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

機率式建模技術與自然語言的標記、認知和教學

計畫類別：■ 個別型計畫 □ 整合型計畫
計畫編號： NSC-97-2221-E-004-007-MY2
執行期間： 2008 年 8 月 1 日至 2010 年 7 月 31 日

計畫主持人：劉昭麟
共同主持人：高照明、蔡介立
計畫參與人員： 賴敏華、田侃文、黃志斌、黃昭憲、莊怡軒、翁睿妤

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 2009 年 5 月 31 日

中文摘要

本年度的工作重點在於持續上一年度的工作內容，在機率式學生建模的工作上，持續發表相關期刊論文。在應用自然語言處理技術於語文教學方面，也持續過去之工作，今年特別著重於中文錯字研究與句子重組的問題上。除了這兩大類的工作之外，也與政大心理系蔡介立教授合作研究中文母語使用者閱讀過程中的眼動路徑。整體而言，本計畫延續過去多年的研究基礎，在過去十個月之中，接受並正式發表的論文數目有十一篇。其中期刊論文兩篇，國際會議論文三篇，國內會議論文六篇。另有兩篇已經投稿之國際會議論文，審查結果尚未公告。

英文摘要

In the first half of this project, we continued what we have been doing in the past few years. We worked on the construction of student models using a probability-based approach, and continued to publish research papers. We have also applied the techniques for natural language processing to computer assisted language learning. In past several months, we have focused on research issues regarding incorrect Chinese words and regarding the reconstruction of scrambled sentences. In addition, in order to offer better assistance in learning languages, we worked with Professor Tsai of the Department of Psychology (National Chengchi University) to study how native speakers of Chinese move their eyes while they read Chinese. Overall, we published 11 papers in the past 10 months. Two of them are journal articles, three are international conference papers, and six are domestic conference papers. Two other submitted papers are still under review.

工作報告

本年度的工作內容分為三個方向：機率式學生模型的建構、應用自然語言處理技術輔助語文教學、人類閱讀歷程之研究。在這三個方向上，本研究計畫，均已陸續發表論文，以下以摘要方式簡報所獲致之成果與經驗，詳細之成果請讀者參考所附之論文。

機率式學生模型的建構

這一研究計畫主要是承繼過去幾年的努力，利用貝氏網路(Bayesian networks)來捕捉學生對於綜合觀念的學習歷程，經過多年的持續努力，研究成果終於發表在國際著名的International Journal of Artificial Intelligence in Education (IJAIED)，內容接近五十頁超過兩萬字。另一部份未能在 IJAIED 中詳述的技術與概念，則發表於 Behaviormetrika 期刊，該期刊是日本行動計算學會(The Behaviormetric Society of Japan)的代表期刊，期刊內容收錄於日本 Scientific Links Japan。IJAIED 是人工智慧與教育學會(International AIED Society)的代表期刊，該學會所主辦的 AIED 研討會、International Conference on Intelligent Tutoring Systems 和 IEEE 的 ICALT 是電腦輔助教育這一領域中最具盛名的一些國際學術研討會。

這一項研究的工作內容是探討學生在學習多個基本觀念時，我們有沒有辦法透過測驗技術，來探究學生如何融合個別基本觀念以習得綜合觀念。由於人們在接受測驗時，常有運氣好答對或者運氣不好答錯的現象，因此人的外顯表現，並不一定反映其真實能力，因此要透過學生的答題表現來探究其內心的細部狀態（即學習歷程）是一件很困難的事情。研究結果顯示，隨著學生碰巧猜對與意外答錯的機率的變化，這一個研究問題的可達成成果有很大的差異。除了這兩個基本變因之外，本研究也發現如何掌握學生族群的能力分佈，也是探究學習歷程的重要因素。

本研究所應用的技術包含貝氏網路、類神經網路(artificial neural networks)、支持向量機(support vector machines)還有一些經驗法則(heuristic rules)。技術的主軸是利用機器學習技術來學習大量資料中的貝氏網路機率模型。

其他細節請參考論文[1,2]。

應用自然語言處理技術輔助語文教學

電腦輔助語文教學(Computer Assisted Language Learning)是國內外許多機構競相發展的技術。世界各國的高度交流，國際化的趨勢與需求高漲，使得語文學習變得比過去更加重要。如何能夠應用資訊通訊科技來提高學習語文的效率與成效，顯然是一件極度吸引人的研究議題。

政大這一個研究群在過去幾年利用處理自然語言的技術，開發過不少有趣的應用環境，在英文試題輔助出題和中文試題輔助出題方面，都有所成就，而且成果都發表在 Annual Meeting of Association for Computational Linguistics (ACL)，ACL 的年度研討會算是計算語言學界第一級的研討會。除了 ACL 的論文之外，部分的成果發表於今年的 IEA/AIE 年度研討會，IEA/AIE 研討會的接受率長年都控制在 30%左右，算是應用領域中比較好的研討會。除了國際學術會議之外，為了與國內學者交流，我們也將成果以中文撰寫，發表於計算語言學會的 ROCLING 研討會。

在這一研究方向上，本年度的重點工作仍和去年一樣，偏重於中文錯字的研究，我們探討了中文錯字的產生的原因，資料顯示中文錯字的產生有高達 80%的機會是因為發音的相同或者相近，另有相當的比例是字體相近引起的。本實驗從中文改錯字的出題輔助，逐漸跨到研究母與使用者為何寫錯字的研究，這兩份研究成果都發表於 ACL 的短文論文。工作內容請參考論文[3,4,5,8]

在錯字研究之外，我們也研究句子重組的出題輔助系統。句子重組的試題，是把一個給定的語句切割成一些字串，然後要求學生利用這一些打散的字串重建原始的語句。切割語句不是困難的工作，但是要確保學生會組合成原來的語句卻不簡單。一個被打散的語句，可能可以組合成許多不同的句子。例如，「this new bike is better than that old car」經過打散，

可以重組為「this new car is better than that old bike」。如何輔助出題教師確保學生所重組的答案只能是符合教學目標的那一些語句，是一件不容易達到的工作。這一部分的研究工作，目前仍然在進行中，部分的成果已經投稿，但是評審結果仍然不知道。

長期而言，本實驗室有計畫重拾計畫主持人過去的機器翻譯研究工作，因此今年度有兩個準備工作。一則是建置中學程度電腦輔助英漢翻譯習作的環境，一則是利用電腦軟體翻譯數理科目的英文試題。這兩個工作的成果都已經發表於 ROCLING [9, 10]。電腦輔助英漢翻譯習作環境的工作，主要是輔導研究生作一些基本研究工作，同時也累積實驗室一些研究資源。技術層次雖然不能說高，但是同時具有訓練與累積研究資源的意義。電腦輔助的試題翻譯則是面對真實的翻譯工作，我們建立了 language model 加上平行語料庫的建立，已經具備研究翻譯系統的雛形。我們期待這一方面的努力，在未來幾年之內，能夠讓實驗室能夠認真地進行機器翻譯的研究工作。

人類閱讀歷程之研究

在研究語文教學與電腦輔助教育的過程中，我們需要學生的學習模型。學生的模型是電腦輔助測驗的重要基礎，即使是 item response theory (IRT, http://en.wikipedia.org/wiki/Item_response_theory)也可以視為是一種間接的學生模型。在 IRT 模型中，我們以統計技術建立學生表現與能力關係的模型，儲存於試題的參數之中。

在自然語言處理和語文教育裡面，我們如果能夠知道人類的閱讀與認知歷程的話，就有可能提高計算機理解人類語文資料的技術層次。瞭解人類認知歷程，也有助於我們設計有效率的語文學習環境。這一些都是我們走向這一研究的原因。

在過去一年之中，透過政大心理系蔡介立教授的協助，我們得以研究真人母語使用者的演動資料，並且加以分析。研究結果顯示人與人的差異性極大，即使我們獲得珍貴的四十個真人的演動資料，目前暫時沒有找到很肯定的模型。儘管如此，我們已經把部分成果撰寫成一篇短文，目前處於審稿階段。

其他自然語言處理技術的應用：資訊檢索與文件分類

延續更久之前的國科會研究計畫，我們今年在 ROCLING 發表了一篇關於中文訴訟文書的檢索系統的論文[6]。這一成果，其實是整合了過去多年研究的功能，在一整個資訊檢索的環境之中。

學術研討會的投稿論文該由誰來擔任評審委員？這是一個很難的問題。我們可以把這一個問題當作是一種文件分類的問題，以個別評審當作是一個類別，我們希望把所收到的稿件分到所屬的類別。這當然是一個簡化的想像，一個研討會的評審指派有許多因素要考慮；例如，有合作關係者不宜互審、同一評審不宜審查過多論文、同時我們也必須考慮評審的興趣等。因此，這一方面的工作，純粹是探索的性質；目前有限的成果發表於 ROCLING[11]。

電腦遊戲學習

這一項工作，主要是訓練一位有潛力的大學部學生，使之能夠對於機器學習技術有初步的應用經驗。我們培養了一位大學部畢業生，習得初步的機器學習技術，應用於黑白棋(或稱蘋果棋、Reversi、Othello)，開發出任意棋盤的Othello的新想法，並且發表一篇TAAI論文[7]。該生即將加入交大吳毅成教授的研究團隊。

計畫成果自評

以政治大學資訊科學系在國內資訊領域的排名所可以收到的研究生，加上計畫主持人在研究、教學與校外學術服務而言，計畫主持人自以為本計畫的執行是成功的。

在論文發表上，雖然我們仍有許多可以再進步的空間，但是也逐漸擠身於國際間頂尖的學術會議與期刊論文。

在學生的培養方面，這兩年這一個實驗室為真正的頂尖大學培育了不少可用之才，就在今年度，就有一位赴台大就讀博士班、一位赴台大就讀碩士班、一位赴交大就讀碩士班的研究生。

本年度至今發表之論文

1. Chao-Lin Liu. Selecting Bayesian-network models based on simulated expectation, *Behaviormetrika*, **36**(1), 1−25. The Behaviormetric Society of Japan, Japan, April 2009.
2. 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.
3. Chao-Lin Liu, Kan-Wen Tien, Min-Hua Lai, Yi-Hsuan Chuang, Shih-Hung Wu. Phonological and logographic influences on errors in written Chinese words, *Proceedings of the Seventh Workshop on Asian Language Resources*, the Forty Seventh Annual Meeting of the Association for Computational Linguistics (**ACL’09**), to appear. Singapore, 2-7 August 2009.
4. Chao-Lin Liu, Kan-Wen Tien, Min-Hua Lai, Yi-Hsuan Chuang, Shih-Hung Wu. Capturing errors in written Chinese words, *Proceedings of the Forty Seventh Annual Meeting of the Association for Computational Linguistics* (**ACL’09**), short papers, to appear. Singapore, 2-7 August 2009.
5. Chao-Lin Liu, Kan-Wen Tien, Yi-Hsuan Chuang, Chih-Bin Huang, Juei-Yu Weng. Two applications of lexical information to computer-assisted item authoring for elementary Chinese, *Lecture Notes in Computer Science* 5579: *Proceedings of the Twenty Second International Conference on Industrial Engineering & Other Applications of Applied Intelligent Systems* (**IEA/AIE’09**), 470−480. Tainan, Taiwan, 24-27 June 2009.
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。臺灣，宜蘭，2008年11月21-22日。(中文內容)
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。臺灣，宜蘭，2008年11月21-22日。(中文內容)
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。臺灣，臺北，2008 年 9 月 4-5 日。(中文內容)

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。臺灣，臺北，2008 年 9 月 4-5 日。(中文內容)
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。臺灣，臺北，2008 年 9 月 4-5 日。(中文內容)
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。臺灣，臺北，2008 年 9 月 4-5 日。(中文內容)

附錄：發表論文之代表作

Chao-Lin Liu. Selecting Bayesian-network models based on simulated expectation, *Behaviormetrika*, **36**(1), 1−25. The Behaviormetric Society of Japan, Japan, April 2009.

SELECTING BAYESIAN-NETWORK MODELS BASED ON SIMULATED EXPECTATION

Chao-Lin Liu[*]

Identifying the best network structure from a myriad of candidates is not an easy task, and we propose a supervised learning method for this task. We test the idea with an instance of learning student models from students' responses to test items, because student models are very important for intelligent tutoring systems. The training data for the classifiers were simulated based on the expectation about students' item responses when students learn in different ways, and the trained classifier was used to select the model from the list of candidate models based on the observed item responses. Experimental results indicate that, even when item responses do not faithfully reflect students' competence in the concepts, our classifiers still help us differentiate very similar models with indirect observations.

1. Introduction

A simple exercise for students who are learning the modeling techniques is to find the distribution over the height of a population. A common strategy for solving such an exercise is to assume a Gaussian distribution and set out to find the mean and variance for the underlying distribution. The assumption is not arbitrary because many statistics for natural phenomena conform to the Gaussian distribution. The experience-based expectation helps us simplify the problem of model selection by confining the search space to the class of Gaussian distributions. We should be able to employ a similar idea to the problem of selecting Bayesian networks in the model selection task. To examine the applicability of this idea, we attempt to tackle a student modeling problem in this paper.

Furnishing appropriate educational material at appropriate timings is important in educational activities. Teachers, utilizing their professional knowledge and experience, may judge the timing, choose the material, and provide individualized guidance when they interact directly with their students. When building intelligent tutoring systems, we attempt to capture and express the professional knowledge and experience in both student and teacher models so that the resulting products can assist students' learning when teachers are not immediately available.

Building good models of teachers and students is not a simple task. It takes a lot of training for an ordinary person to become a qualified teacher, and it takes even more time and costs for a novice teacher to turn experienced. On the other hand, students vary in a wide range of ways. They differ in their competence and in their interests, for instance; and the needs for the same student may also change over time. As a result, it demands plenty of effort to build computational models of teachers and students, so different research projects focus on diverse aspects of students when they discuss student modeling.

In this paper, we discuss an application of supervised learning methods for selecting candidate Bayesian networks, and test the idea with the problems of modeling how students learn composite concepts. We assume that there are test items which are designed to test the competence in related concepts. We also assume the availability of students' responses to these test items, and use the item responses to select the best model from the candidate models that are provided by domain experts. The domain experts do not need to provide the class tags for students' data as in most supervised learning. Instead, the domain experts specify the structures and ranges of parameters for the candidate models, with which we create simulated data based on the candidate models, and use these simulated data and their class tags to select the candidate model that fits students' item responses.

Experience shows that students' external behaviors do not necessarily reflect their internal states. In educational assessment problems, students may fail to answer some test items correctly when they are competent in the concepts that are being tested, and, conversely, student may answer correctly just because of lucky guesses when they are not competent (e.g., VanLehn et al., 1994; Millán & Pérez-de-la-Cruz, 2002). We will refer to the former cases, *slip*, and the latter cases, *guess*, in this paper. We employ Bayesian networks (Pearl, 1988; Jensen & Nielsen, 2007) to capture this uncertain relationship between students' competence in concepts and their responses to test items.

To acquire *composite concepts*, students need to combine their knowledge about two or more other concepts to form the new concept. These "other" concepts can be basic concepts or composite concepts. For instance, to add two fractions that have different denominators, we have to convert these fractions to have a common denominator and then add the numerators. To convert the original fractions to have a common denominator, students need to comprehend the concept about common multiples which is based on a more fundamental concept—the concept about multiples. Hence, the concepts about multiples, common multiples, making fractions to have a common denominator, and adding two fractions with a common denominator are needed to add two fractions that have different denominators.

Although we may list the prerequisites for composite concepts, we are wondering how students manage to combine the prerequisite concepts to form a higher level concept (Gierl et al., 2007). Let *A*, *B*, *C*, and *D* represent four basic concepts for learning a composite concept *ABCD*. How do we know whether students learn *ABCD* by directly combining all of these basic concepts into the composite concept, or whether they first combine *A* and *B* into an intermediate product (say a composite concept *AB*), combine *C* and *D* into another intermediate product (say *CD*), and then learn *ABCD* by combining *AB* and *CD*?

Before attempting to answer this question, we look into two relevant questions. Are these more detailed models beneficial for the design of intelligent tutoring systems? Wouldn't the professional teachers provide detailed models directly? We cannot benefit from using a detailed student model if we cannot take appropriate actions when we know the values of some variables in the model (Mislevy & Gitomer, 1996). The introduction of the intermediate variables might improve the efficiency of conducting inferences with computational models, i.e., the evaluation of Bayesian networks, but the existence of these variables does not directly improve the quality of educational advises an intelligent tutoring system may produce.

There is positive evidence to the first question. Carmona et al. (2005) showed that they improved the efficiency of an adaptive testing procedure by introducing appropriate prerequisite relationships into a multi-layered Bayesian network. Hence, using more detailed models may be helpful for intelligent tutoring systems, though there are more thorough discussions about this point in the literature (Sleeman, 1989; Nichols et al., 1995; Leighton & Gierl, 2007).

The answers to the second question depend on the level of details of the models that we are concerning. It is not uncommon that professional teachers provide some high level information about the student models, and we apply computational techniques to acquire the parameters for the abstract models. This is the case when we obtain the parameters for models that are recommended or constructed based on the Item Response Theory (van der Linden & Hambleton, 1997) and when we learn the conditional probability distributions for student models that employ Bayesian networks (Mislevy et al., 1999). Hence, there is no doubt that professionals can help and should participate in the model construction process.

Despite that professionals may provide the abstract models for intelligent tutoring systems, some researchers have tried to explore the possibility of using computational techniques to learn the models from students' records. Computational techniques are applicable not only for estimating the parameters and implementing the models that are specified by professional teachers but also for helping the professional teachers to find the best models. Computational techniques can become instrumental for model construction, for instance, when there is no professional information avail-

Key Words and Phrases: Student Modeling, Intelligent Tutoring, Learning Bayesian Networks, Model Selection, Computational Cognition, Causal Modeling, Supervised Learning

[*] Department of Computer Science, National Chengchi University, Wen-Shan, Taipei 11605, Taiwan.
E-mail: chaolin@nccu.edu.tw

able or when the professionals can apply the computational techniques to compare the fitness of alternative models.

Vomlel (2004) applied the techniques for learning Bayesian networks (Heckerman, 1999; Jordan, 1999; Neapolitan, 2003) to obtain an initial Bayesian network from students' records, augmented the network with additional variables based on certain principles that experts provided, and compared the effectiveness of using the resulting networks in assessment tasks. Desmarais et al. (2006) noticed the resulting complexity of considering hidden variables in learning Bayesian networks, and chose to learn networks that included only observable variables. Because the learned structures consist mainly of nodes for test items, they are called *item-to-item knowledge structures*.

The nature of the study reported in this paper is related to but different from Vomlel's and Desmarais' research. Since we set out to find the relationships between the nodes that represent the competence levels in concepts, we must accept the existence of these non-observable nodes in the Bayesian network. Therefore, we are looking for the network structures that include a mixture of observable and unobservable variables, and these variables are only probabilistically related. We do not have to find unknown variables as Vomlel did, and, at the same time, we consider variables that are not directly observable, which Desmariais excluded.

We explore a new venue for model construction with machine learning techniques in this paper. We do not completely rely on machine learning techniques, nor do we completely depend on perfectly specified information about the models. We assume that teachers can provide partial specification about the students and that the teachers would like to find the best possible student model from a set of candidate models. To solve this problem, we apply the partial specification about the students and the candidate models to generate data of simulated students. Then, we compare the simulated data and the records of real students. The candidate model that is used to generate the best matching simulated data is chosen to be the model for the real students. Due to the explorative nature of this study, we use simulated data in place of real data about students as well.

We compare the effects of using the proposed method and a heuristic method, and experimental results show that the proposed method can perform very well and outperform the heuristic method significantly, when the quality of the partial specification about the students is reasonably well. Since the quality of the partial specification is controlled by the professional teachers who we may consult, it should be reasonable to trust the quality of the provided information. In later sections, we will discuss more thoroughly about the experimental results and deliberate on some limitations of the proposed methods.

In Section 2, we provide technical definitions and background information about this research. We also explain how we generate the data for simulated students. In Section 3, we illustrate our ideas with simple examples and analyze the complexity of this research problem. In Section 4, we delineate our approach for finding the student models. In Section 5, we present experimental procedures, results, and analyses. In Section 6, we discuss the experimental results, and in Section 7, we list some related research work and limitations of the presented results.

2. Definitions, Formulation, and Simulation

We assume that we have a set of concepts that are involved in the study and that we have a set of test items that are designed for evaluating students' competence in each of these concepts. Let $\Psi = \{C_1, C_2, \cdots, C_i, \cdots, C_\gamma\}$ denote the set of concepts, where C_i is the *i*-th concept, and let I_i denote the set of test items for C_i. Some of the concepts in Ψ are **composite**, and some are **basic**. In this paper, we use single letters to denote basic concepts and a sequence of concatenated letters to denote composite concepts. We call the direct prerequisite concept of a composite concept the **parent concepts** of the composite concept, and, for convenience, the concepts for which the test items are designed for are also called the parent concepts of the test items. As we have explained in Section 1, students may directly or indirectly integrate their knowledge about the basic concepts to become competent in the composite concept. Hence, both basic concepts and composite concepts

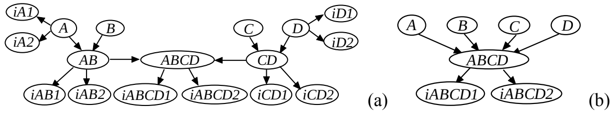

Figure 1. (partial) Bayesian networks for (a) `AB~C~D` (b) `A~B~C~D`

can serve as parent concepts. For instance, if students learn *ABCD* directly from *AB*, *C*, and *D*, then *AB*, *C*, and *D* are, but *A* and *B* are not, the parent concepts of *ABCD*.

Testing is a common way to peek into the competence levels of students, so we employ students' responses to test items to guess how they learn composite concepts. To simplify the presentation, we call a possible way of learning a composite concept a **learning pattern**, and we connect the names of the parent concepts of a composite concept with a tilde ("~") to refer to a particular learning pattern for the composite concept. For instance, we use `AB~C~D` to represent the situation in which students directly integrate *AB*, *C*, and *D* to learn *ABCD*. We assume that students respond to all of the test items that are designed to assess the competence of the concepts in the study, and we use these **item responses** to select the most possible learning patterns in question.

2.1 Formulating Learning Patterns with Bayesian Networks

We use a node to represent the competence level of a concept and a node to represent the correctness of a student's response to a test item. We call these nodes **concept nodes** and **item nodes**, respectively, for simplicity. When there is no risk of confusion, we use the name of the concepts for the names of their nodes. Names for the item nodes have the format ***iNj***: *i* denoting an item, *N* representing the name of the parent concept of the test item, and *j* carrying the identification code for this item. In the current study, both concept nodes and item nodes are Boolean.

Although the directions of links in Bayesian networks do not necessarily suggest the causal directions (Glymour & Cooper, 1999), we follow the recommendations stated in (Russell & Norvig, 2003) to achieve simpler network structures. Since the competence level of a concept directly influences whether students answer correctly to the test items for this concept, we make the node for C_i the parent node for nodes that represent test items in I_i in the Bayesian network. Based on our definition of the parent concepts, we add links to the node for a composite concept from the nodes for its parent concepts.

Figure 1(a) shows a partial Bayesian network that carries the belief that the parent concepts of *ABCD* are *AB* and *CD*. Nodes *iA1* and *iA2* represent the correctness of students' responses to two test items for concept *A*. In this network, we do not show item nodes for all of the concepts to maintain the readability of the network. Figure 1(b) shows a Bayesian network that carries the belief that the parent concepts of *ABCD* are *A*, *B*, *C*, and *D*. These are the two cases for learning the addition of fractions in Section 1.

2.2 Representing Competence Patterns with GC Matrices

Students' item responses should relate to their competence levels in concepts. We assume that students have different **competence patterns** over the concepts in Ψ, and adopt matrices to represent the competence patterns of students who belong to different types. Because the format is similar to the *Q* matrices that Tatsuoka (1983, 1995) used to encode the relationships between the test items and the tested concepts, we name our matrices as *GC* matrices.

Table 1. A *GC* matrix with only two types of students

group	*A*	*B*	*C*	*D*	*AB*	*AC*	*AD*	*BC*	*BD*	*CD*	*ABC*	*ABD*	*ACD*	*BCD*	*ABCD*
g1	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*
g2	*1*	*1*	*0*	*1*	*1*	*1*	*0*	*0*	*0*	*1*	*1*	*1*	*1*	*0*	*1*

Table 1 shows a *GC* matrix that specifies competence patterns for only two types of students. The leftmost column shows the names of the types, and the headings of other columns show the names of concepts. The semantics of the contents of the cells depend on whether the concepts are basic or composite. Let $q_{i,j}$ denote the cell at the intersection of *i*-th row and *j*-th column in a *GC* matrix. If $q_{i,j}$ is 1 and the heading of the *j*-th column is a basic concept, then the *i*-th type of student is competent in the basic concept with a high probability. If $q_{i,j}$ is 1 and the heading of the *j*-th column is a composite concept, then the *i*-th type of student is competent in integrating the parent concepts of the composite concept with a high probability. In both cases, a "0" in the cells suggests a low probability. Note that, when the column heading is a composite concept, a "1" in the cell does not imply that the corresponding type of students is competent in the composite concept. This type of students may not be competent in the composite concept, if they lack the competence in the parent concepts. (In other words, a "1" for basic concepts represents competence, but a "1" for composite concepts denotes just the ability to integrate parent concepts.) The contents of our *GC* matrices are related both the *rule nodes* and the *rule application nodes* that Martin and VanLehn (1995) defined. We will discuss how we control the ranges of the probability values in Section 2.3.

Without considering the uncertain factors such as *guess* and *slip*, the students of the first type in Table 1 are competent in all of the concepts. The students of the second type lack competence in some concepts. Since the composite concepts that involve only two basic concepts must be learned directly from the basic concept, we can infer the competence levels of students based on the information in a *GC* matrix. Therefore, we can infer that students of the second type will not be competent in *AC* because they lack the competence in *C*, although they are competent in integrating the parent concepts of *AC*.

Note that the contents of *GC* matrices do not provide sufficient information for us to infer whether a student is competent in composite concept that involves three or more basic concepts. A "1" for a composite concept that covers three or more basic concepts just indicates that typical students of that group are competent of integrating the parent concepts of that composite concept. Whether typical students of that group are really competent in the composite concept depends on whether the students are competent in the required parent concepts. Take the type *g2* in Table 1 for instance. Although this type of students is competent in integrating the parent concepts of *ABC* with a high probability, the students will be competent in *ABC* with a low probability because they are not competent in *C* with a high probability. In contrast, we cannot tell whether students of *g2* are competent or incompetent in *ABD*. We need the *GC* matrix *and* the learning patterns for this task. Had we known that the parent concepts of *ABD* were *AD* and *B*, then we would be able to tell that students of *g2* would have a low probability of being competent in *ABD*. Had we known that the parent concepts of *ABD* were *AB* and *D*, then we would be able to tell that students of *g2* would have a high probability of being competent in *ABD*. The *GC* matrix alone does not provide complete information about whether a student is competent in *ABD*.

We employ a special node, the *group* node, to represent types of students in a Bayesian network. The *group* node can take on any value that represents a type of student in the study, i.e., *g1* or *g2*. Since students' competence in concepts must be influenced by their types, there is a link from the *group* node to every concept node. We allow a student to perform differently from the stereotypical behavior of the student's group, and control the probability of such a deviation by a simulation parameter, β, that is explained in the next subsection.

2.3 The Simulator

We employ the simulator that was invented for a previous work on student classification, so we will not explain the functions of the simulator in great detail. Interested readers are referred to (Liu, 2005).

The simulator takes as input a Bayesian network structure, a *GC* matrix, and two controlled parameters to produce item responses of a selected number of students. We must create the conditional probability tables (CPTs) for every node in the Bayesian network to make it functioning.

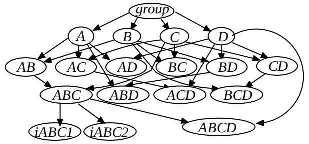

Figure 2. A partial network for the case of learning *ABCD* with

There are three types of nodes in our Bayesian networks: the node that represents types of students (called the *group* node, henceforth), the concept nodes, and the item nodes.

Figure 2 shows a network structure for representing a possible way of learning *ABCD* (i.e., `ABC~D`). The structure only shows the *group* node, the concept nodes, and two item nodes to maintain the clarity of the network. Not all of the item nodes are included, and the links from the *group* node to the composite nodes are not shown either. The structure of Figure 2 is employed in our experiments to examine the feasibility of the proposed method for model selection, and we do not have a specific interpretation for the concept nodes in the network.

We can simulate any prior distributions for the *group* node. At this moment, we assume that a student can belong to any of the types in the given *GC* matrix with equal probabilities. We employ the *noisy-and* models (Pearl, 1988) for the CPTs of the nodes for composite concepts. The CPTs for the item nodes and the concept nodes for basic concepts have only one parent node, so it is simple to create CPTs for these nodes based on the values of the controlled parameters that we explain next.

The first controlled parameter, α, confines the range of the probabilities of *guess* and *slip*. For instance, if we set α to 0.1, the probability of observing *guess* and *slip* will be between [0, 0.1]. The actual probability is chosen with a random number generator that selects a number from the uniform distribution between 0 and α. The second controlled parameter, β, controls the range of the probability of a student's performance deviating from the competence pattern of the type that s/he belongs to (Tatsuoka & Tatsuoka, 1997; Table 2). It works in ways that are similar to how the value of α affects the probability of observing *guess* and *slip*. For instance, if we set β to 0.2, a student may deviate from the standard competence pattern with a probability that is chosen uniformly from the range [0, 0.2]. Note that, although the values of α and β influence the item response pattern of a simulated student, α and β carry different semantics, and their co-existence allows us to control the uncertainty of different sources. The actual values in the CPTs of all of the concept nodes and the item nodes are computed based on the samples that are sampled independently from the specified ranges.

After constructing the CPTs for all of the nodes in the Bayesian network, we can compute the conditional probability of answering to a test item correctly given a student type, and use this information to simulate whether a student actually answer correctly or incorrectly with a simple Monte-Carlo method. For instance, assume that we evaluate the Bayesian network and find that $\Pr(iABC1 = correct \mid group = g2)$ is equal to 0.6. Since a student must respond to a test item either correctly or incorrectly in our current study, we sample uniformly a number from the range [0, 1], and determine that a particular student responds to *iABC1* correctly if this random number is less or equal to 0.6. If this random number is larger than 0.6, we will determine that this student fails to answer correctly. More importantly, we sample a new random number for each test item and for each simulated student, so the simulated results are independent among all of the test items and among all of the students.

3. Motivating Examples and Problem Complexity

Given a *GC* matrix, the values of α and β, and a network structure, we can simulate the item responses for a population of students. To examine the applicability of a machine-learning method,

Table 2. A *GC* matrix for the current experiments

group	*A*	*B*	*C*	*AB*	*BC*	*AC*	*ABC*	*group*	*A*	*B*	*C*	*AB*	*BC*	*AC*	*ABC*
g1	1	1	1	1	1	1	1	*g5*	1	1	1	1	0	0	1
g2	1	1	1	1	1	0	1	*g6*	1	1	1	0	1	0	1
g3	1	1	1	0	1	1	1	*g7*	1	1	1	0	0	1	1
g4	1	1	1	1	0	1	1	*g8*	1	1	1	0	0	0	1

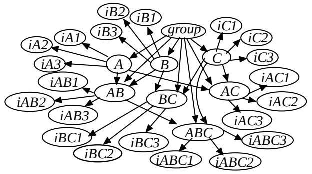

Figure 3. A complete network for the case of learning *ABC* with AB~C

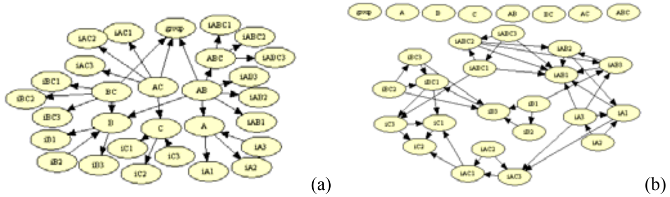

Figure 4. Two structures learned with the PC algorithm under different assumptions

we may then try to recover the network structure by using the simulated data as the input for an algorithm for structure learning.

In this example, we consider only three basic concepts (*cA*, *cB*, and *cC*) and four composite concepts (*dAB*, *dBC*, *dCA*, and *dABC*). Table 2 shows the *GC* matrix with which we generated the item responses for 10000 simulated students. In this illustrative example, we used the AB~C pattern (shown in Figure 3), and set both α and β to 0.1. With the item responses created under this setting, we applied the PC algorithm implemented in Hugin (http://www.hugin.com) to learn the structure of the original network. We manipulated the information available to the PC algorithm to emulate the situations of whether we have non-observable data in the input data to the algorithm. The first obvious choice is to provide complete information about all the nodes to the learning algorithm; and the second is to provide only the information that is truly observable. Namely, in the second alternative, we provided only the information about the item responses. In a normal situation, without human intervention or tagging, we do not know the states of the *group* node and the nodes for the competence, i.e., *AB*, *BC*, *AC* and *ABC*.

Figure 4 shows the network structures that the PC algorithm inferred from the test data. (The nodes were relocated for readability from the network that was generated by Hugin. We attempted to minimize the intersection of links and to move nodes of similar nature close to each other.) We obtained part (a), when we used the complete information, and part (b), when used only the observable information. When we ran the PC algorithm, we set the level of significance to 0.05, which is the default value in Hugin and was also adopted by Vomlel (2004).

Figure 4(a) is not very similar to the original network in Figure 3, which we used to create the simulated data. The nodes for test items are directly connected to their parent concepts, though in diverse directions. The PC algorithm chose to connect the nodes for the concept in ways that we cannot fully justify based on the intended meaning of those concept nodes. Nodes related to the concept *ABC* were just remotely related to nodes related to the concept *C*.

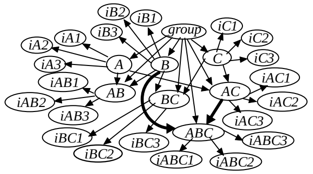

Figure 5. A complete network for the case of learning *ABC* with AC~B

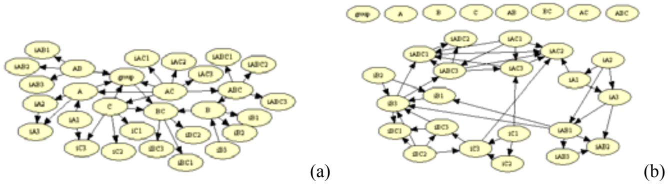

Figure 6. Two structures learned with the data generated with AC~B

In Figure 4(b), the nodes for test items were not connected to the nodes for the concepts because the information about the *group* node and the concept nodes are completely missing. A structure like part (b) is called item-to-item knowledge structures by (Desmarais et al., 2006). The structure showed close relationships between nodes for test items for *ABC* and *AB*, and there was a direct relationship between *iABC1* and *iC3*. However, it is hard to tell the relationships among *ABC*, *AB*, and *C* from this tem-to-item knowledge structure.

We repeated the same procedure for the learning pattern AC~B. Figure 5 shows the network structure which we used to create simulated data. Thicker links emphasize the change in the learning pattern this network structure suggests. We reused the *GC* matrix shown in Table 2, and set α and β to 0.1 again. Figure 6 shows the network structures that the PC algorithm learned when the information about all the nodes (part (a)) and when the information about the observable nodes were provided (part (b)).

Results of such pilot experiments showed that directly applying the machine learning methods may not help us rebuild the network structures very well. The problem is particularly challenging when we have a few completely unobservable nodes in the network. Completely unobservable nodes make it difficult to figure out the relationships among them as Figures 4(b) and 6(b) indicated.

Due to such an observation, we attempt to apply the machine learning methods to help the domain experts select the best network structure from a set of candidate structures. This approach is applicable when we have a set of candidate networks in mind, e.g., (Gierl et al., 2007; page 251), and would like to use real data to find the best network.

In the current study, we assume that students learn composite concepts from parent concepts that do not share any common basic concepts. Hence, for instance, *ABC* and *CD* cannot serve as the parent concepts of *ABCD* because they share the basic concept *C*. This assumption makes the space of possible solutions much smaller than it can be.

Therefore, the number of different ways to learn a composite concept depends on the number of the basic concepts that the composite concept contains, and is related to the Stirling number of type 2 (Knuth, 1973). Assume that we are studying the learning pattern for a composite concept that involves λ basic concepts. There are $\Omega(\lambda)$, shown in (3.1), ways to learn this composite concept (Liu, 2008). The value of $\Omega(\lambda)$ grows very quickly with λ. For example, when λ is equal to 3, 4, 5, and 6, $\Omega(\lambda)$ is equal to 4, 14, 51, and 202, respectively.

$$\Omega(\lambda) = \sum_{i=2}^{\lambda}\left(\frac{1}{i}\sum_{j=0}^{i-1}(-1)^{j}\binom{i}{j}(i-j)^{\lambda}\right) \tag{3.1}$$

The quantity prescribed in formula (3.1) is not yet the worst case scenario. When we consider the task of building a model that includes λ basic concepts, there can be $S(\lambda)$, given in formula (3.2), different models. Here, λ is not smaller than 3 because it does not make too much sense to discuss the learning patterns for only two basic concepts.

$$S(\lambda) = \prod_{k=3}^{\lambda}(\Omega(k))^{\binom{\lambda}{k}} \tag{3.2}$$

Finding the best model from this humongous amount of candidate models purely based on only computational requirements might not be very fruitful. This is suggested by the motivating examples that we showed at the beginning of this section. Allowing domain experts to provide a few meaningful candidate models and employing computational techniques to compare these candidates offer a chance for us to identify a good model for complex problems.

4. Model Selection

We discuss our methods for assisting teachers to select the learning pattern from a set of candidate answers in this section. Experimental results will be presented in the next section.

4.1 Using Mutual Information as Scores for Candidate Models

Consider the sample network shown in Figure 2 again. Let CI(X, Y, Z) denote the situation that X and Z are conditionally independent given Y, where X, Y, and Z are three sets of variables. If we have access to the exact information about the concept nodes, we may choose the learning pattern relatively more easily. In Figure 2, we should have CI({A, B, C}, {ABC, D}, {$ABCD$}). If the parent nodes of $ABCD$ were AB and CD in Figure 2, then we would have CI({A, B, C, D}, {AB, CD}, {$ABCD$}), and we will have precise criteria for judging the fitness of network structures when we have information about the concept nodes. In fact, however, we can collect only the item responses that are probabilistically related to the states of the concept nodes, so we do not have ways to determine whether CI({A, B, C, D}, {AB, CD}, {$ABCD$}) holds or not.

Observe the network structures shown in Figure 1 and Figure 2. We can see that the item nodes are linked directly and only to their parent nodes, so, intuitively, item nodes for concept nodes that are more closely related may be more closely related to each other than to item nodes for other concept nodes. For instance, if the actual learning pattern is `ABC~D`, as shown in Figure 2, then the correctness of answers to test items for $ABCD$ may be more related to the correctness of answers to the test items for ABC and D than to the correctness of answers to the test items for other concept nodes.

We embody the concept of "more related" with the mutual information between two sets of random variables (Cover & Thomas, 2006). Let X and Y denote two sets of random variables, Eq. (4.1.1) shows the mutual information between X and Y, where $\hat{X}$ and $\hat{Y}$ represent the sets of the possible values of X and Y, respectively. It is easy to prove that, if $MI(X;Y)>MI(Z;Y)$, then knowing the information about X will reduce more uncertainty about Y than knowing the information about Z. Specifically, we have $MI(X;Y)>MI(Z;Y) \Rightarrow H(Y|X)<H(Y|Z)$, where $H(\cdot|\cdot)$ represents the conditional entropy.

$$MI(X;Y) = \sum_{x\in\hat{X}}\sum_{y\in\hat{Y}}\Pr(X=x,Y=y)\log\frac{\Pr(X=x,Y=y)}{\Pr(X=x)\Pr(Y=y)} \tag{4.1.1}$$

Consider a more concrete example with the networks in Figure 1 and Figure 2. If $MI(A,B,C,D;ABCD)$ is larger than $MI(ABC,D;ABCD)$, we may believe that the learning pattern `A~B~C~D` is more likely to be the answer than `ABC~D`, and the network shown in Figure 1(b) is a better choice than the network shown in Figure 2.

Having chosen this measure, we face the problem of how we can compute the necessary mutual information. Recall that we do not know the actual states of the concept nodes, which are required in applying Eq. (4.1.1). This time, however, we can use the item responses for the concept nodes to estimate the states of the concept nodes, because we are just computing mutual information not judging conditional independence.

We note that using item responses to estimate the states of competence levels is not a perfect choice, though item responses are the most direct observation we have about student's competence levels. When students respond to a test item correctly, we are not really sure that they are competent of the related concepts. In addition, when students are competent in certain concepts, we are not sure how they will apply these concepts in tests. Hence, there is really a great deal of uncertainty between students' item responses and their competence levels as we have mentioned in this paper and elsewhere in the literature.

We assume that, in a certain examination, we use n test items to evaluate the competence levels of every concept being tested. We also assume that students will respond to every test items in the examination. Figures 3 and 5 show two such examples. In both cases, we have three test items for every concept, i.e., n=3 for all concepts. We also assume that every student will respond to each of these test items, without leaving any of the test times unanswered. Hence, the portion of correct responses to the test items for a concept must be one of {0, 1/n, 2/n, …, 1}, and we can use this portion as an indication of the competence level of the concept. (If different concepts have different numbers of test items, we can adapt this estimation procedure by considering this factor accordingly.) We can generalize this estimation method for a set of concept nodes, and Eq. (4.1.2) shows an example.

$$\Pr(A=1/n, BC=2/n) = \frac{\text{number of students who respond correctly to 1 test item for } A \text{ and 2 test items for } BC}{\text{number of students}} \tag{4.1.2}$$

To avoid the problem of zero probability values that are caused by rarely observed cases in the simulation, we add a very small amount to the counts of basic events to smooth the probability distributions. Currently this small number is 0.001.

After estimating the marginal and joint probability distributions of concept nodes, we can apply Eq. (4.1.1) to compute the mutual information of interest, and use the results as the scores of the learning patterns. When λ is 4, we will have to compute a score for each of the 14 learning patterns as dictated by formula (3.1).

4.2 Using Supervised Learning for Model Selection

Experimental results, which are provided in a later section, indicated that using mutual information as the scores for the learning patterns can be useful. However, there are situations when using mutual information alone could not help us achieve high performance. Recall that we do not have the actual probability distributions of the concept nodes, so we have to use the item responses to estimate these distributions. Consequently, when the relationship between the correctness of item responses and the competence of individual concepts is really uncertain, our estimation can become quite inaccurate. For instance, when the largest and the second largest scores for the learning patterns were quite close, e.g., the ratio was less than 1.2, it was quite common that the raw values of the mutual information did not lead us to the correct answer.

Hence we would like to introduce more features when we apply the machine learning techniques to find the best learning pattern. The previous experience suggests that collecting the ratio between each of the original mutual information and the largest original mutual information can be useful. In addition, the ratio between the largest score and the second largest score and the ratio between the largest score and the average score are also useful. Take the study for the learning pattern of *ABCD* as an example. In addition to the original 14 scores, we obtain 14 ratios between these original scores and the largest score. We also divide the largest score by the second largest score and divide the largest score by the average score to get two more features. Therefore we have 30 features in total. (Note that we did not manually set thresholds for these ratios. Mentioning the value of “1.2” in the preceding paragraph was meant to bring to readers’ attention the usefulness of ratios between raw features.)

In summary, when we have some candidate learning patterns in mind, we can generate simulated data with the learning patterns, a *GC* matrix, and the controlled parameters. Since the learning patterns are known when we generate the simulated data, the learning patterns can be used as the class labels in supervised learning (Witten & Frank, 2005). We can create training instances by associating the features with class labels to train a classifier, use the classifier to classify the item responses of real students, and use the classification result as the learning pattern of the real students. Based on this principle, we can apply support vector machines (SVMs) (Cortes & Vapnik, 1995), artificial neural networks (ANNs) (Bishop, 1995), or any other appropriate classification techniques in our experiments. Experimental results observed in some early explorations showed that we achieved results of similar quality when we used appropriate parameters for the right SVM or ANN models (Liu, 2008). In the next section, we choose to report results of using SVMs in our classifiers.

5. Experimental Studies

In the current study, we assume that all of the students learn a composite concept with the same learning pattern. This allows us to find the candidate pattern that has the highest score and to simplify the procedures of our experiments. If we will consider multiple learning patterns, say *k* learning patterns, for a composite concept in other experiments, we just have to choose those *k* candidate patterns with leading scores.

Because we did not collect students’ data in a real scenario, we use only simulated data in this study, and employ a Bayesian network to represent the true learning pattern. We try to find the learning pattern for *ABCD* in the experiments. We assume that there are professional sources that can provide us the list of candidate learning patterns, and we can represent each of these candidates with a corresponding Bayesian network. We assume that our professional advisors can provide perfect information about the *GC* matrix, which contains information about students’ competence patterns. With a Bayesian network, a *GC* matrix, and two controlled parameters, we can generate simulated data as explained in Section 2.3. After generating the training and test data with due procedures, we can conduct experiments and record the accuracy achieved by our classifiers.

5.1 Bayesian Networks for the Experiments

We conducted five sets of experiments. Each of these experiments uses one pattern in {A~BCD, AB~CD, A~BC~D, A~B~CD, A~B~C~D} as the true learning pattern. These patterns are selected to represent different ways of combinations of the parent concepts of *ABCD*. A~BCD and AB~CD represent two different ways to learn *ABCD* with two parent concepts. They are different because one parent concept in A~BCD has only one basic concept, and, in contrast, both parent concepts in AB~CD have two basic concepts. Using an intuitive interpretation of “similarity”, A~BC~D and A~B~CD are two similar ways to learn *ABCD* with three parent concepts. A~B~C~D represents the direct integration of four basic concepts into the composite concept.

Note that it can be difficult to tell the differences between the item responses of students who employ closely related learning patterns. A~BC~D and A~BCD are closely related because A~BC~D is refined form of A~BCD by splitting BCD into BC and D. Analogously, A~B~CD and A~BCD are closely related, and A~B~CD and AB~CD are closely related. Hence, we believe that we have chosen a challenging task for our tests.

Table 3. A *GC* matrix for the current experiments

group	*A*	*B*	*C*	*D*	*AB*	*AC*	*AD*	*BC*	*BD*	*CD*	*ABC*	*ABD*	*ACD*	*BCD*	*ABCD*
g1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
g2	1	1	1	1	1	0	0	0	0	1	1	1	1	0	1
g3	1	1	1	1	0	1	0	1	0	1	1	1	0	1	1
g4	1	1	1	1	1	0	0	0	0	1	1	1	0	0	1
g5	1	1	1	1	1	0	1	0	1	0	1	0	1	1	1
g6	1	1	1	1	1	0	0	0	0	1	1	0	1	0	1
g7	1	1	1	1	1	1	1	0	0	0	1	0	0	1	1
g8	1	1	1	1	1	0	0	0	0	1	1	0	0	0	1
g9	1	1	1	1	0	0	0	1	1	1	0	1	1	1	1
g10	1	1	1	1	1	0	0	0	0	1	0	1	1	0	1
g11	1	1	1	1	1	1	0	0	1	1	0	1	0	1	1
g12	1	1	1	1	1	0	0	0	0	1	0	1	0	0	1
g13	1	1	1	1	0	0	1	1	0	0	0	0	1	1	1
g14	1	1	1	1	1	0	0	0	0	1	0	0	1	0	1
g15	1	1	1	1	0	0	0	0	1	1	0	0	0	1	1
g16	1	1	1	1	1	0	0	0	0	1	0	0	0	0	1

In order to make our classifier have a chance to find the correct learning pattern, we assume that some professional teachers will provide {A~BCD, AB~CD, A~BC~D, A~B~CD, A~B~C~D} as the list of candidate solutions. Namely, we make sure that the list includes the true learning pattern, no matter which of the learning patterns that we discussed at the beginning of this subsection is used as the true learning pattern.

In all of our experiments, we assume that the parent concepts of *ABC*, *ABD*, *ACD*, and *BCD* are basic concepts. Other situations are discussed in the third paragraph in Section 6. Given this assumption and the learning patterns, we can construct the structures of the Bayesian networks that our simulator needs.

5.2 The GC matrices and Controlled Parameters

In addition to the network structures, our simulator needs a *GC* matrix that specifies students’ competence patterns. When we are using simulated data in the experiments, we need two *GC* matrices. One represents the competence patterns that we obtain from the professional sources, and the other represents students’ actual competence patterns. When the professional sources are highly experienced, we may be able to obtain an accurate *GC* matrix. Hence, we use only one *GC* matrix for generating the training and test data in our experiments for now.

Table 3 shows the *GC* matrix that we used in our experiments. We use the same format that we used in Table 1 to present a *GC* matrix in Table 3. This *GC* matrix was also used in experiments in which a different list of candidate patterns was used (Liu, 2008), so it is possible to compare the experimental results.

There are 16 types of students in Table 3, and we can examine the settings for different types of students. All of these 16 types of students are capable of using A~B~C~D as their learning pattern. When *i* is an odd number, students of types *gi* can use A~BCD. Students of types *gi*, *i* = 1, 2, 4, 6, 8, 10, 11, 12, 14, and 16, can use AB~CD; students of types *gj*, *j* = 1, 3, 9, and 13, can use A~BC~D; and students of types *gk*, *k* = 1, 2, 3, 4, 6, 8, 9, 10, 11, 12, 14, 15, and 16, can use A~B~CD.

We can also examine the settings for different concepts. All of these 16 types are capable of the basic concepts and are capable of integrating the parent concepts of *ABCD*. If we want to study how students learn *ABCD*, we should recruit students that appear to be competent in *ABCD* in our study, so the settings in the *A*, *B*, *C*, *D*, and *ABCD* columns should be reasonable. Treating *ABC*, *ABD*, *ACD*, and *BCD* as a group, we have 16 possible ways to set the values for this group, and we do that in Table 3, which is an important reason why we have 16 types of students in Table 3. There

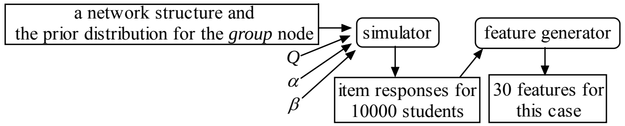

Figure 7. Generating an instance of datum for the experiments

are a total of 64 possible ways to set the capabilities of integrating concepts that include two basic concepts, but we have chosen 16 of them arbitrarily in Table 3.

In Section 2.3, we explained how we use the controlled parameters α and β to modulate the ranges of *slip* and *guess* and to manipulate the possibility of students' deviation from the standard competence patterns of their types. Both α and β can be set to any value from {0.05, 0.10, 0.15, 0.20, 0.25, 0.30}. We do not try values that are larger than 0.30 because those situations are really rare and were not discussed in the literature. As a result, for any pair of a network structure and a *GC* matrix, we repeat the experiment 36 (=6×6) times.

5.3 *Main Steps of the Experiments*

Figure 7 shows the flow that we employed to create an instance of datum for our experiments. Given a network structure along with the prior distribution for the *group* node, a *GC* matrix, and a particular combination of α and β, we used the simulator that we described in Section 2.3 to create item responses of 10000 students. We assumed that students responded to three test items for every concept in the experiments, but this quantity could be changed easily. With the item responses, we applied the methods that we described in Section 4.1 to estimate the mutual information and to compute the derived features.

In the experiments, we repeated this procedure 600 times for every possible solution in {`A~BCD`, `AB~CD`, `A~BC~D`, `A~B~CD`, `A~B~C~D`}, the *GC* matrix in Table 3, and all of the 36 possible combinations of α and β. The conditional probability tables were re-sampled for every one of these 600 instances, so their underlying probability distributions were mutually independent given the setups of the experiments. Hence, for any combination of α and β, we had 600 instances for a possible learning pattern. We could use 500 instances for every learning pattern to train our classifier, and used the remaining 100 instances as the test data. In total, we had 2500 training instances and 500 test instances when we ran an evaluation of our approach.

We employed LIBSVM (Chang & Lin, 2001) for realizing our classifier. We chose the SVMs of type c-SVC, and used the radial basis function as the kernel function. Since there were still free parameters in the SVMs, we had to run some explorative tests to search the best combination of the parameters C and γ in LIBSVM. In these explorative tests, we set C and γ to any value in {0.1, 0.2, …, 1.9}, so we had to run 361 explorative tests. We used the training data as the test data to search the combination of C and γ that helped us achieve the highest accuracy in these explorative tests. This particular combination of C and γ was then used in the evaluation that used the real test data.

5.4 *Preliminary Analysis*

It was possible for us to use the original 14 mutual information, which we discussed in Section 4.1, to guess the learning patterns. A simple procedure was just to select the learning pattern that corresponded to the largest mutual information, and it was easy to verify whether the procedure found the correct pattern because each instance of datum was labeled with the correct learning pattern. Since we had a total of 3000 instances for every combination of α and β, we could use the proportion of correct identification as the accuracy for this simple procedure.

Figure 8(a) shows the results of such experiments for all of the combinations of α and β. The vertical axis shows the accuracy, the horizontal axis shows the values of α, and the legend shows

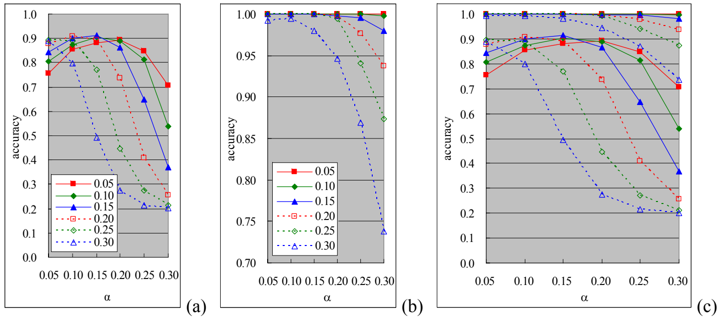

Figure 8. Domain-specific constraints improve the classification accuracy

the values of β. Although using only mutual information for determining the learning patterns can perform well in some cases, the trends of the curves show that the results can change radically with the varying α and β. When α and β are both 0.3, the accuracy is just 0.2.

Note that this result of 0.2 was not due to that the simple procedure randomly selected an answer from five possible answers. This simple procedure did not know that there were only five possible answers. It was allowed to guess any of the possible answers—14 possible answers when λ was 4. This phenomenon was due to the similarity among the competing concepts. For instance, when α was small, this simple procedure frequently chose `AD~BC` as the answer for test instances that had `A~BC~D` as their labels. A procedure that randomly selected an answer from 14 candidates would have led to the result of about 0.0714 in accuracy. The reason for the result of 0.2 was because, when α and β were large, the simple procedure inclined to select `A~B~C~D` as the learning pattern. This should not be very surprising because all of the concepts must be related to the basic concepts. Because `A~B~C~D` was used in 600 instances, we had about 600 correct answers out of 3000 test instances in the experiments, making the accuracy about 0.2.

5.5 *Experimental Results*

Figure 8(b) shows the results that we achieved by using the SVM-based classifiers to guess the learning patterns. The horizontal axis, the vertical axis, and the legend in Figure 8(b) carry the same meaning as their counterparts in Figure 8(a). Figure 8(c) shows all of the curves in Figures 8(a) and 8(b) in one chart to help us compare the experimental results.

The chart in Figure 8(b) indicates that the accuracy achieved by our classifier still depends on the values of α and β. The trends of the curves also suggest that the accuracy reduced as we increased the values of α and β. However, this relationship does not hold all the time, and we will see an example in the next subsection. There are other factors which influence the observed accuracy.

The chart in Figure 8(c) shows that we improved the accuracy a lot by using the domain-specific information to train a classifier and by using this classifier to judge the unobservable learning pattern from the item responses. The positions of curves that we copied from Figure 8(b) lie above their counterparts that we copied from Figure 8(a). In addition, we reduced the variation in accuracy by using the classifiers.

We also used the F measure (Witten & Frank, 2005) to gauge the quality of our classifiers, in addition to using the proportion of correct classification as the measure for the quality. Since we classified the test instances into one of five candidate answers and the true answer of every test instance belonged to the five candidates, we could calculate the precision and recall (Witten & Frank, 2005) for each of the five classes. To obtain the F measure for an experiment, we calculated the average precision based on the precision for the five classes, and calculated the average recall

Table 4. Another *GC mat*rix for the experiments

	A	*B*	*C*	*D*	*AB*	*AC*	*AD*	*BC*	*BD*	*CD*	*ABC*	*ABD*	*ACD*	*BCD*	*ABCD*
g1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
g2	1	1	1	1	**1**	**0**	**0**	**0**	**0**	**1**	1	1	1	0	1
g3	1	1	1	1	**0**	**0**	**1**	**1**	**0**	**0**	1	1	0	1	1
g4	1	1	1	1	**1**	**1**	**0**	**0**	**1**	**1**	1	1	0	0	1
g5	1	1	1	1	**0**	**0**	**0**	**0**	**0**	**0**	1	0	1	1	1
g6	1	1	1	1	**1**	**1**	**1**	**1**	**1**	**1**	1	0	1	0	1
g7	1	1	1	1	**0**	**0**	**1**	**0**	**0**	**0**	1	0	0	1	1
g8	1	1	1	1	**1**	**1**	**0**	**1**	**1**	**1**	1	0	0	0	1
g9	1	1	1	1	**0**	**0**	**0**	**1**	**0**	**0**	0	1	1	1	1
g10	1	1	1	1	**0**	**1**	**1**	**0**	**1**	**0**	0	1	1	0	1
g11	1	1	1	1	**1**	**0**	**1**	**0**	**0**	**1**	0	1	0	1	1
g12	1	1	1	1	**1**	**1**	**0**	**1**	**0**	**1**	0	1	0	0	1
g13	1	1	1	1	**0**	**0**	**0**	**0**	**1**	**0**	0	0	1	1	1
g14	1	1	1	1	**0**	**1**	**1**	**1**	**1**	**0**	0	0	1	0	1
g15	1	1	1	1	**1**	**0**	**1**	**1**	**0**	**1**	0	0	0	1	1
g16	1	1	1	1	**0**	**1**	**0**	**0**	**1**	**0**	0	0	0	0	1

analogously. We then weighed the average precision and the average recall equally when calculating the F measure. We divided the F measure by the accuracy of the same experiment to compare these two measures, and found that the ratio changed in a very narrow range. The largest one was 1.0052, and the smallest was 1.0000. Hence, the F measures were larger than the accuracy in our experiments, but the differences were really ignorable. Hence, we do not show the results here.

5.6 More Experimental Results

We have mentioned, in Section 3, that the contents of the *GC* matrix will influence the experimental results. In this subsection, we replace part of the contents of the *GC* matrix in Table 3, and repeat the same experiment to show the influence of using different *GC* matrices on the experimental results.

Table 4 shows a *GC* matrix that we created based on the *GC* matrix in Table 3. Note that we only changed the numbers that are in boldface in Table 4. We intentionally set the values in the *AB*, *AC*, *AD*, *BC*, *BD*, and *CD* columns so that the numbers of 1s in these columns are equal to 8 and that there were eight types of students who could use A~BCD, AB~CD, A~B~CD, and A~BC~D in this *GC* matrix. Moreover, the types of students that could and could not use AB~CD and A~B~CD were exactly the same.

We replaced the *GC* matrix in Table 3 by the new *GC* matrix, repeated the experiments, used the same list of candidate learning patterns, and conducted a preliminary analysis. Figure 9 contains three charts that were prepared with an analogous procedure for preparing their corresponding charts that are shown in Figure 8. We obtain Figure 9(a) in the preliminary analysis and Figure 9(b) in the experiments that used the trained SVM as the classifier. Figure 9(c) is the result of copying all of the curves in Figures 9(a) and 9(b) into one chart.

Like Figure 8(c), the positions of the corresponding curves in Figure 9(c) show that using the domain-specific information to train the classifier and limiting the possible answers within a selected list helped us to achieve higher accuracy in all of the experiments.

The positions of the corresponding curves in Figure 9(a) and Figure 8(a) show that it became more difficult to find students' learning patterns with just the values of mutual information. The general trends of the curves are similar for the corresponding curves in Figure 9(a) and Figure 8(a). In particular, increasing the value of α with a constant β does not necessarily make finding the learning patterns more difficult. It was found that, even when the values of α and β were small, the scores for AB~CD were still larger than the scores for A~B~CD when the true learning pattern was A~B~CD, and the scores for AD~BC were still larger than the scores for A~BC~D when the true

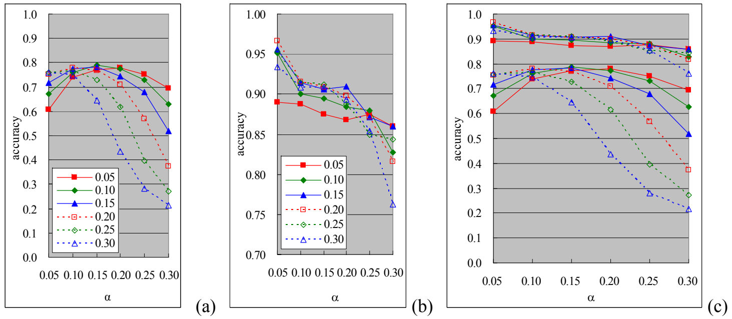

Figure 9. The contents of GC matrices influence the experimental results

learning pattern was A~BC~D. Our program may choose AD~BC as the answer because it was allowed to choose any of the 14 possible learning patterns as the answer in the preliminary analysis.

The positions of the corresponding curves in the charts in Figure 9(b) and Figure 8(b) show that it was also relatively more difficulty to find students' learning patterns with SVMs in the new experiment. The degradation in performance is clear even by visual inspection of these charts, and larger values of α and β do not imply poor performance this time. In addition to searching for *C* and γ in [0.1, 1.9] with the grid search that we described in Section 5.3, we expanded the search range for these parameters for SVMs. For the case in which α and β were both 0.05, we repeated the search twice. The first new search range for both *C* and γ was between 0.1 and 9.9 with a step of 0.2, and the second new search range was between 0.1 and 99 with a step of 2. Hence, we searched 5000 combinations of *C* and γ for the best results. However, we did not find any better results.

We investigated the output of our classifiers, and found two important sources of misclassification. When α and β were relatively large, it was very easy for our classifiers to classify all of the possible learning patterns into A~B~C~D. When α and β were relatively small, it was very easy for our classifiers to misclassify cases for AB~CD into A~B~CD and cases for A~B~CD into AB~CD. It seems that the new *GC* matrix have posted a detrimental challenge to our classifiers. Students who may apply AB~CD may also apply A~B~CD, and vice versa. Students who may not apply AB~CD may not apply A~B~CD, and vice versa. Namely, there seems no obvious way to differentiate AB~CD and A~B~CD in the settings for the new experiments, and this may have caused the difficulty for telling them apart when we had just indirect evidence about students' true competence levels for the related concepts.

6. Discussions

The results reported in the previous subsections clearly indicate that the range of deviations that were controlled by the values of α and β influence the accuracy that can be achieved by the simple method and by the SVM-based classifiers. The contents of the *GC* matrices that are used to generate the training and test data also affected the experimental results. When the *GC* matrix admitted multiple learning patterns in the candidate list, the proposed method would not be able tell these candidates apart very well, as the results reported in Section 5.6 have shown. When the *GC* matrix provided certain chances for distinguishing the learning patterns, the proposed method may offer a reasonable chance for finding the correct learning pattern even if the candidates were related as A~BCD, AB~CD, and A~B~CD. Similar results were observed when we used {A~B~CD, AB~C~D, ACD~B, ABD~C, A~B~C~D} as the candidate list and exactly the same *GC* matrix in the experiments (Liu, 2008). Although we have reported results of using the SVM-based classifiers in this paper, some previous experience showed that the ANN-based classifiers could attain results of similar quality, after we re-trained the ANN-based classifiers to find the best weights with the random restart method for classifying the training data (Liu, 2008).

As we can analyze in Section 3 and in (Liu, 2008), we are facing a myriad of different choices of the candidate networks and the *GC* matrices. The “right” choices of the candidate networks and the matrix should depend on domain dependent expertise, and the choices certainly influence the achieved results. Results reported in the previous subsections assumed that professional teachers can provide perfect information about the *GC* matrices and a candidate list that includes the actual learning pattern. If the candidate list does not include the actual learning pattern, any supervised learning method cannot help us find the right answer. If the *GC* matrix that we used to generate the simulated expectation does not match with the *GC* matrix of the real students, results observed in more experiments indicate that the results achieved by our classifiers will degrade. To put the proposed method in field tests, finding a good professional and reliable source for the competence patterns will help. If this is not possible, we may apply the clustering techniques to find the competence patterns from students’ records first.

The best way to apply the proposed method to learn network structure is to start from simpler sub-networks. If we directly try to find a network that considers a large number of basic concepts, we will have to face a very large search space that is dictated by formula (3.2). If we start from simpler networks and incrementally consider one more basic concept for some composite concepts, the search space can be reduced to what formula (3.1) prescribes. When λ is large, both (3.1) and (3.2) lead to large quantities, but the (3.1) leads to a smaller quantity. For this reason, we have chosen to illustrate our ideas with cases in which λ are either 3 or 4. This does not mean that we must confine the applications of the proposed methods with the same limits.

Using the average performance of students’ item responses to estimate students’ competence in the concepts can cause problems. If there is a small group of students who apply a special way to learn a composite concept, it may be difficult to find their behavior in the statistics, and, as a consequence, it become difficult to find how they learn the composite concepts with the proposed method.

Finally, simulated results may not provide sufficient support for us to adopt a technology in real world problems. Hence, although the observed results show the promise of the proposed method, a field test of the proposed method is in demand.

7. Concluding Remarks

Chickering et al. (2004) show that learning the structures of Bayesian networks is NP-hard, and discuss a list of relevant work. More recently, researchers have proposed a variety of methods for searching the space of possible solutions for the best model. The search heuristics may consider domain-specific scores, constraints, or both of them, e.g., (Guo & Schuurmans, 2006; Teyssier & Koller, 2005; Tsamardinos et al., 2006), and some of the reported methods need to limit the number of nodes in the candidate networks, e.g., less than 33 nodes as stated in (Silander & Myllymäki, 2006). The method proposed in this paper aims at helping domain experts to select candidate models. It requires relatively more specific domain-dependent information than the previously proposed methods did, not just constraints as in (Vomlel, 2004). The reported work is also special in how we apply the domain-specific information to compare the candidate models.

We have seen an example of using the simulated expectation about the data to select the best model from a list of candidate models for the data, in which we create the simulated expectation based on the domain-specific expertise. We illustrate the main idea with the problem of using students’ item responses to learn the hidden structures that consist of unobservable variables in Bayesian networks. The hidden structures reflect how students learn the composite concepts. We can estimate the competence of students only indirectly through students’ item responses that relate to students’ competence levels probabilistically. We must admit that the aforementioned experiments and simulations need to be strengthened by expertise of educational psychology and assessments, and we have just shown a technical possibility of learning the hidden structures.

Experimental results show that, if generated with well guided information, the simulated expectation can help us to confine the scope of possible solutions within a huge solution space and to discriminate competing models that can be difficult to tell apart otherwise. Given the humongous search space prescribed in formulas (3.1) and (3.2), we consider that the experimental results are not as pessimistic as they might appear. The machine learning method provides a much better quality of decisions than the heuristics, and achieves reasonably good results when both α and β are smaller than 0.25.

Example problems used in this paper can be improved by incorporating realistic consideration of cognition and learning theories for assessment problems. We employed the examples just to show that using simulated expectation can be useful for differentiating models that have subtle differences when we have only indirect observation about the subjects. Whether students hold a stable model (or procedure) for a task (Ben-Zeev & Ronald, 2002) and how the learning models might help instruction (Sleeman, 1989) require more serious interdisciplinary studies for decisive answers (Nichols et al, 1995; Tatsuoka & Tatsuoka, 1997; Leighton & Gierl, 2007).

Acknowledgements

The author owes a lot to the anonymous reviewers for their comments on previously published papers about this work. This research was supported in part by the grants NSC-94-2213-E-004-008 and NSC-95-2213-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

Ben-Zeev, T. & Ronald, J. (2002). Is procedure acquisition as unstable as it seems?, *Contemporary Educational Psychology*, **27**(4), 529−550.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford: 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. software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Chickering, D. M., Heckerman, D., & Meek, C. (2004). Large-sample learning of Bayesian networks is NP-hard, *Journal of Machine Learning Research*, **5**(Oct), 1287−1330.

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 edition). New Jersey: John Wiley & Sons.

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.

Gierl, M. J., Leighton, J. P., & Hunka S. M. (2007). Using the attribute hierarchy method to make diagnostic inferences about examinees’ cognitive skills, In (Leighton & Gierl, 2007), 242−274.

Glymour, C. & Cooper, G. F. (Eds.) (1999). *Computation, Causation, and Discovery*. California: AAAI Press.

Guo, Y. & Schuurmans, D. (2006). Convex structure learning for Bayesian networks: Polynomial feature selection and approximate ordering, *Proceedings of the Twenty-Second Annual Conference on Uncertainty in Artificial Intelligence*.

Heckerman, D. (1999). A tutorial on learning with Bayesian networks, In (Jordan, 1999), 301−354.

Jensen F. V. & Nielsen, T. D. (2007). *Bayesian Networks and Decision Graphs*. New York: Springer-Verlag.

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

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

Leighton, J. P. & Gierl, M. J. (2007). *Cognitive Diagnostic Assessment for Education*. Cambridge: 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. (2008). 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.

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

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.

Neapolitan, R. E. (2003). *Learning Bayesian Networks*. New Jersey: Prentice Hall.

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

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

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

Silander, T. & Myllymäki, P. (2006). A simple approach for finding the globally optimal Bayesian network structure, *Proceedings of the Twenty-Second Annual Conference on Uncertainty in Artificial Intelligence*, 445−452.

Sleeman, D., Kelly, A. E., Martinak, R., Ward, R. D., & Moore, J. L. (1989). Studies of diagnosis and remediation with high school algebra students, *Cognitive Science*, **13**(4), 551−568.

Teyssier, M. & Koller, D. (2005). Ordering-based search: A simple and effective algorithm for learning Bayesian networks, *Proceedings of the Twenty-First Annual Conference on Uncertainty in Artificial Intelligence*, 584−590.

Tsamardinos, I., Brown, L. E., & Aliferis, C. F. (2006). The max-min hill-climbing Bayesian network structure learning algorithm, *Machine Learning*, **65**(1), 31−78.

Tatsuoka, K. K. (1983). Rule space: An approach for dealing with misconceptions based on item response theory, *Journal of Educational Measurement*, **20**(4), 345−354.

Tatsuoka, K. K. (1995). Architecture of knowledge structures and cognitive diagnosis: A statistical pattern recognition and classification approach, In (Nichols et al, 1995), 327−360.

Tatsuoka, K. K. & Tatsuoka, M. M. (1997). Computerized cognitive diagnostic adaptive testing: Effect on remedial instruction as empirical validation, *Journal of Educational Measurement*, **34**(1), 3−20.

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

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.

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

Chao-Lin Liu, Kan-Wen Tien, Min-Hua Lai, Yi-Hsuan Chuang, Shih-Hung Wu. Phonological and logographic influences on errors in written Chinese words, *Proceedings of the Seventh Workshop on Asian Language Resources*, the Forty Seventh Annual Meeting of the Association for Computational Linguistics (**ACL'09**), to appear. Singapore, 2-7 August 2009.

Phonological and Logographic Influences on Errors in Written Chinese Words

Chao-Lin Liu[1] Kan-Wen Tien[1] Min-Hua Lai[1] Yi-Hsuan Chuang[1] Shih-Hung Wu[2]
[1]National Chengchi University, [2]Chaoyang University of Technology, Taiwan
{chaolin,96753027,95753023,94703036}@nccu.edu.tw, shwu@cyut.edu.tw

Abstract

We analyze a collection of 3208 reported errors of Chinese words. Among these errors, 7.2% involved rarely used character, and 98.4% were assigned common classifications of their causes by human subjects. In particular, 80% of the errors observed in the writings of middle school students were related to the pronunciations and 30% were related to the logographs of the words. We conducted experiments that shed light on using the Web-based statistics to correct the errors, and we designed a software environment for preparing test items that need incorrect replacements of certain words. Experimental results show that using Web-based statistics can help us correct only about 75% of these errors. In contrast, Web-based statistics are useful for recommending incorrect characters for composing test items for "incorrect character identification" tests about 93% of the time.

1 Introduction

Incorrect writings in Chinese are related to our understanding of the cognitive process of reading Chinese (e.g., Leck et al., 1995), to our understanding of why people produce incorrect characters and our offering corresponding remedies (e.g., Law et al., 2005), and to building an environment for assisting the preparation of test items for assessing students' knowledge of Chinese characters (e.g., Liu and Lin, 2008).

Chinese characters are composed of smaller parts that can carry phonological and/or semantic information. A Chinese word is formed by Chinese characters. For example, 新加坡 (Singapore) is a word that contains three Chinese characters. The left (土) and the right (皮) part of 坡, respectively, carry semantic and phonological information. The semantic information, in turn, is often related to the logographs that form the Chinese characters. Evidences show that production of incorrect characters are related to phonological, logographic, or the semantic aspect of the characters. Although the logographs of Chinese characters can be related to the lexical semantics, not all errors that are related to semantics were caused by the similarity in logographs. Some were due to the context of the words and/or permissible interpretations.

In this study, we investigate issues that are related to the phonological and logographical influences on the occurrences of incorrect characters in Chinese words. In Section 2, we present the details about the sources of the reported errors. We have collected errors from a published book and from a group of middle school students. In Section 3, we analyze the causes of the observed errors. Human subjects were asked to label whether the observed errors were related to the phonological or the logographic reasons. In Section 4, we explore the effectiveness of relying on Web-based statistics to correct the errors. We submitted an incorrect word and a correct word separately to Google to find the number web pages that contained these words. The correct and incorrect word differed in just one incorrect character. We examine whether the number of web pages that contained the words can help us find the correct way of writing. In Section 5, we employ Web-based statistics in the process of assisting teachers to prepare test items for assessing students' knowledge of Chinese characters. Experimental results showed that our method outperformed the one reported in (Liu and Lin, 2008), and captured the incorrect characters better than 93% of the time.

2 Data Sources

We obtained data from three major sources. A list that contains 5401 characters that have been believed to be sufficient for everyday lives was obtained from the Ministry of Education (MOE) of Taiwan, and we call the first list the **Clist**, henceforth. We have two lists of word pairs, each including both a correct and an incorrect way to write certain words. The first list is from a book published by MOE (1996). The MOE provided the correct words and specified the incorrect characters which were mistakenly used to replace the correct characters in the correct words. The second list was collected, in 2008, from the written essays of students of the seventh and the eighth grades in a middle school in Taipei. The incorrect words were entered into computers based on students' writings, ignoring those characters that did not actually exist and could not be entered.

We will call the first list of word pairs the **Elist**, and the second the **Jlist** from now on. Elist and Jlist contain, respectively, 1490 and 1718 entries. Each of these entries contains a correct word and the incorrect character. Hence, we can reconstruct the incorrect words easily. Two or more different ways to incorrectly write the same words were listed in different entries and considered as if they were different for simplicity of presentation.

3 Error Analysis of Written Words

Two subjects, who are native speakers of Chinese and are graduate students in Computer Science, examined Elist and Jlist and categorized the causes of errors. They compared the incorrect characters with the correct characters to determine whether the errors were **pronunciation-related** or logographs-related. Referring to an error as being "semantics-related" is ambiguous. Two characters might not contain the same semantic part, but are still semantically related. In this study, we have not considered this factor. For this reason we refer to the errors that are related to the sharing of logographic parts in characters as **composition-related**.

Among the 1490 and 1718 words in Elist and Jlist, respectively, the two human subjects had consensus over causes of 1441 and 1583 errors. It is interesting to learn that native speakers had a high consensus about the causes for the observed errors, but they did not always agree. Hence, we studied the errors that the two subjects had agreed categorizations.

The statistics changed when we disregarded errors that involved characters not included in Clist. An error would be ignored if the correct or the incorrect character did not belong to the Clist. It is possible for students to write a rare character in an incorrect word just by coincidence.

After ignoring the rare characters, there were 1333 and 1645 words in Elist and Jlist, respectively. The subjects had consensus over the causes of errors for 1285 and 1515 errors in Elist and Jlist, respectively.

Table 1. Error analysis for Elist and Jlist

	C	*P*	*C&P*	*NE*	*D*
Elist	66.09%	67.21%	37.13%	0.23%	3.60%
Jlist	30.70%	**79.88%**	20.91%	2.43%	7.90%

Table 1 shows the percentages of five categories of errors: ***C*** for those composition-related errors, ***P*** for those pronunciation-related errors, ***C&P*** for the intersection of *C* and *P*, ***NE*** for those errors that belonged to neither *C* nor *P*, and ***D*** for those errors that the subjects disagreed on the error categories. There were, respectively, 505 composition-related and 1314 pronunciation-related errors in the Jlist, so we see 505/1645=30.70% and 1314/1645=79.88% in the table. Notice that *C&P* represents the intersection of *C* and *P*, so we have to deduct *C&P* from the sum of *C*, *P*, *NE*, and *D* to find the total probability, namely 1.

It is worthwhile to discuss the implication of the statistics in Table 1. For the Jlist, similarity between pronunciations accounted for nearly 80% of the errors, and the ratio for the errors that are related to compositions and pronunciations is 1:2.6. In contrast, for the Elist, the corresponding ratio is almost 1:1. The Jlist and Elist differed significantly in the ratios of the error types. It was assumed that the dominance of pronunciation-related errors in electronic documents was a result of the popularity of entering Chinese with pronunciation-based methods. The ratio for the Jlist challenges this popular belief, and indicates that even though the errors occurred during a writing process, rather than typing on computers, students still produced more pronunciation-related errors than composition-related errors. Distribution over error types is not as related to input method as one may have believed. Nevertheless, the observation might still be a result of students being so used to entering Chinese text with pronunciation-based method that the organization of their mental lexicons is also pronunciation related. The ratio for the Elist suggests that editors of the MOE book may have chosen the examples with a special viewpoint in their minds – balancing pronunciation and composition related errors.

4 Reliability of Web-based Statistics

In this section, we examine the effectiveness of using Web-based statistics to differentiate correct and incorrect characters. The abundant text material on the Internet gives people to treat the Web as a corpus (e.g., webascorpus.org). When we send a query to Google, we will be informed of the number of pages (**NOPs**) that possibly contain relevant information. If we put the query terms in quotation marks, we will find the number of pages that literally contain the query terms.

Hence, it is possible for us to compare the NOPs for two competing phrases for guessing the correct way of writing. At the time of this writing, Google found 107000 and 3220 pages, respectively, for "strong tea" and "powerful tea". (When conducting such advanced searches with Google, the quotation marks are needed to ensure the adjacency of individual words.) Hence, "strong" appears to be a better choice to go with "tea". How does this strategy serve for learners of Chinese?

4.1 Field Tests

We verified this strategy by sending the words in Elist and Jlist to Google to find the NOPs. We can retrieve the NOPs from the documents returned by Google, and compare the NOPs for the correct and the incorrect words to evaluate the strategy. Again, we focused on those common words that the human subjects had consensus about their error types. Recall that we have 1285 and 1515 such words in Elist and Jlist, respectively. As the information available on the Web changes all the time, we also have to note that our experiments were conducted during the first half of March 2009. The queries were submitted at reasonable time intervals to avoid Google's treating our programs as malicious attackers.

Table 2 shows the results of our investigation. We considered that we had a correct result when we found that the NOP for the correct word was larger than the NOP for the incorrect word. If the NOPs were equal, we recorded an ambiguous result; and when the NOP for the incorrect word was larger, we recorded an incorrect event. We use '**C**', '**A**', and '**I**' to denote "correct", "ambiguous", and "incorrect" events in Table 2.

The column headings of Table 2 show the setting of the advanced searches with Google and the set of words that were used in the experiments. We asked Google to look for information from web pages that were encoded in traditional Chinese (denoted **Trad**). We could add another restriction on the source of information by asking Google to inspect web pages from machines in Taiwan (denoted **Twn+Trad**). We were not sure how Google determined the languages and locations of the information sources, but chose to trust Google. The headings "**Comp**" and "**Pron**" indicate whether the words whose error types were composition and pronunciation-related, respectively.

Table 2. Reliability of Web-based statistics

		Trad		Twn+Trad	
		Comp	Pron	Comp	Pron
Elist	C	**73.12%**	73.80%	69.92%	68.72%
	A	**4.58%**	3.76%	3.83%	3.76%
	I	**22.30%**	22.44%	26.25%	27.52%
Jlist	C	76.59%	74.98%	69.34%	65.87%
	A	2.26%	3.97%	2.47%	5.01%
	I	21.15%	21.05%	28.19%	29.12%

Table 2 shows eight distributions, providing experimental results that we observed under different settings. The distribution printed in bold face showed that, when we gathered information from sources that were encoded in traditional Chinese, we found the correct words 73.12% of the time for words whose error types were related to composition in Elist. Under the same experimental setting, we could not judge the correct word 4.58% of the time, and would have chosen an incorrect word 22.30% of the time.

Statistics in Table 2 indicate that web statistics is not a very reliable factor to judge the correct words. The average of the eight numbers in the 'C' rows is only 71.54% and the best sample is 76.59%, suggesting that we did not find the correct words frequently. We would made incorrect judgments 24.75% of the time. The statistics also show that it is almost equally difficult to find correct words for errors that are composition and pronunciation related. In addition, the statistics reveal that choosing more features in the advanced search affected the final results. Using "Trad" offered better results in our experiments than using "Twn+Trad". This observation may arouse a perhaps controversial argument. Although Taiwan has proclaimed to be the major region to use traditional Chinese, their web pages might not have used as accurate Chinese as web pages resided in other regions.

4.2 An Error Analysis for the Field Tests

We have analyzed the reasons for why using Web-based statistics did not always find the correct words. Frequencies might not have been a good factor to determine the correctness of Chinese. However, the myriad amount of data on the Web should have provided a better performance.

The most common reason is that some of the words are really confusing such that the majority of the Web pages actually used the incorrect words. Some of errors were so popular that even one of the Chinese input methods on Windows XP offered wrong words as possible choices, e.g., "雄赳赳" (the correct one) vs. "雄糾糾". It is interesting to note that people may intentionally used incorrect words in some occasions, for instance, people may choose to write homophones in advertisements.

Another popular reason is that whether a word is correct depends on its context. For instance, "小斯" is more popular than "小廝" because the former is a popular nickname. Unless we had provided more contextual information about the queried words, checking only the NOPs of "小斯" and "小廝" led us to choose "小斯", which happened to be an incorrect word when we meant to find the right way to write "小廝". Another difficult pair of words to distinguish is "紀錄" and "記錄".

Yet another reason for having a large NOP of the incorrect words was due to errors in segmenting Chinese character strings. Consider a correct character string "WXYZ", where "WX" and "YZ" are two correct words. It is possible that "XY" happens to be an incorrect way to write a correct word. This is the case for having the counts for "花海繽紛" to contribute to the count for "海繽" which is an incorrect form of "海濱".

5 Facilitating Test Item Authoring

Incorrect character correction is a very popular type of test in Taiwan. There are simple test items for young children, and there are very challenging test items for the competitions among adults. Finding an attractive incorrect character to replace a correct character to form a test item is a key step in authoring test items.

Our analysis of the errors listed in the MOE book and the errors produced by real students led us to building a software environment for assisting the authoring of test items for incorrect character correction. It should be easy to find a lexicon that contains pronunciation information about Chinese characters. In contrast, it might not be easy to find visually similar Chinese characters with computational methods. Liu and Lin (2008) expanded the original Cangjie codes (**OCC**) (Chu, 2009), and employed the expanded Cangjie codes (**ECC**) to find visually similar characters.

With a lexicon, we can find characters that can be pronounced in a particular way. However, this is not enough for our goal. We observed that there were different symptoms when people used incorrect characters that are related to their pronunciations. They may use characters that could be pronounced exactly the same as the correct characters. They may also use characters that have the same pronunciation and different tones with the correct character. Although relatively infrequently, people may use characters whose pronunciations are similar to but different from the pronunciation of the correct character.

As Liu and Lin (2008) reported, replacing OCCs with ECCs to find visually similar characters could increase the chances to find similar characters. Yet, it was not clear as to which components of a character should use ECC.

5.1 Formalizing the Extended Cangjie Codes

We analyzed the OCCs for all the words in Clist to determine the list of basic components. We treated a basic Cangjie symbol as if it was a word, and computed the number of occurrences of n-grams based on the OCCs of the words in Clist. Since the OCC for a character contains at most five symbols, the longest n-grams are 5-grams. Because the reason to use ECCs was to find common components in characters, we disregarded n-grams that repeated no more than three times. After obtaining this initial list of n-grams, we removed those n-grams that were substrings of longer n-grams in the list.

In addition, the n-grams that appeared more than three times might not represent an actual part in any Chinese characters. This may happen by chance because we considered only frequencies of n-grams when we generated the initial list at the previous step. Hence, we manually examined all of the n-grams in the initial list, and removed such n-grams from the list.

In addition to considering the frequencies of n-grams formed by basic Cangjie codes to determine the list of components, we also took advantage of radicals that are used to categorize Chinese characters in typical printed dictionaries. Radicals that are standalone Chinese words were included in our list.

After selecting the list of basic components with the above procedure, we encoded the words in Elist with these basic components. We adopted the 12 ways that Liu and Lin (2008) employed to decompose Chinese characters. There are other methods for decomposing Chinese characters into components. Juang et al. (2005) and their team at the Sinica Academia propose 13 different ways for decomposing characters.

At the same time when we annotated individual characters with their ECCs, we may revise the list of basic components. If a character that actually contained a "common" part and that part had not been included in the list of basic component, we would add this part into the list to make it a basic component and revised the ECC for all characters accordingly. The judgment of being "common" is subjective, but we still maintained the rule that such common parts must appear in more than three characters. When defining the basic components, not all judgments are completely objectively yet, and this is also the case of defining the original Cangjie codes. We tried to be as systematic as possible, but intuition sometimes stepped in.

We repeated the procedure described in the preceding paragraph five times to make sure that we were satisfied with the ECCs for all of the 5401 characters. The current list contains 794 components, and we can revise the list of basic components in our work whenever necessary.

5.2 Recommending Incorrect Alternatives

With the pronunciation of Chinese characters in a dictionary and with our ECC encodings for words in the Elist, we can create lists of candidate characters for replacing a specific correct character in a given word to create a test item for incorrect character correction.

There are multiple strategies to create the candidate lists. We may propose the candidate characters because their pronunciations have the **s**ame **s**ound and the **s**ame **t**one with those of the correct character (denoted ***SSST***). Characters that have **s**ame **s**ounds and **d**ifferent **t**ones (***SSDT***), characters that have si**m**ilar **s**ounds and **s**ame **t**ones (***MSST***), and characters that have si**m**ilar **s**ounds and **d**ifferent **t**ones (***MSDT***) can be considered as candidates as well. It is easy to judge whether two Chinese characters have the same tone. In contrast, it is not trivial to define "similar" sound. We adopted the list of similar sounds that was provided by a psycholinguistic researcher (anonymity for submission) at the Sinica Academia.

In addition, we may propose characters that look similar to the correct character. Two characters may look similar for two reasons. They may contain the same components, or they contain the same **r**adical and have the same total number of **s**trokes (***RS***). When two characters contain the same component, the shared component might or might not locate at the same position within the bounding boxes of characters.

In an authoring tool, we could recommend a limited number of candidate characters for replacing the correct character. We tried two different strategies to compare and choose the visually similar characters. The first strategy (denoted ***SC1***) gave a higher score to the shared component that located at the same location in the two characters being compared. The second strategy (***SC2***) gave the same score to any shared component even if the component did not reside at the same location in the characters, e.g., 其 in 斯 and 廝 that we mentioned in Section 4.2. When there were more than 20 characters that receive nonzero scores, we chose to select at most 20 characters that had leading scores as the list of recommended characters.

We had to set a bound on the number of candidate characters, i.e., 20. As we mentioned in Section 4.1, a frequent continual submission of queries to Google will make Google treat our programs as malicious processes. Without the bound, it is possible to offer a very long list of candidates. On the other hand, it is also possible that our program does not find any visually similar characters for some special characters, and this is considered a possible phenomenon.

5.3 Evaluating the Recommendations

We examined the usefulness of these seven categories of candidates with errors in Elist and Jlist. The first set of evaluation (the inclusion tests) checked whether the lists of recommended characters contained the incorrect character in our records. The second set of evaluation (the ranking tests) was designed for practical application in computer assisted item generation. Only for those words whose actual incorrect characters were included in the recommended list, we replaced the correct characters in the words with the candidate incorrect characters, submitted the incorrect words to Google, and ordered the candidate characters based on their NOPs. We then recorded the ranks of the incorrect characters among all recommended characters.

Since the same character may appear simultaneously in *SC1*, *SC2*, and *RS*, we computed the union of these three sets, and checked whether the incorrect characters were in the union. The inclusion rate is listed under ***Comp***, representing the inclusion rate when we consider only logographic influences. Similarly, we computed the union for *SSST*, *SSDT*, *MSST*, and *MSDT*, checked whether the incorrect characters were in the union, and recorded the inclusion rate under ***Pron***, representing the inclusion rate when we consider only phonological influences. Finally, we computed the union of the lists created by the seven strategies, and recorded the inclusion rate under ***Both***.

The second and the third rows of Table 3 show the results of the inclusion tests. The data show the percentage of the incorrect characters being included in the lists that were recommended by the seven strategies. Notice that the percentages were calculated with different denominators. The number of composition-related errors was used for *SC1*, *SC2*, *RS*, and *Comp* (e.g., 505 that we mentioned in Section 3 for Jlist); the number of pronunciation-related errors for *SSST*, *SSDT*, *MSST*, *MSDT*, and *Pron* (e.g., 1314 mentioned in Section 3 for the Jlist); the number of either of these two types of errors for *Both* (e.g., 1475 for Jlist).

Table 3. Incorrect characters were contained and ranked high in the recommended lists

	SC1	*SC2*	*RS*	*SSST*	*SSDT*	*MSST*	*MSDT*	*Comp*	*Pron*	*Both*
Elist	73.92%	76.08%	4.08%	91.64%	18.39%	3.01%	1.67%	81.97%	99.00%	93.37%
Jlist	67.52%	74.65%	6.14%	92.16%	20.24%	4.19%	3.58%	77.62%	99.32%	97.29%
Elist	3.25	2.91	1.89	2.30	1.85	2.00	1.58			
Jlist	2.82	2.64	2.19	3.72	2.24	2.77	1.16			
Elist	19.27	17.39	11.34	19.13	8.29	19.02	9.15			
Jlist	17.58	16.24	12.52	22.85	9.75	22.11	7.68			

The results recorded in Table 3 show that we were able to find the incorrect character quite effectively, achieving better than 93% for both Elist and Jlist. The statistics also show that it is easier to find incorrect characters that were used for pronunciation-related problems. Most of the pronunciation-related problems were misuses of homophones. Unexpected confusions, e.g., those related to pronunciations in Chinese dialects, were the main reason for the failure to capture the pronunciation-related errors. *SSDT* is a crucial complement to *SSST*. There is still room to improve our methods to find confusing characters based on their compositions. We inspected the list generated by *SC1* and *SC2*, and found that, although *SC2* outperformed SC1 on the inclusion rate, *SC1* and *SC2* actually generated complementary lists and should be used together. The inclusion rate achieved by the *RS* strategy was surprisingly high.

The fourth and the fifth rows of Table 3 show the effectiveness of relying on Google to rank the candidate characters for recommending an incorrect character. The rows show the average ranks of the included cases. The statistics show that, with the help of Google, we were able to put the incorrect character on top of the recommended list when the incorrect character was included. This allows us to build an environment for assisting human teachers to efficiently prepare test items for incorrect character identification.

The sixth and the seventh rows show the average number of candidate characters proposed by different methods. Statistics shown between the second and the fifth rows are related to the recall rates (cf. Manning and Schütz, 1999) achieved by our system. For these four rows, we calculated how well the recommended lists contained the reported errors and how the actual incorrect characters ranked in the recommended lists. The sixth and the seventh rows showed the costs for these achievements, measured by the number of recommended characters. The sum of the sixth and the seventh rows, i.e., 103.59 and 108.75, are, respectively, the average numbers of candidate characters that our system recommended as possible errors recorded in Elist and Jlist.

There are two ways to interpret the statistics shown in the sixth and the seventh rows. Comparing the corresponding numbers on the fourth and the sixth rows, e.g., 19.27 and 3.25, show the effectiveness of using the NOPs to rank the candidate characters. The ranks of the actual errors were placed at very high places, considering the number of the originally recommended lists. The other way to use the statistics in the sixth and the seventh rows is to compute the average precision. For instance, we recommended an average 19.13 characters in *SSST* to achieve the 91.64 inclusion rate. The recall rate is very high, but the averaged precision is very low. This, however, is not a very convincing interpretation of the results. Having assumed that there was only one best candidate as in our experiments, it was hard to achieve high precision rates. The recall rates are more important than the precision rates, particularly when we have proved that the actual errors were ranked among the top five alternatives.

Notice that the numbers do not directly show the actual number of queries that we had to submit to Google to receive the NOPs for ranking the characters. Because the lists might contain the same characters, the sum of the rows showed just the maximum number of queries that we submitted. Nevertheless, they still served as good estimations, and it is fair to say that we had submitted 103.59×1441(=149273) and 108.75×1583 (=172151) queries to Google for Elist and Jlist in experiments from which we obtained the data shown in the fourth and the fifth rows. These quantities explained why we had to be cautious about how we submitted queries to Google. When we run our program for just a limited number of characters, the problems caused by intensive queries should not be very serious.

5.4 Discussions

Dividing characters into subareas proved to be crucial in our and Liu and Lin's experiments (Liu and Lin, 2008), but this strategy is not perfect, and could not solve all of the problems. The way we divided Chinese characters into subareas like (Juang et al., 2005; Liu and Lin, 2008) sometimes contributed to the failure of our current implementation to capture all of the errors that were related to the composition of the words. The most eminent reason is that how we divide characters into areas. Liu and Lin (2008) followed the division of Cangjie (Chu, 2009), and Juang et al. (2005) proposed an addition way to split the characters.

The best divisions of characters appear to depend on the purpose of the applications. Recall that each part of the character is represented by a string of Cangjie codes in ECCs. The separation of Cangjie codes in ECCs was instrumental to find the similarity of 苗 and 福 because "田" is a standalone subpart in both 苗 and 福. The Cangjie system has a set of special rules to divide Chinese characters (Chu, 2009; Lee, 2008). Take 副 and 福 for example. The component 畐 is recorded as an standalone part in 副, but is divided into two parts in 福. Hence, 畐 is stored as one string, "一口田", in 副 and as two strings, "一口" and "田", in 福. The different ways of saving 畐 in two different words made it harder to find the similarity between 副 and 福. An operation of concatenation is in need, but the problems are that it is not obvious to tell when the concatenation operations are useful and which of the parts should be joined. Hence, using the current methods to divide Chinese characters, it is easy to find the similar between 苗 and 福 but difficult to find the similar between 副 and 福. In contrast, if we enforce a rule to save 畐 as one string of Cangjie code, it will turn the situations around. Determining the similarity between 苗 and 福 will be more difficult than finding the similarity between 副 and 福.

Due to this observation, we have come to believe that it is better to save the Chinese characters with more detailed ECCs. By saving all detailed information about a character, our system can offer candidate characters based on users' preferences which can be provided via a good user interface. This flexibility can be very helpful when we are preparing text materials for experiments for psycholinguistics or cognitive sciences (e.g., Leck et al, 1995; Yeh and Li, 2002).

6 Summary

The analysis of the 1718 errors produced by real students show that similarity between pronunciations of competing characters contributed most to the observed errors. Evidences show that the Web statistics are not very reliable for differentiating correct and incorrect characters. In contrast, the Web statistics are good for comparing the attractiveness of incorrect characters for computer assisted item authoring.

Acknowledgments

not provided for anonymous submission

References

B.-F. Chu. 2009. *Handbook of the Fifth Generation of the Cangjie Input Method*, available at http://www.cbflabs.com/book/ocj5/ocj5/index.html. Last visited on 30 April 2009.

D. Juang, J.-H. Wang, C.-Y. Lai, C.-C. Hsieh, L.-F. Chien, J.-M. Ho. 2005. Resolving the unencoded character problem for Chinese digital libraries, *Proc. of the 5th ACM/IEEE Joint Conf. on Digital Libraries*, 311–319.

S.-P. Law, W. Wong, K. M. Y. Chiu. 2005. Whole-word phonological representations of disyllabic words in the Chinese lexicon: Data from acquired dyslexia, *Behavioural Neurology*, **16**, 169–177.

K. J. Leck, B. S. Weekes, M. J. Chen. 1995. Visual and phonological pathways to the lexicon: Evidence from Chinese readers, *Memory & Cognition*, **23**(4), 468–476.

H. Lee. 2008. *Cangjie Input Methods in 30 Days*, http://input.foruto.com/cjdict/Search_1.php, Foruto Company, Hong Kong. Last visited on 30 April 2009.

C.-L. Liu, J.-H. Lin. 2008. Using structural information for identifying similar Chinese characters, *Proc. of the 46th ACL*, short papers, 93−96.

C. D. Manning, H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press. 1999.

MOE. 1996. *Common Errors in Chinese Writings* (常用國字辨似), Ministry of Education, Taiwan

S.-L. Yeh, J.-L. 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.

Chao-Lin Liu, Kan-Wen Tien, Yi-Hsuan Chuang, Chih-Bin Huang, Juei-Yu Weng. Two applications of lexical information to computer-assisted item authoring for elementary Chinese, *Lecture Notes in Computer Science* 5579: *Proceedings of the Twenty Second International Conference on Industrial Engineering & Other Applications of Applied Intelligent Systems* (IEA/AIE’09), 470−480. Tainan, Taiwan, 24-27 June 2009.

Two Applications of Lexical Information to Computer-Assisted Item Authoring for Elementary Chinese

Chao-Lin Liu, Kan-Wen Tien, Yi-Hsuan Chuang, Chih-Bin Huang, Juei-Yu Weng
Department of Computer Science, National Chengchi University, Taiwan
{chaolin, g9627, s9436, g9614, s9403}@cs.nccu.edu.tw

Abstract.[†] Testing is a popular way to assess one's competence in a language. The assessment can be conducted by the students for self evaluation or by the teachers in achievement tests. We present two applications of lexical information for assisting the task of test item authoring in this paper. Applying information implicitly contained in a machine readable lexicon, our system offers semantically and lexically similar words to help teachers prepare test items for cloze tests. Employing information about structures and pronunciations of Chinese characters, our system provides characters that are similar in either formation or pronunciation for the task of word correction. Experimental results indicate that our system furnishes quality recommendations for the preparation of test items, in addition to expediting the process.

Keywords: Computer assisted test-item authoring, Chinese synonymy, Chinese-character formation, natural language processing

1 Introduction

The history of applying computing technologies to assisting language learning and teaching can be dated back as least 40 years ago, when the Programmed Logic for Automatic Teaching Operations, which is referred as PLATO usually, was initiated in 1960 [6; p. 70]. The computing powers of modern computers and the accessibility to information supported by the Internet offer a very good environment for language learning that has never been seen before.

The techniques for natural language processing (NLP) [11] are useful for designing systems for information retrieval, knowledge management, and language learning, teaching, and testing. In recent years, the applications of NLP techniques have received attention of researchers in the Computer Assisted Language Instruction Consortium (often referred as CALICO, http://calico.org/, instituted in 1983) and the researchers in the computational linguistics, e.g., in United States of America [13] and in Europe [5]. Heift and Schulze [6] report that there are over 100 documented projects that employed NLP techniques for assisting language learning.

The applications of computing technologies to the learning of Chinese language can also be traced back as far as 40 years ago, when researchers applied computers to collate and present Chinese text for educational purposes [14]. The superior computing powers of modern computers offer researchers and practitioners to invent more complicated tools for language learning. As a result, such computer-assisted language learning applications are no longer limited to academic laboratories, and have expanded their existence into real-world classrooms [1, 16].

In this paper, we focus on how computers may help teachers assess students' competence in Chinese, and introduce two new applications for assisting teachers to prepare test items for elementary Chinese. Students' achievements in cloze tests provide a good clue to whether they learned the true meanings of the words, and the ability to identify and correct a wrong word in the so-called word-correction tests is directly related to students' ability in writing and reading. Our system offers semantically or lexically similar words for preparing the cloze items, and provides characters that are visually or phonetically similar to the key characters for the word-correction items. Experimental results indicate that the confusing characters that our system recommended were competitive in quality, even when compared with those offered by native speakers of Chinese.

We explain how to identify semantically and lexically similar Chinese words in Section 2, elaborate how to find structurally and phonetically similar Chinese characters in Section 3, and report an empirical evaluation of our system in Section 4. Finally, we make concluding remarks in Section 5.

2 Semantically and Lexically Similar Words for Cloze Tests

A cloze test is a multiple-choice test, in which one and only one of the candidate words is correct. The examinee has to find the correct answer that fits the blank position in the sentence. A typical item looks like the following. (A translation of the sentence used in this test item is "The governor officially ___ the Chinese teachers' association yesterday, and discussed with the chairperson about the education problems for the Chinese language in California." The four choices, including the answer, are different ways to say "visit" or "meet" in Chinese.)

州長於昨日正式____中文教師學會，與會長深入討論加州的中文教育問題。(a) 見面 (b) 走訪 (c) 拜訪 (d) 訪視

Cloze tests are quite common in English tests, such as GRE and TOEFL. Applying techniques for word sense disambiguation, Liu et al. [10] reported a working system that can help teachers prepare items for cloze tests for English. Our system offers a similar service for Chinese cloze tests.

To create a cloze test item, a teacher determines the word that will be the answer to the test item, and our system will search in a corpus for the sentences that contain the answer and present these sentences to the teacher. The teacher will choose one of these sentences for the test item, and our system will replace the answer with a blank area in the sample sentence (the resulting sentence is usually called *stem* in computer assisted item generation), and show an interface for more authoring tasks.

[†] Due to the subject matter of this paper, we must show Chinese characters in the text. Whenever appropriate, we provide the Chinese characters with their Romanized forms and their translations in English. We use traditional Chinese and Hanyu Pinyin, and use Arabic digits to denote the tones in Mandarin. (http://en.wikipedia.org/wiki/Pinyin) For readers who do not read Chinese, please treat those individual characters as isolated pictures or just symbols.

A cloze item needs to include distracters, in addition to the stem and the correct answer in the choices. To assist the teachers prepare the distracters, we present two types of candidate word lists to the teachers. The first type of list includes the words that are semantically similar to the answer to the cloze item, and the second type of list contains the words that are lexically similar.

We have two sources to obtain semantically similar words. The easier way is to rely on a Web-based service offered by the Institute of Linguistics at the Academia Sinica to find Chinese words of similar meanings [3], and present these words to the teachers as candidates for the distracters. We have also built our own synonym finder with HowNet (http://www.keenage.com). HowNet is bilingual machine readable lexicon for English and Chinese. HowNet employs a set of basic semantic units to explain Chinese words. Overlapping basic semantic units of two Chinese words indicate that these words share a portion of their meanings. Hence, we can build a synonym finder based on this observation, and offer semantically related words to the teachers when they need candidate words for the distracters of the cloze items. In addition, words that share more semantic units are more related than those that share fewer units. Hence, there is a simple way to prioritize multiple candidate words.

When assisting the authoring of cloze items, we can obtain lists of Chinese words that are semantically similar to the answer to the cloze item with the aforementioned methods. For instance, “造訪” (zao(4) fang(3)), “拜會” (bai(4) hui(4)), and “走訪” (zou(3) fang(3)) carry a similar meaning with “拜訪” (bai(4) fang(3)). The teachers can either choose or avoid those semantically similar, yet possibly contextually inappropriate in ordinary usage, words for the test items.

It is the practice for teachers in Taiwan to use lexically similar words as distracters. For this reason, our system presents words that contain the same characters with the answer as possible distracters. For instance, both “喝酒” (he(1) jiu(3)) and “奉茶” (feng(4) cha(2)) can serve as a distracter for “喝茶” (he(1) cha(2)) because they share one character at exactly the same position in the words. We employ HowNet to find candidate words of this category.

3 Visually and Phonetically Similar Words for Word Correction

In this section, we explain how our system helps teachers prepare test item for “word correction.” In this type of tests, a teacher intentionally replaces a Chinese character with an incorrect character, and asks students to identify and correct this incorrect character. A sample test item for word correction follows. (A translation of this Chinese string: The wide varieties of the exhibits in the flower market dazzle the visitors.)

> 花市中各種展品讓人眼花繚亂 (“繚” is incorrect, and should be replaced with “撩”)

Such an incorrect character is typically similar to the correct character either visually or phonetically. Since it is usually easy to find information about how a Chinese character is uttered, given a lexicon, we turn our attention to visually similar characters. Visually similar characters are important for learning Chinese. They are also important in the psychological studies on how people read Chinese [12, 15]. 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 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.

3.1 Examples of Visually Similar Chinese Characters

士土工干千 戊戍成 田由甲申
母毋 勿匆 人入 未末 采釆 凹凸

Fig. 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

Fig. 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

Fig. 3. Homophones with a shared component

We show three categories of similar 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 component, but the shared component is not the radical of these characters. Similar characters in every group in the second row in Figure 2 share a common component, 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 common component in Figure 3. Characters that are similar in both pronunciations and internal structures are most confusing to new learners.

It is not difficult to list all of those characters that have the same or similar pronunciations, e.g., “試” and “市”, 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 [2].

In contrast, it is relatively difficult to find characters that are written in similar ways, e.g., “構” with “購”, with an efficient manner. 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 Chinese [7], so directly computing the similarity between images of these characters demands a lot of computation. There can be more than 242 million combinations of character pairs. The Ministry of Education in Taiwan suggests that about 5000 characters are needed for everyday communication. In this case, there are about 12.5 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 component 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 component of the characters in the rightmost 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 3 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 3 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.

3.2 Encoding the Chinese Characters with the Cangjie Codes

The Cangjie method is one of the most popular methods for people to enter Chinese into computers. The designer of the Cangjie method, Mr. Chu, selected a set of 24 basic elements in Chinese characters, and proposed a set of rules to decompose Chinese characters into these elements [4]. Hence, it is possible to define the similarity between two Chinese characters based on the similarity between their Cangjie codes.

Table 1 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 “visual similarity” for characters that appeared in Figures 2 and 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”.

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 others 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 average number of key strokes needed to enter a character is a critical factor in designing input methods for Chinese. The longest Cangjie code among all Chinese characters contains five elements. As shown in Table 1, the component “巠” is represented by “一女一” in the Cangjie codes for “脛” and “徑”, but is represented only by “一一” in the codes for “頸” and “勁”. The simplification makes it relatively harder to identify visually similar characters by comparing the actual Cangjie codes.

Table 1. Cangjie codes for some characters

	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

3.3 Engineering the Cangjie Codes for Practical Applications

Though useful for the design of an 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 the complete 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”.

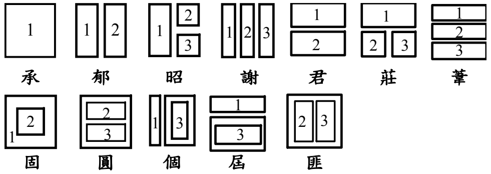

Fig. 4. Layouts of Chinese characters (used in Cangjie)

The information about the structures of the Chinese characters [7, 9] can be instrumental as well. Consider the examples in Figure 3. 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 illustrates the layouts of the components in Chinese characters that were adopted by the Cangjie method [9]. A sample character is placed below each of these layouts. A box in a layout indicates a component, and there can be at most three components in a character. We use digits to indicate the ordering the components. Due to space limits in the figure, we do not show all digits for the components.

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. The layouts are numbered from left to right and from top to bottom in Figure 4. 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. Elements that do not

Table 2. Some annotated and expanded codes

	Layout	Comp. 1	Comp. 2	Comp. 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	
莊	6	廿	女中一	土
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

belong to the original Cangjie codes of the characters are shown in a bounding box.

Recovering the elements that were dropped out by the Cangjie method and organizing the sub-sequences of elements into components 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.

Computing the similarity between characters using a database of such strengthened Cangjie code is very efficient. In the worst case, we need to compare nine pairs of code sequences for two characters that both have three components. Since we are just doing simple string comparisons, computing the similarity between characters is simple. It takes less than one second to find visually similar characters from a list of 5000 characters on a Pentium IV 2.8GHz CPU with 2G RAM. Moreover, we can offer a search service that allows psycholinguistics researchers to look for characters that contain specific components that locate at particular places within the characters.

3.4 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 the 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. It appears that we need to mark the similarity among such special characters manually, perhaps with the interactive assistance of the methods proposed in this paper.

Except for the characters that use the first layout, the Cangjie method splits characters into two or three components, and considers one of these components more important than the others. For the characters that use layouts 2, 3, 4, 5, 6, 7, 10 or 11, the component that locates at the left-most side or at the top of the layout is the most important. For the characters that use layouts 8, 9, or 12, the bounding box is the most important. In Figure 4, components that are marked "1" are the most important ones.

Due to this design principle 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.

Just like that we can choose not to simplify codes for components as we discussed in Section 3.3, we can choose not to follow this Cangjie's rule about layouts of Chinese characters. This is a feasible design choice, and we plan to implement the idea.

4 Evaluation with Real Test Items

We put into field tests the methods reported in Section 3 to show their practicability. Table 3 shows 20 Chinese words that can be included in test items for the task of word correction. These words were arbitrarily chosen from a book that was written for learning correct Chinese [17]. The underlined character in each of these 20 words is the character that is often written incorrectly. For each of these 20 words, the book provides the incorrect character that is most commonly used to replace the correct character. To make the explanation succinct in this section, we refer to these underlined characters as *target characters*. Given the words in Table 3, 21 native speakers of Chinese were asked to write down one character for each of these target characters.

We employed the method reported in Section 3 to find visually similar characters from 5000 Chinese characters. We selected phonetically similar characters for the target characters based on the list of confusing Chinese sounds that is provided by a psycholinguist of the Academia Sinica [8].

4.1 Prioritizing the Candidate Characters with Real-World Information

In the experiments, we had the flexibility to allow our system to recommend a particular number of incorrect characters for the target characters. We must have a way to prioritize the candidate characters so that we can recommend a particular number of candidate characters that may meet the teachers' expectation. More specifically, if the teachers want our system to recommend no more than 10 candidate characters and if our system has more than 10 candidate characters, how does our system choose from all the candidate characters?

This is an interesting question that requires the expertise in psycholinguistics. Facing this problem, a typical psycholinguist will probably refer us to the literature on how human read Chinese text. Indeed, computer scientists may try to do some experiments and apply machine learning techniques to learn the concept of "degree of confusion" from the collected data.

We take advantage of Google for this task. Take the number 3 item in Table 3 for example. Assume that we want to prioritize "僚", "瞭", and "繚" for "撩". We can search "眼花僚亂", "眼花瞭亂", and "眼花繚亂" in Google, and see the number of pages that used these words. Note that we must use the double quotations in our queries so that Google searches the words verbatim. On the date of this writing, Google reports that there are, respectively, 2650, 37100, and 118000 pages for these three queries. Hence, when asked for just one recommendation, our system will return "繚"; and, when asked for two recommendations, our system will return "繚" and "瞭".

Table 3. Chinese words used in the evaluation

item #	word	item #	word	item #	word	item #	word
1	一**剎**那	2	一**炷**香	3	眼花**撩**亂	4	相形見**絀**
5	作**踐**	6	剛**愎**自用	7	可見一**斑**	8	和**藹**可親
9	**彗**星	10	**委**靡不振	11	**穠**纖合度	12	待價而**沽**
13	獎**券**	14	意興闌**珊**	15	**罄**竹難書	16	**搔**首弄姿
17	根深**柢**固	18	**椿**萱並茂	19	煩**躁**	20	璀**璨**

4.2 Experimental Results: Competing with Native Speakers

Evidence showed that the native speakers who participated in our experiments did not agree with each other very well. Since every human subject returned a list of 20 characters for the words in Table 3, we collected a total of 21 lists of 20 characters. We used one of these lists as the “correct” answer and checked how the remaining 20 lists agreed with the correct answer. We repeated this process 21 times, and recorded the number of agreed character pairs. Because there were 21 human subjects, there were (21×20÷2)=210 pairs of human subjects. The agreement between any human subject pair ranged between 0 and 20.

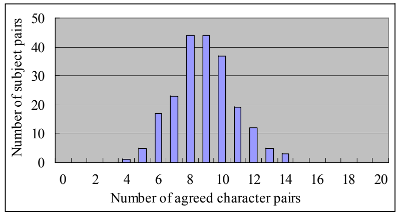

Fig. 5. Human subjects did not agree very well

Figure 5 shows how these human subjects agreed with each other. The horizontal axis shows the number of characters that appeared in the lists of a pair a human subject, and the vertical axis shows the number of human subject pairs that agreed on a particular number, indicated on the horizontal axis, of character pairs. On average, a human subject agreed with other human subjects only on 8.905 characters.

How well did our system perform? We allowed our system to recommend only one character for the words in Table 3, and compared the recommended characters with the lists provided by the book [17] that we relied on. Out of the 20 characters, our system provided 13 perfect answers. In contrast, answers provided by the best performing human subjects contained 14 perfect answers, but the average of all 21 human subjects achieved only 10.66 characters.

We also evaluated the performance of our system with the precision and recall measures that are popular in the literature on information retrieval [11]. Again, we used the answers provided in the book [17] as the perfect answers. Asked to provided five candidate characters for the words in Table 3, our system achieved 0.19 and 0.875, respectively, in precision and recall rates. Notice that a precision of 0.19 is very good in this experiment, because our system was forced to offer five candidate characters while there was only one perfect answer. A precision of 0.19 suggests that our system almost caught the perfect answer with just five candidate characters.

5 Concluding Remarks

We report the applications of phonetic, lexical, and semantic information to the design of a computer-assisted item authoring environment. Our experience indicates that the environment can improve the efficiency for item authoring. Experimental results further show that the resulting system is useful for preparing quality test-items. Nevertheless, we must note that the quality of the compiled test items depends highly on the experts who actually prepare the test items. The capability of finding visually similar characters is also very useful for conducting psycholinguistic studies, and we have begun a joint research project in this direction.

It is important to study the effects of employing the incorrect characters in real-world applications that involve human subjects to investigate their reactions. We have taken steps toward this research, and will report the findings in an extended report.

Acknowledgments. We thank anonymous reviewers for their invaluable comments, and we will respond to the comments in an extended report. This research project was funded in part by the grants NSC-95-2221-E-004-013-MY2 and NSC-97-2221-E-004-07-MY2 from the National Science Council of Taiwan.

References

1. Bourgerie, D. S.: Computer assisted language learning for Chinese: A survey and annotated bibliography, *J. of the Chinese Language Teachers Association*, **38**(2), 17–48 (2003)
2. Chen, M. Y.: *Tone Sandhi: Patterns across Chinese Dialects*. (Cambridge Studies in Linguistics 92) Cambridge: Cambridge University Press (2000)
3. Cheng, C.-C.: Word-focused extensive reading with guidance, In *Selected Papers from the Thirteenth International Symposium on English Teaching*, 24–32. Taipei, Taiwan: Crane Publishing. http://elearning.ling.sinica.edu.tw/ (2004) Last visited on 11 Nov. 2008
4. Chu, B.-F.: *Handbook of the Fifth Generation of the Cangjie Input Method*, available at http://www.cbflabs.com/book/ocj5/ocj5/index.html. Last visited on 11 Nov. 2008
5. Ezeiza, N, Maritxalar, M, Schulze, M. (eds): *Proc. of the Workshop on Natural Language Processing for Educational Resources*, International Conference on Recent Advances in Natural Language Processing (2007)
6. Heift, T., Schulze, M.: *Errors and Intelligence in Computer-Assisted Language Learning*. NY, USA: Routledge (2007)
7. Juang, D., Wang, J.-H., Lai, C.-Y., Hsieh, C.-C., Chien, L.-F., Ho, J.-M.: Resolving the unencoded character problem for Chinese digital libraries. *Proceedings of the Fifth ACM/IEEE Joint Conference on Digital Libraries*, 311–319 (2005)
8. Lee, C.-Y.: personal communication, Institute of Linguistics, Academia Sinica (2008)
9. Lee, H.: *Cangjie Input Methods in 30 Days*, http://input.foruto.com/cjdict/Search_1.php, Foruto Company, Hong Kong. Last visited on 11 Nov. 2008
10. Liu, C.-L., Wang, C.-H., Gao, Z.-M.: Using lexical constraints for enhancing computer-generated multiple-choice cloze items, *International Journal of Computational Linguistics and Chinese Language Processing*, **10**(3), 303–328 (2005)
11. Manning, C. D., Schüetze, H.: *Foundations of Statistical Natural Language Processing*. MA, USA: The MIT Press (1999)
12. Taft, M., Zhu, X, Peng, D.: Positional specificity of radicals in Chinese character recognition, *Journal of Memory and Language*, **40**, 498–519 (1999)
13. Tetreault, J., Burstein, J., De Felice, R. (eds.): *Proc. of the Third Workshop on Innovative Use of NLP for Building Educational Applications*, the Forty Sixth Annual Meeting of the Association for Computational Linguistics (2008)
14. Wang, F. F. Y.: Report on Chinese language concordances made by computer, *Journal of the Chinese Language Teachers Association*, **1**(2), 73−76 (1966)
15. Yeh, S.-L., Li, J.-L.: 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 (2002)
16. Zhang, Z.-S.: CALL for Chinese—Issues and practice, *Journal of the Chinese Language Teachers Association*, **33**(1), 51−82, (1998)
17. Tsay, Y.-C., Tsay, C.-C. : *Diagnoses of Incorrect Chinese Usage*, Pan-Chiao, Taiwan: Firefly Publisher（蔡有秩及蔡仲慶：*新編錯別字門診*，臺灣，板橋：螢火蟲出版社）(2003)