面的研究經驗，常常也觸及更深層的科學研究理念，如果聽者本身沒有相當廣博的知識和很好的英文聽力，這樣高階的演講可能是不容易立即吸收。附件三是 Wilks 的演講資料。

關於報告者關於個別論文的聽講心得對於本報告的讀者或許沒有特別的吸引力，ACL 所有的論文都公開在網路上面，請參閱附件四的議程，與網路上的電子版論文(http://aclweb.org/anthology-new/)。其他例如六個教學課程和十個工作坊的資料，請分別參考附件一和附件二的簡介。

除了參與學術會議之外，由於出訪經費的拮据，因此報告人所暫住的旅店距離會議的飯店有相當的距離，每次來回開會與住所之間，單程就須要步行大約二十幾分鐘，也因此有許多天的機會來觀察哥倫布市的日常街景。此次由美國而起的世界金融海嘯對於美國人確實有不小的影響，哥倫布市的大眾運輸系統的使用率看起來相當地高，上下班時間有不少等車的民眾。這可能不是一般美國中小型城市所常見的景象。

3 結論

我國致力於推展學術研究國際化，近年以來資訊科學這一方面的國際學術研討會如雨後春筍般的蓬勃發起，除了國際學術會的頂級會議之外，例如 AAAI、IJCAI、ACL、ICML、UAI、ITS、AIED、COLING、ACM 各 SIG 的年會等等，我國參與其他的新興的學術研討會的必要性似乎可以做一個整體性的規劃。新興的學術研討會雖然學術知名度不高，但是常常是培養新領域的搖籃，學術價值不可謂不高；然而，如果長期投注在這一類新領域的研討會的邊際效用則是可以檢討的。相對地，參與具有傳統聲譽的學術研討會，則有立竿見影的觀摩效果，可以刺激參與者更加努力、以追求在這一類研討會發表更好論文的機會。

參考附件

附件一：ACL 2008 教學課程簡介
附件二：ACL 2008 工作坊簡介
附件三：http://www.companions-project.org/downloads/Wilks_ACL08.pdf
附件四：ACL 2008 論文議程
附件五：報告人所發表之論文

附件一

ACL 2008 教學課程簡介

The 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies
Columbus, OH June 15-20, 2008

Home
Registration
Accommodation
Travel
Area Info
Presentation instructions
Poster instructions
Call for Papers
Schedule At-A-Glance
Full Schedule
Workshops
Tutorials
Student Research Workshop
Related Events
Food
Reception
Banquet
Student Lunch

ACL-08:HLT Tutorials

Morning	Introduction to Computational Advertising	Building Practical Spoken Dialog Systems	Semi-supervised Learning for Natural Language Processing
Afternoon	Advanced Online Learning for Natural Language Processing	Speech technology from research to industry	Interactive Visualization for Computational Linguistics

Tutorial Schedule

9:00-10:30	Morning tutorial part 1
10:30-11:00	morning break
11:00-12:30	Morning tutorial part 2
2:00-3:30	Afternoon tutorial part 1
3:30-4	Afternoon break
4:00-5:30	Afternoon tutorial part 2

Introduction to Computational Advertising

(Evgeniy Gabrilovich, Vanja Josifovski, and Bo Pang)

Short abstract:

Web advertising is the primary driving force behind many Web activities, including Internet search as well as publishing of online content by third-party providers. A new discipline - Computational Advertising - has recently emerged, which studies the process of advertising on the Internet from a variety of angles. A successful advertising campaign should be relevant to the immediate user's information need as well as more generally to user's background and personalized interest profile,

Sponsors
Contact

Hosted by:

The Ohio State University Department of Linguistics

The Ohio State University Department of Computer Science and Engineering

be economically worthwhile to the advertiser and the intermediaries (e.g., the search engine), as well as be aesthetically pleasant and not detrimental to user experience.

The tutorial does not assume any prior knowledge of Web advertising, and will begin with a comprehensive background survey of the topic. In this tutorial, we focus on one important aspect of online advertising, namely, contextual relevance. It is essential to emphasize that in most cases the context of user actions is defined by a body of text, hence the ad matching problem lends itself to many NLP methods. At first approximation, the process of obtaining relevant ads can be reduced to conventional information retrieval, where one constructs a query that describes the user's context, and then executes this query against a large inverted index of ads. We show how to augment the standard information retrieval approach using query expansion and text classification techniques. We demonstrate how to employ a relevance feedback assumption and use Web search results retrieved by the query. This step allows one to use the Web as a repository of relevant query-specific knowledge. We also go beyond the conventional bag of words indexing, and construct additional features using a large external taxonomy and a lexicon of named entities obtained by analyzing the entire Web as a corpus. Computational advertising poses numerous challenges and open research problems in text summarization, natural language generation, named entity extraction, computer-human interaction, and others. The last part of the tutorial will be devoted to recent research results as well as open problems, such as automatically classifying cases when no ads should be shown, handling geographic names, context modeling for vertical portals, and using natural language generation to automatically create advertising campaigns.

Tutorial outline

- Introduction
- Advertising on the Web
  - The evolution of Web advertising
  - Advertese (introduction of terminology)
  - Main scenarios of online advertising
    - Sponsored search
    - Content match
    - Exact match vs. broad match
    - The economics of Web advertising
- Main technical challenges for NLP and IR

- Bibliography survey
- Break
- IR modeling
  - Ads as information supply and reduction to search
  - A unified approach to Web advertising
  - Using search results as external knowledge
  - Text classification
  - Named entities
- The research frontier
  - Text summarization / just-in-time advertising
  - When not to advertise / ad spam
  - Location awareness / geo-targeting
  - Context modeling
  - Natural language generation / automatic ad creation
- Discussion and questions from the audience

Short biographical description of presenter(s)

Evgeniy Gabrilovich gabr@yahoo-inc.com
Vanja Josifovski vanjaj@yahoo-inc.com
Bo Pang bopang@yahoo-inc.com

Affiliation: Yahoo! Research, Computational Advertising and Search Technology Group

Contact information:
2821 Mission College Blvd, Santa Clara, CA 95054
Fax: 408-349-2270

Evgeniy Gabrilovich is a Senior Research Scientist at Yahoo! Research. His research interests include information retrieval, machine learning, and computational linguistics. He serves on the program committees of ACL-08:HLT, AAAI '08, JCDL '08, CIKM '08 and WWW '08, and in the past he served on the program committees of AAAI, EMNLP-CoNLL, COLING-ACL, served as a mentor at SIGIR '07, as well as reviewed papers for ACM TOIT, IP&M, JNLE, CACM, AAAI, AAMAS, WWW and CIKM. Evgeniy earned his MSc and PhD degrees in Computer Science from the Technion - Israel Institute of Technology.

Vanja Josifovski is a Principal Research Scientist at Yahoo! Research, where he

works on search and advertisement technologies for the Internet. He is currently exploring designs for the next generation ad placement platforms for contextual and search advertising. Previously, Vanja was a Research Staff Member at the IBM Almaden Research Center working on several projects in database runtime and optimization, federated databases, and enterprise. He earned his MSc degree from the University of Florida at Gainesville, and his PhD from the Linkoping University in Sweden. Vanja published over thirty peer reviewed publications, authored around 20 patent applications, and was on the program committees of WWW, SIGIR, ICDE, VLDB and other major conferences in the database, information retrieval, and search areas.

Bo Pang is a Research Scientist at Yahoo! Research. Her primary research interests are in natural language processing, machine learning, and information retrieval. She obtained her PhD in Computer Science from Cornell University, where she worked on automatic analysis of sentiment in text and paraphrase extraction and generation in the context of machine translation. She has served on the program committees of ACL, HLT-NAACL, EMNLP, and AAAI, and reviewed for journals including ACM TOIS, JMLR, JAIR, Computer Speech and Language, and Computational Linguistics.

back to top

Building Practical Spoken Dialog Systems

(Antoine Raux, Brian Langner, Maxine Eskenazi, Alan Black)

Abstract:

This tutorial will give a practical description of the free software Carnegie Mellon Olympus 2 Spoken Dialog Architecture. Building real working dialog systems that are robust enough for the general public to use is difficult. Most frequently, the functionality of the conversations is severely limited - down to simple question-answer pairs. While off-the-shelf toolkits help the development of such simple systems, they do not support more advanced, natural dialogs nor do they offer the transparency and flexibility required by computational linguistic researchers. However, Olympus 2 offers a complete dialog system with automatic speech recognition (Sphinx) and synthesis (SAPI, Festival) and has been used, along with previous versions of Olympus, for teaching and research at Carnegie Mellon and elsewhere for some 5 years. Overall, a dozen dialog systems have been built using various versions of Olympus, handling tasks ranging from providing bus schedule information to guidance through maintenance procedures for complex machinery, to personal calendar management. In addition to simplifying the development of dialog systems, Olympus provides a transparent platform for teaching and conducting research on all aspects of dialog systems, including speech recognition and synthesis, natural language understanding and generation, and dialog and interaction management.

The tutorial will give a brief introduction to spoken dialog systems before going into detail about how to create your own dialog system within Olympus 2, using the Let's Go bus information system as an example. Further, we will provide guidelines on how to use an actual deployed spoken dialog system such as Let's Go to validate research results in the real world. As a possible testbed for such research, we will describe Let's Go Lab, which provides access to both the Let's Go system and its genuine user population for research experiments.

Attendees will receive a CD with the latest version of the Olympus 2 architecture, along with several tutorials and example systems.

Tutorial Outline:

- Introduction
- Overview of current spoken dialog system architectures
- Description of the Olympus2 dialog architechture
- How to build an Olympus2 dialog system (text I/O)
- Short break
- Expanding an Olympus2 system to use speech - a true spoken dialog system
- Discussion of installation requirements and practical system-building issues, including:
  - telephony
  - system backend
  - ASR (re)training / (re)tuning
  - improving synthesis output
  - dialog strategies & parameters
  - monitoring / logging
- Using Olympus2 for research and applications
  - Let's Go Lab: a test platform for dialog systems with real users -

- Final summary

Presenter Bios:

Antoine Raux
Language Technologies Institute
Carnegie Mellon University
website: http://www.cs.cmu.edu/~antoine/
email: antoine@cs.cmu.edu

Antoine Raux is a PhD student at the Language Technologies Institute at Carnegie Mellon University. He has been conducting research and published more than 15 reviewed papers on several aspects of dialog systems, including speech recognition, speech synthesis, dialog and interaction management, and system building. His teaching experience includes two teaching assistantships in natural language-related graduate courses, as well as the ongoing design of online tutorials for the Olympus architecture.

Brian Langner
Language Technologies Institute
Carnegie Mellon University
website: http://www.cs.cmu.edu/~blangner/
email: blanger@cs.cmu.edu

Brian Langner is a PhD student at the Language Technologies Institute at Carnegie Mellon University. He has been conducting research and published more than 12 reviewed papers on speech synthesis, natural language generation, and spoken dialog systems. He has six semesters of experience as a teaching assistant for graduate and undergraduate computing- or natural language- related courses, including some course design, in addition to continuing work for the Olympus architecture tutorials.

Dr. Alan W Black
Language Technologies Institute
Carnegie Mellon University
website: http://www.cs.cmu.edu/~awb/
email: awb@cs.cmu.edu

Alan W Black is an Associate Research Professor in the Language Technologies

Institute at Carnegie Mellon University. He previously worked in the University of Edinburgh, and before that at ATR in Japan. He received his PhD in Computational Linguistics from Edinburgh University in 1993. He is one of the principal authors of the Festival Speech Synthesis System. In addition to speech synthesis, he also works on two-way speech-to-speech translation systems and, telephone-based spoken dialog systems. He also has served on the IEEE Speech Technical Committee (2003-2006), is on the editorial board of Speech Communications and is a board member of ISCA. He teaches a number of graduate and undergraduate courses and has taught a number of short term tutorials on speech synthesis, speech technology and on rapid support for new languages.

Dr. Maxine Eskenazi
Language Technologies Institute
Carnegie Mellon University
website: http://www.cs.cmu.edu/~max/
email: max@cs.cmu.edu

Maxine Eskenazi is on the faculty of the Language Technologies Institute at Carnegie Mellon University. She has a BA from Carnegie Mellon University in French and Education and a These de Troisieme Cycle from the Universite de Paris 11 in Computer Science. She has extensive publications on the use of automatic speech processing for spoken dialog systems and on the use of language technologies for computer-assisted language learning. She is the Principal Investigator on the NSF Let's Go project.

back to top

Semi-supervised Learning for Natural Language Processing

(John Blitzer and Xiaojin (Jerry) Zhu)

Website for SSL-NLP tutorial

Statistical natural language processing tools are being applied to an ever wider and more varied range of linguistic data. Researchers and engineers are using statistical models to organize and understand financial news, legal documents, biomedical abstracts, and weblog entries, among many other types of data. Creating high-coverage, accurate labeled datasets for so many different types of data can be prohibitively expensive, but for many tasks we have large amounts of unlabeled data that we can exploit.

This tutorial covers semi-supervised learning for natural language processing. Semi-supervised learning methods use a large amount of unlabeled data and a small amount of labeled data to estimate a statistical model [1]. Our emphasis is on practical application, and we will treat semi-supervised learning methods as tools for building effective models from limited training data. An attendee will leave our tutorial with basic knowledge of the general classes of semi-supervised learning, as well as the ability to decide which class will be useful in her research and where to find detailed information on several methods within each class. We will cover three main classes: self-training, graph-based methods, and structural learning. From each general class we will choose one specific model (or two, in the case of self-training) to cover in detail, with a demo and a detailed discussion of known success and failure cases. There will also be a high-level description of several other methods within each class.

References: [1] Xiaojin (Jerry) Zhu. Semi-supervised Learning Literature Survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison 2005.

Outline:

1. Introduction and overview:
2. Self-training:
   - Overview:
   - Co-training:
   - Prototype-driven learning:
3. Graph methods:
4. Structural learning:
5. Wrapup and pointers to external references:

Biographical information:

John Blitzer
3330 Walnut Street
Philadelphia, PA 19104
email: blitzer@cis.upenn.edu

John Blitzer is currently a PhD student in computer science under Fernando Pereira at the University of Pennsylvania. Beginning in February 2008, he will be a visiting researcher at Microsoft Research Labs Asia, and in August 2008, he will start a position as a postdoctoral fellow under Dan Klein at the University of California, Berkeley. John's research area is machine learning for natural language processing, with a primary focus on unsupervised dimensionality reduction of text. Recently, he has worked on empirical and theoretical analyses of structural learning for semi-supervised domain adaptation. He has been a teaching assistant for courses in cognitive science and numerical linear algebra at the University of Pennsylvania.

Xiaojin Zhu
University of Wisconsin, Madison
1210 West Dayton Street
Madison, WI 53706-1685
email: jerryzhu@cs.wisc.edu

Xiaojin Zhu is an Assistant Professor in Computer Sciences at University of Wisconsin, Madison. His research interests are statistical machine learning (in particular semi-supervised learning), and its applications to natural language analysis. He received a Ph.D. in Language Technologies from CMU in 2005, with thesis research on graph-based semi-supervised learning. His current research projects aim at bridging the different approaches in semi-supervised learning, and making them more effective for practitioners. He has taught several graduate and undergraduate courses in AI, machine learning and NLP at the University of Wisconsin, Madison.

back to top

Advanced Online Learning for Natural Language Processing

(Koby Crammer)

Most research in machine learning has been focused on binary classification, in which the learned classifier outputs one of two possible answers. Important fundamental questions can be analyzed in terms of binary classification,but real-world natural language processing problems often involve richer output spaces. In this tutorial, we will focus on classifiers with a large number of possible outputs with interesting structure. Notable examples include information retrieval, part-of-speech tagging, NP chucking, parsing, entity extraction, and phoneme recognition.

Our algorithmic framework will be that of online learning, for several reasons. First, online algorithms are in general conceptually simple and easy to implement. In particular, online algorithms process one example at a time and thus require little working memory. Second, our example applications have all been treated successfully using online algorithms. Third, the analysis of online algorithms uses simpler mathematical tools than other types of algorithms. Fourth, the online learning framework provides a very general setting which can be applied to a broad setting of problems, where the only machinery assumed is the ability to perform exact inference, which computes a maxima over some score function.

The goals of the tutorial:

1. To provide the audience systematic methods to design, analyze and implement efficiently learning algorithms for their specific complex-output problems: from simple binary classification through multi- class categorization to information extraction, parsing and speech recognition.
2. To introduce new online algorithms which provide state-of-the-art performance in practice backed by interesting theoretical guarantees.

Theory and Algorithms

- The online learning paradigm
- Major concepts : loss function, large margin
- The perceptron algorithm and variants
- The passive-aggressive approach
- The general-margin extension to the passive-aggressive approach for complex problems

Implementation and Practice

- Applications

- Multi-class multi-label text classification
- Speech processing
- Information extraction
- Parsing
- Alternative algorithms : Logistic regression, CRFs
- Improvements : Top-k inference, Averaging
- Conclusion an Questions from the audience

Koby Crammer is a research associate at the University of Pennsylvania (PhD Hebrew University). His research focuses on the design, analysis and implementation of machine learning algorithms for complex prediction problems, and applying them for various natural language processing tasks and other structured problems.

back to top

Interactive Visualization for Computational Linguistics

(Christopher Collins, Gerald Penn, and Sheelagh Carpendale)

Interactive information visualization is an emerging and powerful research technique for understanding models of language, and their abstract representations. Much of what computational linguists fall back upon to improve natural language processing and to model language "understanding" is structure that has, at best, only an indirect attestation in observable data. An important part of research progress depends on our ability to fully investigate, explain, and explore these structures, both empirically and as outcomes of grammar design relative to accepted linguistic theory. The sheer complexity of these abstract structures, and the observable patterns on which they are based, however, usually limits their accessibility, often even to the researchers creating or attempting to learn them.

To aid in understanding, visual 'externalizations' are used in CL for presentation and explanation - traditional statistical graphs and custom-designed data illustrations fill the pages of ACL papers. Such visualizations do provide insight into the representations and algorithms designed by researchers, but visualization can also be used as an aid in the process of research itself. There are special

statistical methods, falling under the rubric of "exploratory data analysis", and visualization techniques just for this purpose, in fact, but these are not widely used or even known in CL. These novel data visualization techniques, which we have used successfully in the CL domain, offer the potential for creating new methods that reveal structure and detail in data. Instructed by a team of computational linguists and information visualization researchers, this tutorial will bridge computational linguistic and information visualization expertise, providing attendees with a basis from which they can begin to accelerate their own research.

Tutorial Objectives:

This tutorial will equip participants with:

- An understanding of the importance and applicability of information visualization techniques to computational linguistics research;
- Knowledge of the basic principles of information visualization theory;
- The ability to identify appropriate visualization software and techniques that are available for immediate use and for prototyping;
- A working knowledge of research to date in the area of linguistic information visualization.

Tutorial Outline:

1. Introduction
2. Information Visualization Theory
   - Representational theory, cognitive psychology, preattentitive processing
   - Interaction & animation
   - Assessing and validating visualization
     i. Evaluation challenges
     ii. Measuring insight
     iii. Metrics for evaluation
     iv. Heuristic approaches to evaluation
3. Review of Linguistic Visualization
   - Document content visualizations
   - Text collection analysis
   - Literary analysis
   - Streaming data visualization
   - Convergence of language and other data

  - Corpora exploration
  - Visualization of statistical NLP outputs
  - Linguistic analysis
  - Visualization of non-textual linguistic data

4. Tools for Visualization
   - Software solutions
   - Programming toolkits
   - Online tools
   - Collaborative visualization tools in development
5. Case Studies in Linguistic Visualization
6. Open Research Problems
7. Closing

Tutorial Instructors

Christopher Collins
PhD Candidate, University of Toronto Computer Science

Christopher Collins received his M.Sc. in the area of Computational Linguistics from University of Toronto in 2004. His PhD research focus is inter-disciplinary, combining computational linguistics and information visualization. He is currently in his final year of PhD studies, investigating interactive visualizations of linguistic data with a focus on convergence and coordination of multiple views of data to provide enhanced insight. He has developed various methods for generating, reading, and comparing visual summaries of document thematic content for everyday users and data analysts. Recent publications include a new method for revealing relationships amongst visualizations, and a system for exposing the uncertainty in statistical natural language systems. He recently embarked on a study of visualization use in a team of machine translation researchers and plans to continue collaboration with language engineers to provide them with an enhanced ability to analyse and improve their algorithms.

Gerald Penn
Associate Professor, University of Toronto Computer Science

Gerald Penn's research interests are in computational linguistics, theoretical computer science, programming languages, spoken language processing, and human-computer interaction. He is probably best known as the co-designer and maintainer of the ALE programming language, and has published widely on topics

pertaining to logics and discrete algorithms for natural language processing applications. He is a member of the advisory board to Computational Linguistics and the editorial board of Linguistics & Philosophy, and is a past president of the ACL Mathematics of Language Society.

Sheelagh Carpendale
Associate Professor, University of Calgary Computer Science
Canada Research Chair: Information Visualization

Sheelagh Carpendale holds a Canada Research Chair in Information Visualization and an NSERC/SMART/iCORE Industrial Research Chair in Interactive Technologies at the University of Calgary. She is the recipient of several major awards including the British Academy of Film and Television Arts Award (BAFTA) for Off-line Learning, and has been involved with successful technology transfer to Idelix Software Inc. Her research focuses on the visualization, exploration and manipulation of information. Current research includes: visualizing uncertainty particularly in medical data, visualizing biological data, developing visualizations to support computational linguistic research and the development of methodologies to support collaborative data analysis with visualization. Sheelagh Carpendale's research in information visualization and interaction design draws on her dual background in Computer Science (Ph.D. Simon Fraser University) and Visual Arts (Sheridan College, School of Design and Emily Carr, College of Art).

back to top

Speech Technology from Research to Industry

Roberto Pieraccini
CTO, SpeechCycle, Inc.,
26 Broadway, 11th floor
New York, NY 10004
Tel.: (646) 792 2744
roberto@speechcycle.com

This tutorial is about the evolution of speech technology from research to a mature industry. Today, spoken language communication with computers is becoming part of everyday life. Thousands of interactive applications using spoken language technology - known also as "conversational machines" - are only a phone call away, allowing millions of users each day to access information, perform

transactions, and get help. Speech recognition, language understanding, text-to-speech synthesis, machine learning, and dialog management enabled this revolution after more than 50 years of research. The industry of speech continues to mature with its evolving standards, platforms, architectures, and business models within different sectors of the market. In this tutorial I will briefly trace the history of speech technology, with a special focus on speech recognition and spoken language understanding, from the early attempts to today's commercial deployments. I will summarily describe the most successful ideas and algorithms that brought to today's technology. I will discuss the struggle for ever increasing performance, the importance of data for training and evaluation, and the role played by government funded projects in creating effective evaluation benchmarks. I will then describe the birth of the speech industry in the mid 1990s, with the role played by the Voice User Interface and dialog engineering disciplines in bringing speech recognition from a laboratory "accuracy challenge" to an enabler of usable interfaces. I will describe the rising of standards (such as VoiceXML, SRGS, SSML, etc.) and their importance in the growth of the market. I will proceed with an overview of the current architectures and processes utilized for creating commercial spoken dialog systems, and will provide several case studies of the use of speech technology. I will conclude with a discussion on the current open problems and challenges. The tutorial duration will be of about 3 hours with a short break. Several audio and video samples will be shown during the tutorial. The tutorial is directed to a general HLT audience with no prior knowledge of speech technology.

Tutorial Outline

- What is speech and why it is difficult to recognize it.
- The history of speech recognition from the early attempts to Hidden Markov Models
- The struggle for performance and the importance of data
- Short break
- Spoken language understanding and dialog
- The birth of the "spoken dialog" industry
- Industrial standards and architectures
- Case studies
- Open issues and future research

Roberto Pieraccini spent more than 25 years in the area of speech and language technologies. He worked at research labs such as CSELT in Torino, Italy (the

research center of the Italian telephone company in the 1980s), Bell Laboratories, AT&T Shannon Labs, and IBM T.J. Watson Research. He was director of R&D at SpeechWorks, one of the companies that had a major impact in the definition of the current speech technology market in the late 1990s and early 2000s. He is now the Chief Technology Officer of SpeechCycle, a company specialized in complex spoken language interaction systems for technical support and customer care. He is the author of more than 100 publications and book chapters. He is senior member of IEEE, the current chair of the IEEE Speech and Language Technical Committee, and a member of the IEEE Signal Processing Society's Conference Board.

back to top

Tutorial Speaker Responsibilities

Accepted tutorial speakers will be notified by February 4, 2008, and must then provide abstracts of their tutorials for inclusion in the conference registration material by February 21, 2008. The description should be in two formats: an ASCII version that can be included in email announcements and published on the conference web site, and a PDF version for inclusion in the electronic proceedings (detailed instructions to follow). Tutorial speakers must provide tutorial materials, at least containing copies of the course slides as well as a bibliography for the material covered in the tutorial, by May 5, 2008.

Chairs:

Ani Nenkova (Univ of Pennsylvania, USA) nenkova@seas.upenn.edu
Marilyn Walker (Univ of Sheffield, UK) m.a.walker@sheffield.ac.uk
Eugene Agichtein (Emory University, USA) eugene@mathcs.emory.edu

Please send inquiries concerning ACL-08 tutorials to tutorials-acl08@lists.seas.upenn.edu.

附件二

ACL 2008 工作坊簡介

The 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies
Columbus, OH June 15-20, 2008

Home
Registration
Accommodation
Travel
Area Info
Presentation instructions
Poster instructions
Call for Papers
Schedule At-A-Glance
Full Schedule
Workshops
Tutorials
Student Research Workshop
Related Events
Food
Reception
Banquet
Student Lunch

ACL-08: HLT Workshops

The following workshops will be held in conjunction with the 46th Annual Meeting of the Association for Computational Linguistics (ACL-08: HLT), June 15-20, 2008, in Columbus, Ohio, United States. The workshops will be held on June 19 and June 20, 2008 at the main conference venue.

1. SIGDIAL, ACL08-SIGDIAL
2. The Third Workshop on Issues in Teaching Computational Linguistics, ACL08-Teaching-CL
3. Third Workshop on Statistical Machine Translation, ACL08-SMT
4. SSST-2: Second Workshop on Syntax and Structure in Statistical Translation, ACL08-SSST
5. Software engineering, testing, and quality assurance for natural language processing, ACL08-NLP-Software
6. BioNLP 2008, ACL08-BioNLP
7. Computational Morphology and Phonology(SIGMORPHON)
8. ACL2008 Workshop on Mobile Language Processing
9. The 4th Workshop on Innovative Use of NLP for Building Educational Applications, ACL08-NLP-Education
10. Workshop on Parsing German

Changes in plan

1. Semantic Evaluations: Recent Achievements and Future Directions,ACL08-Semantic-Evaluation CANCELLED

ACL08 Workshop Dates

19th	20th
ACL08-SigDial	SigDial-Continued
ACL08-SMT	ACL08-SSST
ACL08-BioNLP	ACL08-NLP-Software
ACL08-SIGMORPHON	ACL08-Mobile-NLP
ACL08-Teaching-CL	Teaching-CL contd.
ACL08-NLP-Education	ACL08-Parsing-German

Sponsors
Contact

Hosted by:

The Ohio State University Department of Linguistics

THE OHIO STATE UNIVERSITY

The Ohio State University Department of Computer Science and Engineering

WORKSHOPS PROGRAM COMMITTEE

Ming Zhou (Chair), Microsoft Research Asia
Chengxiang Zhai (Co-Chair), University of Illinois at Urbana-Champaign
Helen Meng (Co-Chair), Chinese University of Hong Kong

附件三

Yorick Wilks 演講資料

ACL Lifetime Achievement Award

On Whose Shoulders?

Yorick Wilks*
University of Sheffield

Introduction

The title of this piece refers to Newton's only known modest remark: "If I have seen farther than other men, it was because I was standing on the shoulders of giants." Since he himself was so much greater than his predecessors, he was in fact standing on the shoulders of dwarfs, a much less attractive metaphor. I intend no comparisons with Newton in what follows: NLP/CL has no Newtons and no Nobel Prizes so far, and quite rightly. I intend only to draw attention to a tendency in our field to ignore its intellectual inheritance and debt; I intend to discharge a little of this debt in this article, partly as an encouragement to others to improve our lack of scholarship and knowledge of our own roots, often driven by the desire for novelty and to name our own systems. Roger Schank used to argue that it was crucial to name your own NLP system and then have lots of students to colonize all major CS departments, although time has not been kind to his many achievements and originalities, even though he did build just such an Empire. But to me one of the most striking losses from our corporate memory is the man who is to me the greatest of the first generation and still with us: Vic Yngve. This is the man who gave us COMIT, the first NLP programming language; the first random generation of sentences; and the first direct link from syntactic structure to parsing processes and storage (the depth hypothesis). I find students now rarely recognize his name, and find that incredible.

This phenomenon is more than corporate bad memory, or being too busy with engineering to do the scholarship. It is something endemic in the wider field of Computer Science and Artificial Intelligence, although bottom–up wiki techniques are now filling many historical gaps for those who know where to look, as the generation of pioneers has time to reminisce in retirement.[1] There are costs to us from this general lack of awareness, though: a difficulty of "standing on the shoulders" of others and acknowledging debts, let alone passing on software packages. Alan Bundy used to highlight this in the *AISB Quarterly* with a regular column where he located and pilloried reinventions in the field of AI; he also recommended giving obituaries for one's own work, and this paper could be seen in that way, too.

[*] Department of Computer Science, The University of Sheffield, Regent Court, 211 Portobello Street, Sheffield, S1 4DP, UK. E-mail: Y.Wilks@dcs.shef.ac.uk. This article is the text of the talk given on receipt of the ACL's Lifetime Achievement Award in 2008.

[1] See the video interview with Victor Yngve on my Web site at `http://www.dcs.shef.ac.uk/~yorick/YngveInterview.html`.

Early Academic Life

My overwhelming emotion on getting this honor was, after surprise, a feeling of inadequacy in measuring up to previous honorees, but nonetheless, I want to grasp at this moment of autobiography, or at what in his own acceptance paper Martin Kay called: "but one chance for such gross indulgence." I was born in 1939 in London at just about the moment the Second World War started in Europe; this was, briefly, a severe career slowdown. However, the British Government had a policy of exporting most children out of the range of bombs and I was sent to Torquay, a seaside town in southwest England that happened to have palm trees on all the main streets, a fact it is often difficult to convince outsiders of. The town had, and has, a Grammar School for Boys, which had a very good Cambridge-trained mathematician as its headmaster, and eventually I made my way back across England to Pembroke College, Cambridge, to study mathematics, a college now for ever associated with my comedian contemporaries: Peter Cook, Clive James, Eric Idle, Tim Brooke-Taylor, and similar wastrels. I began a series of changes of subject of study, downhill towards easier and easier ones: from mathematics to philosophy to (what in the end after graduation became) NLP/AI. It was not that I could not do the mathematics, but rather that I experienced the shock that many do of finding how wide the range of talent in mathematics is, and that being very good in a provincial grammar school does not make one very good at Cambridge. This is a feeling peculiar to mathematics, I think, because the talent range is so much wider than in most subjects, even at the top level.

Margaret Masterman, who was to become the main intellectual influence in my life, was the philosophy tutor for my college, although her main vocation was running the Institute she had founded, outside the University in a Cambridge suburb: CLRU, the Cambridge Language Research Unit. It was an eccentric and informal outfit, housed in what had been a museum of Buddhist art, some of whose sculptures were built into the walls. MMB (as she was known) ran the CLRU from the mid 1950s to the early 1980s on a mix of US, UK, and EU grants and did pioneering work in MT, AI, and IR. Of those honored by the ACL with this award over the last five years, three have been graduates of that little Buddhist shed, and include Martin Kay and Karen Spärck Jones, a remarkable tribute to MMB. The lives and work of we three have been quite different but all in different ways stem from MMB's interests and vision: She had been a pupil of Wittgenstein and, had she known it, would have approved of Longuet-Higgins's remark that "AI is the pursuit of metaphysics by other means." She believed that practical research into the structure of language could give insight into metaphysics, but was in no way other-worldly: She was the daughter of a Cabinet Minister and knew what it was to command.

In a final twist, I found after her death in 1986 that she had made me her literary executor: She had never written a book and wanted me to construct one from her papers posthumously. It took me twenty years to get the required permissions but the volume finally appeared in 2005 (Masterman et al. 2005).

Thesis Building and CLRU

When I started work at CLRU in 1962 to do a doctorate, it had no computer in the normal sense, only a Hollerith card sorter of the sort built for the US census half a century before. Basically, you put a stack of punched cards into one of these things—which looked like a metal horse on four legs—and the cards fell into (I think) 10 slots

depending on how you had plugged in a set of wires at the back to identify destination slots for sorted cards with hole patterns on the cards. With some effort, these could be turned into quite interesting Boolean machines; my first task was to take a notion of Fred Parker-Rhodes that a Hallidayan grammar could be expressed as a lattice of typed classes, and then program the card sorter so that repeated sorts of punched cards could be used to parse a sentence. It was triumph of ingenuity over practicality. Later the CLRU owned an ICL 1202 computer with 1,200 registers on a drum, but it was a so-called bini-ten machine designed for UK cash transactions when there were still 12 pennies to a shilling, and so the 1,202 has print wheel characters for 10, 11, and 12 (as well as 0–9), a fact on which Parker-Rhodes built a whole world of novel print conventions for his research. This was the period at CLRU when Karen Spärck Jones was completing her highly original thesis (published twenty years later as Jones [1986] on unsupervised clustering of thesaurus terms—whose goal was to produce primitives for MT, it is often forgotten—until she had to move her computations to a real computer at the University Computing Laboratory, where she eventually created a new career in IR, essentially using the same clump algorithms—created by Parker-Rhodes and her husband Roger Needham—to do IR.

My own interests shifted to notions in an early Masterman paper titled "Semantic message detection using an interlingua" (Masterman 1961), an area in which Martin Kay had also originally worked on an interlingua for MT. My thesis computation was done in LISP 1.6 on an IBM360 (under a one-man US Air Force contract, administered by E. Mark Gold, who later became famous as the founder of learnability theory), at SDC in Santa Monica, where I was attached loosely in 1966 to the NLP group there run by Bob Simmons. My thesis was to be entitled "Argument and proof in Metaphysics from an empirical point of view" and my advisor was MMB's husband, Richard Braithwaite, Knightbridge Professor of Moral Philosophy at the University. He was a philosopher of science and a logician, and was given the chair of moral philosophy—a subject about which he knew nothing— because it was the only one available at Cambridge at the time. This produced an extraordinary inaugural lecture in which he effectively founded a new subject: "The theory of games as a tool for the moral philosopher."

Unfortunately for me he was not interested in my thesis, and took me on only as a favor to MMB. My interest was the demarcation of metaphysical text: what it was, if anything, that distinguished it from ordinary language text. Wittgenstein had once said that words were "on holiday" in metaphysical text, but also that he wanted to "bring words back from their metaphysical to their everyday usage" (Wittgenstein 1973). This is exactly what I wanted to capture with computation, and the thesis was eventually submitted to the Cambridge Philosophical faculty in 1967—then called Moral Sciences—with a large appendix of LISP program code at the back, something they had never seen before, or since. The thesis was bound in yellow, though the regulations stipulated black or brown bindings; I must have had some extraordinary idea that someone might cruise the long corridors of Cambridge theses looking for one that stood out by color—the arrogance of youth!

The thesis's starting point was Carnap's monumental *Logische Syntax der Sprache* (1937) and his claim that meaningfulness in text could be determined by "logical syntax"—rules of formation and transformation (a notion which may well sound familiar; Chomsky was a student of Carnap). My claim was that this was a bad demarcation and a better criterion of meaningfulness would be to have one interpretation rather than many, namely, that word-sense discrimination (WSD) was possible for a given text. On that view, the "meaningless" text had too many interpretations rather than none (or

one). A word in isolation is thus often meaningless. Preference Semantics was a WSD program to do just that, and to provide a new sense where WSD failed.

The other starting point of the thesis was a slim paper by Bosanquet on the nature of metaphysical discourse, entitled "Some Remarks on Spinoza's Ethics." He argued that Spinoza's logical arguments are all false, but that what Spinoza was actually doing is *rhetorical*, *not logical*: imposing a new sense on the reader. The system as implemented was, of course, a toy system, in the sense that all symbolic NLP systems were in that era. It consisted of an analysis of five metaphysical texts (by Wittgenstein, Spinoza, Descartes, Kant, and Leibniz) along with five randomly chosen passages from editorials in the London *Times*, as some sort of control texts.

The vocabulary was only about 500 words, but this was many years before Boguraev declared the average size of vocabularies in working NLP systems to be 36 words. The semantic structures derived—via what we would now call chunk parsing—consisted of tree structures of primitives (from a set of about 80), one tree for each participating word sense in the text chunk, that fitted into preformed triples called **templates**. These templates were subject–predicate–object triples that defined well-formed sequences of the triples of trees (i.e., the first tree for the sense of the subject, the second for the action and so on), whose tree-heads had to fit those of the template's three primitive items in order. The overall system selected the word senses that fitted into these structures by means of a notion of "semantic preference" (see subsequent discussion), and then declared those to be the appropriate senses for the words, thus doing a primitive kind of WSD.

There was in the thesis an additional "sense constructor" mode, called if the WSD did not work, which tried to identify some sense of a word in the text whose representation would fit in the overall structure derived, and so could be declared a suitable "new" sense for the word which had previously failed to fit in. Unsurprisingly, it identified, say, a sense of "God" in the Spinoza text with an existing sense of "Nature" so that, after this substitution, the whole thing fitted together and WSD could proceed, and thus the passage be declared meaningful, given the criterion of having a single, ambiguity-free, interpretation. This was the toy procedure that allowed me to argue that Spinoza's real aim, whether he knew it or not, was to persuade us that the word "God" could have the sense of "Nature" and that this was the real point of his philosophy— exactly in line with what Bosanquet had predicted.

The philosophy work was never really published, outside an obscure McGill University philosophy journal, although the meaningfulness criterion appeared in *Mind* in 1971 under the title "Decidability and Natural Language" (Wilks 1971). Since publishing in *Mind* was, at the time, the ambition of every young philosopher, I was now satisfied and could move to the simpler world of NLP. The thesis, shorn of the metaphysics, appeared as my first book, *Grammar, Meaning and the Machine Analysis of Language* (Wilks 1972); the title was intended as a variation on the title of some strange German play, popular at the time, and whose actual name I can no longer remember.

Preference Semantics

I returned from California to CLRU but left again for the Stanford AI Lab in 1969. I had fantasized at CLRU about all the things one could do with a methodology of trying to base a fairly complex compositional semantics on a foundation of superficial pattern matching. This had earlier produced speculations like my 1964 CLRU paper "Text searching with templates," procedures that we could not possibly have carried

```
I.1 ((*ANI 1)((SELF IN)(MOVE CAUSE))(*REAL 2))→(1(*JUDG) 2)
      Or, in semi-English:
      [animate-1 cause-to-move-in-self real-object-2]→[1 *judges 2]
I.2 (1 BE (GOOD KIND))↔((*ANI 2) WANT 1)
      Or, again:
      [1 is good]↔[animate-2 wants 1]
```

Figure 1
Inference rules in Preference Semantics.

out with the machines then available, but which I now choose to see as wanting to do Information Extraction: though, of course, it was Naomi Sager who did IE first on medical texts at NYU (see Sager and Grishman 1975).

At Stanford as a post-doc, I was on the same corridor as Winograd, just arrived from MIT; Schank, then starting to build his Conceptual Dependency empire; and Colby and his large team building the PARRY dialogue system, which included Larry Tesler, later the Apple software architect. Schank and I agreed on far more than we disagreed on and saw that we would be stronger together than separately, but neither of us wanted to give up our notation: He realized, rightly, that there was more persuasive power in diagrams than in talk of processes like "preference." It was an extraordinary period, when AI and NLP were probably closer than ever before or since: Around 1972 Colmerauer passed though the Stanford AI Lab, describing Prolog for the first time but, as you may or may not remember, as a tool for machine translation! I spent my time there defining and expanding the coherence-based semantics underlying my thesis, calling it "Preference Semantics" (PS), adding larger scale structures such as inference rules (see Figure 1) and thesauri, and building it into the core of a small semantics-based English-to-French machine translation system programmed in LISP. At one point the code of this MT system ended up in the Boston Computer Museum, but I have no idea where it is now. The principles behind PS were as follows:

- an emphasis on processes, not diagrams;
- the notion of affinity and repulsion between sense representations (cf. Waltz and Pollack's WSD connectionism [1985]);
- seeking the "best fit" interpretation—the one with most satisfied preferences (normally of verbs, prepositions and adjectives);
- yielding the least informative/effort interpretation;
- using no explicit syntax, only segmentation and order of items;
- meaningfulness as being connected to a unique interpretation/sense choice;
- meaning seen as represented in other words, since no other equivalent for the notion works (e.g., objects or concepts);
- gists or templates of utterances as core underlying entities; and
- there is no *correct* interpretation or set of primitive concepts, only the best available.

One could put some of these, admittedly programmatic and imprecise, points as follows:

- Semantics is not necessarily deep but also superficial (see more recent results on the interrelations between WSD, POS, and IE, e.g. Stevenson and Wilks [2001]).
- Quantitative phenomena are unavoidable in language: John McCarthy thought they had no place anywhere in AI, except perhaps in low-level computer vision.
- Reference structures (like lexicons) are only temporary snapshots of a language in a particular state (of expansion or contraction).
- What is important is to locate the update mechanism of language, including crucially the creation of new word senses, which is not Chomsky's sense of the creativity of language.

Constructible Belief Systems

I returned to Europe in the mid 1970s, first to the ISSCO institute in Lugano, where Charniak was and Schank had just left, and then to Edinburgh as a visitor before taking a job at Essex. I began a long period of interest in belief systems, in particular seeking some representation of the beliefs of others, down to any required degree of nesting—for example A's belief about B's belief about C—that could be constructed recursively at need, rather than being set out in advance, as in the pioneering systems emerging from the Toronto group under Ray Perrault (Allen and Perrault 1980). I began thinking about this with Janusz Bien of the University of Warsaw, who had also published a paper arguing that CL/NLP should consider "least effort" methods: in the sense that the brain might well, due to evolution, be a lazy processor and seek methods for understanding that minimized some value that could be identified with processing effort. I had argued in PS for choosing shortest chains of inferences between templates, and that the most connected/preferred template structure for a piece of text should be the one found first. I am not sure we ever proved any of this: It was just speculation, as was the preference for the most semantically connected representation, and the representation with the least information. All this is really only elementary information theory: a random string of words contains the maximum information, but that is not very helpful. Clearly, the preferred interpretation of "He was named after his father" (i.e., named *the same* rather than *later in time*) is not the least informative, since the latter contains no information at all—being necessarily true—so one would have to adapt any such slogan to: "prefer the interpretation with the least information, unless it is zero!"

The belief work, first with Bien, later with Afzal Ballim (Wilks and Ballim 1987) and John Barnden, has not been a successful paradigm in terms of take-up, in that it has not got into the general discourse, even in the way that Fauconnier's "Mental Spaces" (Fauconnier 1985) has. That approach uses the same spatial metaphor, but for strictly linguistic rather than belief and knowledge purposes. But I think the VIEWGEN belief paradigm, as it became, had virtues, and I want to exploit this opportunity to remind people of it. It was meant to capture the intuition that if we want, for language

understanding purposes, to construct X's beliefs about Y's beliefs—what I called the environment of Y-for-X—then:

1. It must be a construction that can be done in real time to any level of nesting required, because we cannot imagine it pre-stored for all future nestings, as Perrault el al. in effect assumed.
2. It must capture the intuition that much of our belief is accepted by default from others: As VIEWGEN expresses it, I will accept as a belief what you say, because I have normally no way of checking, or experimenting on, let alone refuting, the things you tell me, e.g. that you had eggs for breakfast yesterday. As someone in politics once put it, "There is no alternative." Unless, that is, what you say contradicts something I believe or can easily prove from what I believe.
3. We must be able to maintain apparently contradictory beliefs, provided they are held in separate spaces and will never meet as contradictions. I can thus maintain within my-space-for-you beliefs of yours (according to me) that I do not in fact hold.

In VIEWGEN, belief construction is done in terms of a "push down" metaphor: A permeable "container" of your beliefs is pushed into a "container" of my beliefs and what percolates through the membrane, from me to you, will be believed and ascribed to you, unless it is explicitly contradicted, namely, by some contrary belief I already ascribe to you, and which, as it were, keeps mine from percolating through. The idea is to construct the appropriate "inner belief space" at the relevant level of nesting, so that inference can be done, and to derive consequences (within that constrained content space) that also serve to model, in this case, you the belief holder in terms of goals and desires, in addition to beliefs. This approach is quite different not only from the Perrault/Toronto system of belief-relevant plans but also to AI theories that make use of sets-of-support premises since this is about belief-inheritance-by-default. It is also quite distinct from linguistic theories like Wilson and Sperber's Relevance Theory which take no account at all of belief as relative to individuals, but perform all operations in some space that is the same for everyone, which is an essentially Chomskyan ideal competence-style notion of belief that is not relative to individuals—which is of course absurd.

Mark Lee and a number of my students have created implementations of this approach and linked it to dialogue and other applications, but there has been no major application showing its essential role in a functioning conversational theory where complex belief states are created in real time. However, the field is, I believe, now moving in that direction (e.g., with POMDP theories [Williams and Young 2007]) since the possibility of populating belief theories with a realistic base from text by means of Information Extraction or Semantic Web parsing to RDF format is now real (a matter we shall return to subsequently).

There were, for me at least, two connections between the VIEWGEN belief work and Preference Semantics, in terms of meaning and its relation to processes. First, there was the role of choice and alternatives, crucial to PS, in that an assigned meaning interpretation for a text was no more than a choice of the best available among alternatives, because preference implies choice, in a way that generative linguistics—though not of course traditions like Halliday's—always displayed alternatives but considered choice between them a matter for mere performance. What was dispensable

to generative linguistics was the heart of the matter, I argued, to NLP/CL. Secondly, VIEWGEN suggested a view of meaning, consistent locally with PS, dependent on which individuals or classes one chose to see in terms of each other—the key notion here was seeing one thing as another and its consequences for meaning. So, if one chose to identify (as being the same person under two names) Joe (and what one believed about him) with Fred's father (and what one knew about him), the hypothesis was that a belief environment should be constructed for Joe-as-Fred's-father by percolating one set of beliefs into the other, just as was done by the basic algorithm for creating A's-beliefs-about-B's-beliefs from the component beliefs of A and B. This process created a hybrid entity, with intensional meaning captured by the set of propositions in that inner environment of belief space, but which was now neither Joe nor Fred's father but rather the system's point of view of their directional amalgamation: Joe-as-Fred's-father (which might contain different propositions from the result of Fred's-father-as-Joe).

More natural, and fundable, scenarios were constructed for this technique in those days, such as knowledge representations for Navy ships' captains genuinely uncertain as to whether ship-in-my-viewfinder-now was or was not to be identified with the stored representation for enemy-ship-number-X. The important underlying notion was one going back to Frege, and which first had an outing in Winograd's thesis (Winograd 1972), where he showed you could have representations for blocks that did not in fact exist on the Blocks World table. A semantics must be able to represent things without knowing whether they exist or not; that is a basic requirement.

Later, and working with John Barnden and Afzal Ballim, this same underlying process of conflating two belief objects was extended to the representation of "metaphorical objects," which could be described, quite traditionally in the literature, as A-viewed-as-B (e.g., an atom viewed as a billiard ball). The metaphorical object atom-as-billiard-ball was again created by the same push-down or fusion of belief sets as in the basic belief point-of-view procedure. All this may well have been fanciful, and was never fully exploited in published work with programs, but it did have a certain intellectual appeal in wanting to treat belief, points of view, metaphor and identification of intensional individuals—normally quite separate issues in semantics—as being modellable by the same simple underlying process (see Ballim, Wilks, and Barnden 1991). One novel element that did emerge from this analysis was that, in the construction of these complex intensional identifications, such as between "today's Wimbledon winner" and "the top male tennis seed," one could choose directions of "viewing as" with the belief sets that led to objects which were neither the classic *de re* nor *de dicto* outcomes: Those became just two among a range of choices, and the others of course had no handy Latin names.

Adapting to the "Empirical Wave" in NLP

For me, as with many others, especially in Europe, the beginning of the empirical wave in NLP was the work of Leech and his colleagues at Lancaster: CLAWS4 (a name which hides a UK political joke), their part-of-speech tagger based on large-scale annotation of corpora. Such tagging is now the standard first stage of almost every NLP process and it may be hard for some to realize the skepticism its arrival provoked: "What could anyone want that for?" was a common reaction from those still preoccupied by computational syntax or semantics. That system was sold to IBM, whose speech group, under Jelinek, Mercer, and Brown, subsequently astonished the CL/NLP world with their statistical machine translation system CANDIDE. I wrote critical papers about it at the time, not totally unconnected to the fact that I was funded by DARPA on the PANGLOSS project at NMSU (along with CMU and ISI/USC) to do MT by competing, but non-statistical, methods.

In one paper, I used the metaphor of “Stone soup” (Wilks 1996): A reference to the old peasant folk-tale of the traveler who arrives at a house seeking food and claiming to have a stone that makes soup from water. He begs a ham bone to stir the water and stone and eventually cons out of his hosts all the ingredients for real soup. The aspect of the story I was focusing on was that, in the CANDIDE system, I was not sure that the “stone,” namely IBM’s “fundamental equation of MT,” was in fact producing the results, and suggested that something else they were doing was, giving them their remarkable success rate of about 50% of sentences correctly translated. As their general methodology has penetrated the whole of NLP/CL, I no longer stand by my early criticisms; IBM were of course right, and had everything to teach the rest of us.

Early critics of data-driven, alias empirical, CL found it hard to accept, whatever its successes in, say, POS tagging, that its methods could extend to the heartland of semantics and pragmatics. Like others, I came to see this assumption was quite untrue, and myself moved towards Machine Learning (ML) approaches to word-sense disambiguation (e.g., Stevenson and Wilks 2001) and I now work in ML methods applied to dialogue corpora (as I shall mention subsequently). But the overall shift in approaches to semantics since 1990 has not only been in the introduction of statistical methods, and ML in particular, but also in the unexpected advantages that have been gained from what one might call non-statistical empirical linguistics, and in particular Information Extraction (IE; see Wilks 1997).

I referred earlier to the fact that my early work that could be called, in a general sense, semantic parsing, and that it was in fact some form of superficial pattern matching onto language chunks that was then transformed to different layers of compositional semantic representation. There were obvious relations between that general approach and what emerged from the DARPA competitions in the early 1990s as IE, a technology that, when honed by many teams, and especially when ML techniques were added to it later, had remarkable success and a range of applications; it also expanded out into other, traditionally separate, NLP areas such as question answering and summarization. This approach is not in essence statistical at all, however, although it is in a clear sense “superficial,” with the assumption that semantics is not necessarily a “deep” phenomenon but present on the language surface. I believe the IE movement is also one of the drivers behind the Semantic Web movement, to which I now turn, and which I think has brought NLP back to a position nearer the core of AI, from which it drifted away in the 1980s.

Meaning and the Semantic Web

The Semantic Web (SW; Berners-Lee, Hendler, and Lassila 2001) is what one could call Berners-Lee’s second big idea, after the World Wide Web; it can be described briefly as turning the Web into something that can also be understood by computers in the way that it is understood by people now, as a web of texts and pictures. Depending on one’s attitude to this enterprise, already well-funded by the European Commission at least, it can be described as any of the following:

1. As a revival of the traditional AI goal (at least since McCarthy and Hayes [1969]) of replacing language, with all its vagueness, by some form of logical representation upon which inference can be done.

2. As a hierarchy of forms of annotation—or what I shall call augmentation of content—reaching up from simple POS tagging to semantic class annotation (e.g. CITY, PERSON-NAME) to ontology membership and logical forms. DARPA/MUC/NIST competitions have worked their way up precisely this hierarchy over the years and many now consider that content can be “annotated onto language” reliably up to any required level. This can be thought of as extending IE techniques to any linguistic level by varieties of ML and annotation.
3. As a system of access to trusted databases that ground the meanings of terms in language; your telephone or social security number might ground you uniquely (in what is called a URI), or better still—and this is now the standard view—a unique identifying object number for you over and above phones and social systems. This is very much Tim Berners-Lee’s own view of the SW.

There is also a fourth view, much harder to express, that says roughly that, if we keep our heads, the SW can come into being with any system of coding that will tolerate the expansion of scale of the system, in the way that, miraculously, the hardware underpinnings of the World Wide Web have tolerated its extraordinary expansion without major breakdown. This is an engineering view that believes there are no fundamental problems about the meanings and reference of SW terms in, for example, the ontologies within the SW, and everything will be all right if we just hold tight.

This view may turn out to be true but it is impossible to discuss it. Similarly, view (3) has no special privilege because it is the World Wide Web founder’s own view: Marx was notoriously not a very consistent Marxist, and one can find multiple examples of this phenomenon. View (3) is highly interesting and close to philosophical views of meaning expressed over many years by Putnam, which can be summarized as the idea that scientists (and Berners-Lee was by origin a database expert and physicist) are “guardians of meaning” in some sense because they know what terms really mean, in a way that ordinary speakers do not. Putnam’s standard example is that of metals like molybdenum and aluminum, which look alike and, to the man in the street, have the same conceptual, intensional meaning, namely light, white, shiny metal. But only the scientist (says Putnam) knows the real meanings of those words because he knows the atomic weights of the two metals and methods for distinguishing them.

No one who takes Wittgenstein—and his view that we, the users of the language, are in charge of what terms mean, and not any expert—at all seriously can even consider such a view. On the view we are attributing to Wittgenstein, the terms are synonymous in a public language, just as *water* and *heavy water* are, and any evidence to the contrary is a private matter for science, not for meaning.

View (1) of the Semantic Web is a well-supported one, particularly by recycled AI researchers: They have, of course, changed tack considerably and produced formalisms for the SW, some of which are far closer to the surface of language than logic (what is known as RDF triples), as well as inference mechanisms like DAML-OIL that gain advantages over traditional AI methods on the large and practical scale the SW is intended to work over. On the other hand there are those in AI who say they have ignored much of the last 40 years of AI research that would have helped them. This dispute has a conventional flavor and it must be admitted that, in more than 40 years, AI itself did not come up with such formalisms that stood any chance at all of working on a large scale on unstructured material (i.e., text).

This leaves us with View (2), which is my own: namely, that we should see the SW partially in NLP terms, however much Berners-Lee rejects such a view and says NLP is irrelevant to the SW. The whole trend of SW research, in Europe at least, has been to build up to higher and higher levels of semantic annotation—a technology that has grown directly out of IE’s success in NLP—as a way of adding content to surface text. It seems to me obvious that any new SW will evolve from the existing WWW of text by some such method, and that method is basically a form of large-scale NLP, which now takes the form of transducers from text to RDF (such as the recently advertised Reuters API). The idea that the SW can start from scratch in some other place, ignoring the existing World Wide Web, seems to me unthinkable; successful natural evolution always adapts the function of what is available and almost never starts again afresh.

I have set out my views on this recently in more detail (Wilks 2008), but it is important to see that the SW movement—at least as I interpret it herein, and that does seem pretty close to the way research in it is currently being funded, under calls and titles like “semantic content”—is one that links to the themes already developed in this paper in several ways, and which correspond closely to issues in my own early work, but which have not gone away:

1. The SW takes semantic annotation of content as being a method—whether done by humans or after machine learning—of recoding content with special terms, terms close to what have traditionally been called semantic primitives. It is exactly this that was denied by the early forms of, say, statistical MT, where there was nothing available to the mechanism except the words themselves. This is also quite explicit in traditional IR, where, for example, Karen Spärck Jones consistently argued against any form of content recoding, including the SW. As she put it: “One of these [simple, revolutionary IR] ideas is taking words as they stand” (Jones 2003).
2. The SW accords a key role to ontologies as knowledge structures: partially hierarchical structures containing key terms—primitives again under another guise—whose meanings must be made clear, particularly at the more abstract levels. The old AI tradition in logic-based knowledge structuring—descending from McCarthy and Hayes (1969)—was simply to declare what these primitive predicates meant. The problem was that predicates, normally English words written in capital letters (as all linguistic primitives in the end seem to be), became affected by their inferential roles over time and the process of coding itself. This became very clear in the long-term CyC project (Lenat 1995) where the key predicates changed their meanings over 30 years of coding, but there was no way of describing that fact within the system, so as to guarantee consistency. In Nirenburg and Wilks (2000), Nirenburg and I debate this issue in depth, and I defend the position that one cannot simply maintain the meanings of such terms by fiat and independent of their usage—they look like words and they function like words because, in the end, they are words. The SW offers a way out of this classic AI dilemma by building up the hierarchy of annotations with empirical processes like ontology induction from corpora (e.g., ABRAXAS; see Iria et al. 2006); in this way the meanings of higher level terms are connected back directly to text usage. Braithwaite, my thesis advisor, described in his classic “Scientific explanation” (Braithwaite 1953) a process in the philosophy of science he

called “semantic ascent” by which the abstract high-level terms in a scientific theory, seen as a logical hierarchy of deductive processes—terms such as “neutron,“ possibly corresponding to unobservables—acquired meaning by an ascent of semantic interpretation up the theory hierarchy from meanings grounded in experimental terms at the bottom. It is some such grounding process I envisage the SW as providing for the meanings of primitive ontological terms in a knowledge structure.

3. The RDF forms, based on triples of surface items, as a knowledge base—usually with subject–action–object as basic form—can provide a less formal but more tractable base for knowledge than traditional First Order Predicate Logic (FOPL). They have a clear relationship back to the crude templates of my early work and the later templates of IE. I claim no precedence here, but only note the return of a functioning but plausible notion of “superficial semantics.” It seems to me not untrue historically to claim that RDF, the representational base of the SW, is a return of the level of representation that Schank (under the name Conceptual Dependency, in Schank [1975]) and I (under the name Preference Semantics) developed in the late 1960s and early 1970s (Wilks 1975). I remember that at the Stanford AI Lab at that time, John McCarthy, a strong advocate of FOPL as the right level of representation of language content, would comment that formalisms like these two might have a role as a halfway house on a route from language to a full logic representation. On one view of the SW that intermediate stage may prove to be the right stage, because full AI representations have never been able to deliver in terms of scale and tractability. Time will tell, and fairly soon.

The most important interest of the SW, from the point of view of this paper, is that it provides at last a real possibility of a large-scale test of semantic and knowledge coding: One thing the empirical movement has taught us is the vital importance of scale and the need to move away from toy systems and illustrative examples. I mentioned earlier the freely available Reuters API for RDF translation which Slashdot advertised under the title “Is the Semantic Web a Reality at Last?” This is exactly the kind of move to the large scale that we can hope will settle definitively some of these ancient issues about meaning and knowledge.

A Late Interest in Dialogue: The Companions Project

My only early exposure to dialogue systems was Colby’s PARRY: As I noted earlier, his team was on the same corridor as me at Stanford AI Lab in the early 1970s. I was a great admirer of the PARRY system: It seemed to me then, and still does, probably the most robust dialogue system ever written. It was available over the early ARPANET and tried out by thousands, usually at night: It was written in LISP and never broke down; making allowances for the fact it was supposed to be paranoid, it was plausible and sometimes almost intelligent. In any case it was infinitely more interesting than ELIZA, and it is one of the great ironies of our subject that ELIZA is so much better known. PARRY remembered what you had said, had elementary emotion parameters and, above all, had something to say, which chatbots never do. John McCarthy, who ran the AI Lab, would never admit that PARRY was AI, even though he tolerated it under his roof, as it were, for many years; he would say “It doesn’t even know who the President is," as if most of the world's population did! PARRY was in fact a semi-refutation of the claim that you need knowledge to understand and converse, because it plainly knew nothing; what it had was primitive "intentionality," in the sense that it had things "it wanted to say."

My own introduction to practical work on dialogue was when I was contacted in the late 1990s by David Levy, who had written 40 books on chess and ran a company that made chess machines. He already had a footnote in AI as the man who had bet McCarthy, Michie, and other AI leaders that a chess machine would not beat him within ten years, and he won the bet more than once. In the 1990s he conceived a desire to win the Loebner Prize[2] for the best dialogue program of the year, and came to us at Sheffield to fund a team to win it for him, which we did in 1997. I designed the system and drew upon my memories of PARRY, along with obvious advances in the role of knowledge bases and inference, and the importance of corpora and machine learning. For example, we took the whole set of winning Loebner dialogues off the Web so as to learn the kinds of things that the journalist-testers actually said to the trial systems to see if they were really humans or machines.

Our system, called CONVERSE (see Levy et al. 1997), claimed to be Catherine, a 34-year old female British journalist living in New York, and it owed something to PARRY, certainly in Catherine's desire to tell people things. It was driven by frames corresponding to each of about 80 topics that such a person might want to discuss; death, God, clothes, make-up, sex, abortion, and so on. It was far too top–down and unwilling to shift from topic to topic but it could seem quite smart on a good day, and probably won because we had built in news from the night before the competition of a meeting Bill Clinton had had that day at the White House with Ellen de Generes, a lesbian actress. This gave a certain immediacy to the responses intended to sway the judges, as in "Did you see that meeting Ellen had with Clinton last night?"

This was all great fun and gave me an interest in modeling dialogue that has persisted for a decade and is now exercised through COMPANIONS (Wilks 2004), a large EU 15-site four-year project that I run. COMPANIONS aims to change the way we think about the relationships of people to computers and the Internet by developing a virtual conversational "Companion." This will be an agent or "presence" that stays with the user for long periods of time, developing a relationship and "knowing" its owner's preferences and wishes. It will communicate with the user primarily by using and understanding speech, but also using other technologies such as touch screens and sensors.

Another general motivation for the project is the belief that the current Internet cannot serve all social groups well, and it is one of our objectives to empower citizens (including the non-technical, the disabled, and the elderly) with a new kind of interface based on language technologies. The vision of the Senior Companion—currently our main prototype—is that of an artificial agent that communicates with its user on a long-term basis, adapting to their voice, needs, and interests: A companion that would entertain, inform, and react to emergencies. It aims to provide access to information and services as well as company for the elderly by chatting, remembering past conversations, and organizing (and making sense of) the owner's photographic and image memories. This Companion would assume a user with a low level of technical knowledge, and who might have lost the ability to read or produce documents themselves unaided, but who might need help dealing with letters, messages, bills, and getting information from the Internet. During its conversations with its user or owner, the system

[2] See `http://www.loebner.net/Prizef/loebner-prize.html`.

builds up a knowledge inventory of family relations, family events in photos, places visited, and so on. This knowledge base is currently stored in RDF, the Semantic Web format, which has two advantages: first, a very simple inference scheme with which to drive further conversational inferences, and second, the possibility, not yet fulfilled, of accessing arbitrary amounts of world information from Wikipedia, already available in RDF, which could not possibly have been pre-coded in the dialogue manager, nor elicited in a conversation of reasonable length. So, if the user says a photo was taken in Paris, the Companion should be able to ask a question about Paris without needing that knowledge pre-coded, but only using rapidly accessed Wikipedia RDFs about Paris. An ultimate aim of this aspect of the Senior Companion is the provision of a life narrative, an assisted autobiography for everyone, one that could be given to relatives later if the owner chose to leave it to them. There is a lot of technical stuff in the Senior Companion: script-like structures—called DAFs or Dialogue Action Forms—designed to capture the course of dialogues on specific topics or individuals or images, and these DAFs we are trying to learn from tiled corpora. The DAFs are pushed and popped on a single stack, and that simple virtual machine is the Dialogue Manager, where DAFs being pushed, popped, or reentered at a lower stack point are intended to capture the exits from, and returns to, abandoned topics and the movement of conversational initiative between the system and the user. We are halfway through the project and currently have two prototype Companions: The other, based not at Sheffield but at Tampere, is a Health and Fitness Companion (HFC).[3] It is more task-oriented than the Senior Companion and aims to advise on exercise and diet. The HFC is on a mobile phone architecture as well as a PC, and we may seek to combine the two prototypes later. The central notion of a Companion is that of the same "personality," with its memory and voice being present no matter what the platform. It is not a robot, and could be embodied later in something like a chatty furry handbag, being held on a sofa and perhaps reminding you about the previous episodes of your favorite TV program.

Finale

This article has had something of the form of a life story, and everyone wants to believe their life is some kind of narrative rather than a random chase from funding agency to funding agency, with occasional pauses to carry out a successful proposal. But let us return to Newton for a moment in closing; for us in CL he is the great counter-example, of why we do not do science or engineering in that classic solitary manner:

> . . . where the statue stood
> Of Newton, with his prism and silent face,
> The marble index of a mind for ever
> Voyaging through strange seas of Thought, alone.
>
> — William Wordsworth (1770–1850)
> *The Prelude*, book iii, line 61

The emphasis there for me is on *alone*, which is pretty much unthinkable in our research world of teams and research groups. Our form of research is essentially corporate and cooperative; we may not be sure whose shoulders we are standing on, but we know whose hands we are holding. I have worked in such a way since my thirties and, at

[3] An early demo of a Companion can be seen on YouTube at `http://www.youtube.com/watch?v=SqIP6sTt1Dw`.

Sheffield, my work would not have been possible without a wide range of colleagues and former students in the NLP group there over many years and including Louise Guthrie, Rob Gaizauskas, Hamish Cunningham, Fabio Ciravegna, Mark Stevenson, Mark Hepple, Kalina Bontcheva, Christopher Brewster, Nick Webb and many others. In recent years, what one could call "DARPA culture"—of competitions and cooperation subtly mixed—as well as the great repositories of software and data like LDC and ELRA, have gone a long way to mitigate the personal and group isolation in the field.

But we do have to face the fact that, in many ways, we do not do classic science: We have no Newtons and will never have any. That is not to deny that we need real ideas and innovations, and now may be a time for fresh ones. We have stood on the shoulders of Fred Jelinek, Ken Church, and others for nearly two decades now, and the strain is beginning to tell as papers still strive to gain that extra 1% in their scores on some small task. We know that some change is in the air and I have tried to hint in this article as to some of the places where that might be, even if that will mean a partial return to older, unfashionable, ideas; for there is nothing new under the sun. But locating them and exploiting them will not be in my hands but in yours, readers of *Computational Linguistics*!

Acknowledgments
First of course to all those who have worked with me over many years and to whom I owe so much, particularly in connection with this award. Then to my current sponsor: This work was funded by the Companions project (`www.companions-project.org`) sponsored by the European Commission as part of the Information Society Technologies (IST) programme under EC grant number IST-FP6-034434.

References

Allen, James F. and C. Raymond Perrault. 1980. Analyzing intention in utterances. *Artificial Intelligence*, 15:143–178.

Ballim, Afzal, Yorick Wilks, and John A. Barnden. 1991. Belief ascription, metaphor, and intensional identification. *Cognitive Science*, 15(1):133–171.

Q1 Berners-Lee, T., J. Hendler, and O. Lassila. 2001. The semantic web. *Scientific American*, 28–37.

Braithwaite, Richard Bevan. 1953. *Scientific Explanation. A Study of the Function of Theory, Probability and Law in Science*. Cambridge University Press, Cambridge, UK.

Carnap, Rudolf. 1937. *The Logical Syntax of Language*. Kegan Paul, London.

Fauconnier, Gilles. 1985. *Mental Spaces*. Cambridge University Press, Cambridge, UK.

Iria, José, Christopher Brewster, Fabio Ciravegna, and Yorick Wilks. 2006. An incremental tri-partite approach to ontology learning. In *Proceedings of the Language Resources and Evaluation Conference (LREC-06)*, 22–28 May.

Jones, Karen Spärck. 1986. *Synonymy and semantic classification*. Edinburgh University Press, Edinburgh, Scotland.

Jones, Karen Spärck. 2003. Document retrieval: Shallow data, deep theories; historical reflections, potential directions. In *Advances in Information Retrieval*, Lecture Notes in Computer Science. Springer, Berlin/Heidelberg. Q2

Lenat, Douglas B. 1995. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38.

Levy, D., R. Catizone, B. Battacharia, A. Krotov, and Y. Wilks. 1997. Converse: A conversational companion. Q3

Masterman, Margaret. 1961. Semantic message detection for machine translation, using an interlingua. In *Proceedings of the First International Conference on Machine Translation of Languages and Applied Language Analysis*, pages 438–475. HMSO. Q4

Masterman, Margaret, Yorick Wilks, Branimir Boguraev, Steven Bird, Don Hindle, Martin Kay, David McDonald, and Hans Uszkoreit. 2005. *Language, Cohesion and Form (Studies in Natural Language Processing)*. Cambridge University Press, New York.

McCarthy, J. and P. J. Hayes. 1969. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence*, volume 4. Edinburgh University Press, Edinburgh, pages 463–502.

Nirenburg, Sergei and Yorick Wilks. 2000. Machine translation. *Advances in Computers*, 52:160–189.

Sager, Naomi and Ralph Grishman. 1975. The restriction language for computer grammars of natural language. *Communications of the ACM*, 18(7):390–400.

Schank, Roger C. 1975. *Conceptual Information Processing*. Elsevier Science Inc., New York.

Stevenson, Mark and Yorick Wilks. 2001. The interaction of knowledge sources in word sense disambiguation. *Computational Linguistics*, 27(3):321–349.

Waltz, David L. and Jordan B. Pollack. 1985. Massively parallel parsing: A strongly interactive model of natural language interpretation. *Cognitive Science*, 9(1):51–74.

Wilks, Y. 1975. Preference semantics. In E. L. Keenan, editor, *Formal Semantics of Natural Language*. Cambridge University Press, Cambridge, pages 329–348.

Wilks, Yorick. 1971. Decidability and natural language. *Mind*, 80:497–520.

Wilks, Yorick. 1972. *Grammar, Meaning and Machine Analysis of Language*. Routledge and Kegan Paul, London.

Wilks, Yorick. 1996. Statistical versus knowledge-based machine translation. *IEEE Expert: Intelligent Systems and Their Applications*, 11(2):12–18.

Wilks, Yorick. 1997. Information extraction as a core language technology. In *International Summer School on Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology*, volume 1299 of *Lecture Notes In Computer Science*, pages 1–9. Q5

Wilks, Yorick. 2004. Artificial companions. In *Machine Learning for Multimodal Interaction: First International Workshop*, pages 36–45.

Wilks, Yorick. 2008. The semantic web: Apotheosis of annotation, but what are its semantics? *IEEE Intelligent Systems*, 23(3):41–49.

Wilks, Yorick and Afzal Ballim. 1987. Multiple agents and the heuristic ascription of belief. In *Proceedings of the International Joint Conference Artificial Intelligence (IJCAI-87)*, pages 118–124. Q6

Williams, Jason D. and Steve Young. 2007. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech and Language*, 21(2):393–422.

Winograd, Terry. 1972. *Understanding Natural Language*. Academic Press, Orlando, FL.

Wittgenstein, Ludwig. 1973. *Philosophical Investigations*. Blackwell Publishers. Q7

附件四

ACL 2008 論文議程

Monday, June 16, 2008

9:00 – 9:10 Opening Session

9:10 – 10:10 Invited Talk: Marc Swerts, "Facial Expressions in Human-Human and Human-Machine Interactions"

10:10 – 10:40 Break

Session 1A: Information Extraction 1

10:40 – 11:05: Richman, Alexander E.; Patrick Schone *Mining Wiki Resources for Multilingual Named Entity Recognition*
11:05 – 11:30: Bergsma, Shane; Dekang Lin; Randy Goebel *Distributional Identification of Non-Referential Pronouns*
11:30 – 11:55: Pasca, Marius; Benjamin Van Durme *Weakly-Supervised Acquisition of Open-Domain Classes and Class Attributes from Web Documents and Query Logs*
11:55 – 12:20: Banko, Michele; Oren Etzioni *The Tradeoffs Between Open and Traditional Relation Extraction*

Session 1B: Language Resources and Evaluation

10:40 – 11:05: Mírovsk\'y, Jirí *PDT 2.0 Requirements on a Query Language*
11:05 – 11:30: Miyao, Yusuke; Rune Stre; Kenji Sagae; Takuya Matsuzaki; Jun’ ichi Tsujii *Task-oriented Evaluation of Syntactic Parsers and Their Representations*
11:30 – 11:55: Chan, Yee Seng; Hwee Tou Ng *MAXSIM: A Maximum Similarity Metric for Machine Translation Evaluation*
11:55 – 12:20: Voorhees, Ellen M. *Contradictions and Justifications: Extensions to the Textual Entailment Task*

Session 1C: Machine Translation 1

10:40 – 11:05: Cherry, Colin *Cohesive Phrase-Based Decoding for Statistical Machine Translation*
11:05 – 11:30: Deng, Yonggang; Jia Xu; Yuqing Gao *Phrase Table Training for Precision and Recall: What Makes a Good Phrase and a Good Phrase Pair?*
11:30 – 11:55: Zhang, Dongdong; Mu Li; Nan Duan; Chi-Ho Li; Ming Zhou *Measure Word Generation for English-Chinese SMT Systems*
11:55 – 12:20: Zhang, Hao; Chris Quirk; Robert C. Moore; Daniel Gildea *Bayesian Learning of Non-Compositional Phrases with Synchronous Parsing*

Session 1D: Speech Processing

10:40 – 11:05: Kaufmann, Tobias; Beat Pfister *Applying a Grammar-Based Language Model to a Simplified Broadcast-News Transcription Task*
11:05 – 11:30: Bisani, Maximilian; Paul Vozila; Olivier Divay; Jeff Adams *Automatic Editing in a Back-End Speech-to-Text System*
11:30 – 11:55: Fleischman, Michael; Deb Roy *Grounded Language Modeling for Automatic Speech Recognition of Sports Video*
11:55 – 12:20: Fleck, Margaret M. *Lexicalized Phonotactic Word Segmentation*

12:20 – 2:00 Lunch

Session 2A: Information Retrieval 1

2:00 – 2:25: Fang, Hui *A Re-examination of Query Expansion Using Lexical Resources*

2:25 – 2:50: Cao, Guihong; Stephen Robertson; Jian-Yun Nie *Selecting Query Term Alternations for Web Search by Exploiting Query Contexts*
2:50 – 3:15: Duan, Huizhong; Yunbo Cao; Chin-Yew Lin; Yong Yu *Searching Questions by Identifying Question Topic and Question Focus*

Session 2B: Language Generation

2:00 – 2:25: Mairesse, François; Marilyn Walker *Trainable Generation of Big-Five Personality Styles through Data-Driven Parameter Estimation*
2:25 – 2:50: Lee, John; Stephanie Seneff *Correcting Misuse of Verb Forms*
2:50 – 3:15: Espinosa, Dominic; Michael White; Dennis Mehay *Hypertagging: Supertagging for Surface Realization with CCG*

Session 2C: Machine Translation 2

2:00 – 2:25: Mi, Haitao; Liang Huang; Qun Liu *Forest-Based Translation*
2:25 – 2:50: Blunsom, Phil; Trevor Cohn; Miles Osborne *A Discriminative Latent Variable Model for Statistical Machine Translation*
2:50 – 3:15: Zhang, Hao; Daniel Gildea *Efficient Multi-Pass Decoding for Synchronous Context Free Grammars*

Session 2D: Semantics 1

2:00 – 2:25: Koller, Alexander; Michaela Regneri; Stefan Thater *Regular Tree Grammars as a Formalism for Scope Underspecification*
2:25 – 2:50: Davidov, Dmitry; Ari Rappoport *Classification of Semantic Relationships between Nominals Using Pattern Clusters*
2:50 – 3:15: Mitchell, Jeff; Mirella Lapata *Vector-based Models of Semantic Composition*

3:15 – 3:45 Break

Session 3A: Information Extraction 2

3:45 – 4:10: Arnold, Andrew; Ramesh Nallapati; William W. Cohen *Exploiting Feature Hierarchy for Transfer Learning in Named Entity Recognition*
4:10 – 4:35: Ji, Heng; Ralph Grishman *Refining Event Extraction through Cross-Document Inference*
4:35 – 5:00: Branavan, S.R.K.; Harr Chen; Jacob Eisenstein; Regina Barzilay *Learning Document-Level Semantic Properties from Free-Text Annotations*
5:00 – 5:25: Feng, Yansong; Mirella Lapata *Automatic Image Annotation Using Auxiliary Text Information*

Session 3B: Sentiment Analysis

3:45 – 4:10: Szarvas, György *Hedge Classification in Biomedical Texts with a Weakly Supervised Selection of Keywords*
4:10 – 4:35: Andreevskaia, Alina; Sabine Bergler *When Specialists and Generalists Work Together: Overcoming Domain Dependence in Sentiment Tagging*
4:35 – 5:00: Nomoto, Tadashi *A Generic Sentence Trimmer with CRFs*
5:00 – 5:25: Titov, Ivan; Ryan McDonald *A Joint Model of Text and Aspect Ratings for Sentiment Summarization*

Session 3C: Syntax & Parsing 1

3:45 – 4:10: Agirre, Eneko; Timothy Baldwin; David Martinez *Improving Parsing and PP Attachment Performance with Sense Information*
4:10 – 4:35: Hoyt, Frederick; Jason Baldridge *A Logical Basis for the D Combinator and Normal Form in CCG*
4:35 – 5:00: Vadas, David; James R. Curran *Parsing Noun Phrase Structure with CCG*

5:00 – 5:25: Vickrey, David; Daphne Koller *Sentence Simplification for Semantic Role Labeling*

3:45 – 5:50 Session 3D: Student Research Workshop

3:45 – 4:10: Hagiwara, Masato *A Supervised Learning Approach to Automatic Synonym Identification Based on Distributional Features*
4:10 – 4:35: Banik, Eva *An Integrated Architecture for Generating Parenthetical Constructions*
4:35 – 5:00: Eidelman, Vladimir *Inferring Activity Time in News through Event Modeling*
5:00 – 5:25: Liao, Shasha *Combining Source and Target Language Information for Name Tagging of Machine Translation Output*
5:25 – 5:50: Sun, Shuqi; Yin Chen; Jufeng Li *A Re-examination on Features in Regression Based Approach to Automatic MT Evaluation*

Poster Session Student Research Workshop (6:00-8:30)

Fossati, Davide *The Role of Positive Feedback in Intelligent Tutoring Systems*
Heintz, Ilana *Arabic Language Modeling with Finite State Transducers*
Kersey, Cynthia *Impact of Initiative on Collaborative Problem Solving*
McInnes, Bridget *An Unsupervised Vector Approach to Biomedical Term Disambiguation: Integrating UMLS and Medline*
Messiant, Cédric *A Subcategorization Acquisition System for French Verbs*
Trnka, Keith *Adaptive Language Modeling for Word Prediction*
Zhang, Yitao *A Hierarchical Approach to Encoding Medical Concepts for Clinical Notes*

5:25 – 6:00 Break

Poster and Demo Session (6:00-8:30)

Batista, Fernando; Nuno Mamede; Isabel Trancoso *Language Dynamics and Capitalization using Maximum Entropy*
Boston, Marisa Ferrara; John T. Hale; Reinhold Kliegl; Shravan Vasishth *Surprising Parser Actions and Reading Difficulty*
Carenini, Giuseppe; Raymond T. Ng; Xiaodong Zhou *Summarizing Emails with Conversational Cohesion and Subjectivity*
Chali, Yllias; Shafiq Joty *Improving the Performance of the Random Walk Model for Answering Complex Questions*
Chen, Wei *Dimensions of Subjectivity in Natural Language*
Chitturi, Rahul; John Hansen *Dialect Classification for Online Podcasts Fusing Acoustic and Language Based Structural and Semantic Information*
DeNero, John; Dan Klein *The Complexity of Phrase Alignment Problems*
Dickinson, Markus *Ad Hoc Treebank Structures*
de la Chica, Sebastian; Faisal Ahmad; James H. Martin; Tamara Sumner *Extractive Summaries for Educational Science Content*

Dligach, Dmitriy; Martha Palmer *Novel Semantic Features for Verb Sense Disambiguation*
Dredze, Mark; Joel Wallenberg *Icelandic Data Driven Part of Speech Tagging*
Duh, Kevin; Katrin Kirchhoff *Beyond Log-Linear Models: Boosted Minimum Error Rate Training for N-best Re-ranking*
Elsner, Micha; Eugene Charniak *Coreference-inspired Coherence Modeling*
Finkel, Jenny Rose; Christopher D. Manning *Enforcing Transitivity in Coreference Resolution*
Georgila, Kallirroi; Maria Wolters; Johanna Moore *Simulating the Behaviour of Older versus Younger Users when Interacting with Spoken Dialogue Systems*
Goldberg, Yoav; Reut Tsarfaty *A Single Generative Model for Joint Morphological Segmentation and Syntactic Parsing*
Goldwasser, Dan; Dan Roth *Active Sample Selection for Named Entity Transliteration*
Goldwater, Sharon; Dan Jurafsky; Christopher D. Manning *Which Words Are Hard to Recognize? Prosodic, Lexical, and Disfluency Factors that Increase ASR Error Rates*
HaCohen-Kerner, Yaakov; Ariel Kass; Ariel Peretz *Combined One Sense Disambiguation of Abbreviations*
Habash, Nizar *Four Techniques for Online Handling of Out-of-Vocabulary Words in Arabic-English Statistical Machine Translation*
Haertel, Robbie; Eric Ringger; Kevin Seppi; Carroll James; McClanahan Peter *Assessing the Costs of Sampling Methods in Active Learning for Annotation*
Hashimoto, Chikara; Sadao Kurohashi *Blog Categorization Exploiting Domain Dictionary and Dynamically Estimated Domains of Unknown Words*
Henderson, James; Oliver Lemon *Mixture Model POMDPs for Efficient Handling of Uncertainty in Dialogue Management*
Hermjakob, Ulf; Kevin Knight; Hal Daumé III *Name Translation in Statistical Machine Translation - Learning When to*

Transliterate
Hildebrand, Almut Silja; Kay Rottmann; Mohamed Noamany; Quin Gao; Sanjika Hewavitharana; Nguyen Bach; Stephan Vogel *Recent Improvements in the CMU Large Scale Chinese-English SMT System*
Johnson, Mark *Using Adaptor Grammars to Identify Synergies in the Unsupervised Acquisition of Linguistic Structure*
Karakos, Damianos; Jason Eisner; Sanjeev Khudanpur; Markus Dreyer *Machine Translation System Combination using ITG-based Alignments*
Kazama, Jun’ ichi; Kentaro Torisawa *Inducing Gazetteers for Named Entity Recognition by Large-Scale Clustering of Dependency Relations*
Kennedy, Alistair; Stan Szpakowicz *Evaluating Roget’ s Thesauri*
Kulkarni, Anagha; Jamie Callan *Dictionary Definitions based Homograph Identification using a Generative Hierarchical Model*
Li, Wenjie; Peng Zhang; Furu Wei; Yuexian Hou; Qin Lu *A Novel Feature-based Approach to Chinese Entity Relation Extraction*
Li, Zhifei; David Yarowsky *Unsupervised Translation Induction for Chinese Abbreviations using Monolingual Corpora*
Li, Jianguo; Chris Brew *Which Are the Best Features for Automatic Verb Classification*
Liu, Chao-Lin; Jen-Hsiang Lin *Using Structural Information for Identifying Similar Chinese Characters*
Liu, Yandong; Eugene Agichtein *You’ ve Got Answers: Towards Personalized Models for Predicting Success in Community Question Answering*
McClosky, David; Eugene Charniak *Self-Training for Biomedical Parsing*
Miller, Tim; William Schuler *A Unified Syntactic Model for Parsing Fluent and Disfluent Speech*
Moilanen, Karo; Stephen Pulman *The Good, the Bad, and the Unknown: Morphosyllabic Sentiment Tagging of Unseen Words*
Moschitti, Alessandro; Silvia Quarteroni *Kernels on Linguistic Structures for Answer Extraction*
Mrozinski, Joanna; Edward Whittaker; Sadaoki Furui *Collecting a Why-Question Corpus for Development and Evaluation of an Automatic QA-System*
Nakov, Preslav; Marti A. Hearst *Solving Relational Similarity Problems Using the Web as a Corpus*
Olsson, J. Scott; Douglas W. Oard *Combining Speech Retrieval Results with Generalized Additive Models*
Penn, Gerald; Xiaodan Zhu *A Critical Reassessment of Evaluation Baselines for Speech Summarization*
Polifroni, Joseph; Marilyn Walker *Intensional Summaries as Cooperative Responses in Dialogue: Automation and Evaluation*
Roth, Ryan; Owen Rambow; Nizar Habash; Mona Diab; Cynthia Rudin *Arabic Morphological Tagging, Diacritization, and Lemmatization Using Lexeme Models and Feature Ranking*
Saha, Sujan Kumar; Pabitra Mitra; Sudeshna Sarkar *Word Clustering and Word Selection Based Feature Reduction for MaxEnt Based Hindi NER*
Schulte im Walde, Sabine; Christian Hying; Christian Scheible; Helmut Schmid *Combining EM Training and the MDL Principle for an Automatic Verb Classification Incorporating Selectional Preferences*
Syed, Umar; Jason Williams *Using Automatically Transcribed Dialogs to Learn User Models in a Spoken Dialog System*
Talbot, David; Thorsten Brants *Randomized Language Models via Perfect Hash Functions*
Toutanova, Kristina; Hisami Suzuki; Achim Ruopp *Applying Morphology Generation Models to Machine Translation*
Tsuchiya, Masatoshi; Shinya Hida; Seiichi Nakagawa *Robust Extraction of Named Entity Including Unfamiliar Word*
Veale, Tony; Yanfen Hao; Guofu Li *Multilingual Harvesting of Cross-Cultural Stereotypes*
Wan, Stephen; Cecile Paris *In-Browser Summarisation: Generating Elaborative Summaries Biased Towards the Reading Context*
Wang, Qin Iris; Dale Schuurmans; Dekang Lin *Semi-Supervised Convex Training for Dependency Parsing*
Xia, Yunqing; Linlin Wang; Kam-Fai Wong; Mingxing Xu *Lyric-based Song Sentiment Classification with Sentiment Vector Space Model*
Yamangil, Elif; Rani Nelken *Mining Wikipedia Revision Histories for Improving Sentence Compression*
Yang, Fan; Jun Zhao; Bo Zou; Kang Liu; Feifan Liu *Chinese-English Backward Transliteration Assisted with Mining Monolingual Web Pages*
Yuret, Deniz *Smoothing a Tera-word Language Model*
Zapirain, Beñat; Eneko Agirre; Lluís Màrquez *Robustness and Generalization of Role Sets: PropBank vs. VerbNet*
Zhang, Min; Hongfei Jiang; Aiti Aw; Haizhou Li; Chew Lim Tan; Sheng Li *A Tree Sequence Alignment-based Tree-to-Tree Translation Model*

Demos (6:00-8:30)

6:00-8:30: Williams, Jason *Demonstration of a POMDP Voice Dialer*
6:00-8:30: Siddharthan, Advaith; Ann Copestake *Generating Research Websites Using Summarisation Techniques*
6:00-8:30: Versley, Yannick; Simone Paolo Ponzetto; Massimo Poesio; Vladimir Eidelman; Alan Jern; Jason Smith; Xiaofeng Yang; Alessandro Moschitti *BART: A Modular Toolkit for Coreference Resolution*
6:00-8:30: O’ Donnell, Mick *Demonstration of the UAM CorpusTool for Text and Image Annotation*
6:00-8:30: Huggins-Daines, David; Alexander I. Rudnicky *Interactive ASR Error Correction for Touchscreen Devices*
6:00-8:30: Germann, Ulrich *Yawat: Yet Another Word Alignment Tool*
6:00-8:30: Kang, Moonyoung; Sourish Chaudhuri; Mahesh Joshi; Carolyn P. Rosé *SIDE: The Summarization Integrated Development Environment*
6:00-8:30: Yarrington, Debra; John Gray; Chris Pennington; H. Timothy Bunnell; Allegra Cornaglia; Jason Lilley; Kyoko

Nagao; James Polikoff *ModelTalker Voice Recorder—An Interface System for Recording a Corpus of Speech for Synthesis*

6:00-8:30: Kaisser, Michael *The QuALiM Question Answering Demo: Supplementing Answers with Paragraphs drawn from Wikipedia*

Tuesday, June 17, 2008

Session: Outstanding Paper Award Presentations

9:00 – 9:10 Presentation of Awards

9:10 – 9:35: Bartlett, Susan; Grzegorz Kondrak; Colin Cherry *Automatic Syllabification with Structured SVMs for Letter-to-Phoneme Conversion*

9:35 – 10:00: Shen, Libin; Jinxi Xu; Ralph Weischedel *A New String-to-Dependency Machine Translation Algorithm with a Target Dependency Language Model*

10:00 – 10:25: Huang, Liang *Forest Reranking: Discriminative Parsing with Non-Local Features*

10:25 – 10:40: Bikel, Daniel M.; Vittorio Castelli *Event Matching Using the Transitive Closure of Dependency Relations*

10:40 – 11:10 Break

Session 4A: Syntax & Parsing 2

11:10 – 11:35: Koo, Terry; Xavier Carreras; Michael Collins *Simple Semi-supervised Dependency Parsing*

11:35 – 12:00: Nesson, Rebecca; Giorgio Satta; Stuart M. Shieber *Optimal k-arization of Synchronous Tree-Adjoining Grammar*

12:00 – 12:25: Dridan, Rebecca; Valia Kordoni; Jeremy Nicholson *Enhancing Performance of Lexicalised Grammars*

Session 4B: Dialogue

11:10 – 11:35: Ai, Hua; Diane J. Litman *Assessing Dialog System User Simulation Evaluation Measures Using Human Judges*

11:35 – 12:00: Lee, Cheongjae; Sangkeun Jung; Gary Geunbae Lee *Robust Dialog Management with N-Best Hypotheses Using Dialog Examples and Agenda*

12:00 – 12:25: Rieser, Verena; Oliver Lemon *Learning Effective Multimodal Dialogue Strategies from Wizard-of-Oz Data: Bootstrapping and Evaluation*

Session 4C: Machine Learning 2

11:10 – 11:35: Milidiú, Ruy Luiz; Cícero Nogueira dos Santos; Julio C. Duarte *Phrase Chunking Using Entropy Guided Transformation Learning*

11:35 – 12:00: Zhu, Xiaojin; Andrew B. Goldberg; Michael Rabbat; Robert Nowak *Learning Bigrams from Unigrams*

12:00 – 12:25: Suzuki, Jun; Hideki Isozaki *Semi-Supervised Sequential Labeling and Segmentation Using Giga-Word Scale Unlabeled Data*

Session 4D: Semantics 2

11:10 – 11:35: Bhagat, Rahul; Deepak Ravichandran *Large Scale Acquisition of Paraphrases for Learning Surface Patterns*

11:35 – 12:00: Szpektor, Idan; Ido Dagan; Roy Bar-Haim; Jacob Goldberger *Contextual Preferences*

12:00 – 12:25: Davidov, Dmitry; Ari Rappoport *Unsupervised Discovery of Generic Relationships Using Pattern Clusters and its Evaluation by Automatically Generated SAT Analogy Questions*

12:25 – 2:00 Lunch

Session 5A: Short Papers 1 (Machine Translation)

2:00 – 2:15: Xiong, Deyi; Min Zhang; Aiti Aw; Haizhou Li *A Linguistically Annotated Reordering Model for BTG-based Statistical Machine Translation*

2:15 – 2:30: Badr, Ibrahim; Rabih Zbib; James Glass *Segmentation for English-to-Arabic Statistical Machine Translation*

2:30 – 2:45: Chen, Boxing; Min Zhang; Aiti Aw; Haizhou Li *Exploiting N-best Hypotheses for SMT Self-Enhancement*

2:45 – 3:00: He, Zhongjun; Qun Liu; Shouxun Lin *Partial Matching Strategy for Phrase-based Statistical Machine Translation*

Session 5B: Short Papers 2 (Speech)

2:00 – 2:15: Varadarajan, Balakrishnan; Sanjeev Khudanpur; Emmanuel Dupoux *Unsupervised Learning of Acoustic Sub-word Units*

2:15 – 2:30: Nenkova, Ani; Agustin Gravano; Julia Hirschberg *High Frequency Word Entrainment in Spoken Dialogue*

2:30 – 2:45: McMillian, Yolanda; Juan Gilbert *Distributed Listening: A Parallel Processing Approach to Automatic Speech Recognition*

Session 5C: Short Papers 3 (Semantics)

2:00 – 2:15: Bethard, Steven; James H. Martin *Learning Semantic Links from a Corpus of Parallel Temporal and Causal Relations*

2:15 – 2:30: Snajder, Jan; Bojana Dalbelo Basic; Sasa Petrovic; Ivan Sikiric *Evolving New Lexical Association Measures Using Genetic Programming*

2:30 – 2:45: Katrenko, Sophia; Pieter Adriaans *Semantic Types of Some Generic Relation Arguments: Detection and Evaluation*

2:45 – 3:00: Roa, Sergio; Valia Kordoni; Yi Zhang *Mapping between Compositional Semantic Representations and Lexical Semantic Resources: Towards Accurate Deep Semantic Parsing*

Session 5D: Short Papers 4 (Generation/Summarization)

2:00 – 2:15: Krahmer, Emiel; Erwin Marsi; Paul van Pelt *Query-based Sentence Fusion is Better Defined and Leads to More Preferred Results than Generic Sentence Fusion*

2:15 – 2:30: Belz, Anja; Albert Gatt *Intrinsic vs. Extrinsic Evaluation Measures for Referring Expression Generation*

2:30 – 2:45: Liu, Feifan; Yang Liu *Correlation between ROUGE and Human Evaluation of Extractive Meeting Summaries*

2:45 – 3:00: Schilder, Frank; Ravikumar Kondadadi *FastSum: Fast and Accurate Query-based Multi-document Summarization*

3:00 – 3:15 Break

Session 5E: Short Papers 1 (Syntax)

3:15 – 3:30: Gabbard, Ryan; Seth Kulick *Construct State Modification in the Arabic Treebank*

3:30 – 3:45: Musillo, Gabriele Antonio; Paola Merlo *Unlexicalised Hidden Variable Models of Split Dependency Grammars*

3:45 – 4:00: Lin, Feng; Fuliang Weng *Computing Confidence Scores for All Sub Parse Trees*

4:00 – 4:15: Foster, Jennifer; Joachim Wagner; Josef van Genabith *Adapting a WSJ-Trained Parser to Grammatically Noisy Text*

Session 5F: Short Papers 2 (Dialog/Statistical Methods)

3:15 – 3:30: Rangarajan Sridhar, Vivek Kumar; Srinivas Bangalore; Shrikanth Narayanan *Enriching Spoken Language Translation with Dialog Acts*

3:30 – 3:45: Kim, Donghyun; Hyunjung Lee; Choong-Nyoung Seon; Harksoo Kim; Jungyun Seo *Speakers’ Intention Prediction Using Statistics of Multi-level Features in a Schedule Management Domain*

3:45 – 4:00: Dredze, Mark; Koby Crammer *Active Learning with Confidence*

4:00 – 4:15: Goldberg, Yoav; Michael Elhadad *splitSVM: Fast, Space-Efficient, non-Heuristic, Polynomial Kernel*

Computation for NLP Applications

Session 5G: Short Papers 3 (Semantics/Phonology)

3:15 – 3:30: Nielsen, Rodney D.; Wayne Ward; James H. Martin; Martha Palmer *Extracting a Representation from Text for Semantic Analysis*

3:30 – 3:45: Regneri, Michaela; Markus Egg; Alexander Koller *Efficient Processing of Underspecified Discourse Representations*

3:45 – 4:00: Brown, Susan Windisch *Choosing Sense Distinctions for WSD: Psycholinguistic Evidence*

4:00 – 4:15: Alfonseca, Enrique; Slaven Bilac; Stefan Pharies *Decompounding query keywords from compounding languages*

Session 5H: Short Papers 4 (IR/Sentiment Analysis)

3:15 – 3:30: Li, Shoushan; Chengqing Zong *Multi-domain Sentiment Classification*

3:30 – 3:45: Trnka, Keith; Kathleen McCoy *Evaluating Word Prediction: Framing Keystroke Savings*

3:45 – 4:00: Elsayed, Tamer; Jimmy Lin; Douglas Oard *Pairwise Document Similarity in Large Collections with MapReduce*

4:00 – 4:15: Sun, Qi; Runxin Li; Dingsheng Luo; Xihong Wu *Text Segmentation with LDA-Based Fisher Kernel*

4:15 – 4:45 Break

Session 6A: Question Answering

4:45 – 5:10: Kaisser, Michael; Marti A. Hearst; John B. Lowe *Improving Search Results Quality by Customizing Summary Lengths*

5:10 – 5:35: Ding, Shilin; Gao Cong; Chin-Yew Lin; Xiaoyan Zhu *Using Conditional Random Fields to Extract Contexts and Answers of Questions from Online Forums*

5:35 – 6:00: Surdeanu, Mihai; Massimiliano Ciaramita; Hugo Zaragoza *Learning to Rank Answers on Large Online QA Collections*

Session 6B: Phonology, Morphology 1

4:45 – 5:10: Adler, Meni; Yoav Goldberg; David Gabay; Michael Elhadad *Unsupervised Lexicon-Based Resolution of Unknown Words for Full Morphological Analysis*

5:10 – 5:35: Snyder, Benjamin; Regina Barzilay *Unsupervised Multilingual Learning for Morphological Segmentation*

5:35 – 6:00: Goldberg, Yoav; Meni Adler; Michael Elhadad *EM Can Find Pretty Good HMM POS-Taggers (When Given a Good Start)*

Session 6C: Machine Translation 3

4:45 – 5:10: Uszkoreit, Jakob; Thorsten Brants *Distributed Word Clustering for Large Scale Class-Based Language Modeling in Machine Translation*

5:10 – 5:35: Avramidis, Eleftherios; Philipp Koehn *Enriching Morphologically Poor Languages for Statistical Machine Translation*

5:35 – 6:00: Haghighi, Aria; Percy Liang; Taylor Berg-Kirkpatrick; Dan Klein *Learning Bilingual Lexicons from Monolingual Corpora*

Session 6D: Semantics 3

4:45 – 5:10: Zhao, Shiqi; Haifeng Wang; Ting Liu; Sheng Li *Pivot Approach for Extracting Paraphrase Patterns from Bilingual Corpora*

5:10 – 5:35: Chambers, Nathanael; Dan Jurafsky *Unsupervised Learning of Narrative Event Chains*

5:35 – 6:00: Diab, Mona; Alessandro Moschitti; Daniele Pighin *Semantic Role Labeling Systems for Arabic using Kernel*

Methods

7:00 – 11:00 Banquet

Wednesday, June 18, 2008

9:00 – 10:10 Invited Talk: Susan Dumais, "Supporting Searchers in Searching"

10:10 – 10:30 Break

Session 7A: Summarization

10:30 – 10:55: Biadsy, Fadi; Julia Hirschberg; Elena Filatova *An Unsupervised Approach to Biography Production Using Wikipedia*

10:55 – 11:20: Mei, Qiaozhu; ChengXiang Zhai *Generating Impact-Based Summaries for Scientific Literature*

11:20 – 11:45: Nenkova, Ani; Annie Louis *Can You Summarize This? Identifying Correlates of Input Difficulty for Multi-Document Summarization*

Session 7B: Discourse & Pragmatics

10:30 – 10:55: Elsner, Micha; Eugene Charniak *You Talking to Me? A Corpus and Algorithm for Conversation Disentanglement*

10:55 – 11:20: Yang, Xiaofeng; Jian Su; Jun Lang; Chew Lim Tan; Ting Liu; Sheng Li *An Entity-Mention Model for Coreference Resolution with Inductive Logic Programming*

11:20 – 11:45: Eisenstein, Jacob; Regina Barzilay; Randall Davis *Gestural Cohesion for Topic Segmentation*

Session 7C: Machine Learning 2

10:30 – 10:55: Reichart, Roi; Katrin Tomanek; Udo Hahn; Ari Rappoport *Multi-Task Active Learning for Linguistic Annotations*

10:55 – 11:20: Mann, Gideon S.; Andrew McCallum *Generalized Expectation Criteria for Semi-Supervised Learning of Conditional Random Fields*

11:20 – 11:45: Liang, Percy; Dan Klein *Analyzing the Errors of Unsupervised Learning*

Session 7D: Phonology, Morphology 2

10:30 – 10:55: Zhang, Yue; Stephen Clark *Joint Word Segmentation and POS Tagging Using a Single Perceptron*

10:55 – 11:20: Jiang, Wenbin; Liang Huang; Qun Liu; Yajuan Lü *A Cascaded Linear Model for Joint Chinese Word Segmentation and Part-of-Speech Tagging*

11:20 – 11:45: Jiampojamarn, Sittichai; Colin Cherry; Grzegorz Kondrak *Joint Processing and Discriminative Training for Letter-to-Phoneme Conversion*

11:45 – 12:50 Lunch

12:50 – 2:20 ACL Business Meeting

Session 8A: Information Retrieval 2

2:30 – 2:55: Bao, Shenghua; Huizhong Duan; Qi Zhou; Miao Xiong; Yunbo Cao; Yong Yu *A Probabilistic Model for Fine-Grained Expert Search*

2:55 – 3:20: Weerkamp, Wouter; Maarten de Rijke *Credibility Improves Topical Blog Post Retrieval*

3:20 – 3:45: Csomai, Andras; Rada Mihalcea *Linguistically Motivated Features for Enhanced Back-of-the-Book Indexing*

3:45 – 4:10: Elsayed, Tamer; Douglas W. Oard; Galileo Namata *Resolving Personal Names in Email Using Context Expansion*

Session 8B: Syntax & Parsing 3

2:30 – 2:55: Nivre, Joakim; Ryan McDonald *Integrating Graph-Based and Transition-Based Dependency Parsers*

2:55 – 3:20: Finkel, Jenny Rose; Alex Kleeman; Christopher D. Manning *Efficient, Feature-based, Conditional Random Field Parsing*

3:20 – 3:45: Gómez-Rodríguez, Carlos; John Carroll; David Weir *A Deductive Approach to Dependency Parsing*

3:45 – 4:10: Bender, Emily M. *Evaluating a Crosslinguistic Grammar Resource: A Case Study of Wambaya*

Session 8C: Machine Translation 2

2:30 – 2:55: Ganchev, Kuzman; João V. Graça; Ben Taskar *Better Alignments = Better Translations?*

2:55 – 3:20: Lin, Dekang; Shaojun Zhao; Benjamin Van Durme; Marius Pasca *Mining Parenthetical Translations from the Web by Word Alignment*

3:20 – 3:45: Marton, Yuval; Philip Resnik *Soft Syntactic Constraints for Hierarchical Phrased-Based Translation*

3:45 – 4:10: Dyer, Christopher; Smaranda Muresan; Philip Resnik *Generalizing Word Lattice Translation*

Session 8D: Semantics 4

2:30 – 2:55: Zhao, Shiqi; Cheng Niu; Ming Zhou; Ting Liu; Sheng Li *Combining Multiple Resources to Improve SMT-based Paraphrasing Model*

2:55 – 3:20: Srikumar, Vivek; Roi Reichart; Mark Sammons; Ari Rappoport; Dan Roth *Extraction of Entailed Semantic Relations Through Syntax-Based Comma Resolution*

3:20 – 3:45: de Marneffe, Marie-Catherine; Anna N. Rafferty; Christopher D. Manning *Finding Contradictions in Text*

3:45 – 4:10: Kozareva, Zornitsa; Ellen Riloff; Eduard Hovy *Semantic Class Learning from the Web with Hyponym Pattern Linkage Graphs*

4:40 – 6:10 Lifetime Achievement Award Presentation and Closing Session

附件五

報告人所發表的論文

Using Structural Information for Identifying Similar Chinese Characters

Chao-Lin Liu **Jen-Hsiang Lin**
Department of Computer Science, National Chengchi University, Taipei 11605, Taiwan
{chaolin, g9429}@cs.nccu.edu.tw

Abstract

Chinese characters that are similar in their pronunciations or in their internal structures are useful for computer-assisted language learning and for psycholinguistic studies. Although it is possible for us to employ image-based methods to identify visually similar characters, the resulting computational costs can be very high. We propose methods for identifying visually similar Chinese characters by adopting and extending the basic concepts of a proven Chinese input method--Cangjie. We present the methods, illustrate how they work, and discuss their weakness in this paper.

1 Introduction

A Chinese sentence consists of a sequence of characters that are not separated by spaces. The function of a Chinese character is not exactly the same as the function of an English word. Normally, two or more Chinese characters form a Chinese word to carry a meaning, although there are Chinese words that contain only one Chinese character. For instance, a translation for "conference" is "研討會" and a translation for "go" is "去". Here "研討會" is a word formed by three characters, and "去" is a word with only one character.

Just like that there are English words that are spelled similarly, there are Chinese characters that are pronounced or written alike. For instance, in English, the sentence "John plays an important roll in this event." contains an incorrect word. We should replace "roll" with "role". In Chinese, the sentence "今天上午我們來**試場**買菜" contains an incorrect word. We should replace "試場" (a place for taking examinations) with "市場" (a market). These two words have the same pronunciation, shi(4) chang(3)[†], and both represent locations. The sentence "經理要我**構買**一部計算機" also contains an error, and we need to replace "構買" with "購買". "構買" is considered an incorrect word, but can be confused with "購買" because the first characters in these words look similar.

Characters that are similar in their appearances or in their pronunciations are useful for computer-assisted language learning (cf. Burstein & Leacock, 2005). When preparing test items for testing students' knowledge about correct words in a computer-assisted environment, a teacher provides a sentence which contains the character that will be replaced by an incorrect character. The teacher needs to specify the answer character, and the software will provide two types of incorrect characters which the teachers will use as distracters in the test items. The first type includes characters that look similar to the answer character, and the second includes characters that have the same or similar pronunciations with the answer character.

Similar characters are also useful for studies in Psycholinguistics. Yeh and Li (2002) studied how similar characters influenced the judgments made by skilled readers of Chinese. Taft, Zhu, and Peng (1999) investigated the effects of positions of radicals on subjects' lexical decisions and naming responses. Computer programs that can automatically provide similar characters are thus potentially helpful for designing related experiments.

2 Identifying Similar Characters with Information about the Internal Structures

We present some similar Chinese characters in the first subsection, illustrate how we encode Chinese characters in the second subsection, elaborate how we improve the current encoding method to facilitate the identification of similar characters in the third subsection, and discuss the weakness of our current approach in the last subsection.

2.1 Examples of Similar Chinese Characters

We show three categories of confusing Chinese characters in Figures 1, 2, and 3. Groups of similar characters are separated by spaces in these figures. In Figure 1, characters in each group differ at the stroke level. Similar characters in every group in the first row in Figure 2 share a common part, but the shared part is not the radical of these characters. Similar characters in every group in the second row in Figure 2 share a common part, which is the radical of these characters. Similar characters in every group in Figure 2 have different pronunciations. We show six groups of homophones that also share a component in Figure 3. Characters that are similar in both pronunciations and internal structures are most confusing to new learners.

[†] We use Arabic digits to denote the four tones in Mandarin.

Proceedings of ACL-08: HLT, Short Papers (Companion Volume), pages 93–96,
Columbus, Ohio, USA, June 2008. ©2008 Association for Computational Linguistics

士土工干千 戌戍成 田由甲申
母毋 勿匆 人入 未末 采釆 凹凸

Figure 1. Some similar Chinese characters

頸勁 搆溝 陪倍 硯現 裸棵 搞篙
列刑 盆盎盂盅 因困囚 間閒閃開

Figure 2. Some similar Chinese characters that have different pronunciations

形刑型 踵種腫 購構搆 紀記計
園圓員 脛逕徑痙勁

Figure 3. Homophones with a shared component

It is not difficult to list all of those characters that have the same or similar pronunciations, e.g., "試場" and "市場", if we have a machine readable lexicon that provides information about pronunciations of characters and when we ignore special patterns for tone sandhi in Chinese (Chen, 2000).

In contrast, it is relatively difficult to find characters that are written in similar ways, e.g., "構" with "購", in an efficient way. It is intriguing to resort to image processing methods to find such structurally similar words, but the computational costs can be very high, considering that there can be tens of thousands of Chinese characters. There are more than 22000 different characters in large corpus of Chinese documents (Juang et al., 2005), so directly computing the similarity between images of these characters demands a lot of computation. There can be more than 4.9 billion combinations of character pairs. The Ministry of Education in Taiwan suggests that about 5000 characters are needed for ordinary usage. In this case, there are about 25 million pairs.

The quantity of combinations is just one of the bottlenecks. We may have to shift the positions of the characters "appropriately" to find the common part of a character pair. The appropriateness for shifting characters is not easy to define, making the image-based method less directly useful; for instance, the common part of the characters in the right group in the second row in Figure 3 appears in different places in the characters.

Lexicographers employ radicals of Chinese characters to organize Chinese characters into sections in dictionaries. Hence, the information should be useful. The groups in the second row in Figure 2 show some examples. The shared components in these groups are radicals of the characters, so we can find the characters of the same group in the same section in a Chinese dictionary. However, information about radicals as they are defined by the lexicographers is not sufficient. The groups of characters shown in the first row in Figure 2 have shared components. Nevertheless, the shared components are not considered as radicals, so the characters, e.g., "頸"and "勁", are listed in different sections in the dictionary.

2.2 Encoding the Chinese Characters

The Cangjie[‡] method is one of the most popular methods for people to enter Chinese into computers. The designer of the Cangjie method, Mr. Bong-Foo Chu, selected a set of 24 basic elements in Chinese characters, and proposed a set of rules to decompose Chinese characters into elements that belong to this set of building blocks (Chu, 2008). Hence, it is possible to define the similarity between two Chinese characters based on the similarity between their Cangjie codes.

Table 1, not counting the first row, has three sections, each showing the Cangjie codes for some characters in Figures 1, 2, and 3. Every Chinese character is decomposed into an ordered sequence of *elements*. (We will find that a subsequence of these elements comes from a major *component* of a character, shortly.) Evidently, computing the number of shared elements provides a viable way to determine "visually similar" characters for characters that appeared in Figure 2 and Figure 3. For instance, we can tell that "搞" and "篙" are similar because their Cangjie codes share "卜口月", which in fact represent "高".

	Cangjie Codes		Cangjie Codes
士	十一	土	土
工	一中一	干	一十
勿	心竹竹	匆	竹田心
未	十木	末	木十
頸	一一一月金	勁	一一大尸
硯	一口月山山	現	一土月山山
搞	手卜口月	篙	竹卜口月
列	一弓中弓	刑	一廿中弓
因	田大	困	田木
間	日弓日	閒	日弓月
踵	口一竹十土	種	竹木竹十土
腫	月竹十土	紀	女火尸山
購	月金廿廿月	構	木廿廿月
記	卜口尸山	計	卜口十
圓	田口月金	員	口月山金
脛	月一女一	逕	卜一女一
徑	竹人一女一	痙	大一女一

Table 1. Cangjie codes for some characters

[‡] http://en.wikipedia.org/wiki/Cangjie_method

Unfortunately, the Cangjie codes do not appear to be as helpful for identifying the similarities between characters that differ subtly at the stroke level, e.g., "士土工干" and other characters listed in Figure 1. There are special rules for decomposing these relatively basic characters in the Cangjie method, and these special encodings make the resulting codes less useful for our tasks.

The Cangjie codes for characters that contain multiple components were intentionally simplified to allow users to input Chinese characters more efficiently. The longest Cangjie code for any Chinese character contains no more than five elements. In the Cangjie codes for "脛" and "徑", we see "一女一" for the component "巠", but this component is represented only by "一一" in the Cangjie codes for "頸" and "勁". The simplification makes it relatively harder to identify visually similar characters by comparing the actual Cangjie codes.

2.3 Engineering the Original Cangjie Codes

Although useful for the sake of designing input method, the simplification of Cangjie codes causes difficulties when we use the codes to find similar characters. Hence, we choose to use the complete codes for the components in our database. For instance, in our database, the codes for "巠", "脛", "徑", "頸", and "勁" are, respectively, "一女女一", "月一女女一", "竹人一女女一", "一女女一一月山金", and "一女女一大尸".

The knowledge about the graphical structures of the Chinese characters (cf. Juang et al., 2005; Lee, 2008) can be instrumental as well. Consider the examples in Figure 2. Some characters can be decomposed vertically; e.g., "盅" can be split into two smaller components, i.e., "中" and "皿". Some characters can be decomposed horizontally; e.g., "現" is consisted of "王" and "見". Some have enclosing components; e.g., "人" is enclosed in "口" in "囚". Hence, we can consider the locations of the components as well as the number of shared components in determining the similarity between characters.

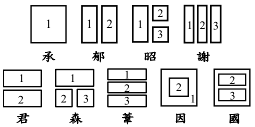


Figure 4. Arrangements of components in Chinese

Figure 4 illustrates possible layouts of the components in Chinese characters that were adopted by the Cangjie method (cf. Lee, 2008). A sample character is placed below each of these layouts. A box in a layout indicates a component in a character, and there can be at most three components in a character. We use digits to indicate the ordering the components. Notice that, in the second row, there are two boxes in the second to the rightmost layout. A larger box contains a smaller one. There are three boxes in the rightmost layout, and two smaller boxes are inside the outer box. Due to space limits, we do not show "1" for this outer box.

After recovering the simplified Cangjie code for a character, we can associate the character with a tag that indicates the overall layout of its components, and separate the code sequence of the character according to the layout of its components. Hence, the information about a character includes the tag for its layout and between one to three sequences of code elements. Table 2 shows the annotated and expanded codes of the sample characters in Figure 4 and the codes for some characters that we will discuss. The layouts are numbered from left to right and from top to bottom in Figure 4. Elements that do not belong to the original Canjie codes of the characters are shown in smaller font.

	Layout	Part 1	Part 2	Part 3
承	1	弓弓手人		
郁	2	大月	弓中	
昭	3	日	尸竹	口
謝	4	卜一一口	竹難竹	木戈
君	5	尸大	口	
森	6	木	木	木
葦	7	廿	木一	手
因	8	田	大	
國	9	田	戈	口一
頸	2	一女女一	一月山金	
徑	2	竹人	一女女一	
員	5	口	月山金	
圓	9	田	口	月山金
相	2	木	月山	
想	5	木月山	心	
箱	6	竹	木	月山

Table 2. Annotated and expanded code

Recovering the elements that were dropped out by the Cangjie method and organizing the subsequences of elements into parts facilitate the identification of similar characters. It is now easier to find that the character (頸) that is represented by "一女女一" and "一月山金" looks similar to the character (徑) that is represented by "竹人" and "一女女一" in our database than using their original Cangjie codes in Table 1. Checking the codes for "員" and "圓" in Table 1 and Table 2 will offer an additional support for our design decisions.

In the worst case, we have to compare nine pairs of code sequences for two characters that both have three components. Since we do not simplify codes for components and all components have no more than five elements, conducting the comparisons operations are simple.

2.4 Drawbacks of Using the Cangjie Codes

Using the Cangjie codes as the basis for comparing the similarity between characters introduces some potential problems.

It appears that the Cangjie codes for some characters, particular those simple ones, were not assigned without ambiguous principles. Relying on Cangjie codes to compute the similarity between such characters can be difficult. For instance, "分" uses the fifth layout, but "兌" uses the first layout in Figure 4. The first section in Table 1 shows the Cangjie codes for some character pairs that are difficult to compare.

Due to the design of the Cangjie codes, there can be at most one component at the left hand side and at most one component at the top in the layouts. The last three entries in Table 2 provide an example for these constraints. As a standalone character, "相" uses the second layout. Like the standalone "相", the "相" in "箱" was divided into two parts. However, in "想", "相" is treated as an individual component because it is on top of "想". Similar problems may occur elsewhere, e.g., "森焚" and "恩因". There are also some exceptional cases; e.g., "品" uses the sixth layout, but "闆" uses the fifth layout.

3 Concluding Remarks

We adopt the Cangjie alphabet to encode Chinese characters, but choose not to simplify the code sequences, and annotate the characters with the layout information of their components. The resulting method is not perfect, but allows us to find visually similar characters more efficient than employing the image-based methods.

Trying to find conceptually similar but contextually inappropriate characters should be a natural step after being able to find characters that have similar pronunciations and that are visually similar.

Acknowledgments

Work reported in this paper was supported in part by the plan NSC-95-2221-E-004-013-MY2 from the National Science Council and in part by the plan ATU-NCCU-96H061 from the Ministry of Education of Taiwan.

References

Jill Burstein and Claudia Leacock. editors. 2005. *Proceedings of the Second Workshop on Building Educational Applications Using NLP*, ACL.

Matthew Y. Chen. 2000. *Tone Sandhi: Patterns across Chinese Dialects*. (Cambridge. Studies in Linguistics 92.) Cambridge: Cambridge University Press.

Bong-Foo Chu. 2008. *Handbook of the Fifth Generation of the Cangjie Input Method*, web version, available at http://www.cbflabs.com/book/ocj5/ocj5/index.html. Last visited on 14 Mar. 2008.

Hsiang Lee. 2008. *Cangjie Input Methods in 30 Days*, http://input.foruto.com/cjdict/Search_1.php, Foruto Company, Hong Kong. Last visited on 14 Mar. 2008.

Derming Juang, Jenq-Haur Wang, Chen-Yu Lai, Ching-Chun Hsieh, Lee-Feng Chien, and Jan-Ming Ho. 2005. Resolving the unencoded character problem for Chinese digital libraries. *Proceedings of the Fifth ACM/IEEE Joint Conference on Digital Libraries*, 311–319.

Marcus Taft, Xiaoping Zhu, and Danling Peng. 1999. Positional specificity of radicals in Chinese character recognition, *Journal of Memory and Language*, **40**, 498–519.

Su-Ling Yeh and Jing-Ling Li. 2002. Role of structure and component in judgments of visual similarity of Chinese characters, *Journal of Experimental Psychology: Human Perception and Performance*, **28**(4), 933–947.