

國立交通大學

理學院網路學習學程

碩士論文

英語文試題檢測與答題驗證系統設計與實作

The Design and Implementation of a Marking and Validation System
for the Computerized Base English Test

研究生：張鈞凱

指導教授：陳登吉 教授

中華民國九十四年六月

英語文試題檢測與答題驗證系統設計與實作

The Design and Implementation of a Marking and Validation System
for the Computerized Base English Test

研究生：張鈞凱

Student : Chun-Kai Chang

指導教授：陳登吉

Advisor : Deng-Jyi Chen

國立交通大學
理學院網路學習學程
碩士論文



A Thesis

Submitted to Degree Program of E-Learning
College of Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Degree Program of E-Learning

June 2005

Hsinchu, Taiwan, Republic of China

中華民國九十四年六月

英語文試題檢測與答題驗證系統設計與實作

學生：張鈞凱

指導教授：陳登吉 博士

國立交通大學

理學院網路學習碩士在職專班



在近年來全國不僅開始總動員學習英語，並且掀起了一股英語檢定風，造成統計至九十三年底，共有三十二萬人次參加全民英語檢定熱潮。在如此繁重的試務工作中，要如何讓檢定具有一定水準，讓試題的分級有一可靠的機制及參考標準，讓批改答題的過程更有效率，這些都是我們所關心的，而且深深的影響到檢定單位的公信力。

因此本論文嘗試去研究電腦化英語檢定測驗過程，如何讓試題列入適當的檢定級數，又如何能讓電腦來協助閱卷老師，分擔閱卷批改過程的負荷，進而使得評分閱卷老師的閱卷基礎較一致。進行試題檢測是因為要達到試題分級的目的，因此我們提出利用詞彙表、字頻及詞彙數分析的方法，在命題委員命題完畢後，能夠經過試題檢測客觀的分析，讓分級的過程中，有一個分級評量標準，使得試題分級更具有公信力。在答題驗證部份，我們實作出拼字檢查、文法檢查及語義檢查三大部份，輔助閱卷老師在批改過程中的各項檢查，以減少人工閱卷之時間及錯誤。

關鍵字：自然語言處理、詞性標記、文法分析、詞彙表、字頻。

The Design and Implementation of a Marking and Validation System for the Computerized Base English Test

Student : Chun-Kai Chang

Advisor : Dr. Deng-Jyi Chen

Degree Program of E-Learning

College of Science

National Chiao Tung University

Abstract

The English language skill training (teaching), learning, and assessment is a big industry around the world, especially in the territory of Asia countries. The traditional pen-paper based test delivery approach for English language skill assessment is still domination. However, the computer based test delivery approach has gradually received its attentions due to its advantages on using multimedia such as video, animation, voice, and color image to simulate the question presentation in the real world. Also, using computer and network to conduct a test delivery is less labor intensive and is relative efficiency for the test delivery management activities. In any test delivery approaches, test questions have to be designed and validated and answers from test takers must be graded. From test management point of views, each pen-paper based test sheet requires a qualified score marker to grade it. The grading process usually is very time-consuming. Considering that there are million copies of test sheets needed to be graded, a lot of qualified score markers are demanded. And, usually, the grading criteria of each score marker may not be able to retain a consistent way after a long grading hour.

In this thesis study, we attempt to design and implement 1) an English test question validation system and 2) a score marking system for English writing test under a computer based test delivery approach. Especially, the proposed English test question validation system is designed to help test question creators to verify if a created test question is met with the level of difficulty by analyzing several test item attributes associated with the question. Based on the report generated from the test question, test question creators can adjust the level of difficulty before the test question is submitted

to test question bank. The proposed scoring marking system for English writing test is designed to help score marker to give a more consistent and accurate grading result by analyzing the vocabulary, grammar, and possibly semantic errors found in the answer sheet. Each score marker can have a report analyzed by the proposed system for reference before a formal score is given. This will avoid problems mentioned above. We integrate several current tools (such as Aspell, Apple Pie parser, Worldnet, and so on) into our score marking system and test question validation system. The proposed system has been used to the NETPAW English test to demonstrate its applicability and feasibility.

Keyword: Natural Language Processing, Tag, Grammar Analysis, Word List, Word Frequency



誌謝

本論文承蒙指導教授陳登吉博士耐心指導與教誨，陳老師不僅在學術上給予指導，在日常生活上也給我非常多的關照，也讓我兩年的研究生生涯過的相當的積極充實，老師在做研究的方法和精神，深深的影響了學生，本論文亦在老師不厭其煩的更正與修改下，得以順利完成，在此特對師恩致上無限的感激。

其次，感謝論文口試委員台大鄭恆雄教授、師大何榮桂教授、交大劉美君教授及班主任莊祚敏教授，細心指正論文謬誤、不妥之處，並提供修正意見。

亦要感謝在生活上以及課業上給予我協助的同學，尤其是實驗室涂仲箎、李嘉彪…等多位學長的經驗傳授，還有同窗兩年同學李玉珍、黃吉楠，以及學弟王向榮、蔡東杰，在程式與技術上對我的指導與幫助。

研究所這兩年是我感到人生最忙碌也最充實的時光。最後，感謝養育我、栽培我的父母，還有老婆在背後的支持與鼓勵，使我得以專心完成論文無後顧之憂，也才能有今日的我，謝謝。



目次

摘要	i
Abstract.....	ii
目次	iv
表目錄	vii
圖目錄	viii
一、緒論	1
1.1 研究背景與動機	1
1.2 研究目的與範圍	2
1.3 研究方法與步驟	2
1.4 章節概要	3
二、相關研究	4
2.1 英語詞彙資料庫—WordNet	5
2.1.1 WordNet發展概況	6
2.1.2 WordNet的內容	6
2.1.3 WordNet的設計	7
2.1.4 WordNet的實例	8
2.2 詞彙詞性標記	9
2.3 Parser軟體—Apple Pie Parser	11
2.4 GNU Aspell	13
三、系統需求分析	16
3.1 系統概念	16
3.2 試題檢測需求分析	19
3.2.1 輸入的檢測資料	19
3.2.2 檢測後的輸出資料	20
3.3 答題驗證需求分析	22
3.3.1 拼字檢查	22
3.3.2 文法檢查	23
3.3.3 語義檢查	23
四、系統設計與實作	25
4.1 前處理	25
4.1.1 正規化(normalization)	25

4.1.2 斷句(sentence segmentation)	26
4.1.3 斷詞(tokenization).....	26
4.2 試題檢測設計與實作	27
4.2.1 詞彙表分析	28
4.2.2 字頻分析	31
4.2.3 詞彙數分析	32
4.2.4 實作程式流程	32
4.3 答題驗證設計與實作	34
4.3.1 系統概念流程圖	34
4.3.2 使用態連結函式庫	36
4.3.3 系統程式架構	37
4.3.4 Module程式說明.....	44
五、系統展示與應用實例	49
5.1 試題檢測系統	49
5.1.1 範例一	49
5.1.2 範例二	52
5.2 答題驗證系統	54
5.2.1 範例一	54
5.2.2 範例二	56
5.2.3 範例三	57
5.2.4 範例四	58
5.2.5 範例五	59
5.2.6 本系統的限制	62
六、總結	64
6.1 總結	64
6.2 未來發展方向	64
參考文獻	66



表目錄

表 1 WORDNET中詞彙、同義詞集及詞義的數量.....	7
表 2 具有多重意義的單字在TREE-BANK CORPUS 中分布的情形.....	10
表 3 THE PENN TREEBANK 詞集分類一覽表.....	12
表 4 ASPELL與其他拼字軟體比較表.....	14
表 5 原型詞彙表暨字頻資料庫結構一覽表.....	28
表 6 衍生詞彙表資料庫結構一覽表.....	30
表 7 專有名詞資料庫結構一覽表.....	31
表 8 詞彙數資料庫結構一覽表.....	32
表 9 WORDNET_INDEX資料庫結構一覽表.....	42
表 10 WORDNET_DATA資料庫結構一覽表.....	42
表 11 WRODNET同義字資料庫查詢「HAPPY」同義字結果一覽表.....	43
表 12 詞性輔助資料庫資料庫結構一覽表.....	44



圖目錄

圖 1：三層關係的名詞語義網路圖例	8
圖 2：WORDNET 查詢 HAPPY 結果顯示畫面	9
圖 3：APPLE PIE PARSER 解析樹	13
圖 4：ASPELL 與其他拼字檢查程式核心測試比較圖.....	15
圖 5：ASPELL 在 UNIX 上執行畫面.....	15
圖 6：同等級試題適性值的驗證與兩段式的動態調整流程圖	16
圖 7：即時試卷的全部試題的 ICC 與 IIC 圖比較.....	17
圖 8：試卷訊息曲線圖(TIC)	17
圖 9：系統概念示意圖	18
圖 10：試題檢測系統概念示意圖	19
圖 11：試題檢測後輸出資料圖	22
圖 12：前處理系統概念示意圖	25
圖 13：網路英檢擬模試題畫面	26
圖 14：試題檢測系統流程概念示意圖	27
圖 15：試題檢測系統內資料庫關連示意圖	31
圖 16：試題檢測系統程式流程圖	33
圖 17：試題檢測系統程式輸出畫面	34
圖 18：答題驗證系統概念流程圖	35
圖 19：答題驗證系統程式架構圖	37
圖 20：APPLE PIE PARSER'S PARSER 建立出解析樹	40
圖 21：語義檢查 MOUDLE 程式流程圖	44
圖 22：形容詞與名詞單複數不一致 MODULE 程式流程圖	45
圖 23：主詞與動詞單複數不一致 MODULE 程式流程圖	47

圖 24：試題檢測範例一輸入畫面	49
圖 25：試題檢測範例一輸出畫面之一	50
圖 26：試題檢測範例一輸出畫面之二	51
圖 27：試題檢測範例一輸出畫面之三	52
圖 28：試題檢測範例二輸出畫面	53
圖 29：答題驗證系統輸入畫面	54
圖 30：答題驗證系統範例一輸出畫面	55
圖 31：答題驗證系統範例二輸出畫面	56
圖 32：答題驗證系統範例三輸出畫面	57
圖 33：答題驗證系統範例四輸出畫面	58
圖 34：答題驗證系統人工閱卷輸出畫面之一	59
圖 35：答題驗證系統人工閱卷輸出畫面之二	60
圖 36：答題驗證系統人工閱卷輸出畫面之三	61
圖 37：答題驗證系統人工閱卷輸出畫面之四	62

一、緒論

1.1 研究背景與動機

2002年初，台灣加入世界貿易組織(World Trade Organization)，成為該組織的正式成員之一，促成國內企業與外國企業更頻繁之貿易關係與接觸，而英語為國際通用之主要語言，它不論在通訊、溝通與資訊傳遞等各領域，都有著強勢的影響力，故其重要性更為提高。據估計目前世界上約有 7.5 億至 10 億的人口使用英語，而其中有 3 億人口的母語是英語。在全球化的競爭壓力之下，政府以及學術界的領導人一再提出台灣必須提昇國民英語能力的呼籲。因此各級教育單位不斷推出提昇國民的英語能力的方案。

身為我國最高行政單位的行政院，目前已提出「挑戰 2008 年：國家發展重點計畫」，認為人的投資是所有建設和發展的基礎與目標，所以第一項就是「E 世代人才培育計畫」，其目標是：培育具有創意活力及國際對話能力的新世代，也就是能夠嫻熟應用「資訊與英語」的新世代。面對未來的挑戰，首先就要強調國民適應全球化與國際化的能力，同時也要營造一個國際化的環境和全民學習的條件。使用外語(特別是英語)和網路通訊能力的培養乃成為這項計畫的重點，尤其是英語，已經成為與世界接軌的主要工具，政府在六年內應該將英語的地位提升為準官方語言，積極推動擴大英語應用的範圍，讓英語成為生活中的一部份。

「E 世代人才培育計畫」中更明確提出營造英語生活環境、平衡城鄉英語教育資源、大專院校教學國際化、提高公務人員英語能力、推動英語與國際文化學習…等五項具體的政策，而在推動英語與國際文化學習項目中，辦理「推廣全民外語學習」方案，持續推動全民社區英語學習各方案；推行「全民英語能力檢定測驗」、「全民網路英語能力檢定」。^[17]

就因如此，全國不僅開始總動員學習英語，並且掀起了一股英語檢定風，造成統計至九十三年底，共有三十二萬人次參加全民英語檢定熱潮^[27]。光今年(九十四年)一月及六月舉行的二次全民英語能力分級檢定初級檢定的報考者，居然高達二十五萬人報考^[31]，其熱烈的情形已經不輸給學力測驗。在如此繁重的試務工作中，要如何讓檢定具有一定水準，讓試題的分級有一可靠的機制及參考標準，讓批改答題的過程更有效率，這些都是我們所關心的，而且深深的影響到檢定單位的公信力。

在電腦與網路發達的二十一世紀，由於資訊科技與網際網路的快速發展，電腦已改變人類生活的方式，對於教育與測驗環境亦有很大的改變，測驗的 e 化漸成為現今之趨勢。就因如此，本研究想藉由電腦化為人類帶來便利當中，去研究電腦化英語檢定測驗過程，如何讓試題列入適當的檢定級數，又如何能讓電腦來協助閱卷老師，分擔閱卷批改過程的負荷，進而使得評分閱卷老師的閱卷基礎較一致。

1.2 研究目的與範圍

本研究期望能達到以下的目的：

1. 訂定一個試題檢測標準，讓試題分級更具公信力。
2. 因應檢測結果分級制，試題檢測應依既定試題庫級數分級。
3. 改善人工試題檢測時，較無法達到統一標準的缺點。
4. 利用電腦先進行答題驗證，以減少人工閱卷之時間及錯誤。
5. 簡答案題的數量龐大時，改善以人工方式審題，費時費力的缺點。

此研究範圍主要是在網路英檢上，而應用到全民網路英檢上做為系統驗證。

網路英檢的測驗模式是在電腦上進行施測，而且平時可以 24 小時上網免費學習及自我檢定，就因為可以在電腦上施測及學習，所以能夠大量的使用多媒體素材，如：文字、圖片、動畫、聲音甚至影片，而多媒體素材每個部份的分析，皆有其專業領域的技術，因此本論文研究範圍僅就文字式的試題及簡答案題部份進行檢測及驗證。



1.3 研究方法與步驟

首先我們會先了解試題檢測及答題驗證過程當中的各種需求，接著了解可能的解決方法，之後針對我們提出的方法做需求分析，進而實作系統。詳細的步驟條列如下：

1. 了解目前試題檢測的需求現況。
2. 探討試題檢測的方法。
3. 收集並且整理試題檢測過程當中之資料，並建立資料庫。
4. 設計出試題檢測的過程。
5. 分析答題驗證的需求。
6. 了解答題驗證的分析種類。
7. 設計答題驗證的分析方法。

8. 實作試題檢測及答題驗證系統。

1.4 章節概要

第一章，我們提出撰寫本篇論文的動機與目的，以及研究方法與步驟。

第二章，介紹與本系統相關研究的內容。

第三章，介紹本系統的流程概念，進而提出試題測檢部份的需求分析，再探討答題驗證部份的需求分析。

第四章，提出系統的設計概念，並且詳細介紹如何設計，進而實作出系統的方法。

第五章，以實際範例示範試題檢測與答題驗證系統的結果，並且說明結果畫面中各項數據的意義。

第六章，作一個總結，並對未來的發展擔出一些建議。



二、相關研究

在此研究的背景當中，須使用到大量的詞彙資料庫、標記化語料庫(Tagged Corpus)，還須要有語義、語法及拼字各方面的技術來支援，而目前國外對這些部份的研究，均有相當的成果，因此研究當中就利用其相關技術，來讓系統運作更完整。

1. 語義技術部份：

因為WordNet為國際間詞彙知識庫通用的架構，其是為結合詞彙編纂資訊和高速運算的所產生更多效益而提出的計畫。在多語環境（如WWW 全球資訊網）中，WordNet提供了兼顧概念通性及語言通性的知識表達架構；更因詞彙網路涵蓋了絕大多數富含資訊（information-rich）的語言；正可以擔任不同語言間知識資訊轉換的座標。[18]

有 WordNet 之父之稱的心理學大師米勒（George A. Miller），給了大家的啓發後，此刻許多研究者都在努力建立各種語言的 WordNets：Spanish WordNet、Chinese WordNet…等，WordNet 儼然已經成為語言研究者的必修課；因為 WordNet 可以透過語意關係，連接到其他相關字，社會大眾也多了一個比一般辭典更方便的工具。

而在台灣有許多有名的研究都開始使用WordNet，其中最具代表的是「中央研究院中英雙語知識本體詞網」（簡稱「研究院知識詞網」，Sinica BOW）目前開放的雛型是由中研院與國家數位典藏計畫中的「語言座標」計畫所建構完成。使用到的資料國外的有IEEE批准執行的SUMO(Suggested Upper Merged Ontology)知識本體(teknowledge.com 管理)及普林斯頓（Princeton University）的WordNet。因為「語言座標」採用了「詞彙網路」(WordNet)的架構，為國際間詞彙知識庫通用的架構。找到了英文對譯詞後，可藉EuroWordNet等網路上開放的資料庫，對應到 20 幾種語言。[19]

就因為以上多項的原因，再加上 WordNet 的研究人員一直把 WordNet 視作一個試驗，而不是一個產品。因此當 WordNet 中的詞語足夠多的時候，他們就把 WordNet 向學術界免費公開，所以其程式碼及原始資料庫資料取得容易，只要不盜用普林斯頓大學的商標來宣傳 WordNet 的任何派生產品。就基於這些原因，所以本系統利用 WordNet 來做為我們語義分析的核心。

2. 語法技術部份：

語法的部份可以利用到自然語言處理(Natural Language Processing)技術中，已經研究多年的解析(Parser)方法。而解析軟體在國外非常多，常見的有 Minipar、CG Parser、CASS Partial Parser、CHILL、ISSCO Tools、IMS Stuttgart、Apple Pie Parser、Link Grammar Parser…等。其中有些程式碼不易取得，如 CASS Partial Parser 網站上只有提供 Parser 服務，沒有其他任何的介紹。有些使用程式語言與本研究規畫使用的 C 語言不同，如 CHILL、ISSCO

Tools 這些則用 Prolog 語言，Prolog 語言是電腦人工智慧程式語言之一。若用語言來選擇，其中有 Minipar、Apple Pie Parser(APP)、Link Grammar Parser 是使用 C 語言，而 Link Grammar Parser 是用句法解析(syntactic parser)的方式去解析，對於輸出結果是圖像的表達，對於後續處理較為不易。

最後可以選擇的有 Apple Pie Parser (APP) 及 Minipar，而有文章[9] 提到 APP 分析 15 字以上的句子正確率可達 83%，而 Minipar 正確率可達 88.54%，均是不錯的解析軟體。而其中 APP 支援 Unix 工作站，加上程式原始碼及資料庫方便取得，且詞性分類採 The Penn Treebank 系統，分類適中易懂，所以本系統採用此軟體做為解析之工具。

3. 拼字技術部份：

此處所提到的拼字指的像是 module，將其英文字母拼錯而成 moudle 這樣的拼字。而要像這樣檢查拼字，大家最熟悉的就是 Microsoft Word，如果在使用 Word 時有拼字錯誤，則其會在錯誤單字下標出紅色的波折線，像是上述的錯字就會被顯示成「moudle」。這樣的拼字檢查技術，在國外中被使用的非常普遍，基本上只要文字編輯的軟體，就有拼字檢查的功能，像是上述 Microsoft Word，還有 ConjuGNU、koffice、OpenOffice...等這些都是，但是其要為研究上利用，則不如 Ispell、Aspell 純粹的拼字檢查工具軟體來的方便及容易。

而 Aspell 主要是為超越 Ispell 而做出來的，因此其功能性及方便性都比 Ispell 來的好，加上又有做出程式庫，更方便被其他程式使用。同樣地，他的程式取得也是非常的容易，支援 Unix 工作站，因此本系統採用 Aspell 來為我們做拼字檢查的工作。

接下的章節中，就來介紹這些技術的相關研究，讓大家更為清楚其背景、理論及用途。

2.1 英語詞彙資料庫—WordNet

WordNet 是普林斯頓大學(Princeton University)認知科學實驗室(Cognitive Science Laboratory)所發展出來的。該專案被稱為「英語詞彙資料庫」(a lexical database for the English language)，主要是由 George A. Miller 教授所領導，WordNet 是這樣一個系統，通過將同義詞的集合集中到稱為同義詞集(synonym sets)或同義集(synsets)的組合中來描述和分類單詞和概念。我不能過分吹捧這一重要的長期項目，該項目代表了如此有前景的產業。因為它的開放性意味著實際上任何開發人員都可以使用它，所以它十分重要。WordNet 有“不受限制的”許可證。它很像 BSD 許可證，因為唯一真正的限制是不可以盜用普林斯頓大學的商標來宣傳 WordNet 的任何派生產品。[5]

2.1.1 WordNet 發展概況

關於WordNet的不成熟的想法可以追溯到 20 多年前，而這一想法開始逐漸具體化和清晰化則是 1985 年後才開始的。從 1985 年開始，WordNet作為一個知識工程全面展開。不過，當時的WordNet和經過 20 年後今天的WordNet還是很不一樣的。[24]

這一工程最初的前提之一是“可分離性假設”(Separability hypothesis)，即語言的詞彙成分可以被離析出來並專門針對它加以研究。詞彙編纂學的歷史明確地告訴我們，在詞語水準上可以得到有用的研究成果。詞庫(詞典，lexicon)當然不是完全獨立於其他語言成分的，但它的確是可以從其他成分中分離出來的。例如，儘管語音和語法知識在一個人的早年生活中就成型了，但詞彙量卻可以隨著智力活動的不斷積累而增加。這表明語言的不同成分涉及不同的認知過程。

另一個前提是“模式假設”(patterning hypothesis)：一個人不可能掌握他運用一種語言所需的所有詞彙，除非他能夠利用詞義中存在的系統的模式和詞義之間的關係。這種系統化的心智模式至少從柏拉圖時代就成爲一種進行推測的學問，現代語言學研究開始在自然語言的語義結構中識別這樣的模式。但許多遵循這類路線的出色的研究工作在這一問題上碰到了困難。一個作者可能提出一種語義理論，並以 20 到 50 個英語單詞爲例來展示他的理論，而留下另外 10 萬個單詞讓讀者去做練習。

第三個前提就是所謂的“廣泛性假設”(comprehensiveness hypothesis)：計算語言學如果希望能像人那樣處理自然語言，就需要像人那樣儲存盡可能多的詞彙知識。

建立包含詞語意義描述的大規模詞庫的方式之一是基於語義成分分析的詞彙語義學(componential lexical semantics)的方法(也可譯爲義素分析法)。這種方式把一個詞的意義分析爲更小的概念原子的組合。不過，定義一套概念原子卻非易事。事實上，WordNet 主帥 George.A.Miller 在 1976 年他與 Philip N. Johnson-Laird 合作的《Language and Perception》一書中還躊躇滿志地探索義素分析的語義描寫方法，但直到 1985 年，仍然沒有能夠出籠一個完整的定義清晰的清單，在上面列舉出所有的概念原子。但發展到今天已經有非常大的進展，尤其是 WordNet 的研究人員一直把 WordNet 視作一個試驗，而不是一個產品。因此當 WordNet 中的詞語足夠多的時候，他們就把 WordNet 向學術界免費公開，讓世界上許多國家依照 WordNet 的概念，開始做該國語言的語料庫，讓更多世人受惠。

2.1.2 WordNet 的內容

WordNet 的描述物件包含 compound(複合詞)、phrasal verb(短語動詞)、collocation(搭配詞)、idiomatic phrase(成語)、word(單詞)，其中 word 是最基本的單位。

WordNet 並不把詞語分解成更小的有意義的單位(這是義素分析法 /componential analyses 的方法)；WordNet 也不包含比詞更大的組織單位(如腳本、框架之類的單位)；由於 WordNet 把 4 個開放詞類區分為不同檔加以處理，因而 WordNet 中也不包含詞語的句法資訊內容；WordNet 包含緊湊短語，如 bad person，這樣的語言成分不能被作為單個詞來加以解釋。

人們經常區分詞語知識和世界知識。前者體現在詞典中，後者體現在百科全書中。事實上二者的界限是模糊的。比如hit(“打”)某人是一種帶有敵意的行為，這是百科知識；而hit跟strike(“擊”)多多少少同義，並且hit可以帶一個直接賓語論元，這是詞語知識。但hit的直接賓語應該是固體(而不是像gas這樣的氣體)，這是詞語知識還是百科知識就界限模糊了。不過毫無疑問，要理解語言，這兩部分知識是缺一不可的。Kay(1989)指出我們的大腦詞庫應該包含這兩部分知識。但是百科知識太多難以駕馭，WordNet不試圖包括百科知識。不過，在WordNet中，對於一些不常見的專業概念，比如不常見的植物和動物，詞語知識和百科知識是融合在一起的。[7]

2.1.3 WordNet 的設計

一般的詞典都是按照單詞拼寫的正字法原則進行組織的。但如果為了獲得詞語意義資訊的目的，通過詞語語義屬性來組織詞典就更值得去做了。線上詞典跟傳統的紙張詞典不同，允許使用者從不同的途徑去訪問詞典資訊。

第一個以意義作為組織原則的詞典是羅傑斯同義詞詞林(Roget's Thesaurus)。傳統的詞典是通過提供給用戶關於詞語的資訊來幫助用戶理解那些他們不熟悉的詞的概念意義。WordNet 既非傳統詞典，也非同義詞詞林。它混合了這兩種類型的詞典。

WordNet跟同義詞詞林相似的地方是：它也是以同義詞集合(synset)作為基本建構單位進行組織的。大家在腦子裏如果有一個已知的概念，就可以在同義詞集合中中找到一個適合的詞去表達這個概念。下表表 5是日前WordNet的規模。

表 1 WordNet中詞彙、同義詞集及詞義的數量[5]

POS	Unique Strings	Synsets	Total Word-Sense Pairs
Noun	114,648	79,689	141,690
Verb	11,306	13,508	24,632
Adjective	21,436	18,563	31,015
Adverb	4,669	3,664	5,808
Totals	152,059	115,424	203,145

但 WordNet 不僅僅是用同義詞集合的方式羅列概念。同義詞集合之間是以一定數量的關係類型相關聯的。這些關係包括上下位關係、整體部分關係、繼承關係等。

不同句法詞類中的語義關係類型也不同，比如儘管名詞都動詞都是分層級組織詞語之間的語義關係，但在名詞中，上下位關係是 **hyponymy** 關係，而動詞中是 **troponymy** 關係；動詞中的 **entailment**(繼承)關係有些類似名詞中的 **meronymy**(整體部分)關係。名詞的 **meronymy** 關係下面還分出三種類型的子關係(見“WordNet 中的名詞”部分)。

我們現在用一個如圖 1 所示的名詞語義網路圖例來跟大家略為解說。

1. 下位關係(hyponymy)：像物質(substance)再細分分類下去，也就是其下位關係有生物物質(organic substance)，而生物物質中再細分下去就有肌內(flesh)及骨頭(bone)都算生物物質。
2. 反義關係(antonymy)：像兄弟(borther)及姊妹(sister)之間就是反義關係。
3. 部份和整體關係(meronymy)：像人的身體(body)就是屬於人(person)的一部份，而手臂(arm)和腿(leg)就屬於身體的一部份。

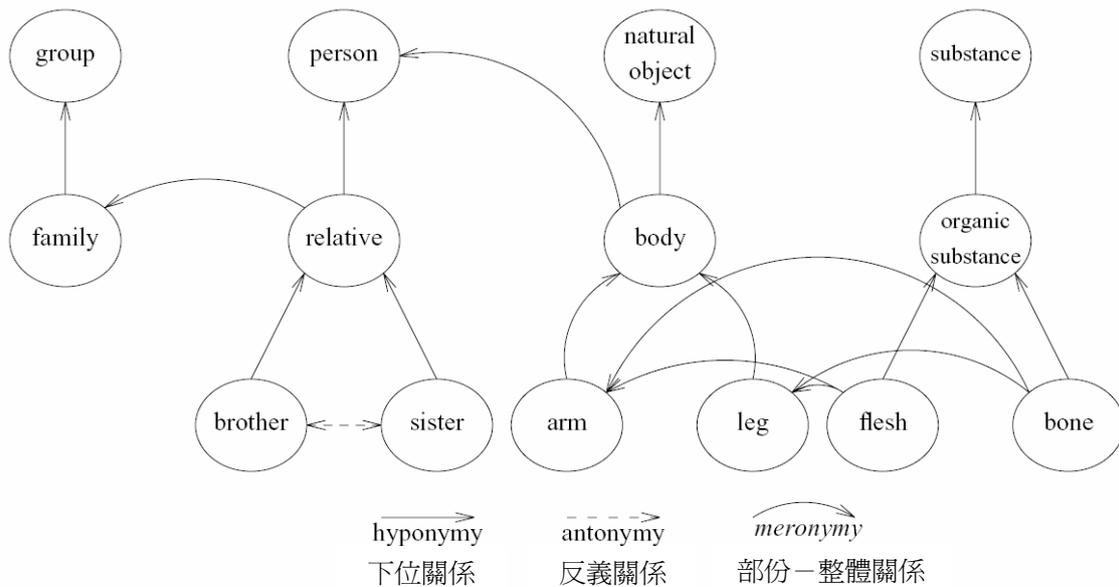


圖 1：三層關係的名詞語義網路圖例[6]

2.1.4 WordNet 的實例

圖 2 是利用目前 WordNet 最新 2.0 版查詢「happy」的結果，其中第一行文字「The adj happy has 6 senses」告訴我們 happy 是一個形容詞(adj)，而有 6 個詞義，其 6 個詞義分別列出 6 個項目做各別說明。在單項說明中同時出現的單字，代表其彼此

之間是同義字的關係，例如項次 2. 中的「happy」及「pleased」是屬同義字的關係。另外單項說明中括弧部份的內容，則是對該詞義做更詳細的說明。

而本論文中是利用 WordNet 來做同義字的分析資料，所以我們就只取出各項次出現的同義字來利用，像此例就取出 pleased、felicitous、euphoric、well-chosen 四個字，來當做 happy 的同義字使用。

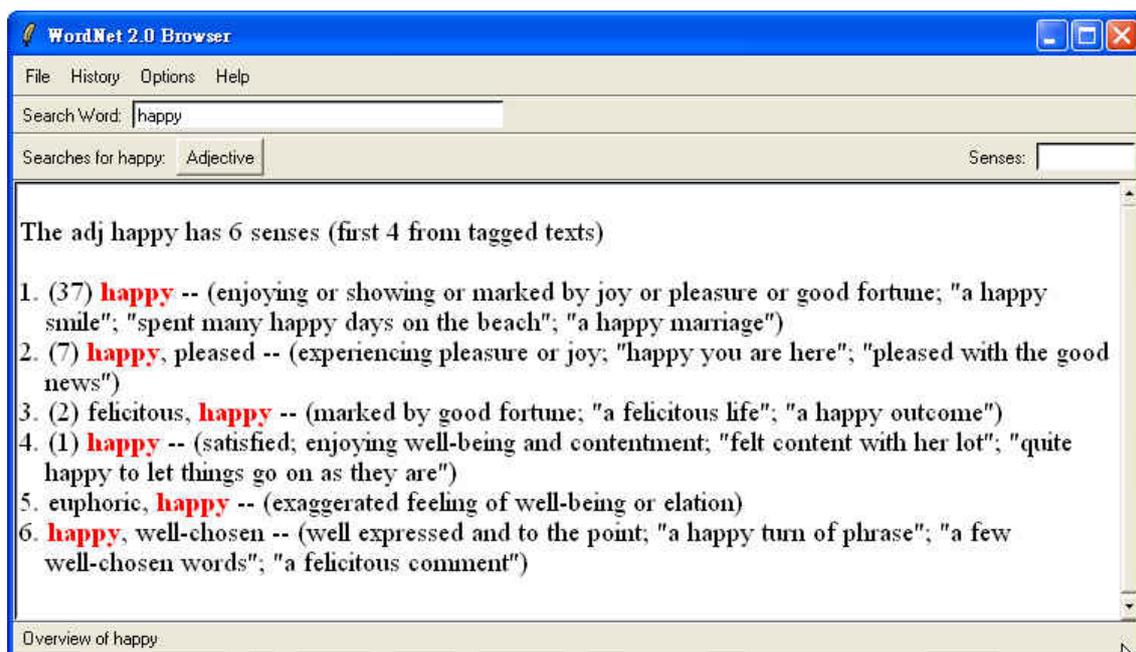


圖 2：WordNet 查詢 happy 結果顯示畫面

2.2 詞彙詞性標記

通常語料庫的資料(每個單字)如果做了詞性標記，那麼它就可以運用在語言學的分析或者是研究句子文法的結構，例如名詞加名詞的組合，形容詞加動詞的組合等等。

詞性標記(Tagging)，就是一次看一個句子的每一個字，然後給每一個字一個標記，也就是該單字的詞性。例如：John sometimes goes to school. 會被標記詞性如下：John：NNP(單數專有名詞)、sometimes：RB(副詞)、goes：VBZ(動詞第三人稱單數)、to：TO(介系詞 to)、school：NN(單數名詞)。

另外，有的英文單字是具有多重意義的單字，在使用時使用者不了解這個單字的用法，以致於他們在使用時，會自己搞不清這個單字，在句子真正所代表的意義。例如：“Good” 這個單字，在下列兩種情形下有不同的詞性標記：

1. He-PP is-VBZ my-PP\$ good-JJ friend-NN . -SENT
2. Christians-NNS believe-VVP the-DT good-NN will-MD go-VV to-TO heaven-NNwhen-WRB they-PP die-VVP .-SENT

第一句的good 標記為JJ(形容詞)，第二句的good 標記為NN(名詞)。在此雖是同一個good，但卻有不同的詞性標記。表 2 是TreeBank 的corpus 中同一個字有幾種詞性的分布情形：

表 2 具有多重意義的單字在Tree-Bank corpus 中分布的情形。[8]

Degree of ambiguity	Total frequency (39,440)
單一詞性 (1 tag)	35,340
多詞性 (2-7 tags)	4,100
2 tags	3,760
3 tags	264
4 tags	61
5 tags	12
6 tags	2
7 tags	1 (still)

[註 1]10.4 percent of the lexicon is ambiguous as to part-of-speech (types)

[註 2]40 percent of the words in the Brown corpus are ambiguous (tokens)

從表 2 還是可以看到有許多單字是具有多重意義的。而詞性標記解決這類問題有二種做法，Rule-based approach、Stochastic approach。

1. Rule-based approach :

- a. 依靠文法規則。像名詞片語就可以由一個限定詞性加上一個或一個以上的形容詞: NP -> DT(JJ*)NN

例如: The pretty woman.

- b. 藉由形態分析(Morphological Analysis)的幫助，去消除語意不清的問題。

例如 X-ing 這類的單字，如果前面接動詞，那就把 X-ing 標記成動詞。

- c. Supervised method : 完成一個已經事先經過詞性標記的語料庫(pre-taggedcorpus)，讓學習者可以從語料庫中抓出他們想要的句子。優點是他們可以找到同一文集中相似的句子，缺點是這種資料庫不能完全符合每一個人的需求，總是會不夠用，要一直加進新的句子。

- d. 額外的規則。這類的規則非常多，例如像” German” 這個單字，頭一個字母是大寫就意味著它可能是專有名詞。”，”,” :”,” !”,” .” ,這些

符號要和單字分開，” ’ ” 可能是所有格，要和前後單字連在一起然後再做詞性標記。

2. Stochastic approach :

通常會一起使用下列的幾種方法:

- a. simple approach : 對於語意模糊不清的字，我們就依照它出現的機率，給它一個特定的詞性。
- b. N-gram approach : 要標記這個字的詞性，是由它的前 N 個字出現的機率所決定。
- c. Viterbi algorithm : 用來找出標記一整句或一連串單字序列(word sequence)詞性的演算法。

2.3 Parser 軟體—Apple Pie Parser

語料庫的句法標注是語料庫語言學研究的前沿課題，它的處理目標是對語料文本進行句法分析和標注，形成樹庫 (treebank) 語料。在這方面，英語語料庫的研究已經做了許多工作，兩個比較大的研究專案是：英國 Lancaster 大學 UCREL 的 Lancaster Treebank[1][2]和美國 Pennsylvania 大學的 Penn Treebank[3]。前者的標記集較大，通過組織成不同的層次描述了詳細的短語句法資訊。而後者的標記集則較為簡節，其最大的特點是增加了四個表明不同空元素 (Null Elements) 的標記。而兩者樹庫規模都達到了二百萬詞以上。在這些工作中提出的骨架 (skeleton) 分析方法、機器自動分析和人工校對相結合的處理思路，為後續的相關研究積累了豐富的經驗。[25][26]

The Penn Treebank [4] 是美國賓夕法尼亞州立大學電腦與資訊科學系的 LINC 實驗室 (LINC Laboratory of the Computer and Information Science Department at the University of Pennsylvania) 所建立的一個附有詞性標記的英文語料庫，它將詞性分成 48 類的詞集 (Tag set)，列出在表 3 The Penn Treebank 詞性集分類一覽表：

解析軟體在國外非常多，常見的有 Minipar、CG Parser、CASS Partial Parser、CHILL、ISSCO Tools、IMS Stuttgart、Apple Pie Parser、Link Grammar Parser 等。而其中 APP 支援 Unix 工作站，加上程式原始碼及資料庫方便取得，且詞性分類採 The Penn Treebank 系統，分類適中易懂，所以本系統採用此軟體做為解析之工具。

APP 是一個 bottom-up probabilistic chart 方法的解析程式，它是用 best-first search 演算法來計算得到 parse tree 中最佳得分。採用和 Penn Treebank 同樣 tagset 的方式，利用 two non-terminals 的技術來解析英文文法。[11][12]

Tag	Description	Example	Tag	Description	Example
CC	Coordin. Conjunction	<i>and, but, or</i>	SYM	Symbol	<i>+, %, &</i>
CD	Cardinal number	<i>one, two, three</i>	TO	“to”	<i>to</i>
DT	Determiner	<i>a, the</i>	UH	Interjection	<i>ah, oops</i>
EX	Existential ‘there’	<i>there</i>	VB	Verb, base form	<i>eat</i>
FW	Foreign word	<i>mea culpa</i>	VBD	Verb, past tense	<i>ate</i>
IN	Preposition/sub-conj	<i>of, in, by</i>	VBG	Verb, gerund	<i>eating</i>
JJ	Adjective	<i>yellow</i>	VBN	Verb, past participle	<i>eaten</i>
JJR	Adj., comparative	<i>bigger</i>	VBP	Verb, non-3sg pres	<i>eat</i>
JJS	Adj., superlative	<i>wildest</i>	VBZ	Verb, 3sg pres	<i>eats</i>
LS	List item marker	<i>1, 2, One</i>	WDT	Wh-determiner	<i>which, that</i>
MD	Modal	<i>can, should</i>	WP	Wh-pronoun	<i>what, who</i>
NN	Noun, sing. or mass	<i>llama</i>	WP\$	Possessive wh-	<i>whose</i>
NNS	Noun, plural	<i>llamas</i>	WRB	Wh-adverb	<i>how, where</i>
NNP	Proper noun, singular	<i>IBM</i>	\$	Dollar sign	<i>\$</i>
NNPS	Proper noun, plural	<i>Carolinas</i>	#	Pound sign	<i>#</i>
PDT	Predeterminer	<i>all, both</i>	“	Left quote	<i>(‘ or “)</i>
POS	Possessive ending	<i>'s</i>	”	Right quote	<i>(’ or ”)</i>
PP	Personal pronoun	<i>I, you, he</i>	(Left parenthesis	<i>([, (, { , <)</i>
PP\$	Possessive pronoun	<i>your, one's</i>)	Right parenthesis	<i>([, (, { , <)</i>
RB	Adverb	<i>quickly, never</i>	,	Comma	<i>,</i>
RBR	Adverb, comparative	<i>faster</i>	.	Sentence-final punc	<i>(. ! ?)</i>
RBS	Adverb, superlative	<i>fastest</i>	:	Mid-sentence punc	<i>(: ; ... - -)</i>
RP	Particle	<i>up, off</i>			

表 3 The Penn Treebank 詞性集分類一覽表

所謂two non-terminals的技術指的是S (sentence) 及NP(noun phrase)兩個元素，因為S及NP兩個元素在Penn Treebank語料庫裡佔了非常大的比率，所以分析出此兩元素的組合規則，就可以解析出絕大多數的英文句子[10]。以下就以一個例子加以說明。

下圖 3 中是英文句子「This apple pie looks good and is a real treat.」的parser tree，其利用三條規則將其解析完畢：

Rule-1：NP -> DT NN NN。

Rule-2：NP -> DT JJ NN。

Rule-3：S -> NP VBX JJ CC VBX NP(其中第一個 NP 即 Rule-1 的 NP，第二個 NP 即 Rule-2 的 NP)，做出:structure “(S <1> (VP (VP <2> (ADJ <3>)) <4> (VP <5> <6>)))”;

最後，將 Rule-1 及 Rule-2 套入 Rule-3 的 structure 中，就得到最後的 parser 結果：(S (NP (NPL (DT This) (NN apple)) (NN pie)) (VP (VP (VBZ looks) (ADJP (JJ good))) (CC and) (VP (VBZ is) (NPL (DT a) (JJ real) (NN treat)))) (. -PERIOD-))

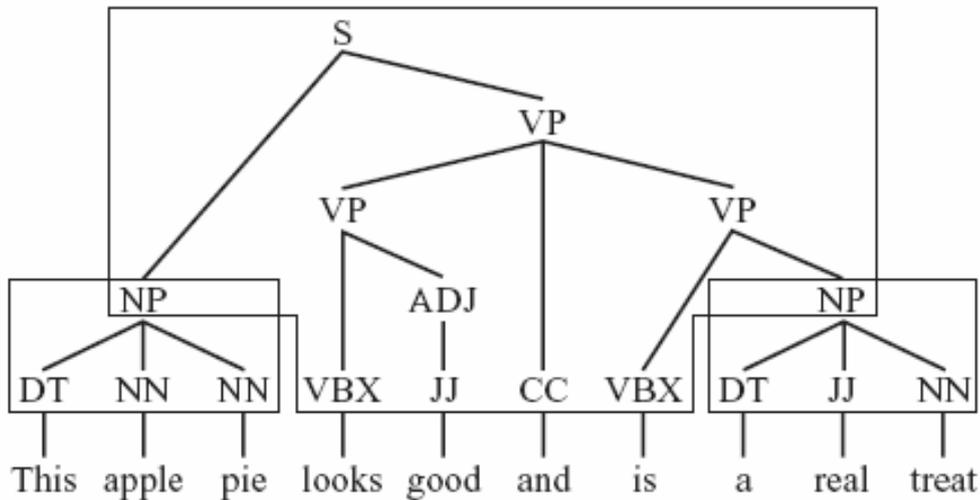


圖 3：Apple Pie Parser 解析樹



2.4 GNU Aspell

對大部份的使用者都非常有用的應用程式，無疑是自動拼字檢查器（automatic spell checkers）。GNU (GNU's Not UNIX) [15] Aspell 就是這樣的一個程式。

直到 Aspell 工程開始之前，只有一個拼字檢查程式是自由軟體：International Ispell。自然地，Ispell 結果在大部份的 Unix 機器上變得更加廣佈，而且也成爲 GNU 系統的標準拼字檢查器。不幸的是，由 Ispell 所作出的知識建議，明顯地比那些由私權拼字檢查器所作出的還要糟糕。爲了這個原因，Kevin Atkinson 在 1998 年開始致力於 Aspell。

在下一年，Kevin 付出了許多他的時間給 Aspell，更加密集的開發。在此過程中，許多觀念都被測試、整理，甚至有時候被捨棄。例如：他尋求創造一個稱爲 Pspell 的原生介面（generic interface），給目前存在系統內的所有拼字檢查器使用。不幸的是，這個原生介面反而把整個系統弄的更複雜，因此他就放棄了。

他付出了很大努力的其他事情還有，根據 Ispell 所提供的字彙列表，製作出更好的字彙列表（word list）。他很仔細地將英文列表分為英國、美國和加拿大的拼法（spelling）。那項工作的結果也被個別地加以公開。

在 2002 年 8 月發行 GNU Aspell 0.5 之後，Aspell 開始成為 GNU 工程的一部份，現在正努力尋求的是，希望在 GNU/Linux 散佈軟體上替換掉 Ispell。

GNU Aspell 不只可以極佳的使用者介面，而作為拼字檢查程式被使用，它也允許被其他程式作為程式庫而被連結。不管是作為一個程式使用或作為程式庫連接，GNU Aspell 都可察覺多進程的使用（multi-process aware）。一個使用者的個人字典，可以為該個使用者的所有 Aspell 進程所取得，而且任何改變，都會自動地在所有的 GNU Aspell 進程間互相增殖（propagated）。儘管如此，任何一個使用者，都可以對字典做出改進的結果，因為 GNU Aspell 會從使用者修正的錯誤中學習，並且讓每個使用者可以擁有數個個人字典。

同時 GNU Aspell 所作出的建議，也比那些由 Ispell 或 Netscape 4.0 或 Microsoft Word 97 所作出的更好。Kevin 為了將這更好的情形，定量地表示而開發了一個測試環境，這測試環境與它所作出的結果都可以在 GNU Aspell 網頁 [14] 上取得。

表 4 Aspell 與其他拼字軟體比較表[14]

	Aspell	Ispell	Netscape 4.0	Microsoft Word 97
Open Source	○	○		
Suggestion Intelligence	88-98	54	55-70?	71
Personal part of Suggestions	○	○	○	
Alternate Dictionaries	○	○	?	?
International Support	○	○	?	?

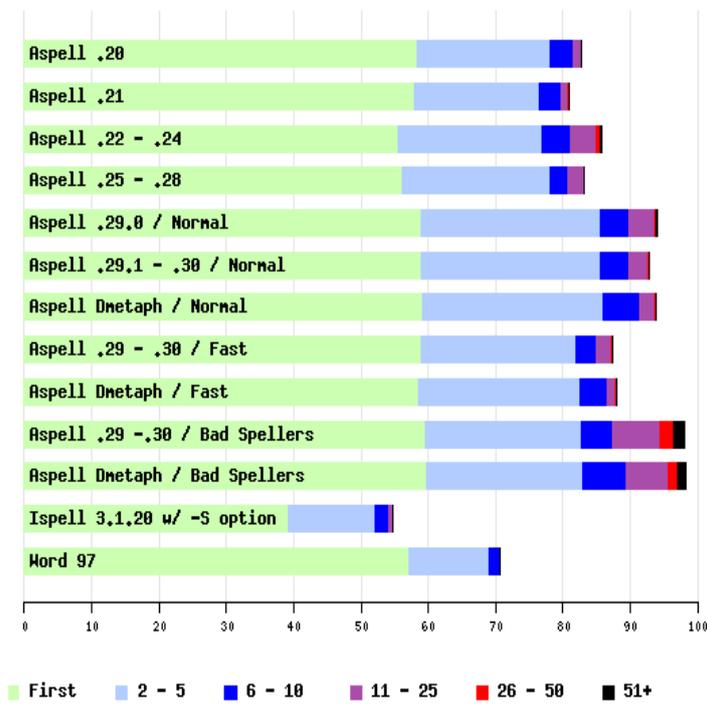


圖 4：Aspell與其他拼字檢查程式核心測試比較圖[14]

圖 5 為Aspell的輸出畫面

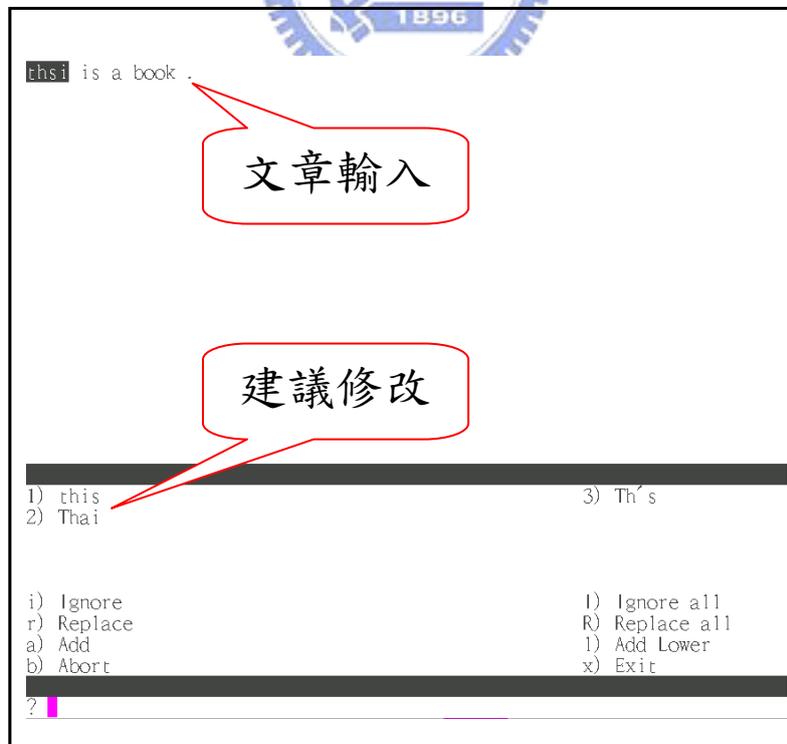


圖 5：Aspell 在 Unix 上執行畫面

三、系統需求分析

3.1 系統概念

本研究系統設計概念與流程主要是配合本實驗室研發之「二階段式試題適性值調整」方式，以使試題更能真實反應受測者之英語能力，其概念如圖 6所示。首先，分析試題並確定試題的評量目標，以多媒體試題開發設計單，確立試題所欲評量的能力層次與學習內容。試題開發設計單設計多媒體試題。透過本實驗室之試題檢測系統及命題專家檢核後，依其審定之適性值置入題庫。第二階段為施測，以本實驗室之IRT分析器分析測驗結果以調整或修正試題，並將所得之反應組型以現代測驗理論分析試題難度、鑑別度、猜測度等試題屬性，依據分析結果重新調整或組合試題以建立適性化題庫。由於進行施測後，立即可以獲得試題特徵曲線(ICC)與試題訊息曲線(IIC)(如圖 7)及試卷訊息曲線圖(TIC)(如圖 8)等各項參考數據，可隨時進行適性值之分析與調整題庫。因此任何一次施測皆可視為下一次測驗之預試，如此經試題檢測系統及專家審題後至入題庫，再經IRT分析器分析測驗結果與調整題庫，即為本實驗室所稱「二階段式試題適性值調整」之方式

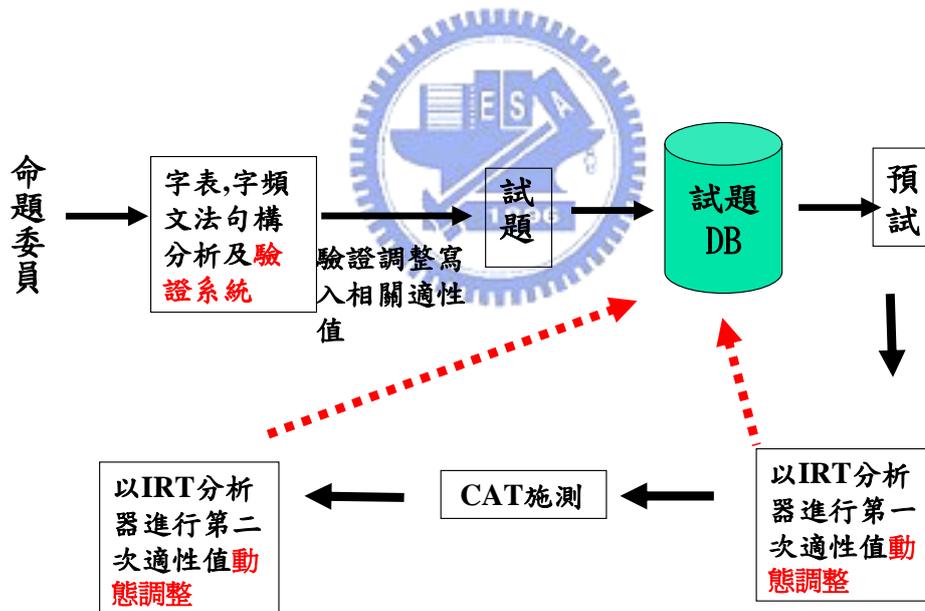


圖 6：同等級試題適性值的驗證與兩段式的動態調整流程圖

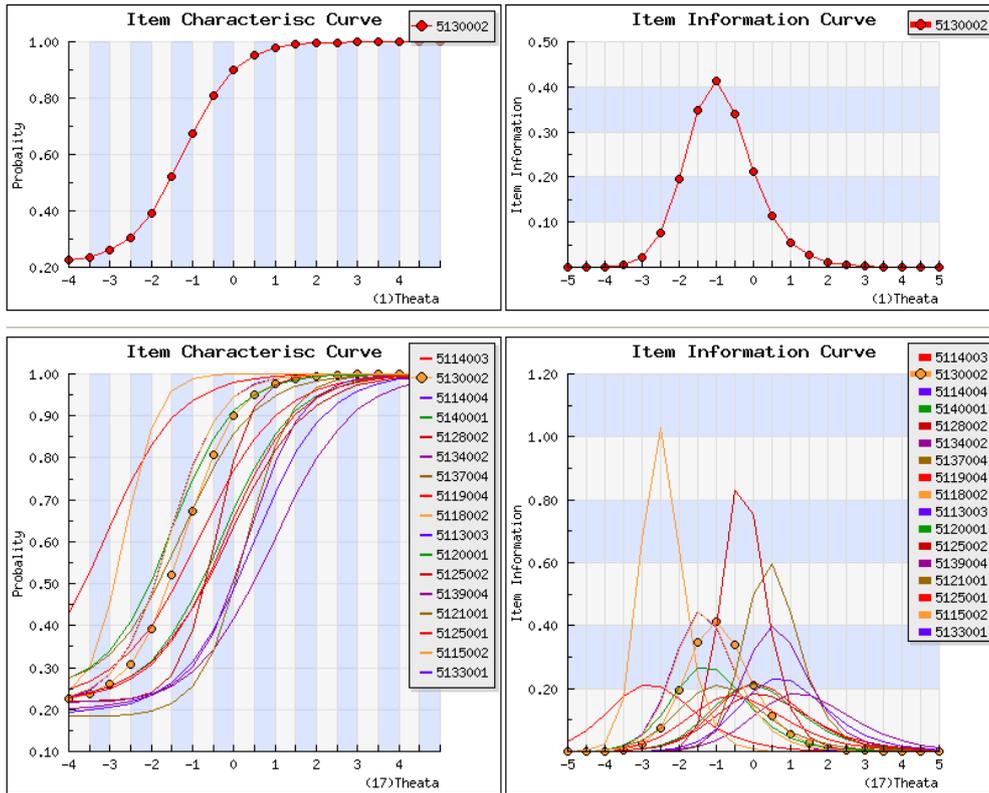


圖 7：即時試卷的全部試題的 ICC 與 IIC 圖比較

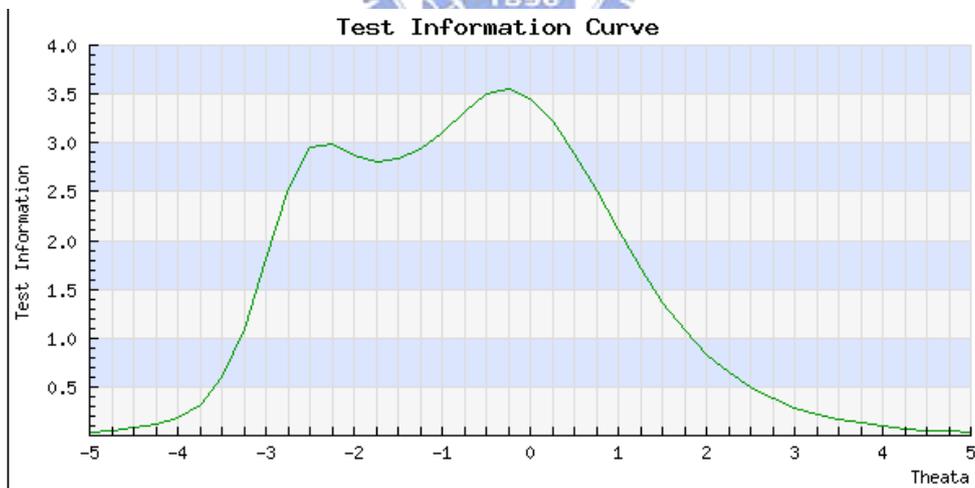


圖 8：試卷訊息曲線圖(TIC)

命題委員在進行試題命題時，可從 1.廣告 2.通告（含指示說明）3.會話（含閒聊、訪問、小組討論、電話會話）4.新聞播報（含敘述性故事、笑話）5.學術性演講（含演說、宣教）6.戲劇台詞（含影片對白）7.歌曲（含詩歌、童謠）…等方向獲得命題教材來源，而產生一個新的試題，至於題型可能是選擇題、填充題或問答題…等型式，但新的試題要如何歸至試題庫當中，放到適合的試題等級，這就是要靠「試題檢測系統」，做出一個判斷的建議值給命題委員，讓委員能夠迅速正確的將試題分級。

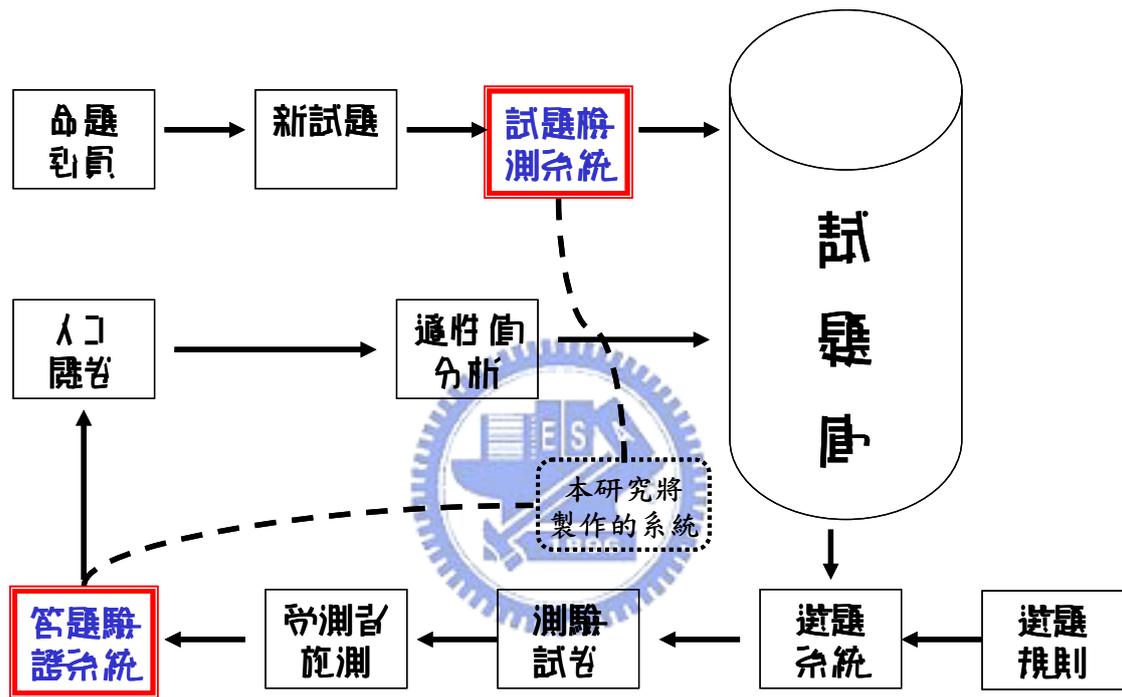


圖 9：系統概念示意圖

經分級後在試題庫中之試題，可以依每次不同等級及特性的測驗，訂定出良好的選題規則，再經由選題系統進行選題，挑出一份適合等級難易度的測驗試卷。此份測驗試卷就做為該場次測驗的依據，受測者則對測驗試卷進行施測，最後作答時間結束後，受測者所答題的部份，會經由「答題驗證系統」利用電腦對龐大的答題資料做初步的批改，系統批改出來的建議資料，會連同受測者的答題部份，一同給閱卷老師進行人工閱卷，閱卷老師可以依據答題驗證系統的建議或是自己的判斷進行給分，最後評定出受測者正確的得分。得到所有受測者的得分，可以再做試題的適性值分析，可能再重新調整試題的等級或難易度，讓試題可以調整到更適合的等級。

整個系統流程的概念如圖 9 所示，而本研究將專注於圖中「試題檢測系統」及「答題驗證系統」兩大部份進行研究，並於下節開始分別詳述之。

3.2 試題檢測需求分析

命題委員出一題新試題時，對於此新試題的題型、題目呈現、試題種類，就已經先行規範好，待試題檢測系統針對詞彙表、字頻、詞彙數進行計算，填入其屬性值後，將新試題納入試題庫內待測。

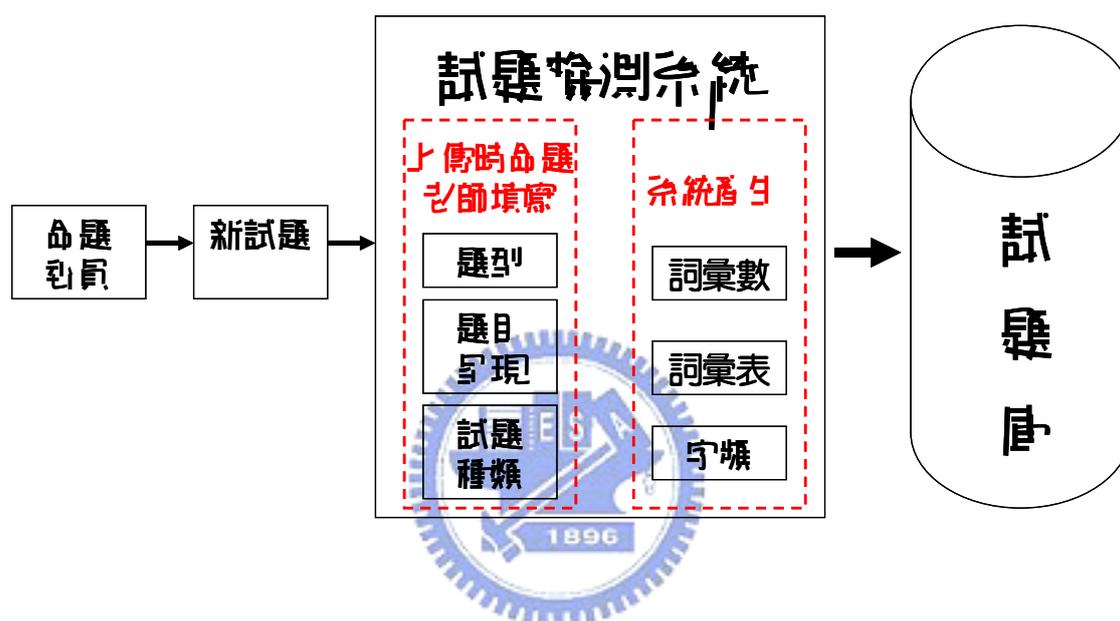


圖 10：試題檢測系統概念示意圖

3.2.1 輸入的檢測資料

在輸入檢測系統時，對於試題已有的資料包含：

1.題型：

選擇、填充、問答、題組。

2.題目呈現方式：

文字、聲音、圖片、動畫、影片。

3.試題種類：

聽力、閱讀、口語、寫作能力測驗。

3.2.2 檢測後的輸出資料

因為命題委員的生活與成長背景不同，所以對於每個試題的難易程度，在認知上大致不會完全相同，因此在此系統中提供三種較為客觀的資料，供命題委員最後判定出試題適當的級數。

1.詞彙表：

有學習過英文經驗的人，在參加任何測驗時，都不難體會到試題詞彙的難易度，會直接影響到該次測驗難易度。

所以教育部讓國中小學習英文時，就會公佈「國中小學英語最基本一千字詞」，這個字彙表的制定是參考民國八十三年教育部發布之「國民中學課程標準」參考字彙表及多國學校的詞彙表，最後再依我國中小學階段學生之生活經驗，及其學習英語之目標，與外語學習環境等因素加以篩選調整完成。

同樣的在負責我國學生最重要考試的大學入學考試中心，對於英文科的詞彙甚為重視，早已針對詞彙部份進行多年的相關研究，計有黃自來等（民 82）的《高中英文 5000 基本單詞》收錄於《大學入學考試中心英文試題與中學英語文教材之相關性參考題庫命題制度建立及命題人才培訓研究報告》、張武昌等（民 87）《高中常用字彙表》收錄於《八十四年度基礎科目英文考科試題研發工作計畫研究報告》、黃春騰等（民 89）《字彙表》收錄於《英文閱讀能力檢定考試規劃研究期末報告》等研究成果。近年來新字不斷擴增，還有其他多項因素考量，鄭恆雄等（民 91）乃參考上述研究成果、國高中教科書、美國小學教科書及世界各國的研究等完成《大考中心高中英文參考詞彙表編修研究計畫報告》。

使用詞彙表的好處：

- (1) 考試範圍明確。
- (2) 對於詞彙難易度有統一的參考標準。
- (3) 更科學地維持測驗歷次施測難易度平行，

2.字頻：

從詞彙研究的角度來看，詞彙統計研究已有很長的歷史了。古印度語言學家在研究婆羅門教的經典《吠陀經》時，就進行過單詞數目的統計。1898 年德國學者 F.W.Kaeding 編制了世界上第一部頻率詞典《德語頻率詞典》。1944 年，英國數學家 G.U.Yule 發表了《文學詞語的統計研究》，大規模地使用概率和統計方法來研究語言。1949 年，法國學者 R.Michea 提出建立“統計詞彙學”。1965 年，德國學者 R.D.Keil 把詞頻統計與現代統計學結合起來，提出了“詞彙計量學（lexicometric）”。

字頻就是一個字的使用頻率，也就是一個單字使用次數與所統計的材料的總字數的比例，一般用百分比表示，這樣可以比較直接地看出一個單字覆蓋面。

例如我們說一篇 2000 字的文章，其中“beauty”用了 78 次，那麼“beauty”的頻率就是： $78/(2000 \times 100) = 3.9\%$ 。3.9% 的概念就是說在一百個字中，就有 3.9 個是單字“beauty”，可見“beauty”的使用頻率是相對高的。摸清楚一個字的使用頻率，意義非常重要，編制教材，編寫字詞典，電腦資訊處理等，都要參照有關單字的使用頻率分析，依此作為依據。比如像小學的語文教材的編寫，首先就要選擇使用頻率高的字，如果遇到低頻字、罕用字，就要注意回避。

而在此處我們是使用字頻，是對單一個英文單字進行統計，而非使用「詞頻」，因為詞頻則須要先做切詞的工作，而切詞的工作困難度較高。

3. 詞彙數：

張郁慧（1996：135）的研究顯示，以英語為母語的人士，所寫的私人信函的英文詞彙平均長度為 6.03[13]。因此如果一個試題句子的詞彙數愈多，則代表此試題的難度亦有可能愈高，同樣的一位英文的初學者，其能閱讀句子的詞彙數是比較少的。所以我們藉由統計試題每句的詞彙數，來提供給命題委員參考，如果詞彙數多則有可能句子較難，如果詞彙數少則有可能句子較容易。

其檢測完成後輸出資料如圖 11 所示，圖中標號 I 的區域，就是詞彙表分析後的輸出資料；圖中標號 II 的區域，就是字頻分析後的輸出資料；圖中標號 III 的區域，就是詞彙數分析後的輸出資料。此三種分析資料輸出後，命題委員就依此分析資料判定出試題適當的級數。

難度檢測 - Mozilla Firefox

檔案(F) 編輯(E) 檢視(V) 瀏覽(O) 書籤(B) 工具(T) 說明(H)

I 句子爲: Nice to meet you. II

單字	字根	詞彙表	字頻	字頻比例
Nice	nice	2	23	0.4‰
to	to	1	1228	20.8‰
meet	meet	1	43	0.7‰
you	you	1	970	16.4‰
最大級數		2	字頻總合	58981

III 句子詞彙數: 4 此詞彙數句數: 491
比例數: $491/7700 = 6.4\%$

建議級數

完成

圖 11：試題檢測後輸出資料圖

3.3 答題驗證需求分析

受測者在簡答題作答完畢後，其答題的部份在閱卷老師批改過程中，是須要經過「拼字檢查」、「文法檢查」、「語義檢查」三大步驟：

3.3.1 拼字檢查

受測者在答題中常有許多拼字上的錯誤，例如：They do not have morden machines but they can still do it well. 其中誤拼成 morden 的字，有可能是 modern。這都是一些有可能犯的拼字錯誤。

受測者對於想使用的單字，可能一時想不起來，而無法確定這個單字的正確拼法，甚至是一時的筆誤，又沒做最後的檢查，以至於造成拼字上的錯誤。

因此我們利用 2.4 節所提到的 GNU Aspell 來幫忙完成此一工作，一個句子輸入 Aspell 後，其可以幫我們檢查出錯誤的拼字，並且列出此錯誤拼字有可能對的拼法。在此部份我們是純做拼字錯誤的檢查，至於字意使用不當或是用錯字，在此部份不做檢查，而列入下一節的文法檢查來完成。例如：I have much apples. 此句有文法使用上的錯誤，但完全沒有拼字錯誤，所以拼字檢查無法檢查出。

3.3.2 文法檢查

曾任暨南國際大學校長李家同在「專門替中國人寫的英文基本文法」一書中提到：

我們發現我們中國人寫英文句子時，會犯獨特的錯誤，比方說，我們常將兩個動詞連在一起用，我們也會將動詞用成名詞，我們對過去式和現在式毫無觀念。更加不要說現在完成式了。而天生講英文的人是不可能犯這種錯的。[22]

進而提出他認為中國人應該注意英文文法最基本規則：

規則(1)：兩個動詞是不能聯在一起用的。

規則(2)：如一定要同時用兩個動詞，後者的前面必須加“to”或者將後者加入“ing”。

規則(3)：主詞如果是第三人稱，現在式及單數，動詞必須加s。

規則(4)：絕大多數的否定的句子，不能直接加“not”。

規則(5)：在不定詞“to”的後面，必須用原形動詞。

規則(6)：英文中有所謂的助動詞。

規則(7)：英文問句要有助動詞。

規則(8)：特殊動詞隨主詞的變化。

諸如此類的英文文法真是非常多，也是在此答題系統最重要的一環。

而我們在此利用 2.3 節所提到的 Apple Pie Parser 程式，為我們解析出句子的句構及標記詞性，再利用程式去檢查其句構上種種的文法問題，進而達到文法檢查的目的。

3.3.3 語義檢查

在答題驗證過程的最後一步就是語義檢查，命題委員在命題時，對於該題的答案一定會有些標準答案或是答題的方向，而我們也必須針對這些部份進行檢查，檢查受測者在答題裡是否有符合標準的部份，有符合該題也才能給分，沒有符合當然該題不能算對。就像考作文一樣，如果洋洋灑灑的寫了一大篇的文章，可是文不對題，那老師一定不會給你分數的。

我們對於語義檢查的部份，就必須先由命題委員對於該試題的答題標準，先給數個答案的關鍵字，來做為語義檢查部份的標準答案，就像是選擇題出完題後，就會有一個標準答案是「C」一樣。再來利用 2.1 節提到的英語詞彙資料庫

—WordNet，針對關鍵字在資料庫中找尋出相對的同義字。最後，我們就將這些關鍵字與關鍵字的同義字來比對答題的內容，如果有用到這些字，表示有達到答題標準，就給予通過語義檢查；如果沒有，則表示沒有達到答題標準，不能通過語義檢查。



四、系統設計與實作

4.1 前處理

現在的電腦速度快，命題老師在試題的出題上，已經不再單單是以前的純文字，一定會爲了試題的生動與活潑，加入大量的圖片、聲音、影片，成爲多媒體試題，而此系統只單單做文字上的檢測，所以原始的試題需經由以下三道前處理的步驟，以簡化系統後續流程的工作。[30]

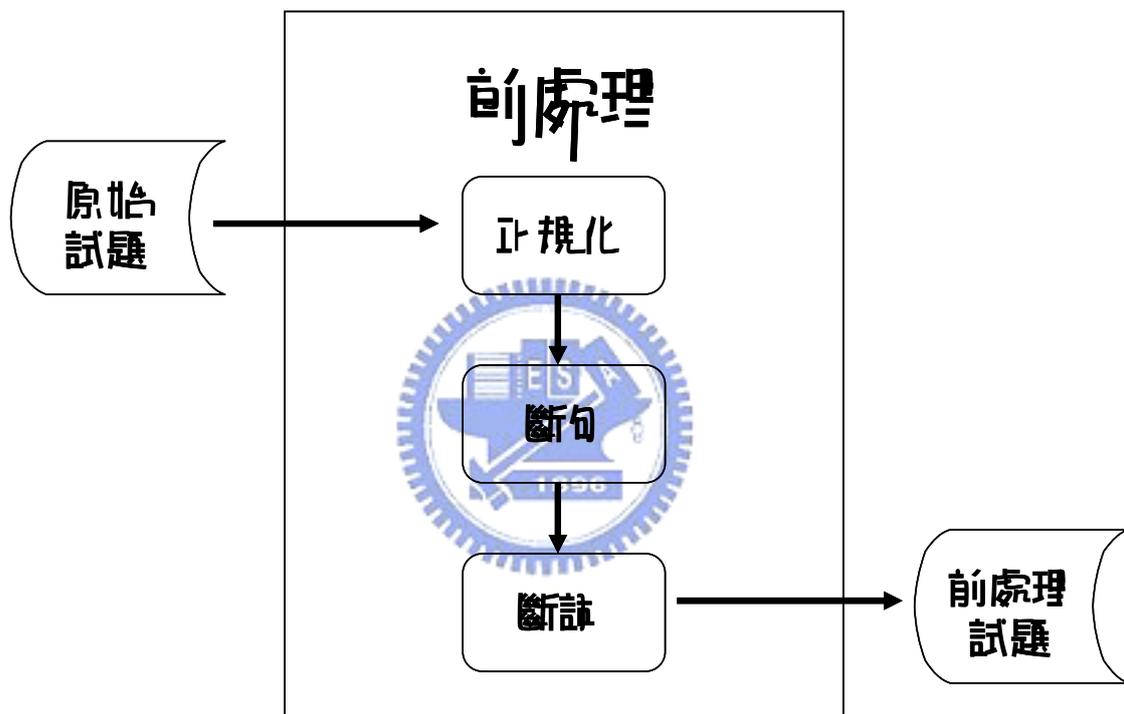


圖 12：前處理系統概念示意圖

4.1.1 正規化(normalization)

我們從系統得到試題或是答題，或許可能含有各式各樣的多媒體內容，我們須要從混合了文件標題、主題文字、圖片與影音檔案的網頁內，將主體文字的部分擷取出來，並以純文字方式儲存這些試題或是答題的文字內容。

而在目前現有的網路英檢試題中，都是在此系統概念提出前，就已經命題完畢，許多試題中的文字部份，都是以圖片格式製作而成(如圖 13)。所以在此次研

究中，無法輕易的將其轉換成純文字格式，我們是利用人工將圖片、影音檔案中，利用人工的方式，將其文字的部份取出來使用。

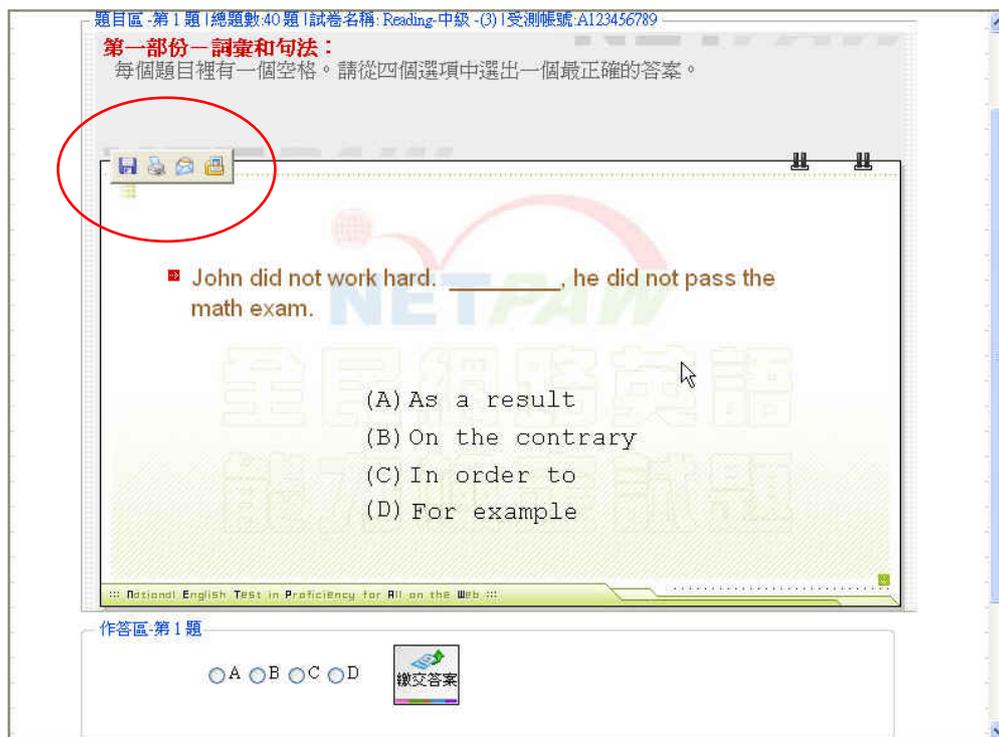


圖 13：網路英檢擬模試題畫面

4.1.2 斷句(sentence segmentation)

每個英文句子的結尾，皆是以句點或是問號來做為結束，就以此為斷句的依據，但須要排除一些例外的情形，例如：“Mr.”，使用到縮寫，則此時的句點並不能代表句子結束，必須要特別處理。

我們利用一小段副程式來做此部份工作，程式流程中主要是以句點及問號來做為斷句的依據。而在程式中如果遇到句點時，會去找尋程式中建立的縮寫資料，如果是縮寫就視為單字，直到有句點時又不是縮寫，則達到斷句條件，進行斷句工作。而目前縮寫資料只有生活中較常用到的縮寫，例如：Dr.、Miss.、Nov.、a.m.。

4.1.3 斷詞(tokenization)

斷詞的工作在英文句子處理中算是相當容易。由於英文的詞與詞之間是以空白(space)作為間隔，例外僅發生於兩個狀況，第一是大部份的文件格式中，標點符號會緊接在詞之後，無空白間隔；其次是縮寫詞的情形，常見的如“I”與“am”縮寫成“I'm”，“is”與“not”縮寫成“isn't”。因此斷詞的工作是

將標點符號與其所鄰接的詞分離，並將縮詞拆開。例如句子 “The stock didn’t rise \$5.” 經斷詞後成爲 “The stock did n’t rise \$5.”。名詞的所有格與阿拉伯數字並不在我們處理斷詞的考慮範圍內。譬如 “Kevin’s” 與 “1500”，皆是當成一個詞處理。

對於斷詞這個部份，我們是利用Apple Pie Parser程式中，有一個斷詞的副程式來幫我們處理。例如上段提到的例句” The stock didn't rise \$5.”，其就會完成斷句而成「(S (NPL (DT The) (NN stock)) (VP (VBD did) (RB n't) (VP (VB rise) (NPL (\$ \$) (CD 5)))) (. -PERIOD-))」

4.2 試題檢測設計與實作

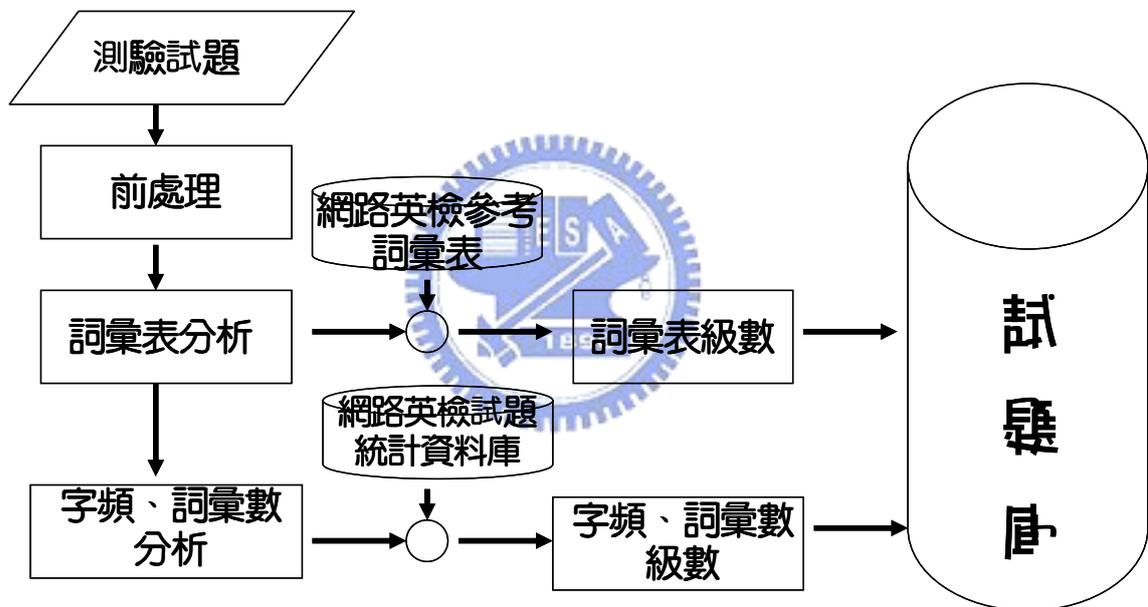


圖 14：試題檢測系統流程概念示意圖

試題檢測系統設計後其流程概念圖如圖 14 所示，命題委員命題好的測驗試題，經過上一節提到的前處理步驟，得到單純的文字模式進行系統檢測。首先依據「網路英檢參考詞彙表」進行詞彙表分析，分析出詞彙表級數，再依據「網路英檢試題統計資料庫」分別對字頻及詞彙數進行分析，分析出字頻及詞彙數的級數，提供給命題委員做爲判斷試題級數的參考數據。

4.2.1 詞彙表分析

依據「網路英檢參考詞彙表」，將字彙分為入門級、基礎級、初級、中級、中高級，「網路英檢參考詞彙表」主要參考兩份詞彙表資料，第一份是大學入學考試中心委託由台灣大學外文系鄭恆雄教授領導的高中英文參考詞彙表編修研究計畫小組，從八十九年九月起至九十一年六月止，以二年時間，編製出的「大學入學考試中心高中英文參考詞彙表」[20]；另一份是國立高雄師範大學英語系莊永山教授研發的「網路英檢入門級 500 字參考字彙表」及「網路英檢基礎級 1000 字參考字彙表」[28]。

我們參考其詞彙表分級方式，將詞彙由 Level 1~6 共分為六級：

Level 1：依據入門級參考詞彙共 500 字，主要參考「網路英檢入門級 500 字參考詞彙表」。

Level 2：依據基礎級參考詞彙共 1000 字，主要參考「高中英文參考詞彙表」之第一級，及「網路英檢基礎級 1000 字參考詞彙表」。

Level 3：依據初級參考詞彙共 2000 字，主要參考「高中英文參考詞彙表」之第二級。

Level 4：依據中級參考詞彙共 4000 字，主要參考「高中英文參考詞彙表」之第三、四級。

Level 5：依據中高級參考詞彙共 6500 字，主要參考「高中英文參考詞彙表」之第五、六級。

Level 6：超出上述 Level 1~5 所列詞彙。

1. 原型詞彙表：

因為高中英文參考詞彙表有 25 項之編輯原則，其中有 9 項原則說明許多詞彙為何不列入詞彙表之原則，所以我們將以上 Level 1~5 所列入詞彙稱之為「原型詞彙表」，建立一個資料庫來處理相關資料，其資料庫欄位格式如下表 5 所示：

表 5 原型詞彙表暨字頻資料庫結構一覽表

欄位	型態	校對	說明
wid	int(11)		記錄詞彙表序號
word	varchar(32)	utf8_general_ci	存放原型詞彙表單字
level	int(11)		記錄原型詞彙表單字的 Level
freqlv1	int(11)		記錄原型詞彙表字頻使用情形

2. 衍生詞彙表：

高中英文參考詞彙表共 25 項之編輯原則中，其中有 10 項原則說明許多詞彙為何不列入詞彙表[21]：

- (1) 由於英文形容詞大部分均可接詞尾-ly 變成副詞，故凡是含有詞尾-ly 之副詞均不列入。(除非意義不同，如 roughly 及 scarcely。)
- (2) 由於英文形容詞有許多可接詞尾-ness 變成名詞，故凡是含有詞尾-ness 之名詞均不列入。
- (3) 由於英文形容詞有許多可接詞頭 in-、im-、ir-、il-、un-等變成否定形容詞，故此種形容詞均不列入。(除非含此種否定詞頭之詞彙比不含此種否定詞頭之詞彙更常用，如 independence、infinite 及 innumerable 等。)
- (4) 由於英文接 non-(非)詞頭的形容詞或名詞數量甚多而且大多屬低頻詞彙，如 nonpoisonous (adj.)及 nonparticipant (n.)，同時其意義均可自其詞頭之意義「非」引伸出來，故除了 nonsense (n.) 及 nonviolent (adj.)兩個常用詞彙之外，其他均不列入。
- (5) 由於英文動詞有許多可接詞頭 re-表示重複動作，故凡是含有表示重複動作詞頭 re-之動詞均不列入。(但是 review 由於和 view again 意義不盡相同，故列入。)
- (6) 由於英文動詞有許多可接詞尾-ment 構成名詞，故凡是以-ment 構成名詞之動詞均把-ment 放在動詞後之括號內，表示該動詞可以接-ment 構成名詞，其詞類則標為 v./(n.)。(如果接-ment 之後，原來的動詞拼法改變，則整個含-ment 的詞彙重列一遍，如 argue (argument。))但是如接-ment 後意義不同，則列為不同詞彙，如 commit 及 commitment。
- (7) 英文有許多動詞可接詞尾-ing 或-ed 變形成容詞，因為相當規律，故均不列入。(除非意義不同，如 promising (有前途的)及 learned (有學問的)。
- (8) 英文有許多名詞可接詞尾-less 變形成容詞，如 fearless 及 homeless，因為相當規律，故均不列入。
- (9) 英文反身代名詞由於詞尾均為-self 或 selves，如 yourself 及 ourselves，相當規則，故不列入。所有格代名詞如 your、her、our 及 their 可以接 -s 成為 yours、hers、ours 及 theirs，故把這些代名詞列為 your(s)、her(s)、our(s) 及 their(s)，不分開列舉。至於 my，其相對應之代名詞為 mine，拼法甚不相同，故分開列舉。而 his 之相對應代名詞同為 his，故只列為 his。
- (10) 複合詞(compounds)因為大多數可從其組成詞彙瞭解其意義，故盡量不列入。但是有些複合詞難以從其組成詞彙瞭解其意義者，則列入本表，如 greenhouse、homework、household、housework 等。ought to 及 used to 因為當作助動詞使用，故各列為一個詞彙。

而這些不列入詞彙表之詞彙，也有被使用到的機會，故在系統中依據參考詞彙表的原型詞彙表，做出一個具有以上所列 10 個不列入原則，及相關詞類變化的衍生詞彙表，讓具有相同級數及詞類變化的詞彙，能夠相對應到適當的原型詞彙。

表 6 衍生詞彙表資料庫結構一覽表

欄位	型態	校對	說明
word	varchar(50)	utf8_general_ci	存放衍生詞彙表單字
original	varchar(50)	utf8_general_ci	衍生詞彙表所對映到原型詞彙表的單字

在衍生資料庫查詢過程中，其處理的原則是：

- (1) 在高中英文參考詞彙表中有的詞彙，則以此詞彙級數為主。例如：
commitment 其動詞是 commit，兩者皆在原型詞彙表中，故算其各自的 Level，所以 commitment 是 Level 5，commit 則是 Level 4。
- (2) 衍生詞彙則與其原型詞彙同等級，例如：abandonment 是 abandon 加上 ment 衍生而來，在原型詞彙中沒有，所以其就與 abandon 同算是 Level 4。
- (3) 複合詞主要是兩個詞彙組合而成，而其級數則是以當中最高詞彙級數為主。例如：bookmark 是由 Level 1 的 book 及 Level 2 的 mark 所組合而成，其級數則是以最高級數 mark 的 Level 2 為主。

3. 專有名詞詞彙表：

高中英文參考詞彙表之編輯原則中，對於專有名詞的處理原則如下：

專有名詞因為數量太多，難以取捨，而且大致不影響文章之理解，故均不列入。(除了五個專有名詞：English，因為本參考詞彙表的詞彙屬於這個語言，故列入；Mandarin 因為是國語，故列入；Confucius (孔夫子)，因為是中國之至聖先師，世界聞名，故列入；Christmas/Xmas，因為已經是普世皆知的節日，又因為 Xmas 常誤拼為 X'mas，故列入；Bible 因為是西方文化之基礎，故列入。)

然而專有名詞的出現也是常有的事，例如：Paris、Italy，如果依照先前詞彙表 Level 的分級，因為沒有列入原型詞彙表 Level 1~5，所以會被列入 Level 6，可是其詞彙的難度又沒有到達實際 Level 6 的難度，所以我們建立一個專有名詞詞彙表，特別將這些專有名詞標記出來，以免讓命題委員誤判試題級數。

其資料庫欄位如表 7，其只有存放專有名詞的單字，如果有查詢到則代表其是專有名詞。

表 7 專有名詞資料庫結構一覽表

欄位	型態	校對	說明
word	varchar(64)	utf8_general_ci	存放專有名詞的單字

而「原型詞彙表」、「衍生詞彙表」及「專有名詞詞彙表」之間的關連性如錯誤! 找不到參照來源。所示。任何一個詞彙要查詢詞彙表時，是先查詢衍生詞彙表資料庫，找到原型詞彙後，才再對映到原型詞彙表中，查詢出詞彙的級數。而專有名詞資料庫在查詢時就較為簡單，直接去查詢資料庫，如果資料庫有列到，則就算題專有名詞。

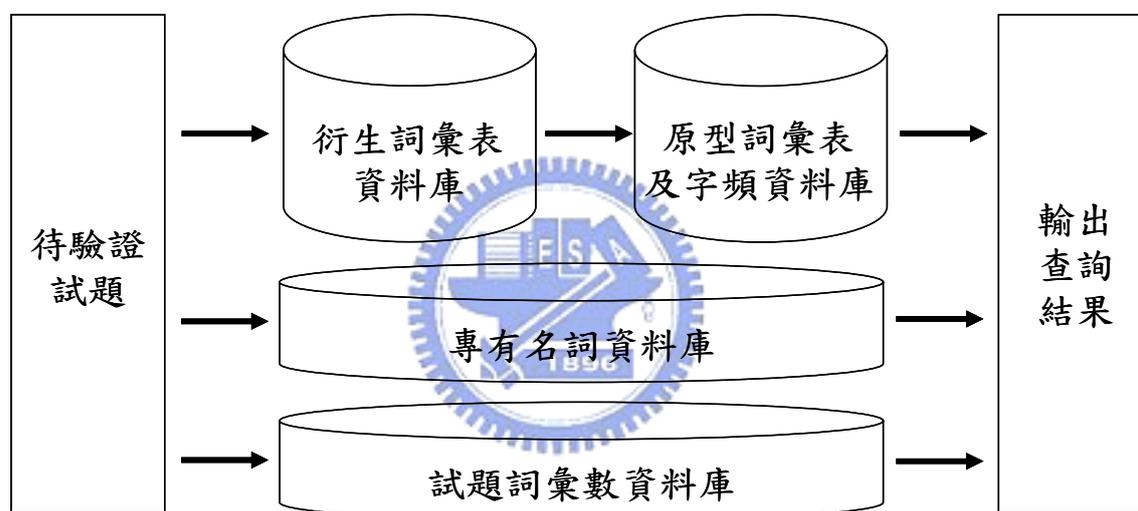


圖 15：試題檢測系統內資料庫關連示意圖

4.2.2 字頻分析

依據現有的網路英檢試題資料庫，先將其試題的資料，依照上一節詞彙表查詢方式，進行字頻統計，做為往後字頻分析的參考數據。只要有對映到原型詞彙表資料庫中的詞彙，則該詞彙的字頻統計數字加一，也就是在表 5 原型詞彙表資料庫中「freqlv1」欄位名稱中的數字加一。例如：某一個試題是「How are you?」，就分別在「how」、「are」、「you」三個單字的字頻數加一。

如果只是單純統計字頻的數字，就只能看到這個單字在網路英檢試題資料庫被使用的次數，例如：this 這個字的字頻是 339，這樣的一個數字，似乎看不出什

麼端倪，339 到底是多還是少呢？因此我們還做出一個字頻統計的比例分析，是以字頻去除以所有原型詞彙表字頻的總合，並以千分一的單位來顯示。

4.2.3 詞彙數分析

經過前處理完畢的試題，就會分出一個一個純粹的單字，利用資料庫來統計每個不同詞彙數試題出現句子的次數，如下表 8 資料庫欄位資料，例如：**Hunting and fishing are my hobbies.**，此句經分析後，得到詞彙數是 6 個字的句子，所以其在資料庫中的wordnum是 6，因此可以在sentnum中查到資料庫中曾經有多少句。

表 8 詞彙數資料庫結構一覽表

欄位	型態	校對	說明
wordnum	int(10)		一個句子的詞彙數
sentnum	int(10)		記錄句子詞彙數的數量

4.2.4 實作程式流程

了解以上對於詞彙表分析、字頻分析及詞彙數分析的設計與實作概念後，現在針對試題檢測如圖 16 所示的整個系統程式流程，做個詳細的說明。

首先，待檢測試題經過前處理後，一整個句子的試題，就會被切出一個個的詞彙，而去計算這個句子中的詞彙數共有幾個，就可以去詞彙數資料庫中，去查詢出這樣詞彙數的句子，在資料庫當中出現過幾句，查出結果後，就直接將此結果輸出，就完成詞彙數分析的部份。

同樣的在經過前處理得到的這些詞彙，接著會將這些詞彙送入衍生詞彙資料庫查詢，查詢出該詞彙的所對映到的原型詞彙，例如：**books** 是名詞的複數，就會查出原型詞彙是 **book**；**swam** 是過去式或過去分詞，就會查出原型詞彙是 **swim**。

從衍生詞彙中對映出原型詞彙後，下一步就可以送至原型詞彙表及字頻資料進行查詢動作，其後就可以查詢出此原型詞彙的詞彙表級數與字頻，而完成詞彙表分析及字頻分析。

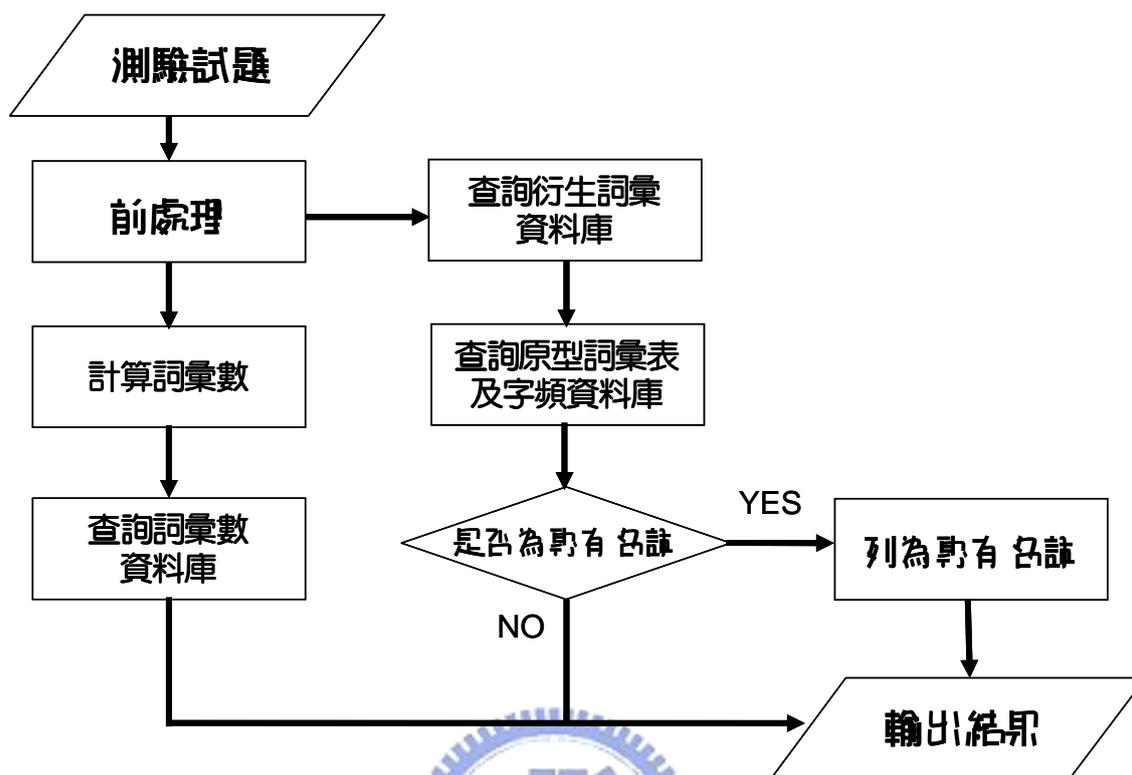


圖 16：試題檢測系統程式流程圖

如果經過上述過程，在原型詞彙表中查詢不到的詞彙，那就有兩種情形：一種是專有名詞；另一種是沒有列入詞彙表中，而算是 Level 6 的詞彙級數。所以我們接著將這些原型詞彙表中查詢不到的詞彙，可以先判斷其字首若為大寫，則可列為專有名詞，送到專有名詞資料庫中查詢，如果有查詢到，則列為專有名詞；如果沒有查詢到，則列為 Level 6 的詞彙級數。

最後試題檢測系統程式執行完成後，其輸出畫面如圖 17 所示，其畫面中各部份內容的意義，將會於第五章系統展示中詳細說明。

難度檢測 - Mozilla Firefox

檔案(F) 編輯(E) 檢視(V) 瀏覽(O) 書籤(B) 工具(T) 說明(H)

句子爲: You swam in the river last Sunday.

單字	字根	詞彙表	字頻	字頻比例
You	you	1	958	16.3‰
swam	swim	2	5	0.1‰
in	in	1	541	9.2‰
the	the	1	2142	36.5‰
river	river	2	23	0.4‰
last	last	1	306	5.2‰
Sunday	Sunday	1	14	0.2‰
最大級數		2	字頻總合	58677
句子詞彙數: 7		字數級: 18	比例數: 2.9%	
建議級數		2	確認	

完成

圖 17：試題檢測系統程式輸出畫面

4.3 答題驗證設計與實作

4.3.1 系統概念流程圖

受測者在簡答題作答完畢後，其答題的部份，須進入答題驗證系統，開始進行閱卷，其流程圖如圖 18說明如下：

1. 前處理：

首先，須經過前處理，主要是要處理斷句及斷詞的部份，處理完成為單一句子後，再做接下來各項的檢驗程序。

2. 拼字檢查：

進行拼字檢查過程中，沒有發現任何一個拼字錯誤視為合格，否則只要有一個錯誤出現，均視為不合格。檢查只要是不合格者，則直接送至評分系統，結束此句接下來的驗證工作。如果檢查合格者送至下一步驟進行標記詞性。

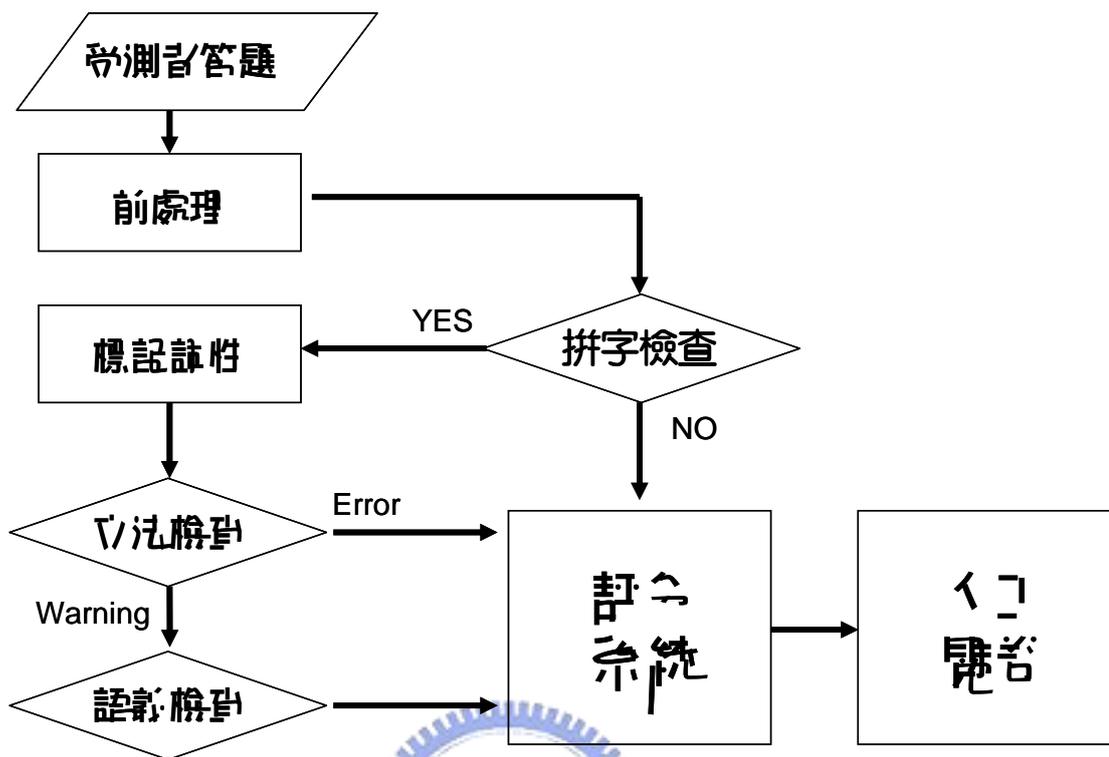


圖 18：答題驗證系統概念流程圖

3. 標記詞性：

標記詞性就是一次看一個句子的每一個字，然後給每一個字一個標記，也就是該單字的詞性。而標記詞性完畢的句子即可送至文法檢查。

4. 文法檢查：

進行文法檢查的結果有分為「error」及「warning」。結果出現 error 時，表示句構上有嚴重錯誤，若送至下一步文法規則檢查會有問題，應立即停止檢查步驟，將結果送至評分系統進行評分；若為 warning 則表示此文法錯誤仍在錯誤容許範圍內，可以繼續送至下一步文法規則檢查，或進行語義檢查。

5. 語義檢查：

語義檢查則是針對受測者答題部份檢查，是否有符合標準答案部份進行檢查，檢查完畢的結果送至評分系統做最後的記分。

6. 評分系統及人工閱卷：

在評分系統中統計出拼字檢查、文法檢查及語義檢查的參考結果，顯示給閱卷老師看，閱卷老師參考其結果後，再進行最後的人工閱卷，確認以上檢查結果，最後再給定確切的評分分數。

4.3.2 使用態連結函式庫

系統程式中採用動態連結函式庫(Dynamic Linking Libraries, DLLs)的概念，使用動態連結函式庫的好處大致可以如下歸類[29]：

1、有效率的重複使用程式碼：

它提供了一個做法，讓我們不需要把應用程式的執行檔變得如此龐大，但一樣可以享用這些使用頻率高的函式。也就是說這些函式會在程式執行時才被載入，而不是直接編譯在執行檔中。這樣一來可以讓我們更有效率使用這些函式。

2、模組化：

用動態連結函式庫使其模組化，若有朝一日發現這些被包裝在動態連結函式庫之中的函式的實作方法有點錯誤，或是發現有更好的做法時，只需要更動部份的動態連結函式庫原始碼，重新將修改過的動態連結函式庫給編譯後就可以達成更新程式的目的，至於應用程式端連動都不需要動一下。

3、節省記憶體的使用量：

由自己的程式來控制動態連結函式庫的使用時機。動態連結函式庫的載入是在應用程式執行時才被載入，甚至還可以是可以在應用程式所需用到函式時才被載入。當沒有用到函式庫時，就會從記憶體裡頭消失，如此是個不錯能夠節省記憶體與系統資源的做法。

4.3.3 系統程式架構

系統程式架構圖如圖 19，其中GNU Aspell是在做拼字檢查的部份；Apple Pie Parser及Apple Pie Parser's Parser 是在做詞性標記及解析句構的工作；利用 WordNet做出語義檢查的資料庫主要存在DataBase中；Module 大部份是在做文法檢查的工作，其中一小部份負責與GNU Aspell及DataBase做溝通。各部份的工作內容及動作詳述如下：

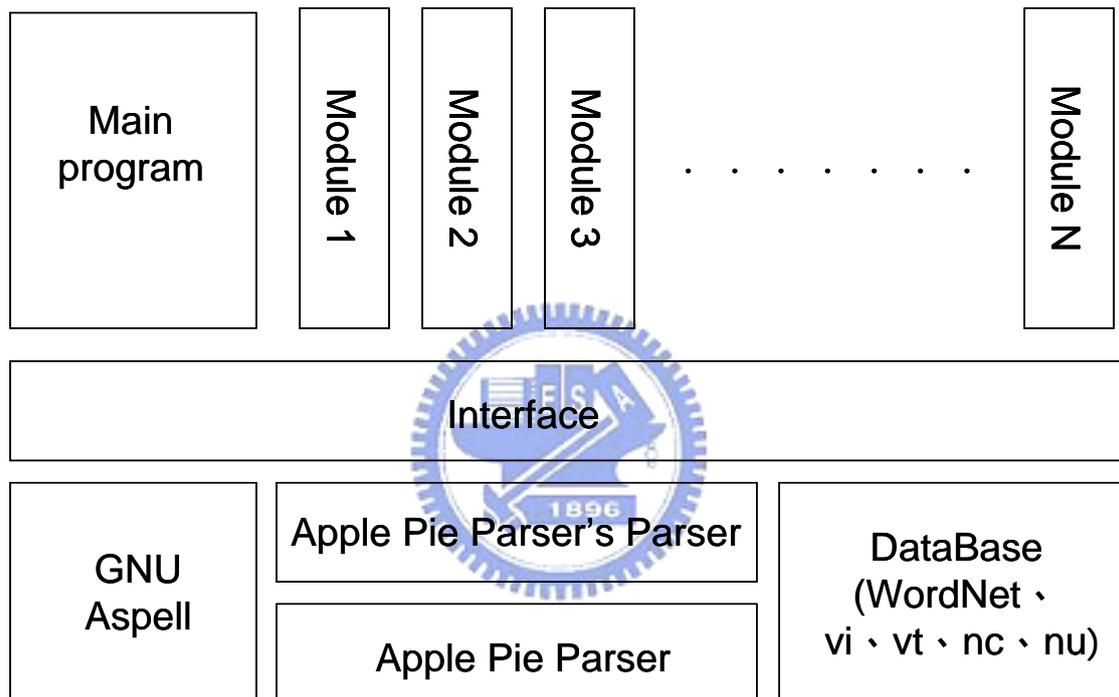


圖 19：答題驗證系統程式架構圖

1. GNU Aspell：

我們利用 Kevin Atkinson 在 1998 年開始製作至今的 GNU Aspell，來為我們做拼字檢查的工作，其最後一個版本是 2004 年 12 月 GNU Aspell 0.60.2 版。拼字檢查主要是在做純粹單字拼字造成錯誤的檢查，例如：應該是 modern，就有可能誤拼成 morden 的拼字錯誤。GNU Aspell 是一個免費而且又是 Open Source 的拼字檢查程式，他既可以單獨當做拼字檢查的軟體，也可以當做一個函式庫供程式呼叫使用，所以在使用上非常的方便。並且提供了一個功能，可以猜測拼錯字可能的結果，給予一些建議的單字，提供給我們使用者更多的參考資訊。

以下幾個是常用到的函式庫：

(1) `aspell_new`

功用：載入一個新辭典。

語法：`int aspell_new(string master, string personal);`

說明：`Aspell_new()` 開放一個新辭典且傳回辭典連結 `identifier` 給其它拼字函式使用

(2) `aspell_check`

功用：檢查一單字。

語法：`boolean aspell_check(int dictionary_link, string word)。`

說明：`Aspell_check()` 檢查一單字的拼法，若拼法是正確的則傳回 `true`，若是錯誤的則傳回 `false`。

(3) `aspell_check-raw`

功用：檢查一單字不改變它的事實或試著去修改它。

語法：`boolean aspell_check_raw (int dictionary_link, string word)。`

說明：`Aspell_check-raw()` 檢查一個單字的拼法，不改變它的事實或試著去修改它，若拼法是正確的則傳回 `true`，若是錯誤的則傳回 `false`。

(4) `aspell_suggest`

功用：建議單字的拼法。

語法：`array aspell_suggest (int dictionary_link, string word)。`

說明：`Aspell_suggest()` 將單字可能的拼法放入陣列變數中傳回。

2. Apple Pie Parser：

此部份主要是 **Apple Pie Parser (APP)** 的主程式，負責為答題句子做解析之用。**APP** 是一個統計式的解析程式，其與傳統的 **rule-based parser** 所有不同，他解析句子的文法規則是由語料庫中統計而得，並且統計出每個文法規則的權重值。其在進行解析時，就是根據每個文法規則的權重值，來決定是增加或減少總權重，最後得到一個適當的總權重值，對映到相關的文法規則，而解析出最後的文法解析樹。由於 **APP** 使用這種統計權重值的特色，因此在解析句子時，並沒有所謂的對與錯，無論如何都可以解析出來。

APP 主要做解析的動作，像是輸入一個句子「**This apple pie looks good and is a real treat.**」，可以得到以下的解析結果：**(S (NP (NPL (DT This) (NN apple))**

(NN pie) (VP (VP (VBZ looks) (ADJP (JJ good))) (CC and) (VP (VBZ is) (NPL (DT a) (JJ real) (NN treat)))) (. -PERIOD-))

3. Apple Pie Parser's Parser :

Apple Pie Parser's Parser (APPP)此部份是將 APP 對一個句子解析完畢後的解析樹，再做進一步的處理，把解析樹處理成我們想要的資料結構。APPP 的過程中主要是利用 lex & yacc 來處理這個部份，把 APP 輸出的解析樹切成各別的 token，再依每個 token 之間的組合方式，建成我們希望的資料結構。

APPP 如此做法還可以達到一個好處，就是往後如果將解析的主程式 APP 換掉後，所有的程式結構不須要重寫，而只要修改 APPP 這個部份，讓其輸出仍然符合現在的資料結構，如此就可以讓整個系統繼續正常運作。

以下利用圖 3 的解析樹來說明 APPP 切出 token 的結構：

- (1) LEAF：為 tree 的最末端，左端是 tag，右端是單字，例如：(NN apple)、(NN pie)、(CC and)皆是。
- (2) NODE：可以是 LEAF 或(tag + NODE_LIST)的結構，例如：(NPL (DT This) (NN apple))就算一個符合(tag + NODE_LIST)的結構，而算是一個 NODE。
- (3) NODE_LIST：可以是 NODE 或 NODE_LIST + NODE 的結構，例如：(DT This) (NN apple)就算一個符合 NODE_LIST + NODE 的結構，而算是一個 NODE_LIST。
- (4) TREE：是最頂端的結構，例如：(S....)。

APP 的輸出結果經過 lex & yacc 用以上的規格，切出 token 後，就可以在 APPP 中自己建立出一個解析結果的樹狀結構，就如圖 20 所示的一個解析樹，其結果和圖 3 說明中的解析樹一樣，如果一來，我們的程式就可以依此樹結構去做許多的動作，另外，有了這個解析樹的顯示，在做程式時可以更容易了解句子的結構，進而找出問題所在之處。

```

This apple pie looks good and is a real treat.
(S)
  (NP)
    (NPL)
      (DT [This])
      (NN [apple])
      (NN [pie])
    (VP)
      (VP)
        (VBZ [looks])
        (ADJP)
          (JJ [good])
      (CC [and])
      (VP)
        (VBZ [is])
        (NPL)
          (DT [a])
          (JJ [real])
          (NN [treat])
  (. [-PERIOD-])

```

圖 20：Apple Pie Parser's Parser 建立出解析樹

4. Interface：

建立出一些 class，做為上層程式與下層程式間的介面，讓上下層程式的工作分開。而上層程式指的是 Main program 及 Moodle，下層程式指的是 GNU Aspell、APP、APPP 及 DataBase。其建立出的 class 介紹如下：

(1) class ConfirmingSentence

一個紀錄每次驗證資訊的容器，其實也就是 Message。可以讀取或新增程式回傳的錯誤訊息，但無法修改已儲存的錯誤訊息資料，並提供許多模組化許須的操作。各個模組與主程式間亦透過此物件來溝通。

(2) class Sentence

基本的一個句子，把一串句子切成 token，並產生出句構。此 class 無法被同類以外初始化，必需透過 ConfirmingSentence::createSentence 產生 factory pattern。這麼做主要是為了模組化的考量。

(3) class Message

紀錄一次驗證失敗的結果，包函被驗證的句子，以及其出錯的 token 位置，還有記錄錯誤訊息。

(4) class CNode

處理句子結構的 tree 中 node 的各項動作，例如：取得 tag、判斷目前 node 是否為 leaf、取得 node 所有的 children。

(5) class CLeaf:public CNode

繼承自 CNode。處理 leaf 中的各項動作，例如：取得 tag、取得 leaf 的單字、得到 leaf 是在句子中第幾個字。

5. Main program :

依照動態連結方式，從設定檔中呼叫每個函式庫，依次序一一執行。

6. Module :



在做拼字檢查、文法檢查及語義檢查時，所須要做的程式動作，都各別放在一個 Module 程式當中。一個 Module 程式是動態連結函式庫模組化程式的一部份。因為採用動態連結方式，所以每一個 Module 程式都算單獨一個程式，如果須要修改其中一個 Module 的程式內容時，則只須要對該 Module 程式重新進行編譯。當日後有需要再新增 Module 時，只要在 Main program 的設定檔增加此 Module 名稱的設定，整個程式即可新增出此 Module 程式的動作，大大的增加日後擴充程式的方便性。

7. DataBase :

存放每個 Module 程式有可能用到的資料庫，主要有兩大類資料庫，一是 WrodNet 同義字的資料庫，另一個詞性輔助資料庫。其說明如下：

(1) WrodNet 同義字資料庫：

就如 2.4 節 WrodNet 相關說明資料中提到，WrodNet 可以查到單字的同義字，而且其資料是開放使用的，所以我們取其同義字的部份建成資料庫，以方便程式在做語義檢查時直接查詢使用，其資料庫欄位資料如表 9 及表 10 所示。

表 9 wordnet_index 資料庫結構一覽表

欄位	型態	校對	說明
word	varchar(70)	utf8_general_ci	單字
relate	int(11)		單字語義代號
type	enum('noun','verb','adj','adv')	utf8_general_ci	單字詞性

表 10 wordnet_data 資料庫結構一覽表

欄位	型態	校對	說明
id	int(11)		單字語義代號
word	varchar(70)	utf8_general_ci	同義字
type	enum('noun','verb','adj','adv')	utf8_general_ci	同義字詞性

其資料庫查詢的方式，首先在 wordnet_index 資料庫 word 欄位查出單字相對映出單字語義的代號，再依此單字語義的代號，可以在 wordnet_data 資料庫中查到此語義的同義字，所以我們就以利用以下的指令查詢出「happy」的同義字。

```
SELECT `wordnet_index`.`type`, `wordnet_data`.`word`
FROM `wordnet_data`, `wordnet_index`
WHERE `wordnet_index`.`word` LIKE 'happy'
AND `wordnet_index`.`relate` = `wordnet_data`.`id`
AND `wordnet_index`.`type` = `wordnet_data`.`type`
ORDER BY `wordnet_data`.`type`, `wordnet_data`.`word`
```

查詢「happy」的同義字的結果如表 11，其結果與圖 2 所示查詢結果一致。查詢結果中有許多重覆的字，可以另外經過程式處理將其省略。

表 11 WrodNet 同義字資料庫查詢「happy」同義字結果一覽表

type(詞性)	word(詞彙)
adj	euphoric
adj	felicitous
adj	happy
adj	pleased
adj	well-chosen

(2) 詞性輔助資料庫：

因為在 APP 中解析出來的詞性標記，是依據 Penn Treebank 所分的 48 類詞集，如此詞集分類的方式，對我們要做更細部的文法檢查而言是不夠的，所以須要一個存放更詳細的英語單字詞性分類的資料庫，以供在做文法檢查時，必須做出更細部判斷資料時使用。舉個例子來說明一下，例如句子 I have much money. 解析為(S (NPL (PRP I)) (VP (VBP have) (NPL (JJ much) (NN money)))) (. -PERIOD-))，其中 money 的詞性標記為 NN(單數名詞)，而對於此句要做「much + 不可數名詞」文法規則的檢查，就須要知道 money 是屬於可數名詞還是不可數名詞，如此就須要這個詞性輔助資料庫的幫忙。而其資料庫有以下三大類：

- A. 名詞(noun)資料庫：資料庫當中對名詞做更詳細的分類標記，其分類標記包含有：專有名詞、專有名詞複數、不可數名詞、可數名詞的單數、可數名詞的複數、可以當成可數或不可數的名詞。
- B. 動詞(verb)資料庫：資料庫當中對動詞做更詳細的分類標記，其分類標記包含有：及物動詞、不及物動詞、可以當做及物或不及物動詞、動詞基本型、過去式、動名詞/現在分詞、過去分詞、現在式、現在式第三人稱單數。
- C. 人稱代名詞(Personal Pronouns)資料庫：資料庫當中對動詞做更詳細的分類標記，其分類標記包含有：第一二三人稱、單數、複數、主格、受格、所有格。

其資料庫結構如表 12 所示：

表 12 詞性輔助資料庫資料庫結構一覽表

欄位	型態	校對	說明
word	binary(30)		單字
type	varchar(4)	utf8_general_ci	詞性代號

4.3.4 Module 程式說明

在此將針對章節4.3.3 系統程式架構中Module的部份，舉出三個Module實例做細部的說明。

1. 語義檢查 Module：

此部份程式流程圖如圖 21所示。語義檢查最主要是跟關鍵字及其同義字做比對，所以程式一開始時，先取出數個關鍵字去查詢出其同義字，並將其查詢結果分別存放在兩個字串陣列中。

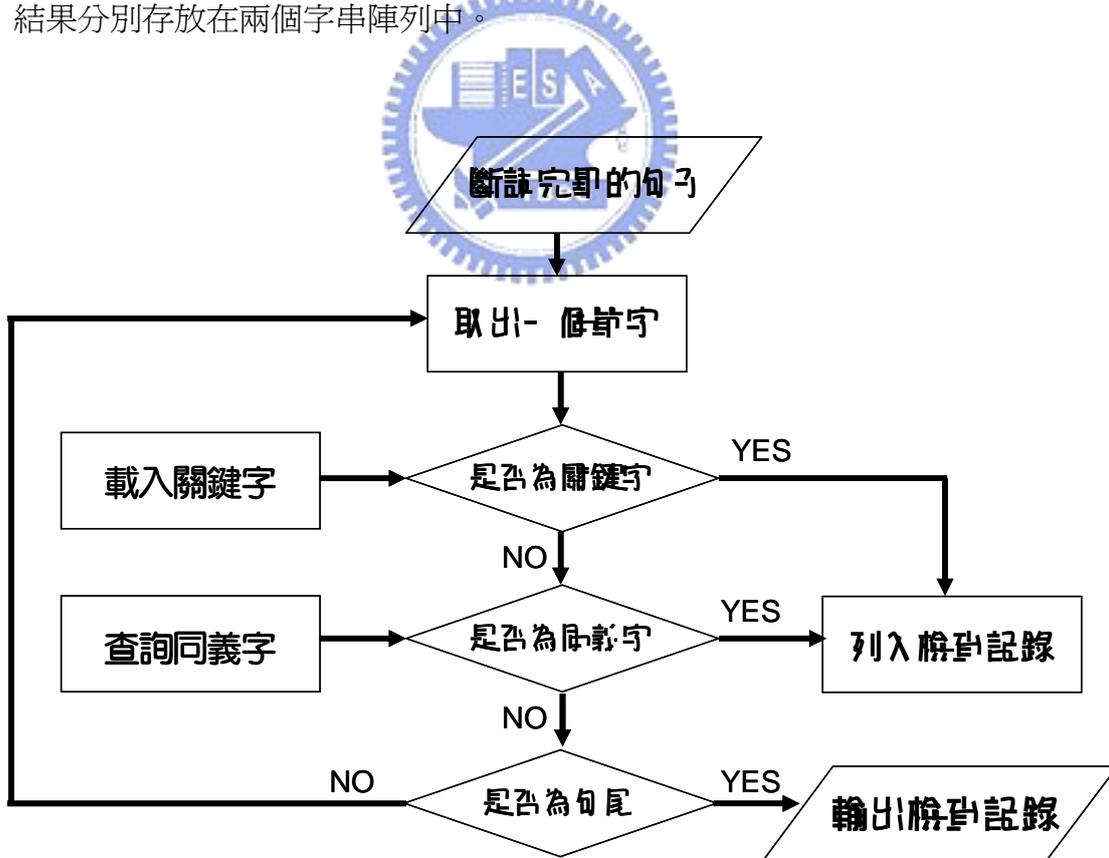


圖 21：語義檢查 Moudle 程式流程圖

接著在已經先斷詞完畢的句子中，一次取出一個單字，去和關鍵字及同義字比對，如果有相同時，記錄其結果是與關鍵字相同或與同義字相同。一個單字比對完畢後，再取出下一個進行比對，直到句子所有單字皆比對完畢，到達句尾時，即結束此一 Module 程式。

實際輸出畫面將在章節 5.2.3 答題驗證系統範例三中，有詳細說明。

2. 形容詞與名詞單複數不一致 Module :

此形容詞與名詞單複數不一致Module是文法檢查其中的一個Module，流程圖如

圖 22所示。形容詞與名詞單複數不一致中，其主要是做以下二個文法規則：

[many, few, a few, several]+ 複數名詞。例句：He knows several languages.

[much, little, a little]+ 不可數名詞。例句：I haven't much money.

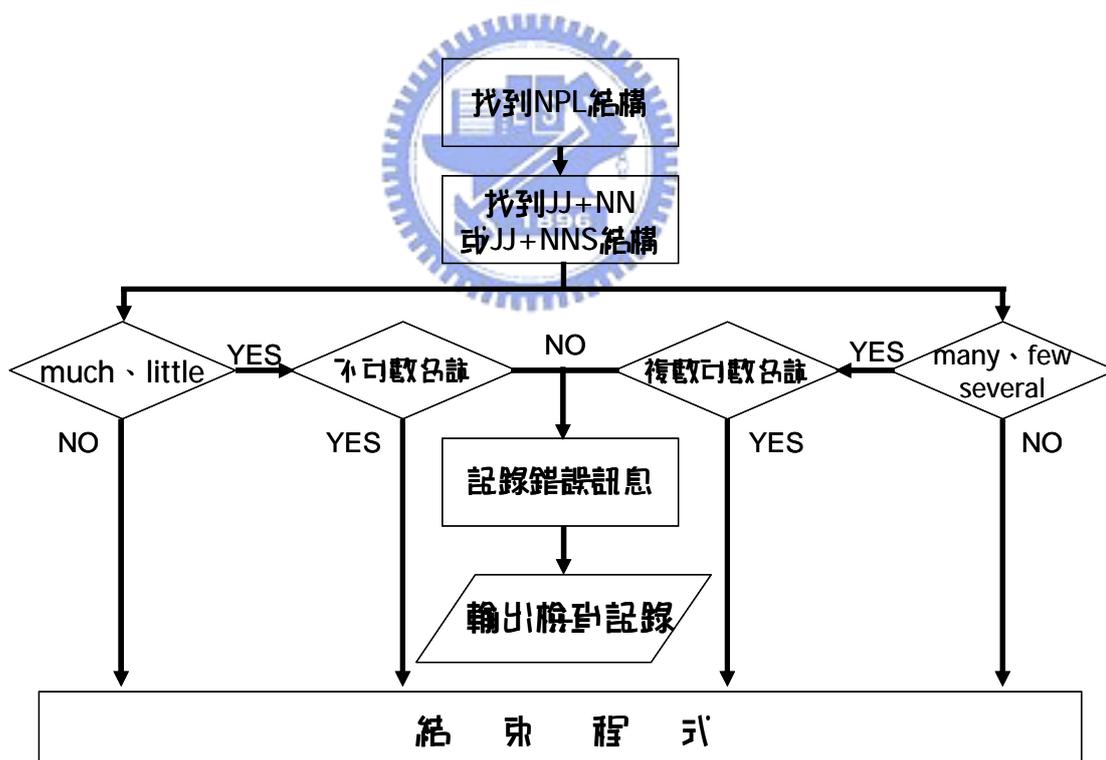


圖 22：形容詞與名詞單複數不一致 Module 程式流程圖

此二個規則在解析後是以「JJ(形容詞)+NN(單數名詞)」或「JJ(形容詞)+NNS(複數名詞)」的詞性標記出現，例如：句子 *There were few apples.* 解析後，解析樹成爲(S (NPL (EX There)) (VP (VBD were) (NPL (JJ few) (NNS apples))) (. -PERIOD-))。而其結構又會被視爲 NPL(最小名詞片語)，像是「(NPL (JJ few) (NNS apples))」此形式，因此我們就利用此解析的特性來做文法檢查。

就如上所述，我們先在解析樹中找到 NPL，如果其結構中有「JJ+NN」或「JJ+NNS」的情形時，即可開始此文法規則的檢查工作。而在此 Module 中的 JJ 是特定的，不是 *many, few, a few, several* 其中之一，就是在 *much, little, a little* 裡面的一個，因此我們必須要確定 JJ 的單字。

找到 JJ 的單字確定是用那一條規則後，接下來就是要確認名詞的單複是不否符合。因爲在解析樹當中我們只能知道名詞是單複數，其他的細節完全不清楚，所以我們就必須利用輔助資料庫，去查出此名詞是屬於可數名詞或不可數名詞，並分辨出單複數。若 JJ 是 *many, few, a few, several* 則名詞就必須是複數可數名詞，至於單數可數或是不可數名詞則都列爲錯誤；同樣的若 JJ 是 *much, little, a little* 則名詞必須是不可數名詞，至於可數名詞就是錯誤。

利用以上的流程就可以完成判斷，若有發現錯誤時，則送出錯誤訊息，若沒有錯誤則通過此 Module 檢查，結束此部份程式。此部份可以參考圖 33 答題驗證系統範例四輸出畫面，其中輸入例句：「*Every boy and every girl find several money with the car.*」，就會找到 *several* 後面用到不可數名詞 *money* 這個錯誤。

3. 主詞與動詞單複數不一致 Module：

此處形主詞與動詞單複數不一致 Module 是文法檢查其中的一個 Module，流程圖如圖 23 所示。在主詞與動詞單複數不一致中，我們目前做出以下三個規則的檢查：

- (1) 單數主詞 + 單數動詞。例句：*John sometimes goes to school by bus.*
- (2) (單數主詞 + 單數主詞) + 複數動詞。例句：*He and I are good friends.*
- (3) 複數主詞 + 複數動詞。例句：*All his sons are doctors.*

以上三個規格句型的主要結構是「主詞 + 動詞」的結構，若是此結構在解析樹中主詞會解析爲 NP(名詞子句)或 NPL(最小名詞片語)，而動詞的部份則解析爲 VP(最小動詞片語)，因此我們在解析樹中有找到「NPL + VP」或「NP + VP」就可進行此部份 Module 檢查。

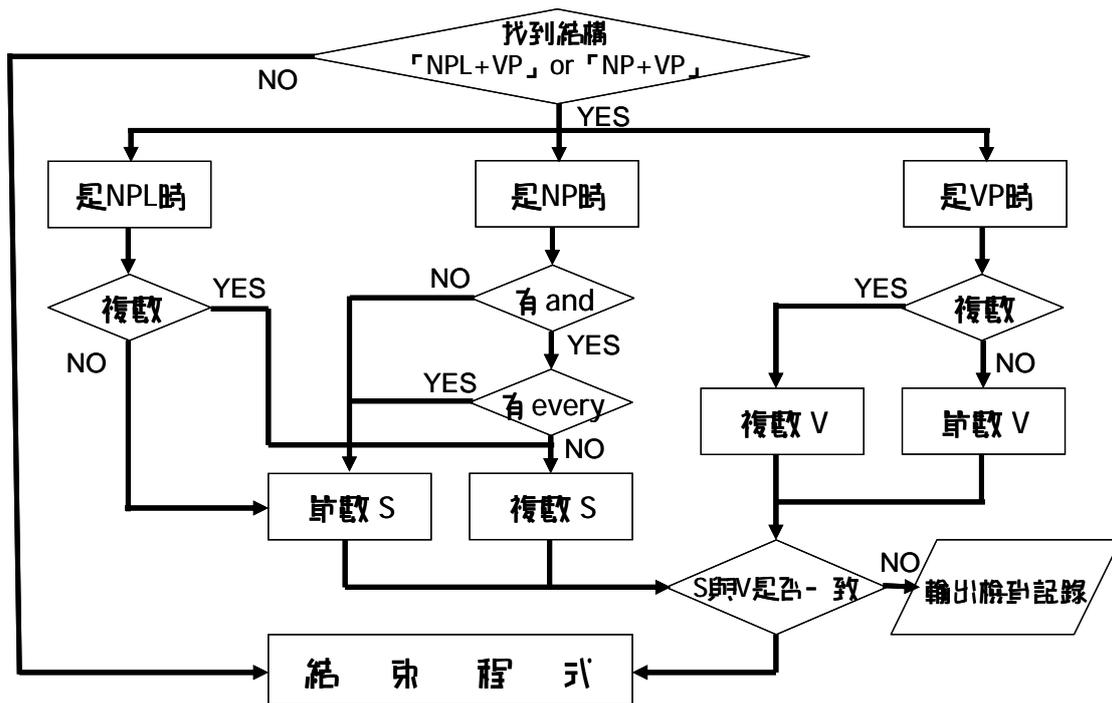


圖 23：主詞與動詞單複數不一致 Module 程式流程圖

我們先對主詞部份做說明。主詞部份會解析成 NPL 的部份較為簡單，例如：(NPL (NNP John))、(NPL (DT All) (PRP\$ his) (NNS sons))，因為其中就只有一個名詞，所以就只要去判斷其為單複數，就可以得到主詞是單複。但是解析成 NP 的部份就比較複雜一點，以上述(2)的例句來舉例說明，此例句的主詞是 He and I，會解析成(NP (NPL (PRP He)) (CC and) (NPL (PRP I)))，其中有二個人稱代名詞，所以其主詞部份是複數，但一個情形就是有 every 時又是單數，例如：Every boy and every girl is fond of it.其主詞「Every boy and every girl」就是單數。因此是 NP 時，就如流程圖中間部份所示，必須分成兩個部份判斷出其是單複數。若在 NP 或 NPL 中是人稱代名詞時，是利用輔助資料庫來查詢其是第幾人稱。

接下來是對動詞部份做判斷。找到 VP 的解析結構時，可以利用詞性標記找到單複數，例如：(VP (VBP are)..)、(VP (VBZ goes)..)，其中 VBP 是代表現在式動詞，VBZ 代表第三人稱單數動詞。我們可以利用此標記來判斷動詞單複數。

經過以上兩大部份後，可以得到主詞的單複數，還有動詞的單複數，就拿這兩大部份進行比對，檢查主詞與動詞的單複數是否一致，若不一致則列出錯誤訊息，若是一致則結束此Module的程式。此部份可以參考圖 33 答題驗證系統範例四輸出畫面，其中輸入例句：「Every boy and every girl find several

money with the car.」，就會找到單數主詞「Every boy and every girl」與複數動詞find不一致這個錯誤。



五、系統展示與應用實例

5.1 試題檢測系統

在這一章節當中我們會以數個英文句子為例子，介紹試題檢測系統檢測出的結果。

5.1.1 範例一

說明一：如圖 24畫面我們首先在 I 處輸入待檢測的句子，在此輸入「Nice to meet you.」為例，然後在 II 處按下「送出查詢」鈕，將此句子送至檢測系統。

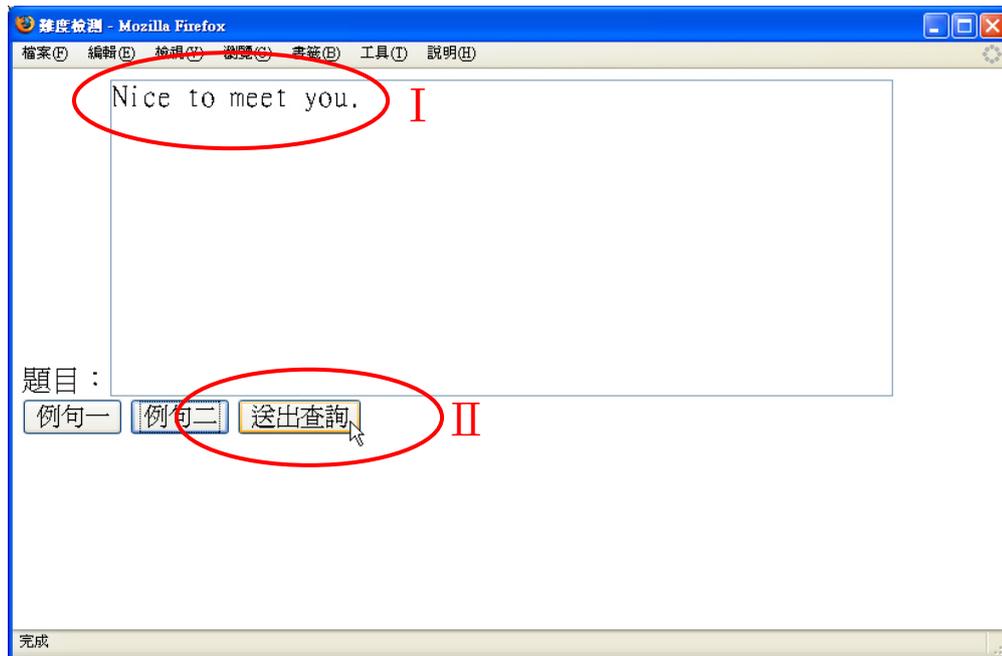


圖 24：試題檢測範例一輸入畫面

說明二：將待測句子送出後，就會得到以下圖 25 的輸出畫面，

標號 I：會顯示出輸入的句子。

標號 II：顯示出句子單字與字根的關係。例如此句中的 meet 如果是用 met，因為 met 是 meet 的過去式，所以 met 的字根就是 meet。

標號 III：是依據字根至詞彙表中查詢其 Level 級數。

標號 IV：所顯示的最大級數是依據每個單字查出的 Level 級數，而取其最大數，因為一個試題的級數，是以該試題中單字出現最高的級數為依據，所以在此取出最大級數為參考。

句子為: Nice to meet you. I				
單字	字根	詞彙表	字頻	字頻比例
Nice	nice	2	23	0.4‰
to	to	1	1228	20.8‰
meet	meet	1	43	0.7‰
you	you	1	970	16.4‰
IV 最大級數		2	字頻總合	58981
句子詞彙數: 4 此詞彙數句數: 491 比例數: $491/7700 = 6.4\%$				
建議級數	2		確認	

圖 25：試題檢測範例一輸出畫面之一

說明三：接下來用圖 26對字頻部份加以說明。

標號 I：是依據網路英檢現有題庫模擬做出的字頻資料庫，所查詢出字根的字頻數，數字愈多代表該字根被使用的次數愈多。

標號 II：此處是模擬網路英檢題庫所有字頻數的總合，以做為一個參考基值。

標號 III：此部份的字頻比例是以字頻數去除以字頻總合所算出的比例，其單位是千分之一，像是 to 表示有千分之 20.8 的出現頻率，相對於 nice 的千分之 0.4 的比例算是蠻高的。

句子為: Nice to meet you. I II				
單字	字根	詞彙表	字頻	字頻比例
Nice	nice	2	23	0.4‰
to	to	1	1228	20.8‰
meet	meet	1	43	0.7‰
you	you	1	970	16.4‰
最大級數		2	字頻總合	58981 III
句子詞彙數: 4 此詞彙數句數: 491 比例數: $491/7700 = 6.4\%$				
建議級數	2	確認		

圖 26：試題檢測範例一輸出畫面之二

說明四：接下來對於詞彙數及其餘部份在圖 27中加以說明。

標號 I：此部份裡面有幾個顯示訊息，句子詞彙數是 4，表示這個句子是使用到 4 個單字，此詞彙數 491，表示在模擬網路英檢題庫中 4 個字的句子有 491 句，而這樣 4 個字的句子在模擬網路英檢題庫佔百分之 6.4 的比例數。

標號 II：是檢測系統建議該試題的級數，命題委員可以綜合以上各種的參考數據，來判定該試題最後的級數，如果命題委員要更改級數，則可以用下拉式的方式輕鬆的更改級數，最後按下確定鈕，將此分級結果送出。

單字	字根	詞彙表	字頻	字頻比例
Nice	nice	2	23	0.4‰
to	to	1	1228	20.8‰
meet	meet	1	43	0.7‰
you	you	1	970	16.4‰
最大級數		2	字頻總合	58981

I 句子詞彙數: 4 此詞彙數句數: 491
比例數: 491/7700 = 6.4%

II 建議級數: 2 (dropdown menu with options 1-5) 確認

圖 27：試題檢測範例一輸出畫面之三

5.1.2 範例二

說明一：我們重新對檢測系統輸入句子「Shafts of sunlight soften the brooding darkness of Stanly Park.」，得到如圖 28的畫面，以下對其的說明。

標號 I：Shafts 是 shaft 的複數。shaft 詞彙不在詞彙表當中，因此在字根的部份我們將其標記成「*NULL*」，在詞彙表資料庫中找不到，所以詞彙表級數中標記成 6。

標號 II：其中 sunlight 的字根是 light，brooding 的字根是 brood，darkness 的字根是 dark，就如本論文章節 4.2.1 詞彙表分析中所提到，那些都算是衍生詞彙的部份，所以對映回來原型詞彙，並且其級數與原型詞彙相同。

標號 III：裡面「Stanly」及「Park」兩字的字首均為大寫。其中 park 在經過詞彙表時，因為在詞彙表有列到此字，所以查詢出其級數為 Level 1。而 Stanly 在詞彙表中查不到，但其字首是大寫，所以可能為專有名詞，因此送至專有名詞資料庫查詢，確認為專有名詞。當然這些專有名詞絕大多數不在詞彙表中，可是又不可算是非常難 Level 6 的級數，因為其雖然沒有看過，但大多都可以猜測到是代表某名稱的地名，所以我們將這些沒有在詞彙表中的專有名詞，在詞彙表的部份列為「*NNP*」的標記，表示不列入詞彙表級數計算。而確認是專有名詞後，我們在字根處用代號「*NNP*」來表示。其實「Stanly Park」號稱是北美最大的市內公園，坐落於半島之上，當中有海洋公園，可看溫哥華標誌之一的獅門橋，日落風景極好。

句子為: Shafts of sunlight soften the brooding darkness of Stanly Park.					
	單字	字根	詞彙表	字頻	字頻比例
I	Shafts	*NULL*	6	-1	-0.0%
	of	of	1	645	10.9%
II	sunlight	light	1	9	0.2%
	soften	soften	5	59	1.0%
	the	the	1	2176	36.9%
	brooding	brood	5	59	1.0%
	darkness	dark	1	67	1.1%
III	of	of	1	645	10.9%
	Stanly	*NNP*	*NNP*	-1	-0.0%
	Park	park	1	67	1.1%
	最大級數		5	字頻總合	58973
句子詞彙數: 10 此詞彙數句數: 817 比例數: 817/7699 = 10.6%					
建議級數	5		確認		

圖 28：試題檢測範例二輸出畫面

5.2 答題驗證系統

在這一節當中我們會以數個英文句子為例子，介紹答題驗證系統驗證的結果。

5.2.1 範例一

說明一：如下圖 29畫面我們首先在 I 處輸入待驗證的句子，在此輸入「They do not have morden machines but they can stil do it well.」為例，然後在 II 處按下「送出查詢」鈕，將此句子送至驗證系統。

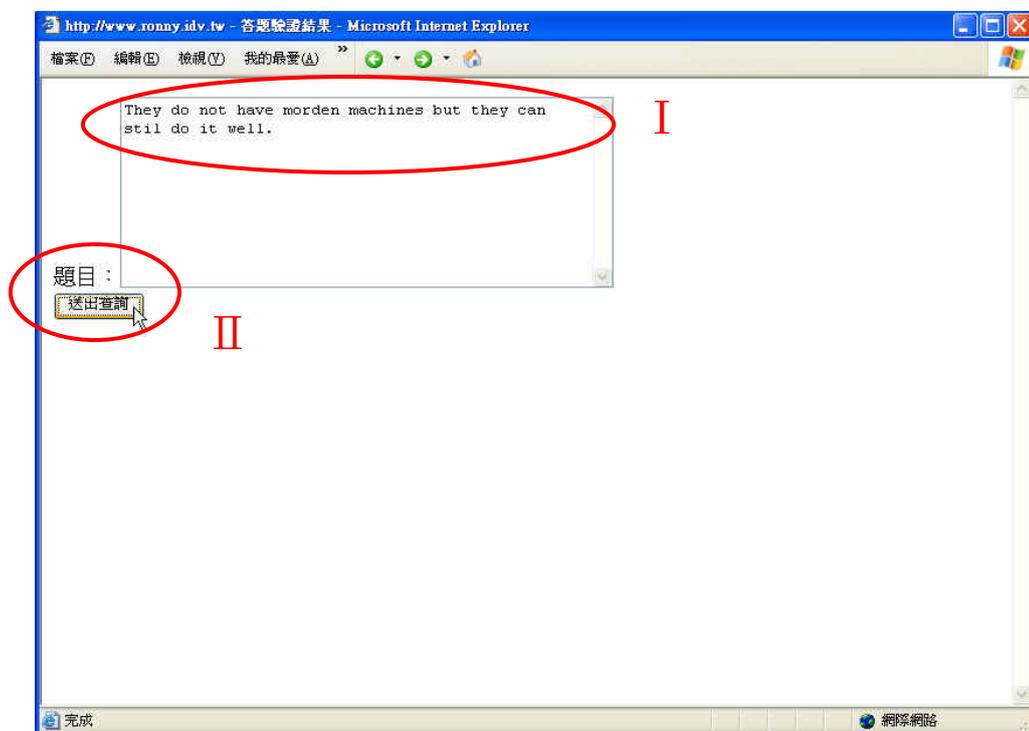


圖 29：答題驗證系統輸入畫面

說明二：將待測句子送出後，驗證系統會先進行拼字檢查，就會得到下圖 30 的輸出畫面。

標號 I：此部份是輸入句子的每個單字，分開來列表。

標號 II：這是拼字檢查的結果，如果左欄的單字是拼字是正確無誤的，則此欄位就是空白；相反的如果有拼字錯誤的話，此欄位就會顯示文字。顯示文字的意思，第一是告訴我們此字是有拼字錯誤，像此例中的「morden」及「stil」兩字是有拼字錯誤；第二顯示文字中是建議給我們有可能的正確拼字。

標號 III：在系統中我們可以顯示出拼字錯誤的建議單字，可是無法利用系統找到正確的單字替換。所以在答題驗證的過程中，如果像此例有拼字錯誤，系統是無法做拼字的更正後，續繼做文法檢查及語義檢查，而整個驗證過程就在此結束。

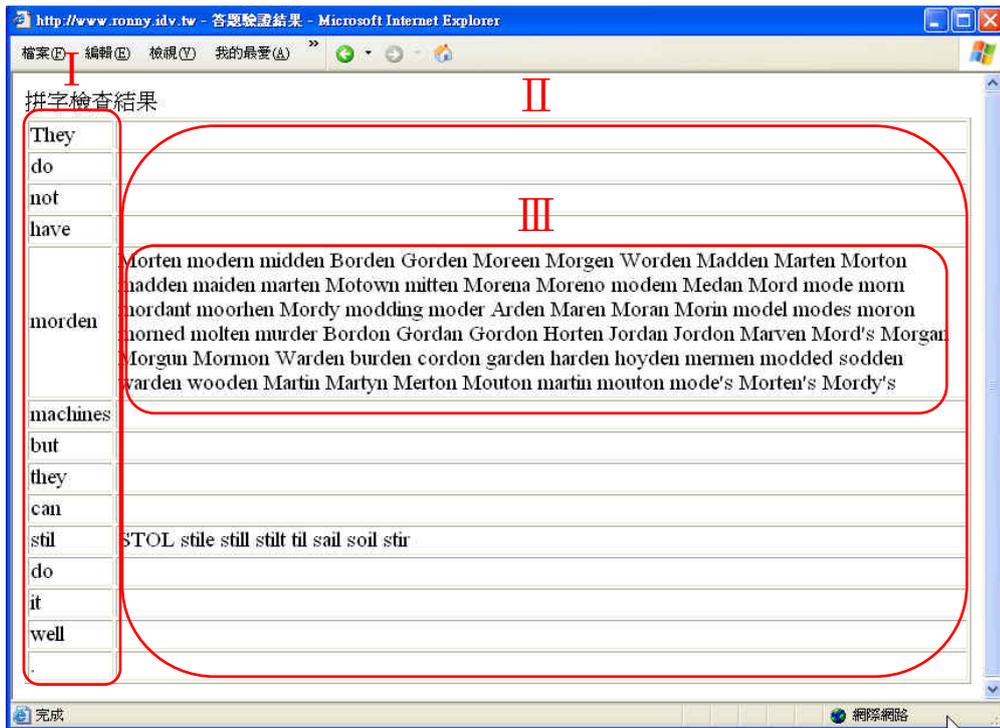


圖 30：答題驗證系統範例一輸出畫面

5.2.2 範例二

說明一：爲了繼續做系統的說明，所以將拼字錯誤去除，讓系統能做文法檢查及語義檢查。因此我們依照上圖 30 拼字檢查的建議，找到正確的拼字後，更正出正確的句子「They do not have modern machines but they can still do it well.」，重新輸入驗證系統。

說明二：重新輸入沒有拼字錯誤的句字後，得到下圖 31 的畫面。

標號 I：此部份告訴我們在拼字檢查中，沒有拼字的錯誤，此處所謂的拼字指的是純粹英文字母的拼字，如果是該用名詞而用成動詞，但其英文字母拼字皆沒錯誤，在此部份是無法檢查出來的，必須要到文法檢查中去處理。

標號 II：顯示出此句有使用到關鍵字的同義字。在展示的系統中關鍵字設定爲 happy、good、people 三個字，所以目前此例句用到 well，而其是 good 的同義字。

標號 III：驗證完畢後，閱卷老師依據系統顯示的建議資料，利用下拉式選單，在此是以 1~10 分的選擇爲例子，選到閱卷老師認爲正確的得分後，將得分送出。

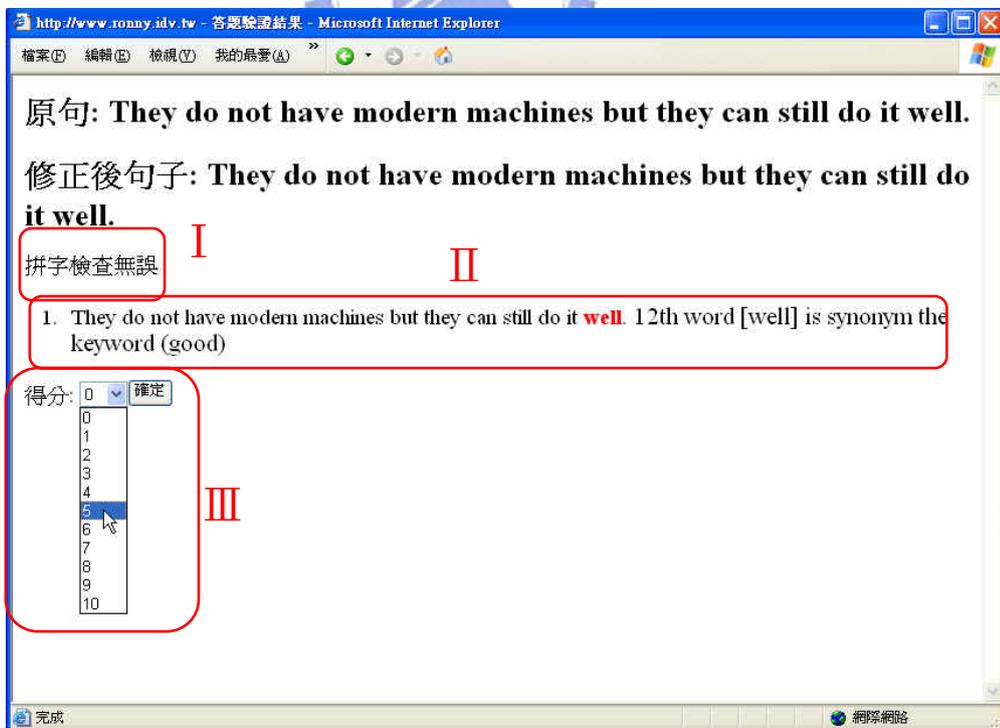


圖 31：答題驗證系統範例二輸出畫面

5.2.3 範例三

說明一：在圖 29 系統輸入畫面中重新輸入例句「He and she talk with he and she, and I am happy and live.」，讓系統進行驗證。

說明二：重新輸入例句後，得到圖 32 的畫面。

標號 I：與上範例相同，例句中沒有拼字錯誤，所以通過拼字檢查。

標號 II：在文法檢查中找到一個錯誤，那就是動詞後加代名詞，就必須用受格代名詞，而不是用主格代名詞。

標號 III：在語義檢查部份，檢查到此例句有使用到關鍵字 good，所以標示出來。同樣的也檢查到此例句有使用到關鍵字 people 的同義字 live。

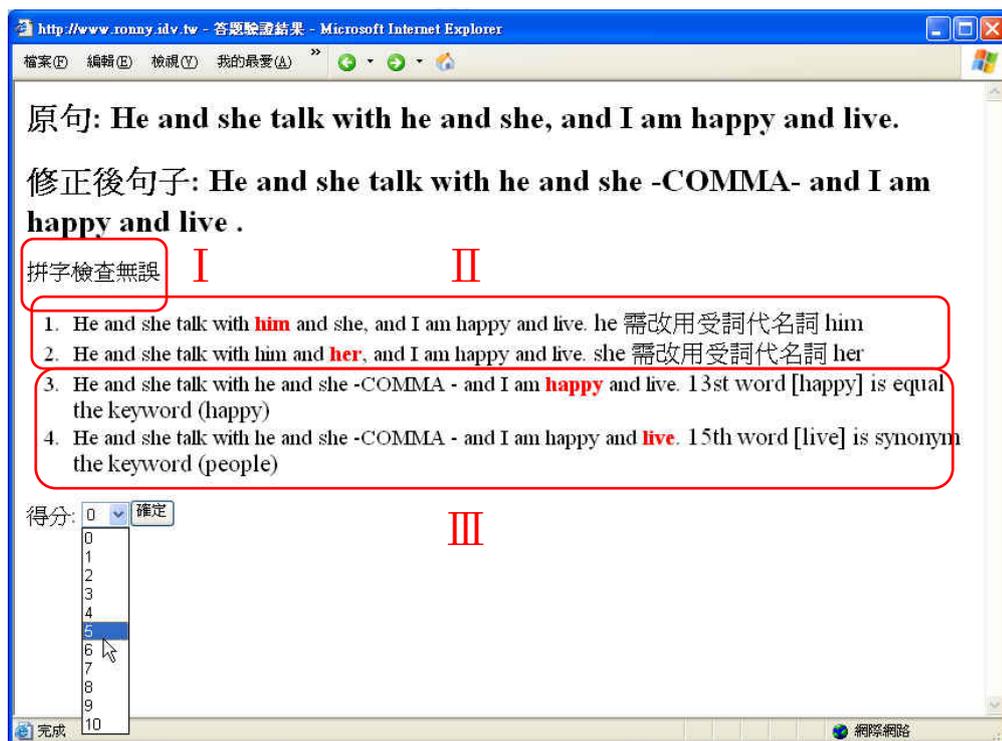


圖 32：答題驗證系統範例三輸出畫面

5.2.4 範例四

說明一：在圖 29 系統輸入畫面中重新輸入例句「Every boy and every girl find several money with the car.」，讓系統進行驗證。

說明二：重新輸入例句後，得到下圖 33 的畫面。

標號 I：此例在文法檢查過程中有一個錯誤，就是 several 接複數的可數名詞，而此例中用到了不可數名詞 money，所以造成錯誤。可是此錯誤不是嚴重的句構問題，所以列為 warning 可以繼續進行下一個規則的檢查。

標號 II：文法檢查到另一個錯誤，例句中的主詞「Every boy and every girl」是算單數，而動詞須使用第三稱單複 finds，而不是用 find。

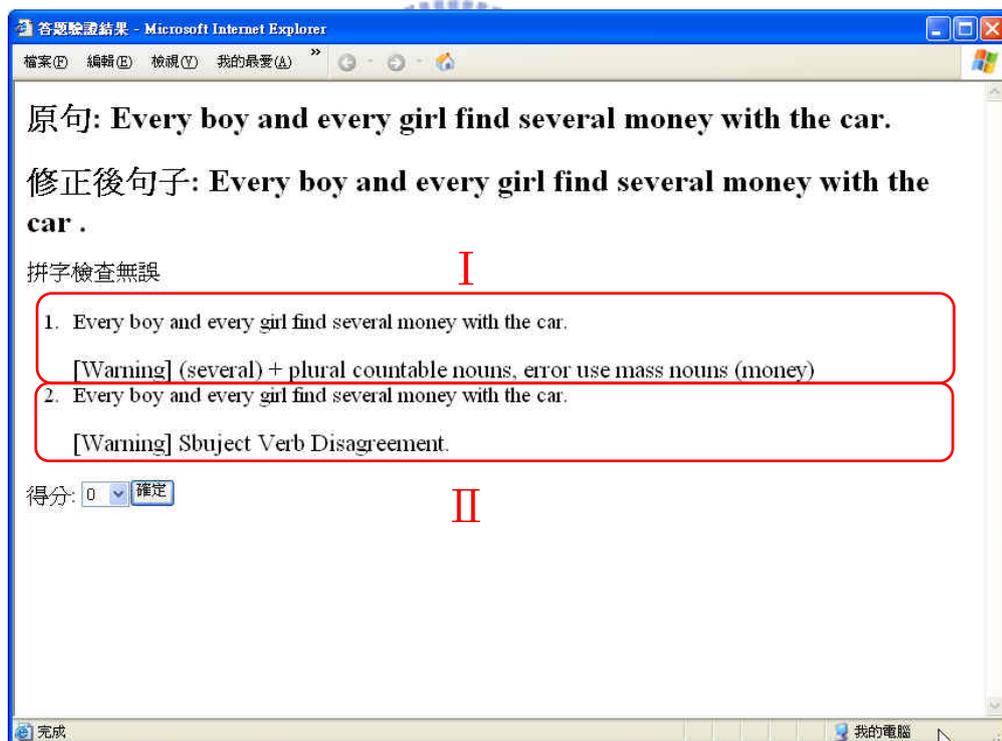


圖 33：答題驗證系統範例四輸出畫面

5.2.5 範例五

說明一：在此範例中將輸入一個多句的短文章，做為人工閱卷部份範例展示。在系統輸入畫面中重新輸入例句「He and she talk with he and she, and I am happy and live. Every boy and every girl find several money with the car. This is a boog.」，讓系統進行驗證。

說明二：重新輸入例句後，得到圖 34 人工閱卷的畫面。

標號 I：在此部份是顯示關於答題的相關訊息，其中有考生准考證號碼、題號、題目及答題的結果。

標號 II：此部份是展示答題驗證系統做完各項檢查後，一個綜合的驗證結果。而閱卷老師可以在參考完畢各項檢查結果的資料後，在「實得分數」處利用下拉表單，更改給該生最後的實得分數，確定要給這個分數後，按下「同意得分」將閱卷老師評定的分數送出。

標號 III：做完評分的動作後，而「確認存檔」可以將此檢查的結果存檔，做為一個記錄備查用。「下一題」是繼續批改，「上一題」是回去重新檢視前一個批改的情形。

結果 - Mozilla Firefox

檔案(F) 編輯(E) 檢視(V) 瀏覽(O) 書籤(B) 工具(T) 說明(H)

考生：9273539
題號：3
題目：What is it ?
答題：He and she talk with he and she, and I am happy and live. Every boy and every girl find several money with the car. This is a boog.

此題配分：10分
實得分數：0

拼字檢查		文法檢查		語義檢查			
錯誤字數	1	錯誤文法數	4	符合關鍵字數	1	符合同義字數	1
總字數	29	總句數	3	關鍵字數	3	同義字數	30

檢查記錄

【語義檢查】
關鍵字：Happy good people
同義字：euphoric、felicitous、pleased、well-chosen、goodness、adept、beneficial、benevolent、dear、dependable、effect(p)、effective、estimable、right、fine、full、citizenry、hoi、mass、完成

圖 34：答題驗證系統人工閱卷輸出畫面之一

說明三：圖 35是人工閱卷畫面中的語義檢查部份。

標號 I：此部份顯示出原命題委員對於此題答案所給的關鍵字，也在同義字的部份將關鍵字的同義字顯示在此，供閱卷老師參考。

標號 II：在此區域顯示為，答題中檢查出那些句子及詞彙使用到關鍵字或同義字。並將檢查結果「符合關鍵字數」及「符合同義字數」顯示出來。

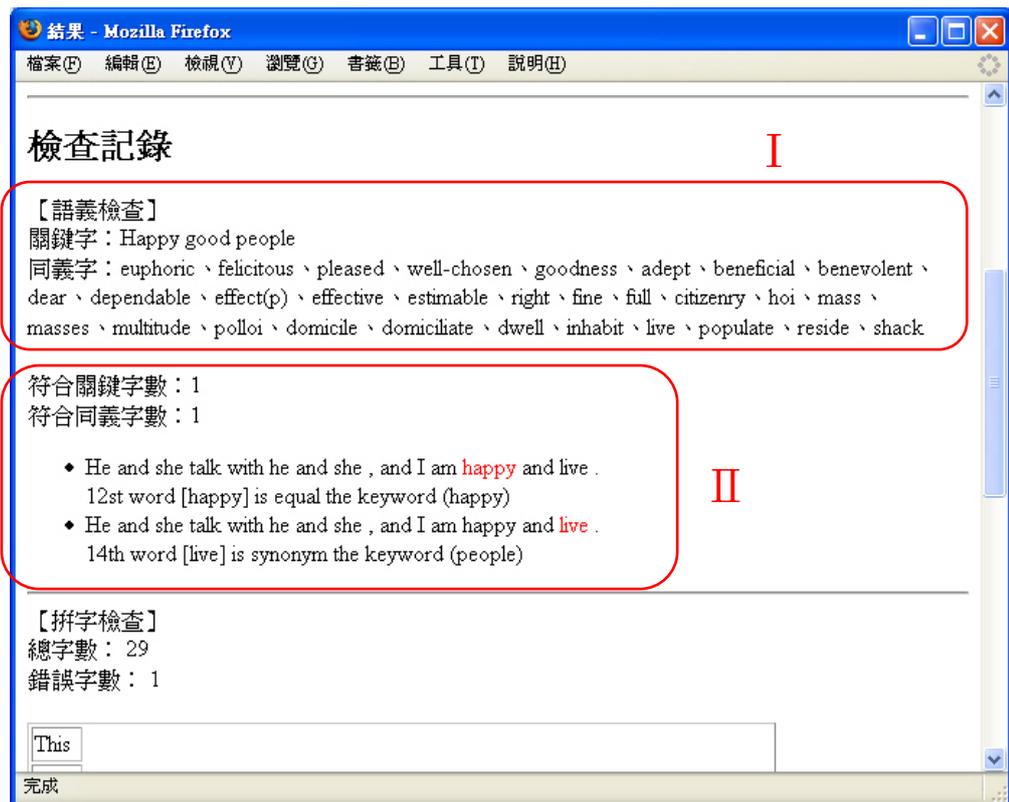


圖 35：答題驗證系統人工閱卷輸出畫面之二

說明四：圖 36是人工閱卷畫面中的拼字檢查部份。

標號 I：拼字檢查的結果會在此部份被顯示出來，例如此例就會顯示總答題的詞彙數是 29，而拼字錯誤字數是 1 個字。

標號 II：在答題的句子中，如果有拼字上的錯誤，就會顯示在此部份。

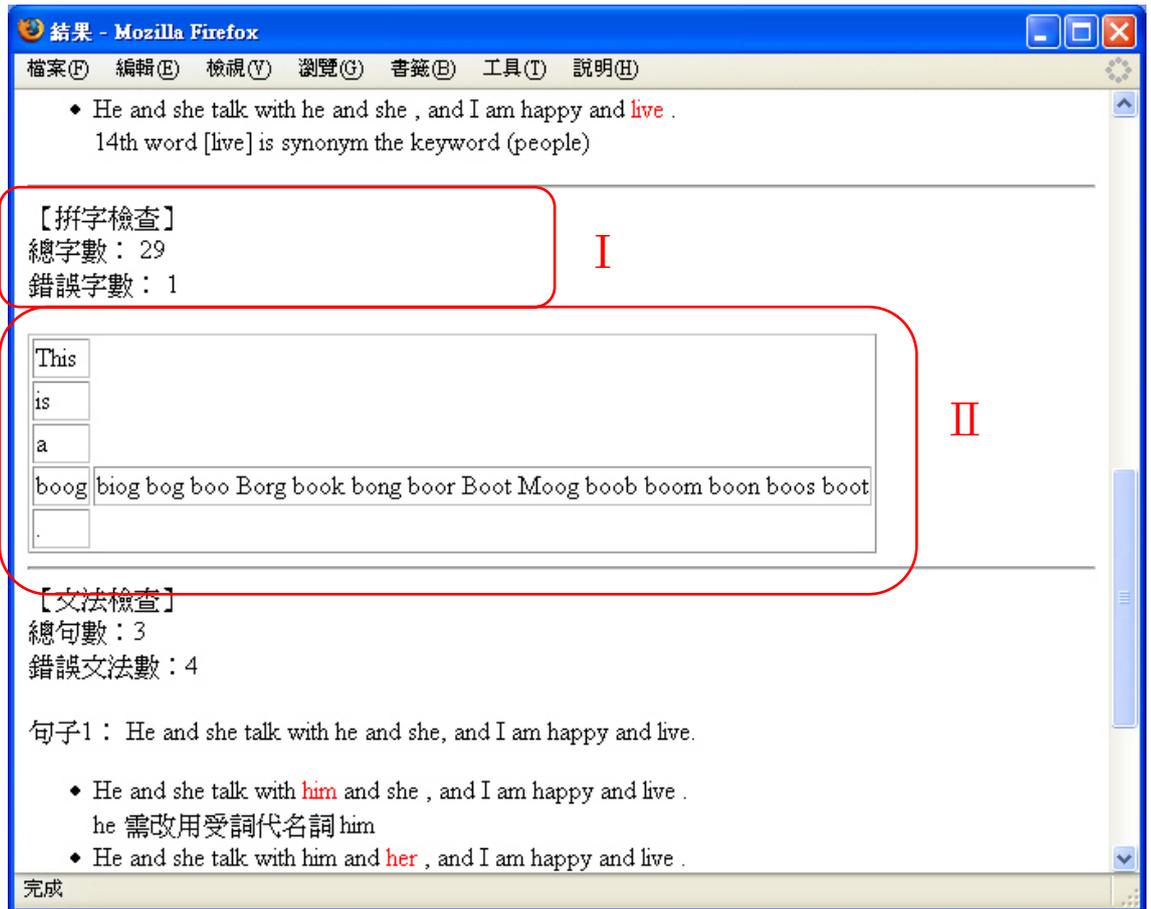


圖 36：答題驗證系統人工閱卷輸出畫面之三

說明五：圖 37是人工閱卷畫面中的文法檢查部份。

標號 I：文法檢查完成後，會將檢查結果顯示出來。像此例就會顯示答題總句數是 3，而總共有 4 個文法錯誤，而句子 3 因為在拼字檢查有錯字，所以無法進行文法檢查。

標號 II：此部份是文法檢查結果錯誤訊息的顯示，給閱卷老師一個參考。

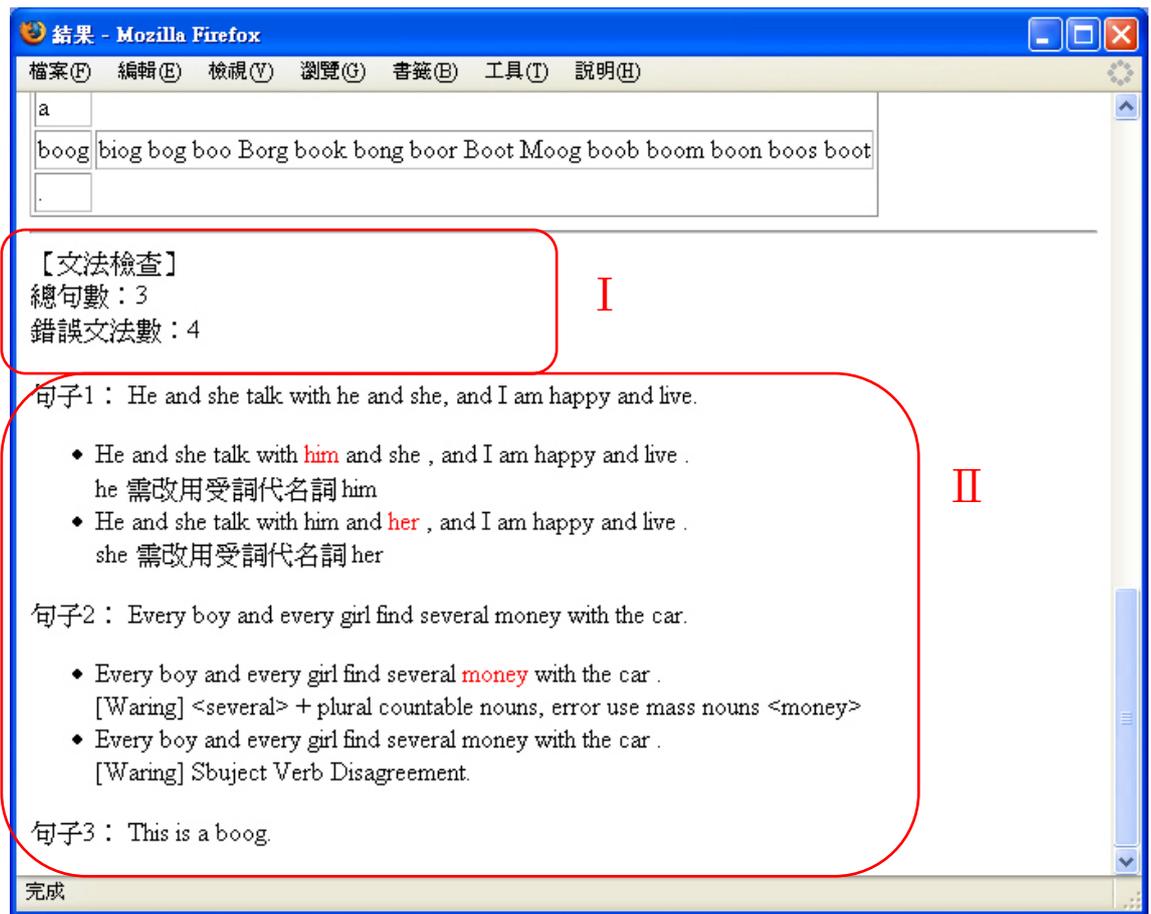


圖 37：答題驗證系統人工閱卷輸出畫面之四

5.2.6 本系統的限制

在拼字檢查部份，是使用到 Aspell 的字典庫，可以至其官方網站更新字典庫，系統目前沒有做到可以讓使用者自行增加或修改字典庫。

語義檢查部份，在 WordNet 裡對於同義字是有許多的分類法，像是詞性的分類、語義的分類。而目前系統對於關鍵字去查詢同義字，是不分其某一語義的同義字，也沒分其某一詞性的同義字。

文法檢查部份可以做那些文法規則，是參考國外已經做好文法檢查軟體「WinProof Grammar Checker」[16]的資料，其中提到此軟體已經做到那些可以檢查的規則。而系統已經做好的規則如下：

- SUBJECT VERB DISAGREEMENT (動詞不一致)
- ADJECTIVE DISAGREES WITH NOUN (形容詞與名詞不一致)
- NOT A QUESTION (非疑問句)
- USE OR INSTEAD OF NOR (or,nor 的使用)
- USE NOR INSTEAD OF OR (or,nor 的使用)
- NEED OBJECTIVE PRONOUN (須要正確的代名詞)
- NEED SUBJECTIVE PRONOUN (須要主格代名詞)
- NEEDS POSSESSIVE (須要所有格)
- USE FEWER INSTEAD OF LESS (使用 fewer 代替 less)
- USE WELL INSTEAD OF GOOD (使用 well 代替 good)
- PRONOUN NOT REFLEXIVE (代名詞不是反身代名詞)
- STARTING WITH AND OR BUT (and、or、but 開頭的句子)
- SENTENCE CAPITALIZATION (句首大寫)
- PUNCTUATION INSIDE OF QUOTATIONS (引號內少了標點)
- MISSING QUOTATIONS (遺漏引號)

未來期待能夠繼續發展的規則如下：

- REFLEXIVE PRONOUN DOES NOT MATCH SUBJECT (反身代名詞與主詞不符合)
- MISSING ARTICLE (遺漏冠詞)
- MISSING INFINITIVE (缺少的不定詞)
- USE AN INSTEAD OF (A 定冠詞 A,AN 的使用)
- USE A INSTEAD OF (AN 定冠詞 A,AN 的使用)
- INCORRECT VERB TENSE (錯誤的動詞時態)
- USE WHO FOR PERSONS(人的連接詞用 who)
- SEXIST LANGUAGE (性別詞的使用)
- USE THAT INSTEAD OF WHICH (使用 that 代替 which)
- NEED ADVERB INSTEAD OF ADJECTIVE (須要副詞代替形容詞)
- AGREEMENT FOR HERE AND THERE (與 here、there(人稱,性別,數,格的)的一致)
- FORMING POSSESSIVE NAMES (名詞所有格的使用)

六、總結

6.1 總結

在本論文期望研究電腦化英語檢定測驗過程，如何讓試題列入適當的檢定級數，又如何能讓電腦分擔閱卷批改過程當閱卷老師的工作負荷。因此提出分別針對試題檢測及答題驗證兩部份提出其解決方法。

在試題檢測部份，我們提出了以詞彙表為主，並參考字頻及詞彙數，來為試題檢測進行分級的概念。其中我們是以台灣大學外文系鄭恆雄教授領導編修「大學入學考試中心高中英文參考詞彙表」為主，加上國立高雄師範大學英語系莊永山教授的詞彙表，所做出的「網路英檢參考詞彙表」為詞彙表分析的核心，再衍生出詞彙表中相關的詞彙，以及收集而來的專有名詞資料庫，構成一個完整的詞彙表分析資料庫。接著利用網路英檢現有的試題資料庫來統計字頻及詞彙數的資料。

最後，系統就可以依據詞彙數資料庫、字頻資料庫、詞彙數資料庫，進行該試題的資料分析，而命題委員就可以依據分析出來的詞彙表、字頻及詞彙數三大類參考資料，為試題做客觀的試題分級。

另外在答題驗證部份，我們也提出拼字檢查、文法檢查及語義檢查三大步驟，來為閱卷工作做初步的驗證，並將驗證結果供閱卷老師參考，提高閱卷的效率。在程式的部份，我們運用了動態連結函式庫的概念，讓往後要增加或修改檢查程式時，能夠更容易更方便。在文法檢查部份，我們提出以自然語言處理技術為基礎的解析方法，來為我們解析英文句子的句構及詞性標記，進而能夠做文法檢查的工作。詞義辨析本身並不是件容易的工作，在自然語言處理的相關研究中已研究了多年。對於文法檢查的部份，雖然我們的方法並不能對其做到百分之百的檢查錯誤，但也提供了一個重要的幫助。畢竟，語言學家與心理學家們普遍認為，詞義辨析仍需要來自上下文甚至是整個段落的資訊，而並非僅僅是單一個句子 [28]。而在語義檢查部份，提出以英語詞彙資料庫 WordNet 來做為我們做同義字檢查的資料庫來源，如此可以運用命題委員設定好關鍵字，再利用同義字資料庫來擴充關鍵字的範圍，為我們達到語義檢查的目標。本論文的研發工具，可用於協助命題委員及閱卷老師較客觀的分級及評分依據。

6.2 未來發展方向

1. 試題檢測增加文法級數檢測

在試題檢測系統中，目前做到詞彙表、字頻及詞彙數三個部份的分析，未來可以增加一個對文法的分析。將文法規則做難易度分級，再去檢測此試題使

用到那個文法規則，分析出其文法的難易度級數，讓命題委員可以更精準的決定試題的級數。

2. 答題驗證持續發展文法檢查的規則數

在目前答題驗證系統只做出數個文法規則的檢查，未來可以持續增加更多的文法規則檢查。在每增加一個新的文法檢查規則的過程中，就會須要克服不同的問題與困難，並且找到新的方法去解決，讓整個系統達到更完美的境界，

3. 答題驗證中加入計分模式

在答題驗證的過程中，可以納入更詳細的計分模式，讓驗證的過程中就能讓電腦直接計分，提出一個計分的建議值，給閱卷老師參考，如此對那一種錯誤就扣多少分，成為所有閱卷老師的標準，不會有的老師認為同一個文法錯誤輕重不一，達到更公平的原則。而此計分模式中的各項權重計分，可以單獨設定在一個部份，可以依各場次考試的須求，去調整權重計分的比例。

4. 答題驗證中輔助資料庫可找權威詞典驗證

因為在整個研究過程中，對於許多資料庫的收集，花費非常多的時間與精力，因此，可以再找具有權威的詞典或資料庫，為系統內的資料庫做驗證工作，讓系統整體的正確度更加提高，更具可信度。



參考文獻

- [1] Garside,R.; Leech,G.; and Sampson,G. (1987). *The Computational Analysis of English*.Longman.
- [2] Leech, G.; and Garside, R. (1991). "Running a grammar factory: The production of syntactically analysed corpora or 'treebanks' ". In *Stig Johansson and Anna-Brita Stenstrom (eds.) English Computer Corpora : Selected papers and Research Guide*. 1991. 15-32.
- [3] Mitchell P.Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini (1993). "Building a Large Annotated Corpus of English: The Penn Treebank", *Computational Linguistics*, 19(2), 313-330
- [4] Santorini, B. (1991). Part of Speech Tagging Guidelines for the Penn Treebank Project(<http://www.cis.upenn.edu/~treebank/home.html>).
- [5] Miller, G. A., "WordNet : a lexical database for the English language", <http://wordnet.princeton.edu/>,2005
- [6] G. Miller, *Nouns in WordNet: a Lexical Inheritance System*, Princeton University, 1993.
- [7] C. Fellbaum, *WordNet, an Electronic Lexical Database*, The MIT Press, 1998.
- [8] Department of Linguistics, University of Victoria, *Linguistics Statistics in Corpus Analysis* (<http://web.uvic.ca/~ling48x/2001f01ling482/2001sept27.html>).
- [9] Jochen L. L., Tiphaine D., Bonnie W., Johan B., Claire G. (2003) "Automatic Multi-Layer Corpus Annotation for Evaluating Question Answering Methods: CBC4Kids", In *4th International Workshop on Linguistically Interpreted Corpora*.
- [10] Sekine, S., Grisman, R. (1995) "A corpus-based probabilistic grammar with only two non-terminals" In *Proceedings of Fourth International Workshop on Parsing Technologies*.
- [11] Satoshi Sekine , (1994)"A New Direction for Sublanguage NLP",International Conference on New Methods in Language Processing .
- [12] Satoshi Sekine , (1994)"Automatic Sublanguage Identification for a New Text", Second Annual Workshop on Very Large Corpora .
- [13] Chang, Claire Hsun-huei (張郇慧). 1996. "A Corpus-based Analysis of Linguistic Features in Written and Spoken English (以語料庫為本探討英語口語及書面語中之語言特徵)." , In *A Collection of Papers Presented in the Third Conference of English Teaching/ Research and Computer Information (第三屆英語文教學研究與電腦資訊研討會論文集)*., Taipei: English Teachers' Association. 129-44.

- [14] Spell Checker Test Kernel Results (<http://aspell.net/test/>).
- [15] The GNU Operating System (<http://www.gnu.org/>).
- [16] WinProof Grammar Checker
(<http://ourworld.compuserve.com/homepages/frankhu/winproof.htm>)
- [17] 2002年2月31日行政院台經字第0910027097號函核定「國家發展重點計畫2002-2007 挑戰2008年」。P.19~45。
- [18] 黃居仁、張如瑩、蔡柏生(民92)。「語意網時代的網路華語教學:兼介中英雙語知識本體與領域檢索介面」。2003年第三屆全球華文網路教育研討會。
- [19] 黃居仁(民92)。「從詞彙庫到知識本體:為專業知識庫許個「語意網」的未來美景」。醫藥衛生圖書資源專題講座暨研習會--網路時代參考服務之應用。
- [20] 鄭恆雄(2002), 大學入學考試中心高中英文參考字彙表, 台北: 大學入學考試中心。
- [21] 鄭恆雄(2003), 大學入學考試中心高中英文參考字彙表之編輯方法及原則。
- [22] 李家同、海柏(民92)。專門替中國人寫的英文基本文法(二版)。台北: 聯經。
- [23] Uche Ogbuji. (2001). 知識管理的基本 XML 和 RDF 技術(<http://www-128.ibm.com/developerworks/cn/xml/rdf/part6/>)。
- [24] 北京大學·北京大學漢語語言學研究中心(<http://ccl.pku.edu.cn/>)。
- [25] 周強、俞士汶, 「漢語短語標注標記集的確定」, 北京大學計算語言學研究所, 北京
- [26] 周強、張偉、俞士汶, 「漢語樹庫的構建」, 北京大學計算語言學研究所, 北京
- [27] 財團法人語言訓練測驗中心, <http://www.lttc.ntu.edu.tw/main.htm>。
- [28] 全民網路英語能力檢定, <http://www.test.org.tw>。
- [29] 蕭永哲(民國88年)「探訪動態連結函式庫」, RUN!PC 1999年九月號。
- [30] 林武聰(2003)「線上英語學習環境」, 雲林: 國立雲林科技大學電子與資訊工程研究所碩士論文。
- [31] 林長順(2005.06.26)「兒童英檢報考近13萬人創新高」, 台灣日報。