

A pragmatic tutorial dialogue  
system: design, implementation  
and evaluation in a health  
sciences domain

Jenny McDonald

a thesis submitted for the degree of  
Doctor of Philosophy  
at the University of Otago, Dunedin,  
New Zealand.

14 March 2014

## **Abstract**

In large undergraduate classes, it is time-consuming, costly and seldom practical for the teacher to provide individualised feedback to students on their written responses to questions. A request for advice in relation to these issues, from the coordinator of a large first-year health sciences course at the University of Otago, motivated the research described in this thesis.

An intelligent tutoring system (ITS) engages students in a dialogue; students enter their contributions to the dialogue as free text and the system gives a response. Such systems, which employ natural language as their interface, are called dialogue-based ITS or tutorial dialogue systems; they offer some promise for supporting and enhancing student understanding of key concepts through the provision of individualised feedback. The appeal of tutorial dialogue is that questions are embedded in a tutorial plan: the questions arise in a meaningful context and concepts and ideas are linked together in a coherent form. This allows each student contribution to be individually assessed.

However, ITS are not currently in widespread use in higher education settings and there has been considerable controversy around their application in this context. Practical issues relating to the time and cost for development, the difficulty of adapting to specific teaching contexts, and pedagogical objections which relate to the idea of student modelling are among some of the barriers to their widespread adoption. In this thesis, a rationale is presented for revisiting tutorial dialogue systems in the context of large-class teaching.

Two broad goals for this research are delineated. The first goal was to design, build and evaluate a new tutorial dialogue system for the cardiovas-

cular section of a first-year undergraduate health-sciences paper. The new system is firmly empirically-based, with both the teaching context and real student responses to questions integral to its design and implementation.

The second goal was to determine whether a tutorial dialogue system, which provides students with the opportunity to practise writing answers to short-answer questions and gives automated feedback about these answers, would result in improved student performance. In order to explore the second goal, two versions of the new system were developed. The first version of the system required students to type their response to questions as free-text; in the second version, students selected the answer they preferred from a menu of options. Student volunteers were divided into three groups: a control group, a free text tutorial group and a menu-based tutorial group. The performance between the two tutorial groups and the control group were compared to test whether free-text entry conferred any performance advantage over selection from a menu of options.

The design and implementation of the new tutorial dialogue system is described in detail and some limitations are discussed. The evaluation of the new dialogue system with 578 student volunteers in a real-class setting is described. Student perceptions of the system were broadly positive and there was strong uptake of the system compared with an earlier prototype. The experiment, which was set up specifically to test the performance of the system overall, as well as to establish if there are differences between free-text and menu-based versions, found student performance gains did occur among students who used either version of the new tutorial dialogue system but no differences were found between the two versions.

The main conclusion to be drawn from this research is that the new system can be deployed in a large-class setting and, at least in the context of first-year health sciences undergraduate courses, is likely to find acceptance with students, in addition to having a positive impact on their performance. The development of a stable platform for the further study of tutorial dialogues and the automated creation of a large corpus of tutorial dialogues are spin-off benefits from the research. Finally, this research is a small contribution towards getting contemporary tutorial dialogue systems back on the educational agenda.

## Acknowledgements

As Pooh Bear would say, ‘Something which feels very Thingish inside you is quite different when it gets out into the open and has other people looking at it’. My sincere thanks are due to many thoughtful people who have looked at this very Thingish enterprise, and especially to:

- Alistair Knott, my primary supervisor. Ali’s depth of knowledge as a linguist, computer scientist and educator has been invaluable as has his care and guidance as a supervisor. Ali has been, and remains, an inspiration and a role model.
- Sarah Stein, my secondary supervisor and colleague from HEDC. Sarah’s support, experience and insights as an educator have also been invaluable.
- Chris Heath, former Director of HEDC. This project would never have started without Chris’s encouragement and his conviction that PhD study was a possibility. The financial assistance and time for study which was supported by subsequent heads of HEDC, Kerry Shephard, Rachel Spronken-Smith and Tony Harland, is also very much appreciated.
- Staff and students of HUBS 192 for their willingness to try something new. Special thanks to: Ruth Napper, Fiona McDonald and John Reynolds (current and past HUBS Course Coordinators), Simon Green and Greg Jones for permission to review their lecture notes and help with script revisions, Zoe Ashley for question development and promotion to students, Philip Kelly for help with data collection; and last but not least Rebecca Bird for help with setting up the experiment and marking student work.

- Colleagues and students from around the University for piloting early versions of the tutor. Especially to Phil Sheard from the Department of Physiology and Phil Blyth from the Faculty of Medicine.
- Fellow postgraduates and staff from the AI Group in Computer Science for their feedback, suggestions and useful discussion.
- Pamela Jordan from the Learning Research and Development Centre at the University of Pittsburgh, USA, for her support and permission to use the TuTalk system in early student trials.
- My colleagues in HEDC for their support and encouragement. Especially: Richard Zeng for his knowledge, skill and insights as a programmer and for his practical support in building the web-application which students used to access the tutor, Ayelet Cohen for designing and creating a simple web-interface and for her support, enthusiasm and thoughtful feedback, Swee Kin Loke for feedback and so many enjoyable discussions, Jo Kennedy and Allen Goodchild for their support, feedback and helpful discussions especially in relation to evaluations.
- Finally, to my family, especially my sister Alison, and to Jenny, Maia, Anna and Debi, thank you for your care, love, patience and understanding. And to Rosemary, this project would never have finished without you.

Note: Parts of Chapters 4, 6 and 7 have appeared in a published paper (McDonald, Knott, and Zeng, 2012). The contributions of my co-authors are described where this work is discussed.

# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduction</b>  | <b>1</b>  |
| 1.1      | Background . . . . .   | 1         |
| 1.2      | Where this project began . . . . .   | 2         |
| 1.3      | Classroom context . . . . .  | 2         |
| 1.4      | Choice of approach . . . . .   | 3         |
| 1.5      | The goals of this study and main contributions . . . . .   | 6         |
| 1.6      | Outline of this thesis . . . . .   | 7         |
| <b>2</b> | <b>Past to present: a multi-perspective review of intelligent tutors</b>   | <b>8</b>  |
| 2.1      | Introduction . . . . .   | 8         |
| 2.2      | A brief history of intelligent tutoring systems . . . . .  | 8         |
| 2.2.1    | The behaviourists . . . . .  | 9         |
| 2.2.2    | The first intelligent tutoring system . . . . .  | 11        |
| 2.2.3    | The empiricists and the rationalists . . . . .   | 12        |
| 2.2.4    | The rationalist approach to building ITS . . . . .   | 13        |
| 2.2.5    | The motivation for ITS . . . . .   | 16        |
| 2.2.6    | The response of educators to ITS . . . . .   | 19        |
| 2.2.7    | The role of educational feedback . . . . .   | 21        |
| 2.3      | The focus of this study . . . . .  | 23        |
| 2.3.1    | The first goal: a pragmatic tutorial dialogue system . . . . .   | 23        |
| 2.3.2    | The second goal: evaluating the new tutorial dialogue system<br>against a similar menu-based alternative . . . . . | 24        |
| 2.4      | Summary . . . . .  | 25        |
| <b>3</b> | <b>Topics in dialogue modelling and text classification</b>  | <b>27</b> |
| 3.1      | Introduction . . . . .   | 27        |
| 3.2      | Models of Human Dialogue . . . . .   | 27        |
| 3.2.1    | A brief introduction to discourse and dialogue . . . . .   | 27        |
| 3.2.2    | Contemporary models of dialogue . . . . .  | 30        |
| 3.2.3    | Human-machine conversational dialogue systems . . . . .  | 32        |
| 3.2.3.1  | Dialogue system architecture . . . . .   | 32        |
| 3.2.3.2  | Dialogue management models . . . . .   | 33        |
| 3.2.3.3  | Natural language understanding . . . . .   | 36        |
| 3.3      | Text classification . . . . .  | 37        |
| 3.3.1    | A general introduction to how classifiers work . . . . .   | 38        |
| 3.3.2    | Classifier performance metrics . . . . .   | 39        |

|          |  |           |
|----------|--|-----------|
| 3.3.3    | Feature selection . . . . .  | 41        |
| 3.4      | Summary . . . . .  | 42        |
| <b>4</b> | <b>A classroom-ready tutor: design and implementation</b>  | <b>43</b> |
| 4.1      | Introduction . . . . .   | 43        |
| 4.2      | Design goals . . . . .   | 43        |
| 4.3      | Choice of domain and approach to system development . . . . .  | 45        |
| 4.3.1    | Domain choice . . . . .  | 45        |
| 4.3.2    | Data collection through fielding a prototype system . . . . .  | 46        |
| 4.3.3    | Revising the dialogue script and building the new dialogue engine                                    | 47        |
| 4.4      | A new surface-based tutorial dialogue system . . . . .   | 48        |
| 4.4.1    | A walk through the tutorial dialogue . . . . .   | 48        |
| 4.4.2    | System architecture . . . . .  | 50        |
| 4.4.3    | Script design . . . . .  | 51        |
| 4.4.3.1  | Contribution node element . . . . .  | 53        |
| 4.4.3.2  | Backward element . . . . .   | 55        |
| 4.4.3.3  | Forward element . . . . .  | 56        |
| 4.4.3.4  | Limited mixed initiative provision . . . . .   | 58        |
| 4.4.4    | Dialogue manager . . . . .   | 59        |
| 4.4.4.1  | Processing a contribution node . . . . .   | 62        |
| 4.4.5    | Tutor server . . . . .   | 65        |
| 4.4.6    | Preprocessor . . . . .   | 66        |
| 4.4.7    | Invoking the Classifier . . . . .  | 68        |
| 4.5      | Approach to Classification . . . . .   | 68        |
| 4.5.1    | Context-specific and context-general classifiers . . . . .   | 69        |
| 4.5.2    | Classifiers for multi-part questions . . . . .   | 71        |
| 4.5.3    | Method for training and testing classifiers . . . . .  | 71        |
| 4.6      | Summary . . . . .  | 74        |
| <b>5</b> | <b>Classifier and script design case-studies</b>   | <b>75</b> |
| 5.1      | Introduction . . . . .   | 75        |
| 5.2      | Binary Questions . . . . .   | 76        |
| 5.2.1    | Case study: Can you feel a pulse in someone's vein? . . . . .  | 76        |
| 5.2.1.1  | Objective of the question . . . . .  | 77        |
| 5.2.1.2  | Training dataset description . . . . .   | 77        |
| 5.2.1.3  | Selecting classes . . . . .  | 77        |
| 5.2.1.4  | Feature selection, bench-testing and script revisions . .  | 79        |
| 5.2.2    | Binary Question Summary . . . . .  | 79        |
| 5.3      | Multi-part Questions . . . . .   | 80        |
| 5.3.1    | Case study: Can you think of the three main factors which affect<br>cardiac contractility? . . . . . | 80        |
| 5.3.1.1  | Objective of the question . . . . .  | 80        |
| 5.3.1.2  | Training dataset description . . . . .   | 81        |
| 5.3.1.3  | Selecting classes . . . . .  | 82        |
| 5.3.1.4  | Feature selection, bench-testing and script revisions . .  | 83        |
| 5.3.2    | Multi-part Questions Summary . . . . .   | 84        |

|          |   |            |
|----------|---|------------|
| 5.4      | Open Questions . . . . .  | 84         |
| 5.4.1    | Case study: What is the pulse? . . . . .  | 86         |
| 5.4.1.1  | Objective of the question . . . . .   | 86         |
| 5.4.1.2  | Training dataset description . . . . .  | 87         |
| 5.4.1.3  | Selecting classes . . . . .   | 88         |
| 5.4.1.4  | An initial revision of the classifier: adjustment of feature sets and entropy threshold . . . . . | 93         |
| 5.4.1.5  | A second revision of the classifier: high-information words . . . . .                             | 94         |
| 5.4.2    | Case study: Can you describe what is meant by contractility? . . . . .                            | 96         |
| 5.4.2.1  | Objective of the question . . . . .   | 96         |
| 5.4.2.2  | Training dataset description . . . . .  | 97         |
| 5.4.2.3  | Selecting classes . . . . .   | 97         |
| 5.4.2.4  | Feature selection, bench-testing and script revisions . . . . .                                   | 99         |
| 5.4.3    | Open Question Summary . . . . .   | 100        |
| 5.4.4    | Classifier bench evaluation results . . . . .   | 101        |
| 5.5      | Summary . . . . .   | 104        |
| <b>6</b> | <b>An in-class evaluation of the tutorial system: description and evaluation methods</b>          | <b>106</b> |
| 6.1      | Introduction . . . . .  | 106        |
| 6.2      | Background and Ethics approval . . . . .  | 107        |
| 6.3      | Tutor Evaluation . . . . .  | 108        |
| 6.3.1    | Classifier evaluation . . . . .   | 108        |
| 6.3.2    | Student experience evaluation . . . . .   | 108        |
| 6.4      | Version Comparisons: Free-text vs Menu-based . . . . .  | 109        |
| 6.4.1    | Experimental Design . . . . .   | 109        |
| 6.4.2    | Data Collection and Management . . . . .  | 111        |
| 6.4.3    | Question Marking . . . . .  | 112        |
| 6.4.3.1  | Pre-test and immediate post-test . . . . .  | 112        |
| 6.4.3.2  | Delayed post-test . . . . .   | 112        |
| 6.4.4    | Summary . . . . .   | 113        |
| <b>7</b> | <b>Results of the in-class evaluation</b>   | <b>114</b> |
| 7.1      | Introduction . . . . .  | 114        |
| 7.2      | Overall Tutor Performance . . . . .   | 114        |
| 7.2.1    | In-class classifier performance data . . . . .  | 115        |
| 7.2.1.1  | Case study classifier performance data . . . . .  | 115        |
| 7.2.1.2  | In-class classifier performance summary and discussion . . . . .                                  | 116        |
| 7.2.2    | Student experiences . . . . .   | 117        |
| 7.2.2.1  | Student completions . . . . .   | 117        |
| 7.2.2.2  | Student evaluations . . . . .   | 118        |
| 7.2.2.3  | Unsolicited feedback . . . . .  | 120        |
| 7.3      | Version Comparisons: Free-text vs Menu-based . . . . .  | 122        |
| 7.3.1    | Immediate post-test . . . . .   | 123        |
| 7.3.2    | Delayed post-test . . . . .   | 124        |
| 7.3.3    | Post-hoc data analysis . . . . .  | 127        |

|          |  |            |
|----------|--|------------|
| 7.3.3.1  | Timing of tutorial . . . . .   | 127        |
| 7.3.3.2  | Implication for overall results . . . . .  | 129        |
| 7.3.3.3  | Time taken for tutorial . . . . .  | 130        |
| 7.3.3.4  | Student cohort . . . . .   | 131        |
| 7.3.3.5  | Immediate post-test question type . . . . .  | 134        |
| 7.4      | Overall summary of the evaluation . . . . .  | 135        |
| <b>8</b> | <b>Discussion</b>  | <b>138</b> |
| 8.1      | Introduction . . . . .   | 138        |
| 8.2      | Rationale for the new tutorial dialogue system . . . . .   | 138        |
| 8.3      | Implications of the in-class evaluation . . . . .  | 140        |
| 8.4      | Limitations of this study and some opportunities for further work . . .  | 141        |
| 8.5      | Main contributions of this thesis . . . . .  | 143        |
| 8.6      | Some closing remarks . . . . .   | 145        |
|          | <b>References</b>  | <b>146</b> |
| <b>A</b> | <b>Dialogue script and dialogue manager source</b>   | <b>156</b> |
| A.1      | Homeostasis script XML file . . . . .  | 156        |
| A.2      | Tutorial dialogue system XML style file . . . . .  | 156        |
| A.3      | Dialogue manager source code . . . . .   | 156        |
| A.4      | Index of classifiers and associated questions . . . . .  | 157        |
| <b>B</b> | <b>Term definitions for Chapter 5: Classifier and script design case-studies</b>                               | <b>158</b> |
| B.1      | Question categories . . . . .  | 158        |
| B.2      | Classifier types . . . . .   | 158        |
| B.3      | Training dataset description and class selection . . . . .   | 159        |
| <b>C</b> | <b>Information for Students and Evaluation Summary</b>   | <b>160</b> |
| C.1      | Information for students . . . . .   | 160        |
| C.2      | Student evaluation summary . . . . .   | 162        |
| <b>D</b> | <b>Pre-test and Post-test Questions and Model Answers</b>  | <b>165</b> |
| D.1      | Pre-test and model answers (in bold) . . . . .   | 165        |
| D.2      | Immediate post-test and model answers (in bold) . . . . .  | 166        |
| D.3      | Delayed post-test: selected CVS questions from HUBS final examination<br>and model answers (in bold) . . . . . | 168        |

# List of Tables

|      |   |     |
|------|---|-----|
| 2.1  | ITS Research Areas vs. Teaching Machine Features . . . . .  | 13  |
| 4.1  | Possible classes for check-hr . . . . .   | 55  |
| 4.2  | Optional backward element tags . . . . .  | 56  |
| 5.1  | Unique Responses & No. of Respondents . . . . .   | 78  |
| 5.2  | ‘Can you feel a pulse in someone’s vein?’ response classes . . . . .  | 79  |
| 5.3  | Unique Responses & No. of Respondents . . . . .   | 82  |
| 5.4  | Inotropic Factors Classifier Bench-Test Metrics . . . . .   | 84  |
| 5.5  | Unique Responses & No. of Respondents . . . . .   | 89  |
| 5.6  | Initial ‘What is the pulse?’ response classes . . . . .   | 90  |
| 5.7  | Final ‘What is the pulse?’ response classes . . . . .   | 92  |
| 5.8  | Unique Responses & No. of Respondents . . . . .   | 98  |
| 5.9  | ‘Can you describe what is meant by contractility?’ response classes . .                                       | 99  |
| 5.10 | Multilabel classifier metrics (refer Appendix A.4 for an index of classifier names and questions) . . . . .   | 103 |
| 5.11 | Generic binary classifier metrics . . . . .   | 104 |
| 5.12 | Multi-binary classifier metrics (refer Appendix A.4 for an index of classifier names and questions) . . . . . | 104 |
| 7.1  | In-class Case-study Classifier Metrics - Open and Binary Questions . .  | 115 |
| 7.2  | In-Class Case-study Classifier Metrics - Multi-part Questions . . . . .                                       | 116 |
| 7.3  | Overall positive ratings (%1s&2s) by condition . . . . .  | 120 |
| 7.4  | Descriptive Statistics . . . . .  | 123 |
| 7.5  | Delayed Post-Test Descriptive Statistics . . . . .  | 125 |
| 7.6  | Delayed Post-Test Between-Subjects ANOVA . . . . .  | 126 |
| 7.7  | Summary of immediate post-test score ranges by condition . . . . .  | 131 |
| 7.8  | Significant Chi-square test results . . . . .   | 137 |
| A.1  | Question, classifier label and classifier type . . . . .  | 157 |

# List of Figures

|      |  |     |
|------|--|-----|
| 3.1  | Dialogue System architecture. (Modified from Jurafsky and Martin (2009, p.821)) . . . . .  | 32  |
| 3.2  | Example FSA . . . . .  | 35  |
| 4.1  | Screenshot of Dialogue System Web Client. . . . .  | 51  |
| 4.2  | Architecture of the Dialogue System. . . . .   | 52  |
| 4.3  | Single Dialogue Contribution . . . . .   | 54  |
| 4.4  | Script Backward Class Illustration . . . . .   | 57  |
| 4.5  | Action Directive Element . . . . .   | 58  |
| 4.6  | Dialogue Manager Pseudocode . . . . .  | 60  |
| 4.7  | Dialogue Session Illustration . . . . .  | 61  |
| 4.8  | Tokeniser Function . . . . .   | 67  |
| 4.9  | Classifier flow diagram. $A_1 \dots A_n$ are alternative categories for the answer to the question to which the student is currently responding. If the classifier's confidence in its assignment of one of these categories falls below some threshold ( $E > e$ ), the student's response is successively input to the 'Question' classifier, the 'Don't know' classifier and the 'Don't understand' classifier. The latter two classifiers are binary classifiers; the 'Question' classifier is currently just a placeholder. . . . . | 70  |
| 4.10 | Feedback when Classification Fails . . . . .   | 73  |
| 4.11 | Sample confusion matrix. . . . .   | 74  |
| 5.1  | Frequency of responses by category. . . . .  | 78  |
| 5.2  | Frequency of responses by category. . . . .  | 81  |
| 5.3  | Frequency of unique textual responses. . . . .   | 88  |
| 5.4  | Confusion Matrix, $e = 1.0$ . . . . .  | 93  |
| 5.5  | Confusion Matrix, $e = 1.5$ . . . . .  | 94  |
| 5.6  | High-Information Word Function Pseudocode . . . . .  | 95  |
| 5.7  | Final Confusion Matrix, $e = 1.5$ . . . . .  | 96  |
| 5.8  | Frequency of responses by category. . . . .  | 97  |
| 5.9  | Contractility Confusion Matrix, $e = 1.0$ . . . . .  | 100 |
| 5.10 | Contractility Confusion Matrix, $e = 0.8$ . . . . .  | 101 |
| 7.1  | Distribution of normalised pre-test minus normalised post-test scores. .   | 124 |
| 7.2  | Distribution of normalised pre-test minus normalised post-test scores. .   | 126 |
| 7.3  | Distribution of student participation in experiment by date. . . . .   | 128 |
| 7.4  | Linear model of normalised scores over time by condition. . . . .  | 129 |

|      |  |     |
|------|--|-----|
| 7.5  | Distribution of Time Spent on Tutorial Task . . . . .  | 130 |
| 7.6  | Linear model of normalised baseline scores over time on task by condition.                                     | 132 |
| 7.7  | Distribution of final grades for the HUBS class. . . . .   | 133 |
| 7.8  | Distribution of final grades for experiment volunteers. . . . .  | 134 |
| 7.9  | Scatter plot and regression line for score correlation: Final exam vs<br>immediate post-test. . . . .          | 135 |
| 7.10 | Scatter plot and regression line for final score correlation with date of<br>experiment participation. . . . . | 136 |
| C.1  | Information for students - Page 1 . . . . .  | 160 |
| C.2  | Information for students - Page 2 . . . . .  | 161 |
| C.3  | Questionnaire Summary - Page 1 . . . . .   | 162 |
| C.4  | Questionnaire Summary - Page 2 . . . . .   | 163 |
| C.5  | Questionnaire Summary - Page 3 . . . . .   | 164 |

# Chapter 1

## Introduction

### 1.1 Background

More than 30 years ago, on a warm February afternoon, I trotted into the University Bookshop and bought a copy of *Best and Taylor's Physiological Basis of Medical Practice*. The book was thick and heavy, the pages were wafer thin and the print was tiny. I was a freshly minted new entrant to Medical School and this book was the prescribed reading for the human physiology course. The book made no sense to me at the start of the year and, supported by sedative lectures and laboratory encounters with pithed toads, perhaps even less by the end. I failed the final exam but to the great relief of family and friends and to my consternation, I was given the opportunity to sit a special exam. This I passed, after a summer spent carefully translating Best and Taylor's work into words which made sense to me. If it was hard to get into medical school, I discovered to my cost that it was very much harder to get out.

It was only towards the final stages of my medical degree, several years later, that my patchy conceptions of human physiology began to gel into an understanding which might find utility in medical practice. Fortunately for me, as well as for the injured and infirm, I finally found the courage to make alternative career arrangements before my practical skills were seriously put to the test. Strangely, a career shift into multimedia and computing and from there into educational technology, has somehow led me back to where I began and to this thesis.

## 1.2 Where this project began

In large undergraduate classes, it is time-consuming, costly and seldom practical for the teacher to provide students with individualised feedback on their written responses to questions. Typically, computer-based marking of formative tests is used as an alternative and examples of this include Learning Management System (LMS) based multiple-choice quizzes or similar. Most computer-assisted assessment, whether formative or summative, involves students being able to recognise a correct response rather than recall and independently generate an answer. In the context of the first-year undergraduate health sciences course that was the subject of this study (see Section 1.3 for details), all computer-assisted assessment takes this form. In 2008, the coordinator of this class asked me about the ways in which technologies might assist students to practise writing answers to short-answer, or constructed response, questions. Anecdotally, students typically performed poorly on these types of questions in the final exam relative to the multiple-choice questions. The course coordinator hypothesised that this was due to the lack of opportunity during the course for students to practise answering such questions: there were simply not enough teaching staff available to mark all the student responses.

As a result of this request, in addition to setting out to explore potential technology-based solutions for providing individualised feedback to students, I also wanted to investigate whether there were performance gains on test scores from formative assessment interventions and to formally test the course coordinator's hypothesis: students who generate free-text answers to practice questions perform better at assessment time than those who select their preferred answer from a menu of options. By chance, I started to ask these questions at a time when early Massive Open Online Courses (MOOCs) were starting to appear. These are now proliferating and ways of both engaging and assessing students in these new learning environments are increasingly needed. The questions considered in this thesis are also among those which need to be addressed by the designers of MOOCs.

## 1.3 Classroom context

The first year health sciences course at the University of Otago is a prerequisite for entry into all the professional health science programmes, including Medicine, Dentistry, Medical Laboratory Science, Pharmacy, and Physiotherapy. Entry into these

professional programmes is highly competitive and is dependent on students achieving excellent grades in their 1st year course. Depending on the specific professional programme there are additional requirements such as a satisfactory interview and an appropriate score in the international Undergraduate Medicine and Health Sciences Admission Test (UMAT). There are seven required papers (or units of study) in the first year health sciences course which cover a wide-range of fields in biological science and science and which aim to provide a strong foundation for subsequent study in any of the health science professions as well in biological science generally. Human Body Systems or HUBS 192 is one of the compulsory papers and provides a comprehensive introduction to the structure and function of the human respiratory, cardiovascular, gastrointestinal, and genito-urinary systems. The HUBS paper consists of 4 one-hour lectures and a 3-hour laboratory session each week and completion of self-directed study modules for each section of the course. Assessment is through a combination of end of year and mid-semester examinations and marks for self-study and laboratory assignments. The number of students enrolled in HUBS 192 varies each year, but ranges from around 1300 to 1800 students. These students are known for being highly competitive, motivated and engaged and tend to welcome any opportunity for additional study support. The cardiovascular section of HUBS 192 was chosen as the specific domain for this research for reasons which are detailed in Section 4.3.1.

## 1.4 Choice of approach

Intelligent tutoring systems (ITS) which employ natural language as their interface (dialogue-based ITS) seemed to offer some promise for supporting and enhancing student understanding of key concepts in the current classroom context. Automated essay-marking systems were considered as an alternative but the appeal of tutorial dialogue was that questions are embedded in a tutorial plan: the questions arise in a meaningful context, and students' free-text contributions can be assessed in relation to this context. In an essay, the way in which students express their knowledge is much less constrained, and assessing this knowledge automatically is correspondingly harder.

The research and development of ITS is an active field in the cognitive sciences, in particular in artificial intelligence (AI) in education. However, I became aware from discussion with colleagues in the academic staff development centre in which I work and from a brief review of the educational literature at the time, that dismissal of ITS as a failed enterprise was a common view among educators (Laurillard, 2002;

Ramsden, 2003). This issue is explored in detail in Chapter 2. It turns out, while very much an active research area, ITS are not in widespread use in educational settings (Reeves and Hedberg, 2003; Shute and Zapata-Rivera, 2010; Mitrovic, Martin, Suraweera, Zakharov, Milik, Holland, and Mcguigan, 2009). There are some exceptions to this including the success of the cognitive tutors from Carnegie Mellon in the United States high school algebra setting (See <http://www.carnegielearning.com/specs/cognitive-tutor-overview/>) and some examples from **STEM** (science, technology, engineering, mathematics) subjects in the wider international higher education context (See, for example VanLehn, Graesser, Jackson, Jordan, Olney, and Rosé (2007), Kumar, Rosé, Wang, Joshi, and Robinson (2007), Mitrovic (2012)). Nonetheless, ITS are hardly on the educational agenda, certainly in Australasia, at the present time.

Aside from educational objections, a key practical issue is that it is seldom easy to adapt an ITS which has been designed for a specific teaching and learning context to another context. If this is to work, domain independence needs to be designed into the system from the outset. *Circsim Tutor* (Evens and Michael, 2006) is an example of an impressive tutorial dialogue system designed expressly to develop student understanding of the baroreceptor reflex in humans (the baroreceptor reflex is one of the mechanisms for maintaining blood pressure). The system probes student understanding with written questions delivered via a computer interface and students respond with typed text input. Questions are related; through dialogue with the tutor the student has the opportunity to develop and practise their understanding of the role of the different but interlinked variables which are involved in the reflex. While *Circsim Tutor* deals with aspects of cardiovascular physiology it deals with them at a level which is too advanced for the broader introductory-level course on the cardiovascular system with which I was dealing. Even if this were not the case, there would likely be important differences in emphasis in terms of the curriculum and it was clear that making significant adaptations to *Circsim Tutor* would be a non-trivial task. While there are certainly ITS which achieve, or which aim to achieve, a level of domain independence, few are readily accessible to teachers without the support of computer programming or AI specialists, not to mention the institutional IT support which is usually required to run any specialised desktop or web application as a routine part of a course.

These difficulties notwithstanding, the idea of tutorial dialogue as an educational tool is both prevalent and powerful and there is considerable evidence of its benefits whether conducted by humans or by machines (this point is discussed in detail in

Section 2.2.5). I therefore chose to persist with the idea of a dialogue-based ITS as an option for providing individualised feedback to students and, for reasons which are described in detail in Section 2.3.1, adopted a **pragmatic** approach to its design. This leads to the purpose of the present study which is summarised in Section 1.5.

## 1.5 The goals of this study and main contributions

The first goal of this study was to design, build and evaluate a pragmatic tutorial dialogue system for the cardiovascular section of the HUBS 192 course. By ‘pragmatic’ I mean a dialogue system which is firmly empirically-based, with both the teaching context and real student responses to questions integral to its design and implementation, and which uses lesson plans that can be understood and authored by teachers without extensive training.

The second goal was to determine whether the opportunity to practise writing answers to short-answer questions and receiving detailed automated feedback would result in performance gains. Two versions of the new system were required in order to explore this: in the first version of the system, students type their response to questions as free-text and in the second version, students select the answer they prefer from a menu of options.

The main contributions of this work are applied rather than theoretical, since the focus is on exploring a tutorial dialogue system in a real classroom setting. Through providing a solid platform for ongoing research and development in a real class context, the stage is set for developing a broader educational design research agenda (Van den Akker, 1999). Design research involves a collaboration between teachers and researchers. Real teaching and learning problems are identified, prototype solutions, based on existing principles, are developed and these are tested and refined ‘until satisfactory outcomes have been reached by all concerned’ (Reeves, 2006, p.59). A strong advocate of design research, Reeves (2006) also notes that,

... the conceptualization of learning theory as something that stands apart from and above instructional practice should be replaced by one that recognizes that learning theory can be collaboratively shaped by researchers and practitioners in context. (p.61)

The tutorial dialogue system described is new in the context of teaching first year undergraduate physiology, was acceptable to both staff and students which is critical for further work and adds to the body of tutorial dialogue corpora available for NLP research. Attaining ‘satisfactory outcomes for all concerned’ is well beyond the scope of this current work; it is but a first step along the way and a contribution towards getting ITS back on the educational agenda in the Australasian context.

## 1.6 Outline of this thesis

This study is situated within at least two broad fields of inquiry: first, educational research, in particular aspects of educational assessment and second, artificial intelligence (AI) research, in particular, intelligent tutoring systems (ITS) and aspects of computational linguistics and natural language processing (NLP). The first two chapters provide coverage of relevant topics from these fields. Chapter 2 begins with a brief history of ITS including its origins in the computing, behavioural and cognitive sciences. The rationalist vs empiricist debate is introduced and the motivation for contemporary efforts to build ITS is explored. Finally, the response of practising teachers and educators to ITS is detailed along with the role of educational feedback. The chapter concludes with a detailed description of the goals of this study and the rationale for these. Chapter 3 introduces two important topics from AI, NLP and computational linguistics which are especially relevant to this study: models of human dialogue and elements of statistical NLP which relate to text classification.

Chapter 4 introduces the new tutorial dialogue system. The design goals and early stages of the project are discussed including the rationale for the choice of domain, data collection and system prototyping. The main focus of the chapter is a detailed description of the design and implementation of the new dialogue-based ITS including the domain-specific dialogue script which determines the content and direction of the dialogue.

Chapter 5 describes a series of case-studies, taken from representative parts of the dialogue script, and discusses some key features of tutorial questions and issues related to recognising student responses to them. The chapter concludes with a summary of bench-test performance data for all classifiers used in the study.

Chapter 6 describes the methodology for conducting both the in-class evaluation of the new system and an experiment to compare the two versions of the system. Chapter 7 presents the results of both the evaluation and the experiment.

Chapter 8 begins with a discussion of the results of the new system evaluation, including what worked in the classroom context as well as some limitations of the system. The experimental results are discussed and specific avenues for further research are identified along with limitations of the study. The chapter concludes with a discussion of the unique contributions of this research and the potential for further development.

# Chapter 2

## Past to present: a multi-perspective review of intelligent tutors

### 2.1 Introduction

As indicated in the previous chapter, intelligent tutoring systems (ITS) have yet to find a place in mainstream education. Yet the promise of educational technology in general, and of computer or machine tutors in particular, retains much of the allure for contemporary educators that it had nearly a hundred years ago. It is important to understand why this is the case and in particular, to better understand the needs and constraints of contemporary higher education settings if a design for a practical solution is to be produced and evaluated.

This chapter describes the context and rationale for the study as a whole and provides a justification for the key research questions and practical aims. The history of ITS is covered in some detail including its origins in the computing, behavioural and cognitive sciences.

### 2.2 A brief history of intelligent tutoring systems

ITS have had an uncomfortable history in the field of education and a turbulent one even in the cognitive sciences. To understand why this is the case, it is worth going back to the behaviourist roots of the forerunner to ITS, the teaching machine.

### 2.2.1 The behaviourists

In the early part of the 20th century when the industrialisation of the West was in full-swing people began to ask whether machines could help us teach. Behaviourism, in New Zealand and the United States at least, is most closely associated with the work of psychologist B.F. Skinner (Corballis, 2006). Skinner had many proposals about educational methods but his work was foreshadowed by earlier psychologists. In the 1920s Sidney Pressey, a psychologist at Ohio State University, invented a mechanical device for automating the recording of answers to multiple choice questions as part of his research in intelligence testing. Pressey (1926) wrote:

the procedures in mastery of drill and informational material were in many instances simple and definite enough to permit handling of much routine teaching by mechanical means. (pp 373-374)

Pressey modified his testing machine to include a lever which, when activated, prevented the user from advancing to the next question until she had chosen the correct answer to the current question. Pressey's insight was recognising the potential of his device as a teaching machine. His modified device provided immediate feedback to the user and eliminated questions from the test set once they had been answered twice in a row. According to Skinner (1958)

Such machines [...] could not only test and score, they could *teach*. When an examination is corrected and returned after a delay of many hours or days, the student's behavior is not appreciably modified. The immediate report supplied by a self-scoring device, however, can have an important instructional effect. (p.969 emphasis in original)

Pressey envisaged an industrial revolution in education to 'free teacher and pupil from educational drudgery and incompetence' (cited in Benjamin, 1988, p.707). His own efforts in this regard failed to come to much although his devices were manufactured and distributed for a short time by Welch Scientific Company, Chicago. Skinner (1958, p.969) suggested two reasons why Pressey's machines failed. For one thing they 'succumbed ... to cultural inertia; the world of education was not ready for them.' In addition, 'Pressey was working against a background of psychological theory which had not come to grips with the learning process.'

Benjamin (1988) argued that Pressey failed for more pragmatic reasons.

Pressey's machines, which promised education at a faster pace and a need for fewer teachers, appeared during the Great Depression when there was a great surplus of teachers and no public pressure to increase the pace of education ... Conditions were very different for Skinner. (p.708)

In the 1940s and 1950s Skinner picked up where Pressey left off, but there were some important differences between Pressey's and Skinner's machines. Skinner developed the idea of introducing very small amounts of information to students one step at a time and allowing them to test their understanding of it before proceeding to more advanced material. He introduced the idea of programmed instruction, proceeding in very small steps until all material on a particular topic was covered. In addition, Skinner felt it was important for students to write out answers to questions rather than simply choose from predetermined choices. Skinner (1958) described five tutor-like affordances of his teaching machine:

1. The student actively engages with the machine.
2. The machine will not allow the student to progress until a point has been mastered.
3. Only material the student is ready for is presented.
4. Hints, prompts and suggestions derived from the analysis of verbal behaviour support the student to come up with the desired response.
5. Feedback from the machine reinforces every correct response.

By the early 1960s, more than 100,000 teaching machines, which were based on Skinner's ideas, had been marketed and sold, mostly to schools in the U.S., by Grolier. There were other versions too although according to Benjamin (1988), Skinner himself failed to successfully market his teaching machine. In spite of the flurry of public and media interest in teaching machines during the early 60s, by the late 60s and early 70s interest in - and the classroom use of - teaching machines had all but disappeared. There are number of possible explanations for this. A frequent concern was the fear that somehow teaching machines threatened teachers' jobs. Skinner himself was clear that the machines should free up teachers for more important tasks rather than replace them (Skinner, 1958). He also emphasised the benefit of the machines as providing personalised attention and immediate formative feedback for students that was impossible in many classroom settings. A more practical problem was the lack of programs for

the machines. Developing fine-grained programmed instruction was then, as it is now, a highly complex and time-consuming enterprise. There simply were not the programs available for the machines.

## 2.2.2 The first intelligent tutoring system

Jaime Carbonell's *Scholar* (Carbonell, 1970) is frequently cited as the earliest intelligent tutor (see for example, Woolf (2008), Evens and Michael (2006), Pea (2004), Shute and Psotka (1994), Merrill, Reiser, Ranney, and Trafton (1992)). *Scholar* produced individualised responses to student statements in a specific domain (for example, South American geography) using a semantic network. A semantic network can be thought of very simply as a model of the relationships between objects. For example, mountains have a number of properties such as location, height and so on. These can be modelled as labelled relations between mountains and other entities. A given mountain could have a 'height' relation to a particular value, or an 'in' relation to a particular country. A semantic network attempts to represent the relationships between objects and their properties in a given domain. In addition, *Scholar* could also respond to student questions (that is, it provided limited mixed-initiative functionality - see Chapter 3 for a detailed description of mixed initiative dialogue) and could change mode between mixed-initiative, question and answer, and test. Carbonell's system parsed natural language input (that is, it could break down a sentence into its component parts and analyse the syntactic role of each part) using a system based on case grammar (Fillmore, 1968). In brief, case grammar allows cases (for example, agent, dative, locative) to be specified for each argument of a verb in a sentence. Logical inferences can then be derived from the sentence. For example, the sentence, 'Jane gave Jack an apple', features the verb 'to give' which takes the following set of cases: the agent case (Jane), the dative case (Jack) and the object case (an apple). This allows us to infer the fact that Jack possessed the apple **after** Jane. Natural language output was template-based, or pre-written, to match specific logical outputs. *Scholar* then translated the parsed input into a logical form for processing by a semantic network.

A number of features follow from this very brief description of the first ITS. First, ITS tend to be highly domain sensitive. In other words they will not handle natural language outside their domain of 'understanding'. *Scholar*, if asked to name the highest mountain in South America might have responded with 'Aconcagua in Argentina'. On the other hand, if asked to name the highest mountain in Europe, it would have no answer. Second, ITS which use a symbolic grammar are not robust to subtle changes in

students' written input. If the question about the highest mountain in South America were subtly rephrased so that it fell outside the systems grammatical coverage, the system would comprehensively fail to interpret it, and simply give an error response. Third, the domain of 'knowledge' represented by the system needs to be mapped out and represented in some way. This presents a whole raft of issues relating to knowledge representation. What to leave in (all mountains in South America?), what to leave out (rivers in South America less than 10 miles long?), what relationships to express (all geographic features in South America discovered by Mr X?), (all geographic features which are located in countries/states/regions/cities? in South America?) and so on. The complexity of the problem of knowledge representation and/or *understanding* rapidly becomes apparent. Fourth, some way of representing the path or progress of the student as they engage with the Tutor would be useful to avoid circular conversations, gauge student progress and potentially model likely actions. Fifth, significant analytical effort is required in order to build an ITS even in a very restricted domain.

### 2.2.3 The empiricists and the rationalists

From a philosophical standpoint, *Scholar* can be thought of as representing a rationalist approach to building an ITS. A set of rules which can take linguistic input from the tutee are fully specified, and an appropriate response is calculated and then produced as linguistic output.

The philosophical distinction between rationalism and empiricism is important in the context of intelligent tutoring systems. Manning and Schütze (1999) characterise the empiricist 'camp' as privileging sensory input over mental organisation and contrast this to the rationalist position which emphasises innate mental structure over sensory input. The poverty of stimulus argument (see for example, Chomsky (1986), Pinker (1984)), essentially the idea that children acquire their language from limited input, is often given as evidence to support the rationalist innate structure hypothesis, however it is still hotly debated (see for example, Pullum (1996), Crain and Pietroski (2001), Clark and Lappin (2010)). Empiricists like Wittgenstein and Skinner simply did not believe that meaning and language could be separated (see Wittgenstein (1968) and Skinner (1957)). For the empiricist, language meaning is use; it takes its shape, structure and meaning from how we use it in practice. The only concession to innateness made by the empiricists is some pre-existing cognitive capacity for recognising patterns and generalising from them (Manning and Schütze, 1999).

The empiricist approach to building an ITS would involve taking linguistic input

from the tutee and looking up the most appropriate linguistic response based on learned knowledge about how to respond. In other words, no calculations or assumptions are made about either mental or machine state; there are no rules. If a particular linguistic pattern or feature-set has been seen before then the machine responds on the basis of a known response to that pattern; if not, it makes no assumptions and simply says (or types) ‘I don’t know’.

The prevailing approach adopted in the development of ITS up until quite recently has been, without a doubt, the rationalist one. Having introduced Scholar, the prototypical rationalist system, in the next section, 2.2.4, additional examples of this style of system are discussed.

## 2.2.4 The rationalist approach to building ITS

Shute and Psotka (1994), in an extensive review, characterise ITS during the 1970s as idiosyncratic experimental systems. In Table 2.1, Shute and Psotka’s list of research areas is extended to note the differences between the early ITS and teaching machines of the 50s and 60s.

Table 2.1: ITS Research Areas vs. Teaching Machine Features

| ITS research area                | Teaching machine feature |
|----------------------------------|--------------------------|
| Real-time generation of problems | Pre-defined problems     |
| Student model development        | None                     |
| Knowledge representation         | Frame-based encoding     |
| Student error modelling          | Correct path             |
| Expert systems                   | None                     |

During the 1980s, efforts became more focused particularly on developing student models and developing authoring tools for ITS (Shute and Psotka, 1994). The relationship to linguistics research was also developing although the language or communication *module* was viewed as separate from the *knowledge* module:

Linguistic research is relevant at two levels. First, there is much interest in endowing intelligent tutoring systems with some capacity to understand and use natural languages. Second, most speech acts can be viewed in a very broad sense as a form of knowledge communication. (Wenger, 1987, p.10)

There was increased interest in ITS, and high expectations, but a key issue was the lack of evaluation of systems that were developed (Shute and Psotka, 1994).

Shute and Psotka (1994) reviewed six evaluation studies that appeared during the 1980s and early 1990s. They concluded that the results were generally positive but noted the problem of selection bias with the publication of unambiguous results of successful interventions.

In contrast, Evens and Michael (2006), writing about their *Circsim-Tutor*, which was mentioned in Chapter 1, noted that during the same period (late 1980s - early 1990s),

...we were aware of the growing body of work in which applications of computer science and artificial intelligence to education were beginning to yield increasingly robust and interesting results. (Evens and Michael (2006) p.13)

Evens and Michael (2006) describe two studies which overlap with those examined by Shute and Psotka (1994); *LISP tutor* (Anderson, Conrad, and Corbett, 1989) and *SHERLOCK* (Lesgold, Lajoie, Bunzo, and Eggan, 1992). It is interesting to compare these two views of similar data. One group of researchers suggest generally positive evaluation results but issue a strong note of caution, while the other group suggest robust and interesting results and an increasing trend.

Shute and Psotka (1994) offered an uncertain outlook for the practical application of ITS but were optimistic about the prospects for research around ITS.

A philosophical shift has been suggested ... away from stand-alone instructional devices and toward using tools to aid in the more collaborative learning process. There are actually very few ITS in place in schools, yet they exist in abundance in research laboratories. We need to move on. (p.50)

By the 1990s and through to the present day, the general architecture of ITS has been resolved to include at least some kind of knowledge base, which might include simulations or an expert system, an expert problem solver (these two together constituting a domain model), a student model, some kind of teaching model (Evens and Michael, 2006), authoring tools to allow teachers, and in particular those without specialist programming knowledge, to create the ITS in context (see Murray, 1999) and increasingly, dialogue modules to facilitate two-way communication in free-text form (for example, *Auto-Tutor* (Graesser, Hu, Susarla, Harter, Person, Louwerse, Olde, *et al.*,

2001), *Why2-Atlas* (VanLehn, Jordan, Rosé, Bhembé, Böttner, Gaydos, Makatchev, Pappuswamy, Ringenberg, Roque, *et al.*, 2002), *Circsim-Tutor* (Evens and Michael, 2006)).

With a few exceptions (for example, the cognitive tutors used in some U.S. schools to support algebra and mathematics teaching (Aleven, McLaren, Sewall, and Koedinger, 2009)) ITS are still very much in the cognitive science research domain rather than in widespread practical use in teaching and learning. There are clearly good practical reasons for this. Murray (1999), in his review of ITS authoring systems, addresses one:

Building an explicit model of anything is not an easy task, and requires analysis, synthesis, and abstraction skills along with a healthy dose of creativity. Authoring tools can significantly decrease the cognitive load involved in various design steps, but it is difficult to reduce the entire design task to low level decisions that yield a quality product [...] Even though an authoring tool might be able to reduce the process to simple atomic steps done in isolation, some degree of holistic understanding and abstract thinking will eventually have to come into play. (p.124)

In a footnote on the same page, Murray continues:

I am not arguing against the merit of special purpose authoring tools, but against the idea that untrained authors can create quality ITSs.

More recently, there have been concerted efforts to address this issue (for example, Aleven *et al.*, 2009; Mitrovic *et al.*, 2009) but even these authors are some way from claiming to have solved the problem:

Although we do not feel we are done yet [...] authoring tools [...] are reaching a mature state where they support development of real-world tutors. [...] It is our strong belief that in years to come, authoring tools that non-programmers can use [...] will contribute greatly to making ITSs widespread. (Aleven *et al.*, 2009, p.145)

In short, ITS have suffered from the same major difficulty as the teaching machine and the programmed instruction agenda. Unless a teacher is able to utilise a pre-authored ITS and use it ‘as-is’ in their own particular teaching context, then a significant commitment in terms of time and effort will be required to achieve learning gains that may be achieved more easily by more conventional means. It is perhaps not

too surprising that even today few ITS are found in classrooms at schools or universities.

In a sense, Carbonell's *Scholar* (Carbonell, 1970) described above in section 2.2.2, as the first ITS, denotes a fork in the road of computer assisted instruction (CAI) where the addition of 'intelligence' to CAI led away from simple drill and practice or programmed instruction. Until that point, teaching systems, including the earlier teaching machines, whether used in teaching practice or confined to research laboratories, typically constituted questions and answers encoded as blocks or *frames* of text which were sequenced and presented to students according to a set of predetermined rules. As Carbonell remarked:

In most CAI systems [of the ad-hoc frame oriented type], the computer does little more than what a programmed textbook can do, and one may wonder why the machine is used at all. Some systems allow some degree of processing of unanticipated answers, time can be measured, and statistics are collected in most cases, but not much more. (Carbonell, 1970, p.194)

Nevertheless, criticisms of CAI as automated page-turners notwithstanding, it is important to point out that while Skinner's teaching machines had largely disappeared by the 1970s, with the advent of personal computers in the late 1980s, frame oriented or similar CAI systems continued to be used in teaching practice. The majority, if not all, higher education institutions in the West today deploy learning management systems (LMS) (Coates, James, and Baldwin, 2005) and all LMS provide, amongst other things, suites of tools comprising MCQ templates or variants thereof for teachers to populate with educational material. While these tools are considerably more sophisticated and flexible than those available in the 1970s they are still based on deterministic drill-and-practice models where the shadow of Skinner and Pressey is unmistakeable.

### 2.2.5 The motivation for ITS

Even if rarely found in educational practice, ITS have persisted in the research domains of cognitive science and educational psychology. In looking for a reason why this might be the case, no reviewer exploring the ITS literature could fail to notice the impact of Bloom's 1984 study and what has become known as the 2 sigma problem. A typical example of how Bloom's paper is regularly positioned is provided by Nielsen, Ward, and Martin (2008).

In 1984, Benjamin Bloom determined that the difference between the amount and quality of learning for a class of thirty students and those who received individualised tutoring was 2 standard deviations (Bloom, 1984). The significant differences in proficiency between those children who enjoy one-on-one tutoring versus those who have little individualised support is testament to the need for further exploration of the individualised tutoring model. (p.204)

The Bloom (1984) study (1854 citations according to Google Scholar at March 11, 2014) claimed an effect size (ES) of 2.0 for human ‘expert’ tutoring and is regularly cited in the ITS literature and textbooks not only as the benchmark against which machine tutors should be compared, but also as the reason why the provision of individualised support is a worthy goal. (For example, Woolf (2008), Evens and Michael (2006), Nielsen *et al.* (2008), Olney, Graesser, and Person (2010)). Effect size is a standardised measure of the the difference between the means of two groups, usually an experimental group and a control group. An effect size of 2.0 literally means that the average score in one group is 2.0 standard deviations higher than the average score in a second group. Amongst other things, effect size provides a standardised tool to compare studies which investigate the same dependent variable. This is particularly useful in meta-analytic studies.

However, similar studies to Bloom’s demonstrate less impressive and highly variable effect sizes. Cohen, Kulik, and Kulik (1982) reviewed 52 studies, found an average effect size of 0.40 and noted that the size of the effect varied widely, the largest being 2.3. Cohen *et al.* went further and described six features of studies which consistently produced strong effects. These included those in the mathematics domain (Average ES = 0.6), instructor developed tests as opposed to standardised tests (Average ES = 0.84), duration of treatment <4 weeks (Average ES = 0.95), structured tutoring (Average ES = 0.51), unpublished studies (Average ES = 0.85) and low-order level of achievement measure (Average ES = 0.76).

Even though the Cohen *et al.* meta-analysis was published prior to Bloom’s study, reported much lower average effect sizes and provided these together with an assessment of confidence interval, the two studies, Bloom (1984) and Cohen *et al.* (1982), are regularly cited together, with Cohen *et al.* (1982) used to support Bloom’s claim that there is a 2 sigma problem to be solved.

More recently, VanLehn (2011) found an average effect size of 0.79 when he reviewed 10 studies comparing human tutoring to no tutoring (ES ranged from -0.24 to 1.95) and

an average effect size of 0.76 for step-based (ITS) tutoring compared to no tutoring (29 studies where ES ranged from -0.32 to 1.35). These averages led VanLehn to suggest that

...ITS are, within the limitations of this article, just as effective as adult, one-on-one human tutoring for increasing learning gains in STEM topics.  
(VanLehn (2011) p.214)

But, setting the average ES values to one side for the moment, the wide range in effect size for human tutoring versus no tutoring, and for ITS tutoring versus no tutoring in VanLehn's analysis is troubling. Is it really possible to draw conclusions about relative effectiveness when ES varies so widely? As a simple check, and using the ES data supplied in the appendix to VanLehn (2011), if the median is calculated in each case instead of the average, the ES for human tutoring versus no tutoring comes down to 0.65 and that for computer tutoring compared to no tutoring rises to 0.78. What this means is that high variability needs to be taken into account in drawing any conclusions. Even if VanLehn's claim were valid it still leaves in doubt what it is about tutoring by either humans or computers which leads to learning gains.

The idea that differences in effect size are accounted for by tutor expertise is also prevalent (see for example Olney *et al.*, 2010; Evens and Michael, 2006); that is, the more expert a tutor, the greater the gains to be made. However, VanLehn (2011) casts doubt on this, at least as far as Bloom is concerned, when he notes that some of the studies conducted by two of Bloom's graduate students and reported in Bloom (1984), involved undergraduate education majors working as tutors under the guidance of the experimenters. Cohen *et al.*, did not list tutor expertise as a feature associated with large effect sizes. Also worth noting is that peer tutoring is both a practical and a currently widespread option for providing tutorial support to large numbers of students in real class settings (see for example Topping, 1996; Falchikov, 2001).

It is not that human tutors cannot be as effective as Bloom claimed; in a few documented instances they have been. Similarly, in some instances, ITS have demonstrated large effect sizes and a few are used, or have been used, in real class settings. However, on the basis of the evidence above, it is clear that both human tutors and ITS vary widely in their effects and they do not consistently produce strong positive effects. In addition, even if they were consistently effective, both humans and appropriately tooled ITS are expensive and time-consuming to train and in short supply for providing individualised support.

Given the wide variability in reported effect size, perhaps it is time to resist the rationalist urge to benchmark ITS against human tutors. The focus could usefully shift to delineating which tutoring or teaching practices or conditions produce the greatest learning effects. Indeed, there are already a number of researchers who are doing just this (see for example Chi, 2009; Chi, VanLehn, and Litman, 2010; VanLehn, Jordan, and Litman, 2007; Chi, Roy, and Hausmann, 2008). It is interesting to note that in a similar vein, Tamim, Bernard, Borokhovski, Abrami, and Schmid (2011) have argued that a more nuanced approach would also be helpful in studies which look at the effect of CAI rather than continuing comparisons between human and automated efforts.

To return to the fork in the CAI road, whatever the presumed shortcomings of computer-based drill-and-practice and programmed instruction, these systems in various forms have for nearly a century secured a place in educational practice, unlike ITS. In order to understand why such systems have stood the test of time it is worth exploring how and why they are typically used in educational contexts. It is also worth exploring the perspective of educational researchers and practitioners on the utility of ITS. These issues are considered in the next section, 2.2.6.

## **2.2.6 The response of educators to ITS**

While early behaviourism is now off the educational agenda, many of the ideas around programmed instruction linger. And, while teaching machines have become a historical curiosity, since the 1960s the desire for the tutor-like affordances embodied in some kind of machine has remained, even among educational researchers and practitioners who would seem to be farthest from the programmed instruction paradigm. The idea that computers *ought* to be able to help with the practical problem of providing one-on-one attention in self-paced learning environments, mediating access to large volumes of information or content and providing formative feedback, is prevalent (see for example Ramsden, 1992; Biggs, 1999).

A specific and important objection from educators, as well as from some educational psychologists and scholars working in the ITS domain, relates to the use of student models in ITS, where the steps taken by the student to solve a problem are compared to those used by an expert and the departure from expert steps or rules is modelled as errors. This can be seen for example in, Laurillard (1988), Collins and Brown (1988) and Scardamalia, Bereiter, McLean, Swallow, and Woodruff (1989). Susanne Lajoie and Sharon Derry compiled an entire volume which compared, contrasted and attempted to find common ground between traditional ITS model builders on the one

hand and non-modelers on the other (Lajoie and Derry, 1993).

Laurillard (1988) presents a compelling case for abandoning models of student errors. She points out that generative student models (that is, models which generate errors in a rule-governed way) can easily fail to capture the real source of student misconceptions. In the following quote she illustrates this point discussing the process of arithmetic subtraction and the use of the **borrow** rule (where a larger digit is subtracted from a smaller digit, 10 is borrowed from the next column). In the particular situation she describes, the student does not know the borrow rule:

The assumption is that the student possesses an incomplete or inaccurate set of procedures, which together with rules such as the impasse-repair mechanism [a rule-based approach to deal with the deficient set of procedures], produce characteristic errors ...the pedagogically important point here is not the absence of the “borrow” rule, but the presence of the impasse-repair mechanism. (Laurillard (1988) p.236)

The assumption Laurillard is challenging is not that the impasse-repair mechanism exists but that teaching should move beyond treating problem solving procedures ‘as a set of uninterpreted rules’. She goes on to argue for a deeper approach to understanding student conceptions and the need for learning environments which provide rich experiences and ones which promote thoughtful questioning and the opportunity to observe, act and practice skills and procedures while supported in context. (See for example, Collins, Brown, and Newman (1989), Brown, Collins, and Duguid (1989)).

Self (1990) too, has argued for a change in role of student models in ITS without going as far as suggesting that they are abandoned altogether.

The general perception ... that student models in ITS are for remediation presents a serious philosophical problem. It is the arrogant, ‘tutor knows best’ style of ITSs which alienates classroom teachers more than any technical shortcomings. The standard ITS approach of first defining a body of certified knowledge and then devising ways to correct students’ understanding so that it conforms to it does not accord with the philosophies of epistemologists, with or without an educational orientation. (p.13)

Self was not opposed to ITS or indeed to student modelling. Rather, he claimed that the student model should reflect student beliefs and therefore,

The role of student models would then be to help ITSs to provoke students to consider the justifications and implications of their beliefs. (p.16)

As Reeves and Hedberg (2003) point out, ‘even the staunchest proponents ... of ITS must acknowledge the lack of impact these computer-as-tutor applications have had on mainstream education and training’ (p.6). Nevertheless, generative models are still used as the benchmark for student models in ITS today (Woolf, 2008). The main alternative, constraint-based student modelling, is arguably a more relaxed approach but it still retains on a focus on the detection and correction of student errors (Ohlsson, 1994). The newer example-tracing tutors (Aleven *et al.*, 2009) appear to represent an easier authoring option but the focus is still step-based error correction.

Perhaps because of these doubts which were raised during the 1980s and early 1990s and perhaps because ITS seldom find utility in educational practice, it is hard to find much reference to ITS in mainstream educational literature. However, if the focus is shifted to educational feedback, of the kind that human teachers and tutors provide, then searching the educational literature provides a good deal of information which is relevant to the design of ITS.

### **2.2.7 The role of educational feedback**

Since the 1920s the positive benefits on student performance of practice questions (that is, formative assessment) have been demonstrated in classroom studies (Frederiksen, 1984). Similar positive effects have been demonstrated in psychology laboratory studies since the 1970s (McDaniel, Anderson, Derbish, and Morrisette, 2007). Large meta-analytic educational studies looking at the impact of practice tests on student outcomes indicate that on average, the provision of practice assessments during a course of study does confer a clear advantage (Bangert-Drowns, Kulik, Kulik, and Morgan, 1991), although the effect of increasing practice-test frequency is less clear (Crooks, 1988). In addition, a more recent meta-analytic educational study which sought to identify the key mediators of positive learning outcomes, found that feedback from student to teacher and feedback from teacher to student, are among the top-ranked factors (Hattie, 2009). Furthermore, studies which examine approaches to feedback generally agree that individualised feedback which is not graded, avoids personal comment (including praise) and which focuses on specific strategies for improvement, results in the largest gains (for example, Hattie and Timperley (2007), Lipnevich and Smith (2009), Shute (2008)).

Gipps (2005) has noted that both evaluative and descriptive feedback is vitally important to support student learning. She goes on to suggest:

If feedback from assessment could be automated, while maintaining quality in assessment, it could certainly be a powerful learning tool. (p.175)

However, she points out that use of computer-marking for anything other than MCQ-style questions, while showing some promise, is seldom used in higher education institutions.

There is some literature which specifically explores whether there is any difference in learning outcome depending on the mode of assessment, specifically MCQ versus free-text or short-answer responses. However the number of studies is limited and the results are inconclusive. Gay (1980) found that in retention tests students who practiced answering short-answer (SA) or free-text questions performed as well as or better than students who practiced MCQs but this effect was also dependent on the mode of retention testing. Specifically, retention test results where the test was conducted using SA were better for both MCQ-practice and SA-practice, whereas there was no difference between the two practice groups where the retention test mode was MCQ. In 1984, reviewing a series of contemporary studies which investigated the influence of test format on student performance, Frederiksen (1984) concluded that

testing increases retention of the material tested and [...] the effects are quite specific to what was tested. There is some evidence that short-answer or completion tests may be more conducive to long-term retention. (p.196)

In a related area in his comprehensive review of classroom evaluation practice, Crooks (1988) suggested

there is no strong evidence...to support widespread adoption of any one [question] item format or style of task. Instead, the basis for selecting item formats should be their suitability for testing the skills and content that are to be evaluated. (p.448)

Support for this view is found in a meta-analysis of 67 empirical studies which investigated the construct equivalence of MCQ and constructed-response (SA) questions (Rodriguez, 2003). Where the content or stem of the MCQ and short-answer questions were the same, Rodriguez found a very high correlation between the different formats. In other words, where the questions relate to the same content, they will measure the

same trait in the student's performance. However, even if the same traits are measured by performance on questions in different formats, this says nothing about whether using practice questions in different formats results in differential learning gains for the students on subsequent retention tests.

Given the large body of evidence for the beneficial effects of formative feedback, given the practical and epistemological issues associated with current rationalist inspired ITS, and finally, given the desire by educators for individualised, intensive and relevant learning environments, it is worth revisiting the 'empiricist design' for a machine that can teach and which was sketched above in Section 2.2.3. The machine should take linguistic input from the tutee and look up the most appropriate linguistic response based on stored empirical evidence about how to respond. No calculations or assumptions are made about either mental or machine state. If a particular linguistic pattern or feature-set has been seen before, then the machine responds on the basis of a known response to that pattern.

## 2.3 The focus of this study

The motivation for the current study was to build and evaluate the kind of machine described above. There were two specific goals and these are outlined below in Sections 2.3.1 and 2.3.2.

### 2.3.1 The first goal: a pragmatic tutorial dialogue system

As discussed in Section 2.2.6, intelligent tutoring systems and in particular tutorial dialogue systems, rarely find utility in real class settings at present. The primary goal of the current research therefore was to create a tutorial dialogue system that will be both responsive and practical in a real class setting; a **pragmatic** tutorial dialogue system. Student responses and teacher feedback should inform system development from the outset and a mechanism for incorporating ongoing student feedback (that is, student responses to the questions posed) should be designed into the system itself. There should be no explicit student model other than what is required to avoid circular questioning and repeated turns, no formal logical or semantic representation of expert knowledge and no explicit instructional model. These features will be **compiled** into a script which is created by the tutorial author: the goal is to support the teacher to teach, rather than try to create explicit representations of domain or student knowledge.

The machine thus described will be a surface-based dialogue system which functions as a tutor. **Surface-based** means that the natural language understanding (NLU) component of the system relies on empirical or statistical NLP techniques rather than deep semantic NLP techniques (these are explained in Section 3.2.3.3). This choice, in addition to sitting well with the empiricist philosophical position outlined in this chapter is also a pragmatic one; statistical NLP techniques are increasingly finding utility in practical applications where traditional NLP methods fail and are relatively straightforward to incorporate into new applications using standard NLP libraries (Manning and Schütze, 1999).

It is important to note that all surface-based ITS are to some extent harking back to the empiricist ideal, the goal here is to develop a pure version of an empiricist system, one that has no explicit student modelling as well as no explicit modelling of the domain of instruction. The new tutor abandons the idea of explicit student models, pre-ordained teaching models and any formal or logical representation of the knowledge domain. But, it does retain the idea of unrestricted free-text input from the student. The family of dialogue systems or conversational agents closest to it are those inspired by Weizenbaum's 'psychotherapist', ELIZA (Weizenbaum, 1966). These dialogue systems or conversational agents, which are not necessarily designed for tutoring, take typed natural language input, attempt to classify the input based on either regular expression matching or surface-based NLP techniques and generate typed output from a pre-defined script. By employing the well understood and longstanding educational practice of engaging in dialogue the system should support teachers to teach and ask no more of them, or their students, than is already asked.

Ideally, given the difficulty and expense of authoring, or customising ITS for specific contexts, a generic structure for creating tutorial dialogues should be designed into the new system in order that it can be readily extended or customised in the future.

### **2.3.2 The second goal: evaluating the new tutorial dialogue system against a similar menu-based alternative**

The absence of conclusive evidence from existing literature that free-text entry of responses to questions confers a benefit over the selection of options from a menu is the basis for the second goal for this study. Here is an opportunity to test explicitly for such a benefit with a dialogue system that can offer either or both input options as part of a system evaluation.

*Geometry Tutor* (Aleven, Ogan, Popescu, Torrey, and Koedinger, 2004) and *Algebra Tutor* (Corbett, Wagner, Lesgold, Ulrich, and Stevens, 2006) are two similar studies. The results from both suggest that there is little difference between the two formats especially on immediate post-test but that the free-text option may yield some advantage for long-term retention and some benefit for performance in subsequent short-answer questions. These results are consistent with the much earlier educational review conducted by Frederiksen (1984). It is important to note that both these studies were conducted with relatively small numbers of students in a laboratory setting. In this project the educational benefits of two versions of the new tutorial dialogue system, free-text and menu-based, are compared in a much larger cohort of students.

Related studies include those which evaluate automated short-answer question marking systems (that is, systems which are not dialogue-based). These are only recently coming into use in classrooms and some, such as Educational Testing Services', c-Rater (Leacock and Chodorow, 2003) were originally developed using deep semantic NLP techniques while others employ surface-based statistical NLP techniques. (See for example, Butcher and Jordan (2010)). The reported results of these studies look promising (accuracy of 85% - 100% compared to human markers) and this work provides a useful benchmark against which to compare the performance of the new system which is implemented in this study.

## 2.4 Summary

This chapter has presented a brief history of ITS from the early 1920s to the present day. The empiricist and rationalists positions in AI have been discussed and a rationale for revisiting the current rationalist-inspired approach to building tutorial dialogue systems has been presented, namely: the practical and epistemological issues associated with current rationalist inspired ITS, the large body of evidence for the beneficial effects of formative feedback, and the desire by educators for individualised, intensive and relevant learning environments. The first goal of this study therefore was to build a new surface-based tutorial dialogue system which aims to find utility in educational practice. The lack of conclusive evidence from existing literature that free-text entry of responses to questions confers a benefit over the selection of options from a menu is the basis for the second goal for this study which was to evaluate the new tutorial dialogue system against a similar menu-based alternative. Chapter 4 describes in detail the approach to designing and implementing the new system. The next chapter, Chapter

3, provides the background for human dialogue modelling, natural language processing (NLP) and machine learning techniques which are required in order to explain the new tutor implementation.

# Chapter 3

## Topics in dialogue modelling and text classification

### 3.1 Introduction

This chapter serves a dual purpose. The first is to provide sufficient background and context in dialogue modelling and text classification in order that subsequent chapters which refer to ideas, methods or techniques from these fields make sense for the non-specialist. The second, is to review the literature, where relevant, from these fields along the way.

### 3.2 Models of Human Dialogue

#### 3.2.1 A brief introduction to discourse and dialogue

Having a conversation is the most natural thing in the world for most people. We spend much of our time doing it and mostly we do it quite naturally, with little effort. Far less natural or intuitive is attempting to understand *how* we have conversations. Yet, developing a model, or set of rules and procedures, which governs how humans have conversations is crucial in order to create a machine that can function as a dialogue partner. To begin, some basic linguistic concepts and ideas relevant to discourse and dialogue are introduced. A number of illustrative examples are drawn from *Winnie-the-Pooh* and *The House at Pooh Corner* (Milne, 1928).

When someone greets us, how do we know how and when to respond and also when to stop?

‘Hallo!’ said Pooh, in case there was anything outside.  
 ‘Hallo!’ said Whatever-it-was  
 ‘Oh!’ said Pooh. ‘Hallo!’  
 ‘Hallo!’  
 ‘Oh, *there* you are!’ said Pooh. ‘Hallo!’  
 ‘Hallo!’ said the Strange Animal, wondering how long this was going on.

Simply repeating a greeting and not knowing when to stop is not something that normally happens. Knowing what to say next, knowing who should say it and knowing how to respond to what has been said are all things we take for granted. And, the excerpt from *Pooh* above is from a dialogue which involves only a rudimentary, predominantly one-word, conversation between two strangers. What of more complicated dialogues?

Before delving more deeply into dialogue, it is helpful to describe the broader term, **discourse**. The word discourse, in the context of computational linguistics, describes passages of text, written or spoken, which have some coherence or connection. More precisely and following Jurafsky and Martin (2009), discourse is a coherent, structured group of sentences. There are two key aspects of coherence in a text: firstly, **coherence relations**, which are specific connections of meaning between sentences in a text and secondly, **entity-based coherence** which refers to how entities in a discourse are introduced and elaborated on.

There are a number of theories of coherence relations (Jurafsky and Martin, 2009). Most attempt to code the relationships of meaning between spans of text (for example as evidence, elaboration, contrast and so on) and then show how these can be hierarchically organised (See for example, Hobbs (1979) and **Rhetorical Structure Theory (RST)**, Mann and Thompson (1988)). The key point for the purpose of this discussion is that a text or dialogue must exhibit coherence in order to make sense. Humans pretty quickly recognise when discourse coherence fails or starts to sound odd. *Pooh* illustrates what happens when one conversational partner becomes incoherent and demonstrates a potential strategy for repair:

‘Just what I feel,’ said Rabbit. ‘What do you say, Pooh?’  
 Pooh opened his eyes with a jerk and said, ‘Extremely.’  
 ‘Extremely what?’ asked Rabbit.  
 ‘What you were saying.’ said Pooh. ‘Undoubtably.’

When caught napping by *Rabbit*, *Pooh*, as an astute and linguistically capable bear, effects a repair by passing the conversational initiative back to his partner with, ‘What you were saying ...’, employing a specific feature of dialogue, namely that it is jointly constructed. *Pooh* relies on *Rabbit* to maintain coherence of the dialogue. The joint construction of dialogue is discussed in more detail in Section 3.2.2.

**Entity-based coherence** involves knowing who or what is being referred to. Entities have to be introduced into a discourse and referred to in such a way that it is clear who or what they are. Consider the following:

‘Well,’ said Rabbit ... ‘I promised Christopher Robin I’d organise a search for him, so come on.’ ... Small’s real name was Very Small Beetle ... He had been staying with Christopher Robin for a few seconds, and he had started round a gorse-bush for exercise ...

The two occurrences of the pronoun *he* in the last sentence of this extract (*He had been staying*, *he had started*) refer to Small, who is the subject of the story (‘The search for Small’). For humans reading this text, it is not too hard to discern that these pronouns refer to Small rather than to Christopher Robin (another possible referent for a third person singular masculine pronoun). But why? The first occurrence of *he* can be resolved on the basis of syntactic constraints alone. It is the subject of a sentence whose object is the full noun phrase *Christopher Robin*; if its antecedent had been Christopher Robin, the object noun phrase would have been realised as a reflexive (*He had been staying with himself*). The second *he* is harder to resolve, since there are no relevant syntactic constraints, and there is no good reason, in terms of meaning, why Christopher Robin might not have started round a gorse-bush for exercise. In this case other factors determine that Small is the antecedent: for instance, the fact that Small is referred to more recently than Christopher Robin, and that Small is the current topic.

A detailed exposition on the general topic of **reference resolution** is beyond the scope of this introductory review. Broadly, reference resolution involves working out to whom pronouns refer (**anaphora resolution**) and when noun phrases are referring to the same thing (**coreference resolution**). The key point, for the purpose of this discussion, is that where referring expressions are used in the course of a dialogue, each dialogue partner must be able to work out to what or whom these expressions refer if the dialogue is to be coherent.

These broad features of discourse, relational coherence and entity-based coherence, are also features of dialogue. Section 3.2.2 explores the specific features of dialogue in

more detail.

### 3.2.2 Contemporary models of dialogue

Dialogue is a special form of discourse which is jointly constructed by two, or sometimes more, participants (Jurafsky and Martin, 2009). In this discussion only the case of two interlocutors is considered. A feature of dialogue is that each of the participants take turns to speak and to listen but what are the ‘rules’ which govern turn-taking, what meaning does each speaker make of the dialogue and how is the dialogue structured?

Early models of conversational dialogue proposed structures (such as greeting-greeting and question-answer) and rules for taking turns in conversation (for example, Schegloff (1968); Sacks, Schegloff, and Jefferson (1974)). But, for a computer-based dialogue system these general ideas need to be more precise. Moving beyond the intuitive notion of rules which govern dialogue, is the idea from Wittgenstein (1968) of language as action.

[There are] countless different kinds of use of what we call ‘symbols’, ‘words’, ‘sentences’. And this multiplicity is not something fixed, given once for all; but new types of language, new language-games, as we may say, come into existence ... Here the term ‘*language-game*’ is meant to bring into prominence the fact that the *speaking* of language is part of an activity ... (p.10, Section 23)

Austin (1975) and Searle (1975) further developed this idea into the concept of a **speech act** or an action which is performed by a speaker, and Searle (1976) proposed several categories of speech act (for example, **directives**: to ask or order someone to do something, **commissives**: the speaker promises or plans to do something, **expressives**: to express gratitude, welcome, apologise ...). An extension of Searle and Austin’s concept of a speech act is the notion of a **dialogue act**. Dialogue acts can be domain specific or more general but include features specific to dialogue, such as grounding and turn-taking. Core and Allen (1997) defined two important functions for dialogue acts, **forward-looking** functions (for example, asking a question, issuing a directive etc.) and **backward-looking** functions (for example, grounding, answering etc.) which relate to the previous utterance of the interlocutor.

If dialogue is a joint activity then this suggests that the speech acts of dialogue partners are related in some way. Interlocutors must continuously establish **common ground** if they are to jointly construct a conversation (Stalnaker, 1978). Clark and

Schaefer (1989) introduced the notion of **grounding** where, by turns, the speaker and hearer make and accept utterances. Continued attention, paraphrasing, acknowledgement of an utterance, and repeating a conversational partner's utterance are all methods for establishing grounding (Clark and Schaefer, 1989).

A remaining and important problem concerns the meanings or content of the dialogue. Meaning in conversation is not necessarily apparent from the composition of dialogue utterances. Grice (1978) proposed his theory of **conversational implicature** which sets out some general rules or **maxims**, by which interlocutors make inferences about meaning.

‘It’s snowing still,’ said Eeyore gloomily.

‘So it is.’

‘And freezing.’

‘Is it?’

‘Yes,’ said Eeyore. ‘However,’ he said, brightening up a little, ‘we haven’t had an earthquake lately.’

‘What’s the matter Eeyore?’

In this situation the inference the hearer (in this case *Christopher Robin*) makes is that Eeyore’s gloomy utterances are relevant to *Eeyore’s* perspective on the world and thus follow Grice’s maxim of relevance. *Christopher Robin*, inferring that the state of *Eeyore’s* world is less than satisfactory enquires, ‘What’s the matter Eeyore’. However, taken out of a conversational context, this particular text composition would make little or no sense at all.

The structure of dialogue as a whole is a large and active research field and again, beyond the scope of this review. Nevertheless, it is worth noting the influence of planning-based models to interpreting utterances in a dialogue (see for example Cohen and Perrault, 1979; Perrault and Allen, 1980), in particular Grosz and Sidner (1986) where it is proposed that the plan-based intentions of the interlocutors make a dialogue coherent. In a very general sense and in the context of a tutorial dialogue system, the intentions of the tutor and tutee and the focus of their attention in relation to each utterance are surely pivotal.

In order to process dialogue by machine, to either understand it or to generate it, the dialogue system used must incorporate at least some, if not all, of the ideas of turn-taking, speech acts, grounding and conversational implicature. Furthermore, dialogue systems need to provide a coherent structure for the dialogue as a whole, either

computationally or, as described fully in Chapter 4, through pre-scripting. Section 3.2.3 provides a brief review of contemporary dialogue systems.

### 3.2.3 Human-machine conversational dialogue systems

#### 3.2.3.1 Dialogue system architecture

Jurafsky and Martin (2009) sketch out a simplified architecture for typical dialogue systems the components of which include speech recognition, natural language understanding (NLU), dialogue manager (DM), natural language generation and text to speech synthesis modules. In the context of text-based dialogue systems, where users interact with the system by typing their input and receive system output as written text, this architecture can be simplified further, as shown in Figure 3.1. The speech recognition module, for converting speech to text, and the text to speech synthesis module are not required.

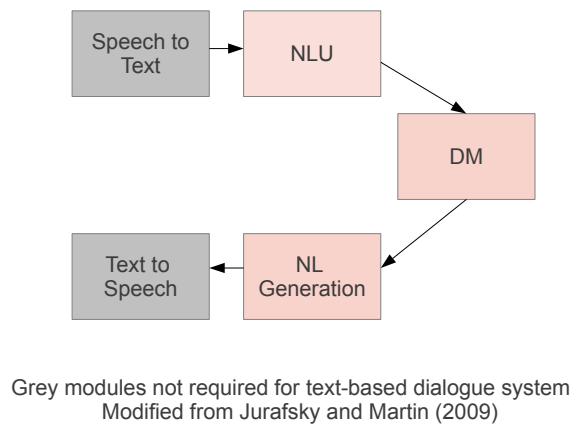


Figure 3.1: Dialogue System architecture. (Modified from Jurafsky and Martin (2009, p.821))

It is important to note that not every feature of human dialogue will be modelled in every human-machine dialogue system. Commercial dialogue systems, or conversa-

tional agents, frequently employ simplified models which take advantage of the specific heuristics associated with particular kinds of dialogue. For example, a flight booking conversational agent will typically ignore input text which it is not specifically looking for and focus on that which it is, such as flight origin, destination, time, date and so on. The fact that a person is using the system for booking a flight will ensure that their intention, to book a flight, matches that of the system, which is to make a booking. The dialogue system can therefore safely ignore the rules of conversation around buying a pair of shoes, or selecting a suitable school for a child, since these conversations in the context of booking a flight would not make any sense at all.

Contemporary models for managing dialogue are described next in Section 3.2.3.2 and natural language understanding models are discussed in Section 3.2.3.3. The new dialogue system presented in this thesis makes minimal use of natural language generation, so this topic is not covered in the current review.

### 3.2.3.2 Dialogue management models

Jurafsky and Martin (2009) suggest that there are currently four common types of dialogue manager design. At the simplest end of the spectrum are finite state machines (for example, *Why2-Atlas* (VanLehn *et al.*, 2002)) and frame-based machines (for example *Scholar* (Carbonell, 1970), *GUS* (Bobrow, Kaplan, Kay, Norman, Thompson, and Winograd, 1977)), then more complex information-state architectures (Traum and Larsson, 2003; Allen, Ferguson, and Stent, 2001) which explicitly model the information states of dialogue participants, and how these are updated by each dialogue act, and finally plan-based systems (Cohen and Perrault, 1979; Perrault and Allen, 1980; Grosz and Sidner, 1986).

Nevertheless, as Traum and Larsson (2003) point out there are many variations on the approach to dialogue management and no universal agreement about what dialogue management is. For example, in a recent review of computational models of dialogue Ginzburg and Fernández (2010) propose three main types of DM architectures: finite-state, frame-based and inferential (which roughly corresponds to systems based on speech act theory and planning). The information-state update architecture (Traum and Larsson, 2003) is viewed more broadly as a framework within which existing models of dialogue might be implemented (Ginzburg and Fernández, 2010).

The finer points of dialogue manager classification notwithstanding, there is broad agreement about what constitutes a finite state system and what constitutes a frame-based system (Jurafsky and Martin, 2009; Ginzburg and Fernández, 2010).

Finite state machines represent the simplest dialogue manager architecture. They are defined by a finite state automaton (FSA): a graph whose nodes correspond to questions posed by the system and whose arcs represent a finite number of possible actions to take depending on user responses. An example is shown in Figure 3.2. The dialogue manager completely controls the interaction with the user and these kinds of systems are generally referred to as **single-initiative** systems. An important limitation of the finite-state approach, as noted by Jurafsky and Martin (2009) is that

[while] it is theoretically possible to create a finite-state architecture that has a separate state for each possible subset of questions that the user's statement could be answering ... this would require a vast explosion in the number of states, making such an architecture difficult to conceptualise.

It is not hard to see why the number of states might explode. In the example FSA shown in Figure 3.2, the number of possible states which correspond to answers to the question, 'Who are you going to have lunch with Pooh?' is arbitrarily restricted to just four. There are many more possible responses Pooh might make (Rabbit, Owl, Christopher Robin and so on ...) and indeed Pooh might make some other utterance altogether, with the intention of returning to the question later. This particular FSA would fail to deal with any other possibility. The same problem occurs with possible responses to each of the subsequent states.

Frame-based systems typically involve filling slots in requisite information frames as required information comes to hand, keeping track of which frames are filled and then taking appropriate action to either move on to next frame or seek further action. Frame-based systems in general provide a little more flexibility in terms of user action than finite state systems and may offer more opportunity for **mixed-initiative** depending on the implementation. For example, the goal of a flight-booking dialogue system is to answer various questions: where do you want to go, which date, do you have luggage, and so on. In a frame-based dialogue system each of these questions corresponds to a slot in the dialogue frame and each user utterance might potentially provide the filler to any possible frame. Thus the user may provide information that the system requires but not necessarily in a prescribed order. A frame-based DM could recognise that information provided might relate to a different frame (the user has taken the initiative), could swap the current frame, fill a slot, then swap back to the original frame and resume the initiative.

Plan-based or inferential architectures are beyond the scope of this review but it

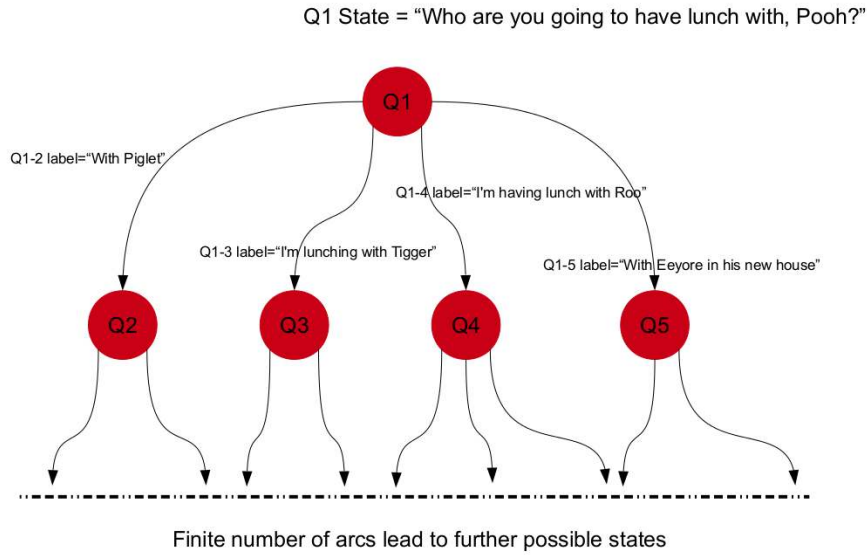


Figure 3.2: Example FSA

is worth summarising the Information State Update (ISU) framework (Traum and Larsson, 2003). ISU specifies the following components:

- A formal representation of the discourse context or information state
- A set of dialogue acts or moves which update the information state
- A set of rules which determine how the information state is updated
- A control strategy for deciding which rules rules to apply and when

The ISU approach, as noted by Ginzburg and Fernández (2010), can be viewed as a framework within which to implement any one of a number of dialogue models, including FSA and frame-based models.

For reasons which are discussed in Chapter 4, the approach to dialogue management taken in this research is a finite-state approach within an ISU framework and this is described in detail in Section 4.4.3. It will be seen that a key difference from more usual information state architectures is a minimalist interpretation of the information state: in this research the information state is simply a representation of the number and type of visits to each dialogue context.

The requirements for natural language generation are minimal in the new system; this is handled by a hand-crafted **tutorial dialogue script** which is processed by the dialogue manager. Script design and development is described in detail in Section 4.4.3. The one outstanding issue which is relevant to this research, is the topic of natural language understanding (NLU) and this is discussed in the next section, 3.2.3.3.

### 3.2.3.3 Natural language understanding

The task of the NLU module in any dialogue system is to generate a semantic representation of the input text (see for example, Jurafsky and Martin (2009)). There are two broad approaches for achieving this. The first, sometimes called a **deep** approach, utilises a **grammar** and involves **parsing** the input text in order to represent its meaning. This is usually done in combination with mapping to meaning through for example, slot-filling in frames, a semantic network or predicate logic methods. The second, sometimes called a **surface** approach involves either matching the input text to a specified pattern, using **regular expressions** for example, or using statistical methods to generate some approximation of the likely meaning of a text based on previously seen training examples. Sometimes a combination of the two approaches is used; for example, where the meaning of a text is ambiguous and generates more than one plausible parse, statistical approaches might be used to determine the most likely parse and therefore the most likely representation of meaning.

Current human-machine tutorial dialogue systems span the range, for their NLU component, from systems which utilise deep semantic processing such as *SCoT* (Pon-Barry, Clark, Schultz, Bratt, and Peters, 2004) to more surface-based systems such as *Auto-Tutor* (Graesser *et al.*, 2001) which relies on statistical NLP methods and *TuTalk* (Jordan, 2007) which utilises simple pattern-matching, with *CarmelTC* (Rosé, Roque, Bhembe, and Vanlehn, 2003) somewhere in between.

Whichever approach or combination of approaches is adopted, some form of pre-processing or **normalising** of the input text will be required in order to tokenise the text (identify individual sentences and words), correct for spelling mistakes and typographical errors, deal with abbreviations, formulae and so on. A preprocessor module is almost always a feature of the larger NLU component and normalising text is a fundamental task for most natural language processing (NLP) tasks (see for example, Jurafsky and Martin (2009)). The design of the preprocessor module used in this research is described in Chapter 4.

One feature of language in general, that is especially important for natural language

understanding, is ambiguity. Ambiguity can occur at the level of discourse in terms of both coherence relations (for example, a text with more than one possible relational meaning) and entity-based coherence (there is potential ambiguity around who ‘he’ refers to in *The Search for Small* example in Section 3.2.1). But there are many other opportunities for ambiguity. Ambiguity can occur at the level of phonology as the following interchange between *Piglet* and *Eeyore* illustrates:

‘Do you know what this is?’  
‘No,’ said Piglet.  
‘It’s an A.’  
‘Oh,’ said Piglet.  
‘Not O - A,’ said Eeyore severely.

Ambiguity can occur at word level. For example, to adapt a famous example, ‘Piglet *flies* like a rabbit’ could mean either, that Piglet is flying like a rabbit, or that flies that hover round piglets like rabbits. And, ambiguity can also occur at sentence level. For example, ‘Pooh put a paw in the cake that was thinly sliced’ creates ambiguity around whether it is the cake or the paw which is thinly sliced. Surface-based approaches to natural language understanding are particularly useful for resolving ambiguity even in the face of a wide range of textual inputs (Manning and Schütze, 1999).

As described in Chapter 2, Section 2.3.1 the approach taken in this research is to develop a surface-based dialogue system which utilises empirical methods to determine the most likely meaning of input text. The family of statistical NLP techniques which can be employed to do this is known as text classification and this is described in detail in Section 3.3.

### 3.3 Text classification

Categorising or **classifying** text is the process of assigning a passage of text or an entire document to a particular theme or topic (Manning and Schütze, 1999). It can have several purposes. If the text is an entire document, the goal of classification might be to identify what type of document it is; for example, is a given email spam or not? For a paragraph, a single sentence, or perhaps even a sentence fragment, the goal might be to identify what topic the text is on. Or it might be to identify a proposition in the text; for instance, a specific class of answer to a question. The statistical classification of text is just one among many applications of **supervised**

**machine learning** which include diverse applications such as image recognition and information retrieval. Following Manning and Schütze (1999), the approach to the supervised classification of text can be broadly summarised as follows: a **training set** of texts is labelled with one or more classes to which they belong. Each text in the training set can then be represented by the **data representation model**,  $(\vec{x}, c)$ , where  $\vec{x}$  is a feature vector (a simple feature vector could be word counts in the text) and  $c$  is the class label. A family of parameterised classifiers (i.e. classifiers where the same set of features is used to characterise the model) is defined and a training procedure picks the classifier from this family which optimises classifier performance. The classifier is then evaluated against a previously unseen **test set** of texts.

There are many families of classifiers and combinations of classifiers and training procedures (for example, Linear, Naive Bayes, Kth-Nearest Neighbour, Decision Trees, Perceptrons, Support Vector Machines and others ...) each with different characteristics and properties which make them suitable for different contexts and classification tasks. Most, but not all, roughly follow the idea of seeking to find the parameters of the classifier function through running training algorithms (for example, logistic regression, gradient descent, generalised iterative scaling ...) on a **training dataset** in order to maximise classifier accuracy on an unseen or heldout **test dataset** (for a discussion of the range of classifiers and training procedures see Manning and Schütze (1999)).

In this research, classifiers are treated largely as a ‘black box’. In other words, the detail of implementation is set aside and the focus is on classifier performance. The choice of classifier for the new dialogue system is discussed in detail in Chapter 4.

The following section, 3.3.1, provides a very brief introduction to how machine learning classifiers work. This is followed by a discussion of classifier evaluation metrics and feature selection (data representation models).

### 3.3.1 A general introduction to how classifiers work

A classifier can be thought of as implementing a function  $f$ , whose inputs are a set of features  $x_0 \dots x_i$  and whose output is a class. To make the idea of training classifiers concrete, consider the family of binary linear classifiers represented by the equation:

$$f(\vec{x}) = \vec{w} \cdot \vec{x} + w_0 = \sum_j w_j x_j + w_0 \quad (3.1)$$

The goal of **training** this binary classifier is to choose the parameters  $\vec{w}$  and  $w_0$  so that when a previously unseen feature vector,  $\vec{x}$ , is given as input to the classifier it is

either assigned a class label, or it is not. When  $f(\vec{x})$  exceeds (or falls below) some real number threshold value this determines whether the class label ( $c$ ) is assigned. For example:

$$f(\vec{x}) > 0, c = \text{Label} \quad (3.2)$$

The issue here is the *correct* assignment of the class label to a previously unseen feature vector. Obviously, in an ideal situation a classifier would always correctly assign class labels. In practice this is seldom the case and there is always the possibility that a feature vector is incorrectly labelled. For this reason, evaluation of classifier performance is a key part of classifier training and selection and is introduced in the next section, 3.3.2.

In order to train a classifier, a training set of labelled feature vectors is required. The performance of the classifier is then evaluated against a test set of previously unseen labelled feature vectors. The specific process used in this research to create labelled training and test sets is described in detail in Chapter 4. A further consideration in developing training sets is the issue of inter-rater reliability. Ideally, where training data relies on human judgements, inter-rater reliability measures should be collected in order to estimate the reliability of the training set; in other words, to determine whether two or more human markers would assign the same label to a particular text. Inter-rater reliability is discussed further in the context of classifier case studies in section 5.4.1.3.

### 3.3.2 Classifier performance metrics

Binary classifiers (that is, classifiers used to make a choice between only 2 categories) are evaluated on the basis of three metrics: accuracy, precision and recall. For evaluating multilabel classifiers (that is, more than 2 possible categories), an averaging procedure (Manning and Schütze, 1999) or MASI distance (Passonneau, 2006) can be used. The metric used to evaluate multilabel classifiers in this research is MASI distance and this is described in detail in Chapter 5.

Before defining accuracy, precision and recall in the context of text classification, it is helpful to review the more general statistical concepts of **Type-I** and **Type-II** errors which are used in formal hypothesis testing. Using the example of a binary classification task, and the label notation from the example in Section 3.3.1, a Type-I error is where a feature vector which is *not* a member of  $c_1$  is incorrectly assigned

the label  $c_1$ . This is also known as a **false positive (FP)**. Conversely, a Type-II error, or **false negative (FN)** is where a feature vector which is a member of  $c_1$  is incorrectly assigned the label  $\text{Not}(c_1)$ . It follows from this description that the terms **true positive (TP)** and **true negative (TN)** refer to the correct identification of the feature vector as a member (or not) of  $c_1$ .

Intuitively, accuracy is the proportion of correctly labelled items in a test set. More formally, and in terms of the error categories described above accuracy is defined as:

$$TP + TN / (TP + TN + FP + FN) \quad (3.3)$$

**Recall**, which in more general terms is a description of how **sensitive** the classifier is, is given by:

$$TP / TP + FN \quad (3.4)$$

Conversely, precision, or how **specific** the classifier is, is given by:

$$TP / TP + FP \quad (3.5)$$

In practice, although it is possible and certainly ideal to achieve 100% precision and recall, there can be a trade-off between the two metrics. So for example, if all the items in a test set are labelled  $c_1$ , and assuming they are not all correctly labelled then the classifier is very sensitive and recall will be 100%. However, precision will be low since there will be a number of false positives. Equally, if only a few of the items in the test set which are labelled  $c_1$  are correctly identified then the number of false negatives increases and recall will be low. Striking the balance between precision and recall is discussed in more detail in Chapter 5 where the evaluation of several case-study dialogue contexts is reviewed.

In the example given so far, the issue of how to select and encode features from text has been glossed over other than to suggest that a feature vector for a specific text might be represented by something like words counts. Feature selection is a central issue for text classification irrespective of the classifier chosen (for example, see Bird, Klein, and Loper, 2009; Manning and Schütze, 1999; Jurafsky and Martin, 2009). In the next Section, 3.3.3, some relevant issues around feature selection are introduced.

### 3.3.3 Feature selection

The simplest and perhaps most intuitive type of data representation model for a passage of text is what is commonly referred to as a **bag of words (BOW)** model. In this model, each word in the text is a feature and the bag of words is literally a set of words where repeated elements are allowed. For example, the sentences, ‘The goat is feeding on my jacket’ and ‘The goat on my jacket is feeding’ can both be represented by the set of words,  $\{goat, jacket, feeding, the, is, on, my\}$ . It is clear from this, that a set representation removes all semantic structure and context from the original sentence. That is, any meaning in the set, is represented only by the meaning of the individual words. Nevertheless, for some contexts, even a simple bag of words approach to feature selection can work well (Manning and Schütze, 1999). In effect, when training a classifier on a BOW feature set, this is in effect a process for identifying specific keywords which are associated with class labels.

The BOW approach, while often a useful point of departure when thinking about which features to select, has limitations. The loss of structure mentioned above is one, the order of words may matter, the impact of negation may be lost (for example, ‘the goat on my jacket is not feeding’) and so on. Knowledge of the domain and context for the text classification task can be helpful in order to determine appropriate features for encoding. For example, rather than encoding all words as features, another approach is to select the words most likely to be informative in a given context (for example, see Manning and Schütze, 1999; Perkins, 2010). This approach is discussed in detail in Chapter 5. Other simple feature selection strategies include selecting first or last words as features, using bigrams (consecutive sequences of two words) or n-grams (word sequences of some other arbitrary length) and in some situations, even looking at the length of text. There are a range of strategies which can be used to create features sets from text and most NLP practitioners seem to agree that the business of feature selection is guided as much by art and intuition as it is by science (for example, see Bird *et al.*, 2009; Manning and Schütze, 1999).

In the end, the choice of features appropriately encoded (a vector of numerical, boolean or text values depending on the classifier family and training strategy used) in combination with the choice of classifier and training strategy will ultimately determine the performance of the specific text classifier which is created. The process used in this research to identify and select features, in combination with evaluation of classifier performance, is detailed in Chapter 4.

## 3.4 Summary

This chapter has introduced ideas from dialogue modelling and text classification which are directly relevant to this research. From the overall architecture of the new tutorial dialogue system to the specific methods used to recognise student input, the system implementation relies on established techniques from the fields of computational linguistics and machine learning. The specific implementation for the new system is the subject of the next chapter.

# Chapter 4

## A classroom-ready tutor: design and implementation

### 4.1 Introduction

This chapter introduces the new tutorial dialogue system. Section 4.2 describes the design goals for the new system and Section 4.3 briefly describes the early stages of the project. Section 4.4 details the specification of the new tutorial dialogue system. Finally, Section 4.5 is devoted to the approach to text classification which is used in the new system and which builds on some of the concepts introduced in Chapter 3.

### 4.2 Design goals

Three design goals for a new tutoring system emerged from the literature review given in Chapter 2.

- Issue 1. Intelligent Tutoring Systems, in particular, tutorial dialogue systems, rarely find utility in real class settings. The first and most important design goal therefore was to create a tutorial dialogue system that would be both responsive and practical in real class settings. It was important to ensure that student and teacher feedback informed system development from the outset and that a mechanism for incorporating ongoing feedback (that is, user perceptions and user responses to the questions posed) was designed into the system itself.
- Issue 2. No conclusive evidence has been found that free-text entry of responses to questions confers a benefit over selection of options from a menu. The second

goal was to create a system to explicitly test whether free-text entry confers an advantage over selection from a menu for any given tutorial dialogue.

- Issue 3. Authoring, or customising ITS for specific contexts is typically time-consuming, expensive and difficult. The third design goal was to ensure that the system could be readily adapted to a wide range of domains. To this end, designing a generic structure for creating tutorial dialogues was pivotal.

While these design goals were reasonably clear from the outset, the detailed specification for implementing the system became clear only as the project progressed. The overall design chosen reflects the commonly understood architecture of a text-based conversational agent which was described in Section 3.2.3.1 including free-text entry from the student, a natural language understanding component, a dialogue manager, and a tutorial dialogue script which obviates the need for a task manager and a natural language generation component. Key features of the final specification are summarised as follows:

- The tutorial dialogue system employs a ‘text chat-style’ interface. This is largely for pragmatic reasons, namely to keep the system simple and because there is no requirement in terms of the research goals for multimedia interfaces.
- A large sample of real student responses is used. There are two reasons for this: the first is to build a model of student responses for the tutorial dialogue script so that it resonates with the target audience. The second is to utilise authentic student response data with surface-based, statistical natural language processing techniques in order to ‘understand’ new typed student input.
- To simplify authoring and promote use, an existing general model of human dialogue guides the design of the tutorial dialogue script and dialogue manager. To simplify matters, there is no natural language generation module or task manager; the script defines the tutorial lesson plan and pre-scripted responses are used to generate written output. Existing corpus formats are used for marking up the tutorial dialogues generated to simplify post-processing and further analysis.
- The dialogue manager uses a FSA architecture (introduced in Chapter 3) to process the tutorial dialogue script on the basis of user input. The tutorial dialogue script has two overarching components: first, a set of questions for the tutor to ask, to probe students’ knowledge of the domain and second, a good

model of the range of common answers which students are likely to provide, so that the tutor can give students immediate individualised feedback.

## 4.3 Choice of domain and approach to system development

The system development methodology involved two stages: a first stage, described in Section 4.3.2, in which a corpus of tutorial dialogue data in the selected domain was gathered, using a prototype dialogue system, and a second stage, described in 4.3.2, in which a new dialogue system was developed, using this corpus for training and benchmark testing. First, the reasons for the choice of dialogue domain are discussed in Section 4.3.1.

### 4.3.1 Domain choice

The first year health sciences course at the University of Otago is a prerequisite for entry into all the professional health science programmes, such as Medicine, Dentistry, Pharmacy, and Physiotherapy. Entry into these professional programmes is highly competitive and is dependent, amongst other things, on students achieving excellent grades in their 1st year courses. There are a number of required papers in the first year health sciences course and one of these, Human Body Systems (HUBS) 192 includes the study of the human cardiovascular system. The number of students enrolled in HUBS 192 varies, but ranges from around 1500 to 1800 students each year.

The domain selected for researching automated tutorial dialogue was first year undergraduate study of the human cardiovascular system, in particular, cardiovascular homeostasis. There were three reasons motivating this choice of domain. First, the domain was the same as at least one other natural language tutor, *Circsim Tutor* (Evens and Michael, 2006), although pitched at a more introductory level. This was helpful in that it provided some confidence that the domain was suitable for automated tutorial dialogue. Second, it was a domain familiar to the researcher and thus obviated the need to find an additional person to lead authoring of the tutorial questions and script. Third, there was interest from teaching staff in automated marking solutions for free text. Fourth, a very large cohort of highly motivated students were potentially available each year to work with the system during design, implementation and evaluation.

### 4.3.2 Data collection through fielding a prototype system

The first stage of the project involved producing a detailed set of questions to probe student understanding of key elements of the tutorial domain and evaluating these questions, in the form of a scripted dialogue, with students. The TuTalk dialogue engine from the Learning Research and Development Centre at the University of Pittsburgh (Jordan, 2007) was chosen to pilot the initial script primarily because it was, at the time, one of the few readily available domain-independent tutorial dialogue systems and provided a relatively easy way to author dialogues using only a text editor. It was also a good fit with the broad design goals. TuTalk scripts can be authored much like writing a natural one-to-one dialogue and the TuTalk scripting language represents each tutor turn as a finite state machine (FSM). The author can define multiple classes of student response that lead from the current tutor turn to the next tutor turn or state (Jordan, 2007). A simple chat-style interface is all that is required for a user to interact with the system.

TuTalk ‘out of the box’ included many features and options that were not required for this project and it became clear during initial testing that this added processing overhead to response times especially for longer typed responses. In addition, the basic natural language understanding module that came with TuTalk was fairly rudimentary and was based on a minimum-edit distance match to model responses. This yielded poor precision and recall when classifying most student responses. Nevertheless, with the addition of a web interface, a large cohort of students could access and use the system.

Questions for the initial cardiovascular homeostasis tutorial script were developed in close consultation with course teaching staff and were written using lecture notes, laboratory manuals and self-directed learning material from the course itself. A prototype tutorial system based on the script was released to students in the 2010 HUBS class (total enrolment=1800), for use on a voluntary basis at the beginning of their module on the human cardiovascular system. Tutorial use was optional. 437 students accessed the system during the course and produced a total of 532 dialogues; several students accessed the dialogue more than once. However, from the total number of dialogues, only 242 dialogues were completed through to the half-way point and only 127 dialogues were completed to the end. A handful of dialogues were interrupted because of system-related problems but the majority that terminated before completion did so because the students simply ended their session. Feedback from course tutors and comments from the students themselves supported researcher intuition that poor

system understanding of student dialogue contributions was probably a key reason for the fall-off in use. This was confirmed when accuracy, precision and recall measures for individual questions were measured: apart from a handful of essentially yes/no questions the majority of these metrics were zero. Nevertheless, the exercise served its purpose in capturing a large quantity of student responses to tutorial questions. These were to serve as training data for the next stage of development.

### **4.3.3 Revising the dialogue script and building the new dialogue engine**

The second stage of the project involved a major shift from using ‘guessed’ categories of student responses to questions to creating categories of student responses based on a study of the responses collected during stage one. In addition, the script developed in the first stage was refined in order to deal appropriately with the newly created categories of response. This process involves creating categories or themes from student responses; it is described in detail in Section 4.4.3. While the process is very familiar to educational researchers using qualitative research methods for identifying student conceptions (for example, phenomenography (Marton and Saljo, 1976) or the methods of content analysis (Stemler, 2001)), it is far less common in the realms of ITS development. As noted in Chapter 2, ITS frequently utilise an expert model of the domain and then trace individual students’ adherence to (overlay model) or departure from (buggy model) the expert model in order to build a student model to inform the next tutor or tutorial dialogue move. In this project, the ‘expert’ model, the student model and the dialogue planner are in effect pre-compiled into the dialogue script. For each question, or dialogue context, the possible states to move to next are completely defined by the categories derived from systematic analysis of previous student responses to the same question.

The central issue however, was to improve the ‘understanding’ performance of the dialogue system in order to ensure the most appropriate next state was selected and thereby provide individualised feedback to free-text input. Two options were considered: either continue to use TuTalk and replace the existing TuTalk natural language understanding (NLU) module along with making adjustments to the script design and dialogue manager (DM), or build a new dialogue system from scratch. In the end the choice was made to start from scratch for three main reasons. First, the natural language toolkit (Bird, 2006) provided many of the functions required in a simple di-

dialogue system such as tokenisers, stemmers and a range of classifiers. Second, some of the features built into TuTalk created performance issues and resulted in sluggish responses to student input; this was likely to be time-consuming to resolve. Third, a new modular system that could be easily extended or adapted, and which was designed from the ground up to meet the specific research goals would provide a solid base from which to do further work in this area.

Iterative development of the new system proceeded over roughly a 2-year period and involved several periods of informal evaluation with teaching staff and colleagues. In addition, 58 second-year student volunteers trialled the new system prior to release for the formal in-class evaluation which is described in Chapter 6.

The next section describes the new dialogue system architecture and provides details of its main components.

## **4.4 A new surface-based tutorial dialogue system**

Section 4.4.1 begins with a basic walk-through from a section of dialogue with a real student to demonstrate how the system works in practice. In Section 4.4.2, the system architecture is described. The dialogue script design is covered in Section 4.4.3, followed by a detailed description of the dialogue manager in Section 4.4.4. The tutor server, preprocessor and classifier modules are described in Sections 4.4.5 - 4.4.7. Given the importance of natural language understanding from both an education-theoretic and a practical perspective, the approach to classifying student input text is described separately in section 4.5.

### **4.4.1 A walk through the tutorial dialogue**

Students connect to the system via a simple chat-style web interface (see Figure 4.1). The system types out statements and questions and the student responds by typing into a dialogue box. The system always waits for a student response before continuing. The following is an annotated excerpt from a real dialogue session between the system and a student. The excerpt begins with an assertion from the tutor:

TUTOR: Stroke Volume (SV) is the volume of blood ejected from the heart with each beat. The average adult heart pumps a total of 5 litres of blood each minute, and the average adult heart rate is 72 beats/minute.

The tutor then asks the student to perform an action which is directly related to this information:

TUTOR: Please calculate the stroke volume for an average adult.

The student's initial response to this is:

STUDENT: 5000/72

The student has not included the units for stroke volume which, even though not explicitly requested, have been modelled by the tutor. The tutor recognises the student's response is numerically correct and responds with an acknowledgement of the student input followed by some scaffolding:

TUTOR: OK. Looks like you have the right idea but you have either forgotten to give the units or used the incorrect units. Remember you can work out the correct units to use like this: You are given HR (beats/minute) and CO (l/minute).  $SV = CO/HR$  which is l/min divided by beats/min = l/beat. Finally don't forget to multiply the result by 1000 in order to convert l/beat to mls/beat.

Next, the tutor asks the student to try again and repeats the question:

TUTOR: Try again. Please calculate the stroke volume for an average adult.

This time the student responds with the same answer they produced the first time but this time in decimal form:

STUDENT: 69.4

Once more, the tutor recognises that the units are missing but this time responds slightly differently. The problem with the student's answer is again stated explicitly but detailed scaffolding is not included since the tutor also indicates that the problem is the same as the student's previous attempt:

TUTOR: Again, looks like you have the right idea but you have either forgotten to give the units or used the incorrect units.

And, once more the tutor repeats the question:

TUTOR: Try again. Please calculate the stroke volume for an average adult.

This time the student provides the correct answer with appropriate units:

STUDENT: 69 ml

The tutor provides positive feedback before moving on to the next dialogue context:

TUTOR: Well done.

This excerpt from a dialogue session illustrates three important features of tutorial dialogue. First, initially the student in this situation is not responding directly to the feedback given. It seems as though some other thinking is going on which results in the same answer being repeated, but just expressed differently. Feedback has to be noticed in order to be responded to. Second, if the dialogue is to be believable, the tutor has to respond appropriately where the same question is repeated. This finessing is one of the features of human dialogue which sets it apart from machine-speak. Humans typically will not just repeat the same utterance over and over, nor should the tutor. This feature is achieved through a mechanism that updates a simple representation of the current dialogue state. Third, there are many ways that students might express the answer to this particular question the tutor needs to be able to recognise most, if not all of them, in order to provide appropriate feedback. Another common error in this dialogue context for example, was problems with algebraic manipulation. The feedback if this were the error is different but the tutor action is the same: the student is asked to try again, and the question is repeated.

The implementation details for the dialogue manager are discussed in Section 4.4.4 where the same excerpt from a student dialogue session is used to describe the main dialogue manager algorithm. However, first, Section 4.4.2 introduces the overall tutorial dialogue system architecture.

## 4.4.2 System architecture

The new dialogue system is written in Python and utilises several NLTK libraries. In addition, Peter Norvig's 'toy' spell checker is used in the preprocessor module and the Asyncore and Asynchat libraries are used to manage multiple simultaneous client connections in the tutor server module.

The tutor server can readily communicate with any web-application front end using XML-formatted messages. A java-based web application through which multiple clients connect to the tutorial dialogue system was created by Richard Zeng, lead

developer in the Educational Technology group within the Higher Educational Development Centre (HEDC). This web application also provided seamless integration with the University central authentication system which was essential to provide easy access for students and staff and retain data integrity without providing system specific logins or identities). Figure 4.1 shows a screenshot of the web interface used by students.

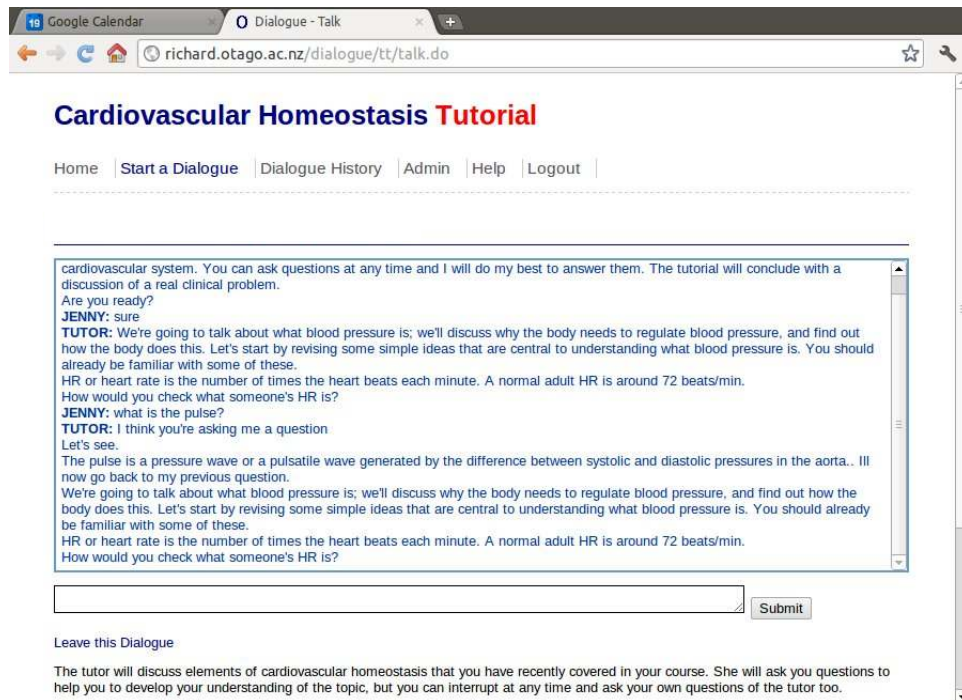


Figure 4.1: Screenshot of Dialogue System Web Client.

Figure 4.2 provides an overview of the system architecture. Each client connection to the dialogue system creates an instance of the dialogue manager which sends tutor contributions to the client according to the preloaded script and receives student contributions which are then preprocessed and classified to determine the next tutor contribution to retrieve from the script.

Sections 4.4.3 to 4.4.5 describe in detail each component of the dialogue system.

### 4.4.3 Script design

The structure of the tutorial dialogue is determined by the dialogue script. The script uses an FSA dialogue model, as introduced in Section 3.2.3.1, which permits an organic authoring process. In other words, the script author can use as many or as few states as required. A key feature of the implementation is that the notion of state as equivalent

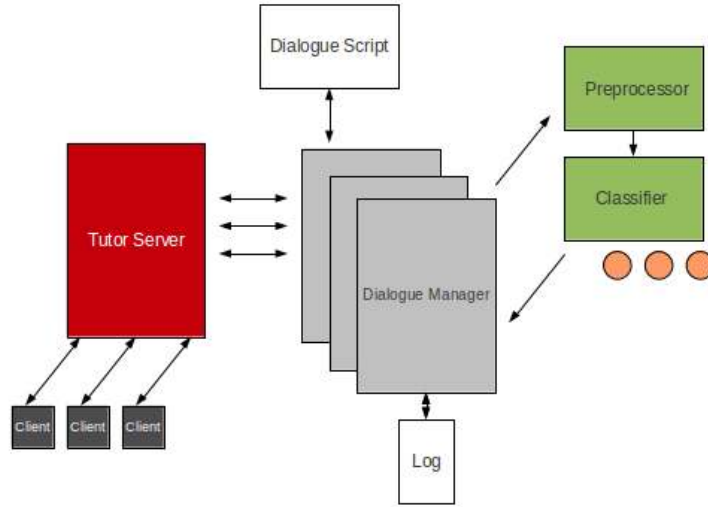


Figure 4.2: Architecture of the Dialogue System.

to a single question is extended to include features which specifically relate to each question posed in the context of a purposeful dialogue - these additional features are implemented through the use of dialogue **contribution nodes**. Through the dialogue contribution node, each state corresponds not just to a question but encapsulates a specific dialogue context and related dialogue acts. In this respect it borrows ideas from the information-state architecture approach of Traum and Larsson (2003) as well as from the *DAMSL* (Dialogue Act Markup in Several Layers) architecture (Core and Allen, 1997).

Using a greatly simplified scheme from that proposed by Core and Allen (1997) each dialogue contribution node in the hand-crafted script contains forward and backward tagsets. Each of these contains relevant dialogue acts or action-directives, for example, an information request or a directive to go to a specified dialogue contribution node.

A potential drawback of this approach as noted by Jurafsky and Martin (2009) and referred to in Section 3.2.3.1, is a potential explosion in the number of possible states. Nevertheless, it was clear from both the student response data and from the context of this tutorial dialogue (a well defined curriculum within the specific domain of cardiovascular homeostasis) that the number of possible states could be adequately contained. The tutorial is in a finite domain, and there are a finite set of issues to be covered, so the potential for follow-up sub-dialogues is limited, or at least, is not explosively large.

The decision to implement a simple finite-state architecture, augmented with a hand-crafted script, is based on a combination of practical and theoretical concerns. First, the finite-state approach is the simplest to implement and this was important in terms of developing the system in a timely manner and in such a way that it was robust enough to use with a large class of students. Second, the finite state approach is consistent with the directed nature of a tutorial dialogue where the dialogue purpose is to revise a specific and well-defined area of the curriculum. Third, the single-initiative question-asking approach provides an opportunity for students to practise tasks they are expected to complete as part of their formal summative assessment.

The script structure itself is loosely based on the Core and Allen (1997) dialogue coding scheme where each dialogue contribution node is divided into forward and backward functional layers (see Section 3.2.2). Each contribution node in the script contains forward and backward elements and each of these contain relevant dialogue acts or directions for action (for example, a request for information or a directive to go to a specified contribution node). The script is an XML file which is defined in the XML schema for the dialogue system and which comprises a series of dialogue contribution nodes. Links to both the XSD schema file for the script and to the final XML script are provided in Appendices A.2 and A.1.

An example of a single contribution node is given in Figure 4.3 below and this is used to illustrate each of the components of a dialogue contribution node in turn. This contribution node becomes the current contribution node after the student makes an affirmative response, such as ‘Yes’ or ‘OK’, to the tutor question, ‘Are you ready?’ This tutor question is contained in the previous contribution node.

#### **4.4.3.1 Contribution node element**

In this example, the unique id of the dialogue contribution is “check-hr”. All contribution nodes have a unique id which is constructed to include an abbreviation of the question for the current dialog context. Thus, “check-hr” is an abbreviation for the tutor question, “How would you check someone’s heart rate (HR)? Apart from the start node of the dialogue, which only contains a forward element, and the end nodes of the dialogue, which only contain a backward element, every contribution node has both backward and forward elements.

The id of the parent node is also a required attribute of each contribution node. An important constraint is that each contribution node can have multiple ‘child’ nodes but can only ever have one parent node. The parent node attribute is used to build a

```

<contribution-node id="check-hr" parent-node="start" default="true">

<backward class="yes">
<acknowledge>OK.</acknowledge>
</backward>

<forward>
<assert>We're going to talk about blood pressure; we'll discuss why the body
needs to regulate blood pressure, and find out how the body does this. Let's
start by revising some simple ideas that are central to understanding what
blood pressure is. You should already be familiar with some of these.</assert>
<assert>HR or heart rate is the number of times the heart beats each minute.
A normal adult HR is around 72 beats/min.</assert>
<info-request value="How would you check someone's HR?" define="You could
measure their pulse."/>
<menu>
<item value="correct">Count the pulse.</item>
<item value="simpler">With a blood pressure cuff and stethoscope.</item>
<item value="simpler">Use an ECG.</item>
<item value="incomplete">Pulse.</item>
<item value="dont-know">I don't know.</item>
</menu>
</forward>

</contribution-node>

```

Figure 4.3: Single Dialogue Contribution

list of possible next nodes for each contribution node in the script. In other words, for the example current contribution node, *check-hr*, the list of next possible contribution nodes is built from all nodes which have *check-hr* as their parent-node.

The remaining optional attribute of the contribution node element is whether the node is a default node. At least one child node for each contribution node which has children must be nominated as the default. In this way, a default path through the entire dialogue script is specified and this will be followed in the event that all utterances from the dialogue partner (the student) fail to be classified appropriately understood.

Each contribution node contains a backward element, which contains responses appropriate to the previous dialogue contribution node, for example an utterance to establish grounding (Clark and Schaefer, 1989), and a forward element which sets up the next dialogue contribution node. The backward element is discussed first, since this is the first element to be processed.

#### 4.4.3.2 Backward element

Each backward element must have a class attribute. When the contribution node is in the list of next possible nodes and the previous student dialogue contribution matches this class, this contribution node becomes the current node. In the example contribution node (Figure 4.3), responses to the parent contribution node *check-hr* are processed by a single classifier into one of the classes listed in Table 4.1. If classification fails then a generic classifier, in this case, *dont-know*, is tried (refer to Section 4.5.1 for details). If generic classification fails then the dialogue contribution node which is specified as the default is chosen from the list of possible next nodes.

Table 4.1: Possible classes for check-hr

| <b>check-hr classifier</b> |
|----------------------------|
| correct                    |
| simpler                    |
| incomplete                 |

Each backward element contains optional dialogue contributions which refer to the previous dialogue contribution node. Each contribution is a dialogue act, whose type is identified by an XML tag. A list of the types currently used is given in Table 4.2. There can be multiple tags used and they always appear in the order in which they

occur in the script. The tags identifying dialogue act types serve three purposes: they act as a guide for the script writer, they provide for the future possibility of the writer specifying tags and the system automatically selecting utterances, and finally they provide for future extensions to the system where for example a speech synthesiser or animated agent might be used and is required to support dialogue acts being delivered in different ways.

Table 4.2: Optional backward element tags

| tag           | meaning   |
|---------------|---|
| acknowledge   | I understand your last utterance                      |
| undefined     | I don't understand your last utterance                |
| agree         | I agree with your last utterance                      |
| disagree      | I disagree with your last utterance                   |
| part-agree    | I partly agree with your last utterance               |
| part-disagree | I partly disagree with your last utterance            |
| repair        | Restatement of question context or answer to question |
| affirm        | Your last utterance is correct                        |
| negate        | Your last utterance is incorrect                      |
| i-dont-know   | I don't know the answer to your question              |
| im-not-sure   | I'm unclear about your question                       |

In the example script given in Figure 4.3, the only tag contained in the backward element is *acknowledge* which in this case prescribes the utterance ‘Ok’. The class attribute for the backward element is ‘yes’ which in this case indicates that the student has read and understood the tutor’s instructions and is ready to begin the tutorial.

#### 4.4.3.3 Forward element

The forward element has no attributes, it only contains other elements. It always advances the dialogue and does this in one of two ways: either through an **info-request** element or through an **action-directive** element, the forward element cannot contain both.

The info-request element has a value attribute which is a specific dialogue act, usually a question or a request for further information. In the example given in Figure 4.3, the info-request *value* attribute contains the question, ‘How would you check someone’s heart rate?’. A model answer for this question is provided in the *define* attribute and

in this case is, ‘You could measure their pulse’. The info-request element plays a pivotal role in the tutorial dialogue script. Once the tutor has performed the dialogue act contained in this element it waits for a response from the student. When the student response is received, this is classified by the classifier for the contribution node and the result of classification determines the next node to visit. In general, a separate classifier is required for each dialogue contribution node. So for example, the dialogue contribution *check-hr*, shown in Figure 4.3, has its own classifier to classify student responses to the question, ‘How would you check someone’s heart rate?’ In this case, one of the possible classes is *simpler*, which classifies responses that while strictly correct are overly complex or require specialised equipment (for instance, you can check someone’s heart rate using an electrocardiograph (ECG) but a simpler method is to count or measure their pulse at the wrist). As described in Section 4.4.3.2, the element in the script which implements this is specified in the backward class attribute value which in this case is *class=simpler* and is shown in Figure 4.4.

```
<contribution-node id="hr-simpler" parent-node="check-hr">

<backward class="simpler">
<acknowledge/>
<part-agree>That is one way to do it.</part-agree>
</backward>

<forward>
<action-directive value="Can you think of a simpler method?"
action="check-hr"/>
</forward>

</contribution-node>
```

Figure 4.4: Script Backward Class Illustration

The action-directive element has two attributes. The first is a statement of the action the tutor is about to take and the second is the id of a specific contribution node to which the dialogue should jump (similar to the programming command, GOTO). Action directives are useful in situations where the script author wants the student to revisit a node and try a question again or where additional prompting is required perhaps in the form of more structured yes or no answers. An example action-directive

element is shown below in Figure 4.5. In this case, a student has responded to a question with an utterance equivalent to ‘I don’t know’. Remedial action is called for and the script jumps to a dialogue context which provides additional prompts for the student to work with.

```
<contribution-node id="baro-response-dont-know" parent-node="baro-response">
</backward>
<forward>
<action-directive value="Let's consider what happens in each case with a
fall in BP." action="baro-symp"/>
</forward>
</contribution-node>
```

Figure 4.5: Action Directive Element

Finally, the **menu** element, shown in Figure 4.3 within the forward element has a special function. A key design goal was to provide a system which can provide a testbed to explore whether free-text entry confers any advantage over menu selection. The menu tag contains selection options denoted with **item** tags for a menu-based version of the system. In the menu-based version of the system, instead of sending user input to the classifier, input is simply matched with one of the available menu options. The menu element is ignored by the free-text version where user choice is ‘guessed’ by the classifier for each dialogue context.

#### 4.4.3.4 Limited mixed initiative provision

While the tutorial system is primarily designed for single-initiative dialogue (described in Section 3.2.3.1), with the tutor taking all the initiatives, there is provision in the script for student initiatives; at any point in the dialogue, instead of answering the tutor’s question, a student can ask his or her own question. If this happens, the system identifies that a question has been asked (using a generic question classifier which is discussed in Section 4.5.1), indicates that it is not able to answer the question at this time and then restates its own question, restoring the dialogue state that held before the student asked their question. Answering student questions is not currently supported in the system however a special dialogue node called *question-match* is visited each time a question is asked by the student and this node is the placeholder for question answering to be implemented in a future enhancement of the system.

#### 4.4.4 Dialogue manager

As already mentioned, the model used for dialogue management is a finite-state model (see Section 3.2.3.1). The general algorithm for the dialogue manager involves opening each element tag for the current dialogue contribution node, implementing the dialogue acts or actions it contains in the specified order, updating the **dialogue state** then waiting for the dialogue partner to respond in order to work out which contribution node to visit next. In the context of this dialogue system, dialogue state is a very simple data structure which is held as a global variable. This variable records a mapping of node id to the number of times each contribution-node in the script has been visited and the number of visits to each contribution node from its ‘child’ contribution nodes is recorded, along with a Boolean to indicate whether a contribution node has been repeatedly visited from a child node.

The pseudocode in Figure 4.6 provides a detailed overview of how the dialogue manager processes each contribution node in the script. An example of the output of one part of a dialogue session is given in Figure 4.7 and this is the same snippet that was used to describe a walk-through the dialogue system in use in Section 4.4.1. The example dialogue in Figure 4.7 begins with a tutor assertion about cardiac stroke volume followed by an information request from the tutor where the student is asked to calculate the stroke volume for an average adult. This excerpt, from a real tutorial dialogue session, occurs approximately halfway through the dialogue script. The following description of the main dialogue manager algorithm should be read with reference to these two figures.

The main function of the dialogue manager is to process contribution nodes in the script. A recursive function called **process\_node** is the heart of the dialogue manager (line 01 in Figure 4.6). The function, **process\_node**, takes three parameters: the node to process, the previous node processed and a Boolean which indicates whether the node to process has been called from an action directive (this is explained further in Section 4.4.4.1 below).

When the tutor asks the question ‘Please calculate the stroke volume for an average adult’, the current node is *adult-sv*. The student response to this question is, ‘5000/72’. Once this user response has been processed by the dialogue manager, there is a new node to be processed called *adult-sv-units* and *adult-sv* becomes the previous node. From this point, Section 4.4.4.1, provides a step-by-step description through the the function, **process\_node** and covers the remainder of the dialogue excerpt listed in Figure 4.7.

```

01 function process_node(current_node, last_node, is_action_directive)
02
03   update_dialogue_state(current_node, last_node)
04
05   if current_node is not start_node AND current_node.visits <= max_visits AND
06   NOT is_action_directive then
07     process_bwd(current_node)
08
09   else if dialogue_state.current_node.visits > max_visits then
10     send user text to indicate we're moving on
11     next_node_list = find_candidate_nodes(current_node)
12     next_node = get_node(default, next_node_list)
13     process_node(next_node, current_node, is_action_directive=TRUE)
14
15   if current_node is not end_node then
16     next_node, is_action_directive = process_fwd(current_node)
17     process_node(next_node, current_node, is_action_directive)
18   else we are done!
19
20
21
22 =====
23 function process_bwd(current_node)
24
25
26   for each element in backward_element of current_node
27     if NOT dialogue_state.local_repeat then
28       send element text to user
29     else if repair_element then
30       send finessed text to user
31
32
33 =====
34 function process_fwd(current_node)
35
36   if NOT is_action_directive then
37     for each assert_element in forward_element of current_node
38       send element text to user
39   if info-request then
40     send value to user
41     response = wait for user response
42     next_node_list = find_candidate_nodes(current_node.id)
43     next_node = classify(preprocess(response), next_node_list)
44     is_action_directive=FALSE
45   else if action-directive then
46     send value to user
47     next_node = get_node(action, all_node_list)
48     is_action_directive=TRUE
49   return next_node, is_action_directive
50
51 =====
52 function update_dialogue_state(current_node, last_node)
53
54   dialogue_state.current_node.successive = FALSE
55   if current_node.id = last_node.parent_node then
56     dialogue_state.last_node.child[current_node].visits += 1
57   if dialogue_state.last_node.child[current_node].visits > 1 then
58     dialogue_state.current_node.successive = TRUE
59
60   dialogue_state.current_node.visits +=1
61

```

Figure 4.6: Dialogue Manager Pseudocode

TUTOR: Stroke Volume (SV) is the volume of blood ejected from the heart with each beat. The average adult heart pumps a total of 5 litres of blood each minute, and the average adult heart rate is 72 beats/minute.

Please calculate the stroke volume for an average adult.

STUDENT: 5000/72

TUTOR: OK. Looks like you have the right idea but you have either forgotten to give the units or used the incorrect units. Remember you can work out the correct units to use like this: You are given HR (beats/minute) and CO (l/minute).  $SV = CO/HR$  which is l/min divided by beats/min = l/beat. Finally don't forget to multiply the result by 1000 in order to convert l/beat to mls/beat.

Try again. Please calculate the stroke volume for an average adult.

STUDENT: 69.4

TUTOR: Again, looks like you have the right idea but you have either forgotten to give the units or used the incorrect units.

Try again. Please calculate the stroke volume for an average adult.

STUDENT: 69 ml

TUTOR: Well done.

Figure 4.7: Dialogue Session Illustration

#### 4.4.4.1 Processing a contribution node

(Note that functions and variables are listed in bold and element tags and the values of variables are italicised. In addition, the first-person plural, *we* is used in this section to aid clarity).

Beginning at line 03, the variable `dialogue.state`, which is initialised at the start of the dialogue where all numeric values are set to zero, is updated by the function `update_dialogue_state`. This function takes the current node and previous node as parameters. Since, for the purpose of this illustration we have not visited *adult-sv-units* previously, the value of **`dialogue.state.current_node.visits`** is now equal to 1. Next, three checks are performed to see whether the **`current_node`** backward element should be processed (line 05): *adult-sv-units* is not the start node for the dialogue, the maximum allowed visits (3 is the default maximum) has not yet been exceeded and the Boolean **`ad`** was set to FALSE when the **`process_node`** function was called, therefore we can process the backward element of *adult-sv-units*.

A new function called **`process_bwd`** is invoked at line 07. **`process_bwd`** takes as its parameters **`current_node`** (value = *adult-sv-units*). **`process_bwd`** checks to see whether this is a successive visit to the current node from one of its child nodes. The global `dialogue.state.current_node.successive` is FALSE therefore this is not a successive visit. Each element contained in the backward element is opened and the text is sent to the student (lines 27-29). In this case there are two elements contained in the backward element, the first is the *acknowledgement*, ‘Ok’, and the second is the *repair*, ‘Looks like you have the right idea ...’. We now return to the main function, **`process_node`**.

At line 16 *adult-sv-units* is not an end node so we can proceed to process the forward element. Another new function **`process_fwd`** is invoked. **`process_fwd`** takes only one parameter, the current node, *adult-sv-units*. There is only one element contained in the forward element for *adult-sv-units* and this is an *action-directive* so we go straight to line 44 in order to process the *action-directive*. The value attribute of the *action-directive* is ‘Try again.’ so this text is sent to the student (line 45). The action attribute of the *action-directive* is the id of contribution node *adult-sv* so this is passed to a utility function **`get_node`**, along with a list of all contribution nodes in the dialogue script and the variable `next_node` is returned. The Boolean variable, **`is_action_directive`** is set to TRUE (line 47) since then next node to go to has been defined as the result of an *action-directive* and we return to the main function, **`process_node`**. **`process_node`** makes the recursive call using **`next_node=adult-sv`**, **`current_node=adult-sv-units`** and **`is_action_directive=FALSE`** as parameter values.

We are now on the second turn through **process\_node**. The current node visits is incremented by 1 so the value for **dialogue\_state.current\_node.visits** is now 2 since we have visited *adult-sv* once before in this illustration. This time round we do not process the backward element of *adult-sv* since **is\_action\_directive** is set to TRUE (line 05).

The current node is not an end node so we can proceed to process the forward element (line 16). There are three elements contained in the forward element for *adult-sv* (two assertions in addition to an info-request - note that these are not shown); however, **is\_action\_directive** is TRUE so we go straight to line 37 to process the *info-request*. The value, ‘Please calculate the stroke volume for an average adult.’ is sent to the student and this time the student responds with, ‘69.4’. The student response is preprocessed and classified and together with a list of possible next nodes for the **current\_node**, this is used to identify the **next\_node**. Given the student’s response, once again the next node turns out to be *adult-sv-units* (lines 39-42). Since we are processing an *info-request*, **is\_action\_directive** is set to FALSE. Finally, the value of **next\_node** and **is\_action\_directive** are returned to the calling function, **process\_node**. **process\_node** makes the recursive call using **next\_node**=*adult-sv-units*, **current\_node**=*adult-sv* and **is\_action\_directive**=FALSE as parameter values.

We are now on the last full turn through **process\_node** in this illustration. The dialogue state is updated and **dialogue\_state.current\_node.visits** is now 2 since we have visited *adult-sv-units* once before and **dialogue\_state.last\_node.child[adult-sv-units].visits** is now also equal to 2 therefore **dialogue\_state.current\_node.successive** is set to TRUE. *adult-sv-units* is not the start node for the dialogue, the maximum allowed visits has not yet been exceeded and **is\_action\_directive** is set to FALSE, therefore we can process the backward element of *adult-sv-units*. Since this is a successive visit we need to finesse the backward looking tutor response which is sent to the student. The simple rule used for all finessed text in the current dialogue is to use only the *repair* element contained in the backward element and concatenate the word ‘Again’ to the start of the text. In this case, finessing the response results in, ‘Again, looks like you have the right idea ...’. We now return to the main function, **process\_node**.

The function **process\_fwd** is invoked and there is only one element contained in the forward element for *adult-sv-units*. This element is an *action-directive* so we go straight to line 44 in order to process it. The value attribute of the *action-directive* is, ‘Try again’ and this text is sent to the student. The action attribute of the *action-directive*

is the contribution node *adult-sv* so the variable **next\_node** takes this value through a call to **get\_node**. *ad* is set to TRUE since we are processing an *action-directive*. Finally, the value of **next\_node**, and **is\_action\_directive** are returned to the calling function, **process\_node**.

The recursive call to **process\_node** is made using **next\_node=adult-sv**, **current\_node=adult-sv-units** and **is\_action\_directive=TRUE** as parameter values. The dialogue state is updated **dialogue\_state.current\_node.visits** is incremented by 1 and the value for **dialogue\_state.current\_node.visits** is now 3 since we have visited *adult-sv* twice before in this illustration.

*adult-sv* is not the start node for the dialogue, the maximum allowed visits (3) has not yet been exceeded but **is\_action\_directive** is set to TRUE therefore we do not process the backward element of *adult-sv*. *adult-sv* is not an end node so we can proceed to process the forward element. **is\_action\_directive** is TRUE so we go straight to line 38 to process the *info-request*. The value, ‘Please calculate the stroke volume for an average adult.’ is sent to the student and the student responds, this time with ‘69 ml’.

This answer is classified as correct and the algorithm moves on to the next appropriate contribution node in the script which has the backward class attribute equal to correct and which issues the backward looking utterance, ‘Well done’.

As noted in Section 4.4.3.3, the menu-based version of the tutorial dialogue system does not send user input to the classifier. Input is matched with one of the available menu options which are read from the dialogue script. This is achieved through a simple change at line 42 of the pseudocode to effectively match input to one of the menu options from the script. In every other respect, the two versions of the dialogue manager are identical. This means that the menu-based system delivers a form of teaching which is quite different from a regular list of multi-choice questions. In the current system, the alternatives presented in each question are derived directly from an analysis of student responses and they may look quite different from alternatives developed by teachers trying to guess likely student misconceptions. In addition, when students select an answer to a question, they receive context-specific tutorial feedback on this answer, just as they do in the free-text version of the system. The subsequent questions they are asked reflect their answers to previous questions. (This form of multi-choice dialogue is of interest in its own right, though it is not the focus of study in the current thesis.)

The only remaining detail of the algorithm which is not covered in the example given is terminating the dialogue. There are two ways this can happen. First, a user

through their web client, simply terminates the connection, in which case the dialogue ends and is incomplete. Second, a contribution node is reached which is one of the possible end nodes for the dialogue. In this latter case, the else-statement at line 20 is entered, a final message is sent to the client and the connection to the client is closed. When the client connection is closed the current state of the dialogue, along with student id information, is passed to the web client. This allows dialogues to be restarted at the point they were stopped if incomplete. This feature was not required for the present study.

A complete copy of the source code for the dialogue manager is available via a web address provided in Appendix A.3. For practical reasons, which relate to the event-driven architecture of the dialogue system, the actual implementation of the dialogue manager is iterative rather than recursive. However, the core functions of the dialogue manager are easier to understand and follow using a recursive explanation.

#### **4.4.5 Tutor server**

The tutor server is solely responsible for establishing, maintaining and ultimately closing the connection to clients. It is really little more than a ‘souped-up’ echo server. Instead of echoing back user input when it receives a string of text, it sends back to the client the message constructed by the dialogue manager in response to the received text. The `asyncore` and `asynchat` python libraries are used to implement the server and maintain a loop which waits for client communication once a connection has been established.

The main advantage of this approach was that it was simple and quick to implement and provided a robust and easy to understand server platform.

A disadvantage of this minimalist approach was that the server could not request information from the client once a connection had been made. One resulting issue was that university student userIDs which were used to login (and ultimately used to correlate with other student data, such as examination scores) could not be passed to the tutorial dialogue system. This was not a major issue since all dialogue interactions and state information were passed from the server to the web client for each connection anyway. It simply meant that userID along with dialogue interactions and state information were recorded to a separate database by the web client. The dialogue manager independently logged each dialogue session to a uniquely identified XML file.

#### 4.4.6 Preprocessor

The preprocessor module comprises two main parts. The first is a tokeniser which takes raw text input from the user and converts it to a set of individual words or tokens, including certain punctuation marks, numeric characters and abbreviations. The extracted tokens are then passed to the second part, the normaliser, which changes all alpha characters to lower case and runs words through a basic spell checker. As noted above, the spell checker used in this system is Peter Norvig’s ‘toy’ spell checker (available at <http://norvig.com/spell-correct.html>). A domain-appropriate dictionary is required for the spell checker and comprises a text file containing representative words for the domain. For this project a dictionary was created comprising:

- The text from pilot student responses,
- The script itself,
- [http://en.wikibooks.org/wiki/Human\\_Physiology/The\\_cardiovascular\\_system](http://en.wikibooks.org/wiki/Human_Physiology/The_cardiovascular_system),
- The NLTK plain text ABC science corpus

The spell checker was not evaluated quantitatively on the tutorial domain. Informal testing appeared to show reasonable performance and Norvig claims 67% accuracy for the spell checker on an unseen test set (Norvig, 2013). For the purpose of this project this seemed adequate. If formal bench testing subsequently demonstrated an issue, the spell checker could easily be swapped out. In the event, this was not the case.

The tokeniser created for the system had some special requirements. Certain questions required students to enter physical quantities, including units. Similarly students often used question marks when asking a question or hyphens for certain expressions and it seemed sensible not to discard this information. An example of a physical quantity is cardiac output, which might take the value of something like 5 litres/minute. It is clear, even from this example, that given the opportunity to input free-text, students may express their answer in a variety of ways (lpm, lit/min, litres per min etc etc). To try to deal with this, the standard NLTK tokeniser was provided with a custom regular expression to ensure domain specific symbols and expressions were correctly captured. This is listed in Figure 4.8.

Regular expressions to extract standard words, the hyphen character, apostrophes, decimal numbers and valuable punctuation marks such as the question-mark, are used in the tokeniser function. These items all aid in interpreting the meaning of student

```

# tokeniser to deal properly with decimals, fractions, units (volume per time)
# and punctuation

def input_tokenise(resp):
    wordr = r'(\w+)'
    hyphen=r'(\w+\-\s?\w+)'
    apostrophe = r'(\w+\'\w+)'
    decimals = r'(\d*(\.)?\d+)'
    punct = r'(\?+)'
    fraction=r'(\d*(\.)?\d+\s?(/|div)\s?\d*(\.)?\d+)'
    units = r'((([lL]i?t?r?e?s?)|([Bb]e?a?t?s?))[\s]?(([pP]e?r?)|/)
[.\s]?[mM]i?n?[.\s]?))'

    pattern = '|'.join([units, fraction, hyphen, apostrophe, decimals,
wordr, punct])

    return(nltk.regexp_tokenize(resp,pattern))

```

Figure 4.8: Tokeniser Function

input. For example, a question-mark is invaluable in determining whether the student has asked a question. The **units** regular expression attempts to capture the wide range of ways that physical units can be expressed in answers to questions which require the student to calculate the value of variables. For example, the variable **heart-rate** or **HR** is expressed in beats per minute. Valid units for **HR** include *bpm*, *beats/min.* and so on.

Only standard words, not units or other tokens are subsequently passed to the normaliser for spell checking. The final output from the preprocessor is a list of tokens.

#### 4.4.7 Invoking the Classifier

Pre-compiled, or in Python terms, **pickled** classifiers, one for each dialogue context plus a small number of generic classifiers, are loaded into memory when the tutorial dialogue system is started on the server. The dialogue manager passes preprocessed input to the classifier module and receives back the result of classification which is either the name of the class chosen for the current context or a flag that classification failed. (Refer to line 17 in the dialogue manager pseudocode given in Figure 4.6).

The classifier module therefore provides the following functions:

- loads pre-compiled classifiers at initialisation
- takes tokenised input from the preprocessor and passes the required features from this input to the relevant classifier
- if the classifier returns a result it passes this back to the dialogue manager, otherwise it tries each generic classifier in turn until either a result is returned or classification fails.

The approach to building classifiers for free-text input received by the dialogue manager and which is tokenised and normalised by the preprocessor is described in detail next, in Section 4.5. The types of classifier used are detailed in Sections 4.5.1 - 4.5.2 and the methods used for training and testing classifiers are described in Section 4.5.3.

### 4.5 Approach to Classification

In general, the supervised classification of text involves labelling texts by-hand by a human expert, whether short passages, sentences, or whole documents, in order to

create a training set. The training set is used to train a classifier so that labels can be automatically applied to an unlabelled test set (see Section 3.3.1). There are several models of machine classification one might use for text categorisation. These include for example, naive Bayes models, perceptrons and neural network models, kth-nearest neighbour, decision trees and maximum entropy models. It was well beyond the scope of this project to explore each of these in detail or to focus on the detail of classifier design and implementation. Naive Bayes, maximum entropy and decision trees are all implemented in the NLTK so it was convenient to start with these as a ‘black-box’ solution to text classification for the tutorial dialogue system. Decision trees, while easy to understand and therefore attractive to this researcher, typically produce variable results for smaller training sets (Manning and Schütze, 1999). It made sense therefore to make the initial choice of classifier from naive Bayes or maximum entropy models.

Initial attempts to classify student data utilised a naive Bayes classifier. On the face of it this produced some reasonable results but the biggest issue in the context of this project was that it ‘guessed’ labels, rather than failing, even where there was no training data. The maximum entropy classifier by contrast does not guess in the absence of data. If it has never seen a particular feature-set previously it assigns equal probabilities to all possible labels. This feature of maximum entropy classifiers is well documented (see for example Manning and Schütze (1999) p.597). Its confidence in its own categorisation can therefore be easily assessed, and if this is low, the tutor can respond by saying it does not understand the student’s utterance; a particularly important trait in a tutorial dialogue system.

#### 4.5.1 Context-specific and context-general classifiers

While the classes or labels derived by hand are specific to each dialogue contribution, there are some responses that a student can make irrespective of the tutor question: ‘I don’t know’ or similar, indicating that the student cannot answer the question, ‘I don’t understand’ or similar, indicating that the student does not understand the question and where a student asks a question instead of responding to the tutor question. For this reason, three generic classes are used throughout the dialogue in situations where context-specific classification fail. These generic classifiers are: *Question*, *Dont-know* and *Dont-understand*. A flow diagram of the classification process is shown in 4.9. There is a classifier for each dialogue contribution (DC-Classifier). The letters  $A_1 \dots A_n$  represent possible classes for student input. If classifier confidence falls below a

certain threshold for assigning input to one of the possible classes then the unclassified input is passed on to the generic classifiers which determine whether the input string is likely to be a question (Q) or some variation on ‘I don’t know’ (DK) or ‘I don’t understand the question’ (DU). (Note that classifier confidence is measured in terms of entropy of the probability distribution for the class labels,  $E$ . A threshold value,  $e$ , is specified for each classifier. If  $E > e$ , then classifier confidence is low. This is discussed further in Section 4.5.3). If the input remains unclassified after each of these generic classifiers has been tried, the dialogue moves to the default next node in the script (Default).

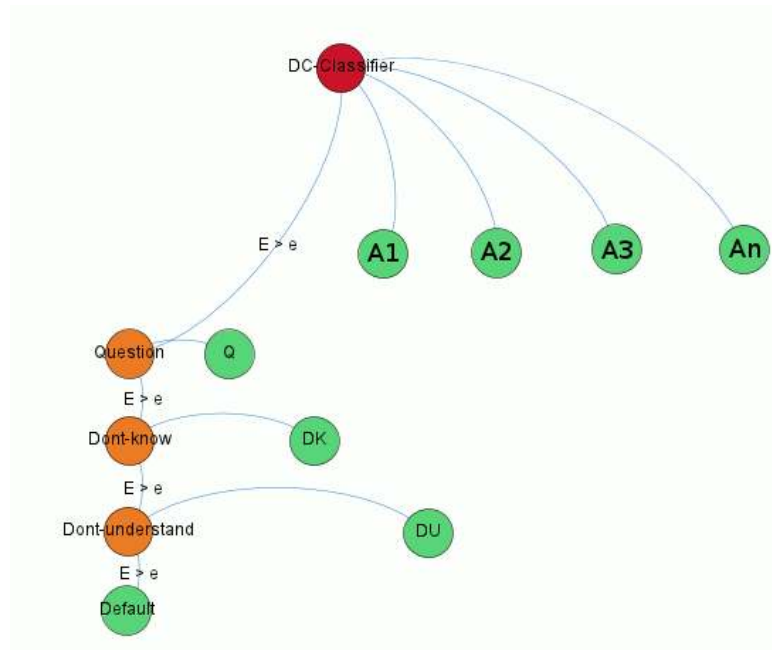


Figure 4.9: Classifier flow diagram.  $A_1 \dots A_n$  are alternative categories for the answer to the question to which the student is currently responding. If the classifier’s confidence in its assignment of one of these categories falls below some threshold ( $E > e$ ), the student’s response is successively input to the ‘Question’ classifier, the ‘Don’t know’ classifier and the ‘Don’t understand’ classifier. The latter two classifiers are binary classifiers; the ‘Question’ classifier is currently just a placeholder.

## 4.5.2 Classifiers for multi-part questions

Tutor questions with multi-part answers, for example, *‘Can you think of three main factors which affect cardiac contractility?’*, lent themselves to chaining together a series of binary classifiers, using the NLTK MultiBinary Classifier wrapper, rather than including all possible classes of response within a single multi-label classifier. This is the best approach given the relatively small amount of training data compared to the large number of possible classes of response. For example, in the question given above, three possible factors to list gives a total of eight possible classes of answer. For some combinations there are only very small, but nevertheless important, training sets and this leads to poor classifier performance overall. Training three binary classifiers which identify each of the factors sought as either present or not and then writing a function which returns a list of the factors found effectively increases the amount of training data per factor. While this approach yielded some improvement, the relatively small amount of training data or possibly a **class imbalance problem** was still evident for some combinations. The class-imbalance problem refers to the situation where the number of training samples for one class outweighs the number of training samples of another class. This has been shown to affect classifier performance for a range of classifiers. (See for example, Japkowicz (2000)).

## 4.5.3 Method for training and testing classifiers

The basic process used for creating classifiers for each context was as follows:

The first 100 student responses for each tutor question in the original dialogue were classified by a human marker (the author). This training set was divided into 5 folds and a **Maximum Entropy** classifier trained on 4/5 folds using simple **bag of words** as the featureset and then tested on the remaining fold. A 5-way cross-validation was carried out and accuracies for each of the 5 test sets calculated. The average accuracy across the 5 test sets and standard deviation was also recorded. This process was repeated using different combinations of featuresets (for example, bag of words, word length, first word, with/without stemming, with/without stopwords etc.) until the highest accuracy and least variability in test set results was achieved. Where initial results were poor, on the basis of accuracy and standard deviation across the 5 test sets, additional student data was marked up and added to the training set. Standard deviation of the accuracy across test sets was found to be especially helpful in providing an indication of the reliability of classifier results. A higher standard deviation indicated that there was

higher variability in accuracy metrics on test sets and therefore reduced likelihood of reliable accuracy on previously unseen data. Classifier **precision** (the index of false positive results for each label) and **recall** (the index of false negative results for each label) data were produced and marked-up student response data were added to the training sets in situations where it was felt the impact of a false negative or false positive result for any given label would be especially troubling for the end user.

Finally, each classifier was evaluated with previously unseen or held-out data and scored relative to a human marker. The entropy of the probability distribution ( $E$ ) was calculated for each unseen response and this was used to determine appropriate thresholds for classification. This may seem like an unnecessary additional step, given that a maximum entropy classifier chooses the best result on the basis of training data and only on training data. However, unless the classifier is reliably 100% accurate there are still false positive and/or false negative results for some labels. In practice the impact on the student of a false negative or false positive result may vary. For example, say that a student provides a correct response, according to a human marker, but in a multilabel classifier this response is incorrectly assigned to another label. The feedback the student gets from the tutor in this situation may actually induce more confusion in the student than the feedback the student receives if the classifier fails completely. The following two transcripts listed in Figure 4.10 illustrate this point. In the first example there is a false negative classification on the correct label. The student response, while tongue in cheek, is nevertheless correct; the flow in the two circulations had indeed ‘better be’ the same. In the second example, where the entropy threshold for classification is increased, classification has failed completely but the tutor acknowledges that it has not understood the student and provides an answer to the question. For the student, example 2 is a far more intelligible response from the tutor.

The confidence of the classifier in its probability distribution for all labels is high if  $E$  is close to zero. Conversely,  $E > 1$  indicates low confidence and less difference between the class labels. The first example above, with a false negative classification on the *correct* label and a false positive classification on the *incorrect* label, produces a confusing tutor response for the student. The second example uses the same classifier but with the entropy threshold increased by hand to effectively reduce the false negative rate. The judgement made here of course is that the effect on students of a concomitant increased false positive rate on the correct label is likely to be less troubling in practice.

The choice of an appropriate entropy threshold is really a subjective judgement and

#Example 1.

TUTOR: Is the flow, or cardiac output, the same in the systemic and pulmonary circulations?

STUDENT: better be

TUTOR: Sorry, no that is incorrect. Have another go.

#Example 2. Increased entropy threshold for same circ-design classifier:

TUTOR: Is the flow, or cardiac output, the same in the systemic and pulmonary circulations?

STUDENT: better be

TUTOR: I don't think I have understood you. The flow in the systemic and pulmonary circuits is the same. All the blood which is oxygenated by the lungs goes to the left heart to be pumped around ....

Figure 4.10: Feedback when Classification Fails

is made based on the classifier performance as well as the dialogue script context and the likely impact of a false negative or false positive classification for a specific label. The use of a confusion matrix helped to visualise false positive and false negative results for multilabel classifiers (that is, non-binary classifiers). A confusion matrix plots the labels assigned by the classifier to an unseen data set against those assigned as a reference to the same data set. An example from a reference set with 20 items for classification is given in Figure 4.11.

100% Correspondence between the reference set and the classifier is achieved where all labels fall on the diagonal. In the example in Figure 4.11, the classifier has achieved an overall accuracy of 80% with precision and recall values of 1.00 on both the **Correct** and **Valve** labels and somewhat lower performance on the remaining labels.

In a sense, the process of assigning an entropy threshold is a rough and ready strategy to try to reduce the potential negative consequences of using the tutor in the absence of unlimited training data and 100% accurate classifiers. There is the potential

|                | Arterial Movt | Correct | Incomplete | Self-ref | Valves | Not classified |
|----------------|---------------|---------|------------|----------|--------|----------------|
| Arterial-movt  |               |         |            | 1        |        |                |
| Correct        |               | 2       |            |          |        |                |
| Incomplete     | 1             |         | 6          | 1        |        | 1              |
| Self-ref       |               |         |            | 7        |        |                |
| Valves         |               |         |            |          | 1      |                |
| Not classified |               |         |            |          |        |                |

*Row = Reference, Column = Classifier*

Figure 4.11: Sample confusion matrix.

to automate this process, however it will require a method to assess the ‘cost’ (in educational terms) of false positive and false negative classification for each class label in each dialogue context.

Once the decision is made that the classifier is good enough to release to students, it is **pickled**, along with its associated featureset parameters and **e** value and saved for use in the tutorial dialogue system itself.

## 4.6 Summary

A new tutorial dialogue system has been introduced in this chapter. The rationale for development and the design goals have been described along with early prototyping of an initial system which was used for data collection. The dialogue script design for the new system and the dialogue manager are covered in detail using illustrative examples from actual dialogue sessions. The remaining essential modules of the dialogue system, including the tutor server, preprocessor and classifier are also described. Finally, a separate section detailing the approach taken to classifying student input provides a segue to the next Chapter 5 where the creation and testing of representative classifiers and associated tutorial dialogue script revision is covered in detail.

# Chapter 5

## Classifier and script design case-studies

### 5.1 Introduction

This chapter describes the creation and testing of representative classifiers from each of the three categories of tutor question type which are included in the tutorial dialogue and summarises the bench test results on held-out data for all classifiers used in the system. The categories of question used in the dialogue are **binary**, **multi-part** and **open**. In brief, a binary question requires exactly one type of response and the response is either there or it is not. Yes/No questions are good examples of a binary type question. By contrast, an open question is one which requires some kind of development of ideas; for example making an inference, justifying a choice or applying a principle. It requires much more than a simple *yes* or *no* response or restatement of facts. Multi-part questions are those where several specific components or features are required in the response. For example, a question beginning with ‘List 3 reasons ...’ is likely to be a multi-part question. The examples in this chapter have been chosen in order to illustrate practical issues that are faced when building natural language classifiers in a relatively constrained discourse domain but with limited training data. Section 5.2 begins with an example of a binary question; these are usually the most straightforward question type to deal with. Section 5.3 deals with one example of a multi-part question and Section 5.4 covers two examples of open questions, the hardest question type to deal with. This chapter concludes with Section 5.4.4 where classifier performance on previously unseen data is summarised for all classifiers used in the dialogue system, not just the examples described in the case studies. For easy reference, the terminology

used in this chapter is summarised in Appendix B.

## 5.2 Binary Questions

In the context of the tutorial dialogue a binary question is one which can be answered with a statement of one of two mutually exclusive options. Typical examples include yes/no questions, questions which seek one of increasing/decreasing or more/less options and so on. A binary question is one of the easiest to ask and also, assuming a reasonably well balanced training set, one of the simplest for a machine classifier to deal with. Binary questions are useful not only to check factual understanding but also, used in the context of this tutorial dialogue in combination with open and multi-part questions, have a place in resolving gracefully situations where more complex classifier options have failed. Certainly they are useful in developing a tutorial dialogue script where there are limited training examples.

Nevertheless, there are at least three important considerations for binary questions. First, binary questions are not necessarily best dealt with by binary classifiers although this may be the case in some situations. A binary classifier will determine the presence or absence of a particular feature set. A binary question however may elicit responses which can take one of two possible labels, may require additional alternative conception labels, or may not be able to be labelled. For this reason the majority of binary-type questions in this dialogue in fact use a **multilabel** classifier. This is illustrated in the following case-study in Section 5.2.1.

Second, just because a question is binary, this does not mean that all student free-text responses will follow the format anticipated for the question-type. Typically there will be a few students, for each binary question, who will elaborate on their basic response, develop ideas, or ask follow-up questions. An example is given in the case study where some attempt is made to capture the meaning of the more common extended responses.

Finally, there is often a 50-50 chance of a student correctly guessing the response even with free-text input. How important this is in terms of the formative goals of the tutorial dialogue should also be considered as part of the overall script design.

### 5.2.1 Case study: Can you feel a pulse in someone's vein?

There are several structural and functional features which distinguish arteries from veins. Thickness and compliance of vessel walls, whether carrying oxygenated blood

and whether the flow of blood is pulsatile are among these. Asking students to state whether a pulse can be felt in the vein is designed as an applied question to encourage students to think further about the nature of the pulse, which they have just been discussing, in addition to considering the function and structure of veins.

#### 5.2.1.1 Objective of the question

The question is posed as a simple yes/no question. The initial assumption made in developing the dialogue script was that most students would approach it as a yes/no question. The initial idea was that this simple starter question would then be followed with a question seeking to elicit student reasoning. However, based on the training data it was clear that a number of students provided more than a simple yes/no response and explained their reasoning unprompted. Of these students, a handful provided excellent explanations while others provided reasonable but incomplete answers to the implicit question, ‘Why can you *not* feel a pulse in someone’s vein?’.

#### 5.2.1.2 Training dataset description

A reasonable first step to take, before attempting to define classes of response to a given question, is to summarise and examine the actual student responses and create a frequency distribution of student responses. This approach was followed for all the classifiers in the dialogue system, and is illustrated in all the case studies in this chapter.

The frequency plot and unique responses to the question ‘Can you feel a pulse in someone’s vein?’ are illustrated in Figure 5.1 and Table 5.1. Table 5.1 lists only the responses where more than one student answered the question in an identical way plus the first (‘gfgdg’) of a total of 52 responses which were unique to a single individual. Figure 5.1 presents all the data graphically. The x-axis represents the number of unique response categories (N=59) and the y-axis represents the number of students who responded in each category.

#### 5.2.1.3 Selecting classes

Four classes of response to the question were designed based on typical answer themes. These are shown in Table 5.2. The ‘yes’ and ‘no’ responses which equate to the *incorrect* and *correct* classes respectively are apparent from the frequency plot and there are many more correct responses than incorrect ones. The *correct-explain* class means a correct response with a good explanation. An actual example is, ‘no because veins are

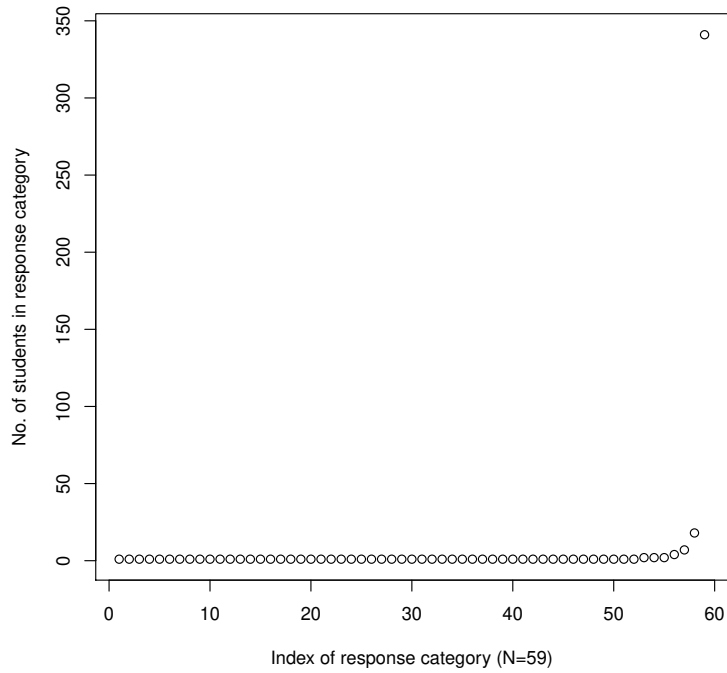


Figure 5.1: Frequency of responses by category.

Table 5.1: Unique Responses & No. of Respondents

| Unique Response | No. of respondents |
|-----------------|--------------------|
| no              | 341                |
| yes             | 18                 |
| nope            | 7                  |
|                 | 4                  |
| d               | 2                  |
| not really      | 2                  |
| s               | 2                  |
| gfgdg           | 1                  |

not pulsatile and the pressure in them is too low’. The *correct-part-explain* class means a correct response with a part explanation. An actual example of this is, ‘no, as the pressure in veins is very low’. The final numbers of responses assigned to each class are shown in Table 5.2.

Table 5.2: ‘Can you feel a pulse in someone’s vein?’ response classes

| Response Class              | No. Responses in Class |
|-----------------------------|------------------------|
| <i>correct</i>              | 354                    |
| <i>incorrect</i>            | 21                     |
| <i>correct-explain</i>      | 3                      |
| <i>correct-part-explain</i> | 22                     |
| <b>Totals</b>               | 400                    |

The very small number of responses in the *correct-explain* class is likely to lead to the class imbalance problem which was introduced in Section 4.5.2. The approach to dealing with this is discussed in the next section, 5.2.1.4.

#### 5.2.1.4 Feature selection, bench-testing and script revisions

Accuracy, precision and recall for each label (except *correct-explain*) of 100% was achieved on unseen data from the test set with the use of word stems and length of response. Stop words were excluded and the entropy value was 1.0. As expected, the accuracy, precision and recall was very poor (zero) for the *correct-explain* class because there were only 3 examples in the entire data set. In order to fix this, 20 hand-crafted *correct-explain* examples were added to the data to achieve a balance with the *correct-part-explain* class and training and testing was repeated. With this adjustment, the accuracy, precision and recall on all four labels for unseen data improved to 100%.

The original prototype script described in Section 4.3.2 was adapted to accommodate the additional classes. If students simply gave a correct, ‘no’ response, they would be explicitly directed to a question which sought their reasoning. A ‘yes’ response, would result in a request to reconsider, a ‘no’ response with a complete explanation would lead to a brief affirmation and skipping to the next default dialogue context, while a part explanation would be noted and a request to reconsider reasoning made.

## 5.2.2 Binary Question Summary

Binary-type questions while leading to generally good classifier performance do raise their own issues. The class imbalance problem can be a feature and in the current domain, binary questions still generally require multilabel classifiers. Binary questions, in particular yes/no, can be useful in order to avoid either an impasse in the

dialogue or giving up too soon on eliciting a response from the student. Hand-crafting additional responses for the under-represented class is an option but it is important to evaluate the classifier in a real-class context to see how viable this solution is in practice.

## **5.3 Multi-part Questions**

A multi-part question is one which is requesting the statement of a number of key facts or ideas. In the context of this project, all the multi-part questions are requests for specific facts rather than requests for elaboration of more open concepts or ideas.

Study of cardiovascular physiology requires students to become familiar with a number of key variables and their functions or effects. Several variables may work together to produce a particular effect or they may be antagonistic in their effects. The inclusion of multi-part questions, together with the facility to identify which individual parts of an answer a student has correctly identified, is therefore potentially useful in this study. The goal with these questions is to help students to revise not only individual variables but also to review how the variables interact with each other and to help them to identify specific variables which they need to review.

### **5.3.1 Case study: Can you think of the three main factors which affect cardiac contractility?**

‘Can you think of the three main factors which affect cardiac contractility?’ is a multi-part question and follows the open question which is described in 5.4.2. One of the useful features of a directed dialogue is that it supports linking related ideas together. The hope is that having established what contractility means, this will help students to think through which factors or variables might directly affect it.

#### **5.3.1.1 Objective of the question**

Sympathetic stimulation has several important effects on the cardiovascular system and these are easily confused. For example, sympathetic stimulation increases heart rate, and peripheral resistance; it also increases cardiac contractility. It is common when students first encounter cardiovascular variables for them to confuse causes with effects. For example, changes in heart rate or peripheral resistance may be suggested as

*causes* of increased cardiac contractility. There is clear evidence for this in the training dataset described in the next section, 5.3.1.2.

In common with a number of other questions in the script, this question seeks to help students to separate the causes of sympathetic stimulation from its effects. Being able to selectively identify causes and effects in student responses and provide appropriate and immediate feedback is therefore a key objective.

### 5.3.1.2 Training dataset description

The same approach described above in Section 5.4.1.2 is used to get a picture of the overall training data set. The top unique textual responses are listed in Table 5.3 and responses plotted against the number of students providing a given response are illustrated in Figure 5.2.

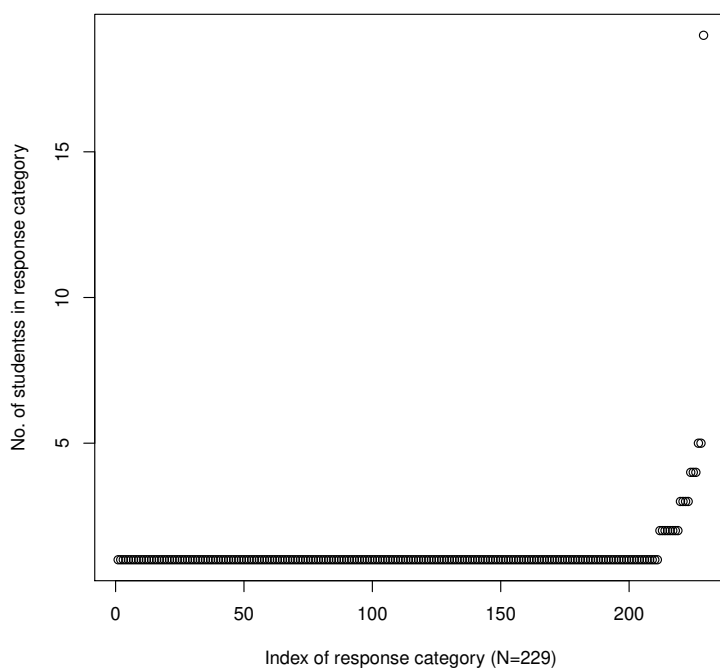


Figure 5.2: Frequency of responses by category.

In terms of the distribution pattern, the expectation might have been for some clustering around the three factors to occur. In fact, what is demonstrated in Table 5.3 is that there is a wide range of student conceptions about the factors which affect cardiac contractility. This provides evidence that the type of question (open, multi-

part or binary) is not necessarily an accurate predictor of the type of response; indeed, similar distribution patterns to this one are found in the open questions which are discussed in Section 5.4.

Table 5.3: Unique Responses & No. of Respondents

| Unique Response                                 | No. of respondents |
|---|--------------------|
| no  | 19                 |
| cardiac output                                  | 5                  |
| stroke volume                                   | 5                  |
| heart rate                                      | 4                  |
| resistance                                      | 4                  |
| venous return                                   | 4                  |
| blood pressure                                  | 3                  |
| pressure and resistance                         | 3                  |
| resistance and pressure                         | 3                  |
| sympathetic stimulation and ventricular filling | 3                  |
| ?   | 2                  |
| compliance                                      | 2                  |
| force   | 2                  |
| muscle  | 2                  |
| nerve innervation                               | 2                  |
| pressure  | 2                  |
| ventricular volume                              | 2                  |
|   | 2                  |
| a   | 1                  |

Once again, a number of students (19/229 or  $\approx 8\%$ ) have answered the question as if it were a binary question and have indicated that they cannot think of any factors which affect cardiac contractility.

### 5.3.1.3 Selecting classes

The choice of classes for multi-part questions like the current question is straightforward; each item or part-answer expected in the response forms the basis for a class centred around a single concept or term. In this case, the factors which affect cardiac

contractility are sympathetic stimulation, ventricular filling and cardiac muscle mass and each of these represents a binary class (the factor is either present or not). As noted in the previous chapter, Section 4.5.2, multi-part questions are best dealt with by chaining together a series of binary classifiers (called a multi-binary classifier), one for each part of the question, rather than including all possible classes of response within a single multilabel classifier.

Nevertheless there are potential drawbacks to this approach. The class-imbalance problem, which was introduced in Section 4.5.2, is one. The inability to deal with specific misconceptions outside of the presence or absence of the three required binary factors is another. For example, it is clear from the training set data that cardiac output, stroke volume, heart rate and blood pressure amongst others, are viewed by a number of students as factors which affect cardiac contractility. The student conceptions behind these are likely to be worth exploring in partnership with students. There are subtle and complex inter-relationships between these variables and the factors sought in answer to the current question. In theory, there is no reason why additional binary classes could not be added to capture these conceptions using the same heuristic as that described for open questions in Section 5.4.1.3. In the current study however, the classes chosen represented just the three factors which affect cardiac contractility.

#### 5.3.1.4 Feature selection, bench-testing and script revisions

The labels used in this case reflect the factors which the question is designed to elicit from students (sympathetic stimulation or *symp*, ventricular filling or *vfill* and muscle mass or *mass*). The next step is to decide which are the most appropriate features for training each of the binary classifiers. Word stems without stop-words turned out to be the best combination when testing on unseen data.

Accuracy is unhelpful as a performance metric for multi-binary classifiers since partial agreement (for example, 2 labels matching and the third different) would be counted as a non-match. For this reason, in addition to precision and recall, MASI-distance, which provides a measure of the degree of overlap between two sets of labels (Passonneau, 2006) is used. MASI distance ranges from 0 to 1 where 0 indicates a complete match between labels (for example, test set = {symp, vfill, mass}, reference set = {symp, vfill, mass}) and 1 indicates no overlap (for example, test set = {}, reference set = {vfill, mass}).

The results of bench-testing are summarised below in Table 5.4.

Table 5.4: Inotropic Factors Classifier Bench-Test Metrics

| <i>Name</i>              | <i>Test Set</i> |            | <i>Unseen</i>    |               |             |
|--------------------------|-----------------|------------|------------------|---------------|-------------|
|                          | $\mu$           | $\sigma^2$ | <i>Precision</i> | <i>Recall</i> | <i>MASI</i> |
| mass                     | 0.81            | 0.24       | 1.00             | 0.67          |             |
| symp                     | 0.88            | 0.08       | 0.50             | 1.00          |             |
| vfill                    | 0.74            | 0.16       | 0.71             | 0.83          |             |
| <b>inotropic factors</b> |                 |            |                  |               | <b>0.23</b> |

### 5.3.2 Multi-part Questions Summary

From the results summarised above it is clear that there is plenty of room for improvement in terms of classifier performance. The accuracies obtained for each of the binary classifiers range between 74% - 88% and two out of three of the binary classifiers show marked variability in terms of their accuracy (*mass* and *vfill*). The MASI distance suggests that around 23% of the time there will be a mismatch in terms of label overlap between test and reference data. Low precision, or a high rate of false positives, is particularly marked for the *symp* label. Part of the issue here is likely to be a lack of suitable training data. For example, there were only 8 examples of references to sympathetic stimulation in the training set which probably accounts for the low precision value (that is, of the samples in the unseen data set which were assigned to the *symp* label, 50% of these were incorrect). An option, would have been to hand-craft some additional examples for the training set; this process was described in the case of the binary classifier in Section 5.2.1.3. For this case however, the decision was taken not to do this and to simply wait and see how the classifier performed in the in-class evaluation.

## 5.4 Open Questions

In the context of the tutorial dialogue an open question is one which cannot be answered with just a yes or no, or with a statement of simple facts. An open question is one which requires some kind of development of ideas; for example making an inference, justifying a choice, applying a principle and so on. The value of open questions in formative assessment is well established (for example, see Black and William, 2004); however creating a good open question is not easy (see Frederiksen, 1984, p.194 for a

brief review) and requires a shift in teaching approach from presenting information to encouraging the exploration of ideas.

Where changes to questioning practices have been made, students have become more active as participants and have come to realize that learning may depend less on their capacity to spot the right answer and more on their readiness to express and discuss their own understanding. The teachers began to realize that in the past they had not planned and conducted classroom dialogue in ways that might help students to learn, and that they had to shift in their role, from presenters of content to leaders of an exploration that develops ideas in which all students are involved. (Black and William, 2004, p.27)

The decision to include open questions in a tutorial dialogue system presents a particular challenge for the system developer. In the face of a small number but potentially wide variety of training examples the likelihood of creating a high performing classifier seems remote. Nevertheless, as noted above, the failure to include at least some open questions in the tutorial dialogue runs counter to well established principles of formative assessment. Furthermore, at least one open question in the form of a mini-essay is included in the final examination for HUBS students and at least part of the original motivation for this study was to see whether technology could be used to help prepare students for this.

With these constraints in mind, all classifiers, not just classifiers for open questions, were built in conjunction with script revisions. The rationale for this was that if the classifier regularly fails on particular classes then either the classes need to be revisited (which would in turn require script revision) or appropriate action should be built into the script to avoid students becoming frustrated and abandoning the tutorial altogether.

Failure to classify an incorrect or incomplete response can usually be finessed by careful scripting. For example, if a classifier fails to classify a student response, then the response would be checked to see whether it could be classified by one of the three generic classifiers (question, dont-know and dont-understand). Assuming each of these (correctly) fails to classify the response then the default position in the script should be to recover the situation with some grace: in this case responding with, ‘Sorry. I’m not understanding you. Let’s try a yes-no question.’ By contrast, if a student has entered a good response to a question and incorrectly receives feedback their answer

is incomplete or just plain wrong and to ‘Have another go’ they could be forgiven for becoming confused. If they then try again and receive a similar response, some students become frustrated. Evidence of this type of frustration was found during pre-experiment trials with staff and second year students. A couple of representative direct quotes from students responding to this situation are given below.

“why dont you listen to me? do i mean nothing to you? the time we’ve spent together? it means nothing?!!??!”

“fuck you”

Indirect evidence of the impact of student frustration from poor ‘tutor’ understanding is provided by the number of students who abandoned the tutor at the ‘What is the pulse?’ question in the 2010 version of the dialogue where all student responses failed to be classified. For this reason, the first open question case study which is described in Section 5.4.1 relates to the open question, ‘What is the pulse?’ and includes a detailed description of the process of creating classes of student responses.

### **5.4.1 Case study: What is the pulse?**

The question ‘What is the pulse?’ follows a brief introduction to the context of the tutorial, a review of the physiological term ‘heart rate’ and a practical question which asks students to state how they would establish the heart rate of a human. The pulse is a palpable sign of an important variable in cardiovascular physiology, the heart rate. Asking students to think about the nature of the pulse is designed as a segue into thinking about several important features of the cardiovascular system as a whole; the heart or pump, the behaviour of fluid in vessels connected to the pump and the nature of the vessels themselves.

#### **5.4.1.1 Objective of the question**

An explanation of the pulse should ideally incorporate two key ideas: first, how the pulse is produced and second, how it travels from the point at which it was produced to distal locations, for example the wrist. A model answer and one which is used in the tutorial dialogue script is as follows:

The pulse is a pressure wave or a pulsatile wave generated by the difference between systolic and diastolic pressures in the aorta.

This model answer incorporates technical terms (systolic, diastolic) and uses some quite domain specific language (pressure wave). Physiology students may or may not have come to grips with the language of physiology but could still produce a thoroughly satisfactory answer to the question, ‘What is the pulse?’ as long as it encapsulates the key ideas. The following are a couple of ‘correct’ student responses to illustrate this point.

Change in pressure as the heart contracts and expels blood into the arteries.

The rhythmic expansion and recoil of the arteries resulting from a wave of pressure produced by contraction of the left ventricle of the heart.

#### 5.4.1.2 Training dataset description

The examples in the previous section also serve to illustrate the difficulty of building a classifier which would reliably detect acceptable responses to this question. For example, apart from the word ‘pressure’ there are no non-stopwords in common between all three responses (model response and student responses). Wave, heart and the stems ‘arter-’ and ‘contract-’ occur in two out of the three examples but beyond that there is little to go on. Looking at the structure of the responses, the first student response adopts a perspective similar to the model answer in that it begins with the notion of pressure. By contrast, the second student response takes the palpable pulse as the starting point and works backwards. Either position is fine and there are certainly many more ways to think about and explain ‘the pulse’.

With an open question one would not expect large numbers of identical answers. Nevertheless, common terms or phrases may become apparent and help with the process of both selecting classes of response, as well as feature selection. A plot of the training data (changed to all lower-case) for ‘What is the pulse?’ is shown in Figure 5.3.

As expected, the majority of responses (407) are, on the face of it, unique. The most common textual responses (4 students each) are, ‘difference between systolic and diastolic pressure’, ‘heart beat’ and ‘heart rate’ (i.e. 3 unique responses). There are 8 unique textual responses with 3 students each and 19 unique textual responses with 2 students each. These top ranking unique responses along with the number of students who gave them are shown in Table 5.5.

Inspection of this list reveals that a number of responses are semantically equivalent. For example, ‘?’, ‘not sure’ and ‘i don’t know’ presumably all mean something like ‘I don’t know the answer to your question’ and effectively collapse three unique responses

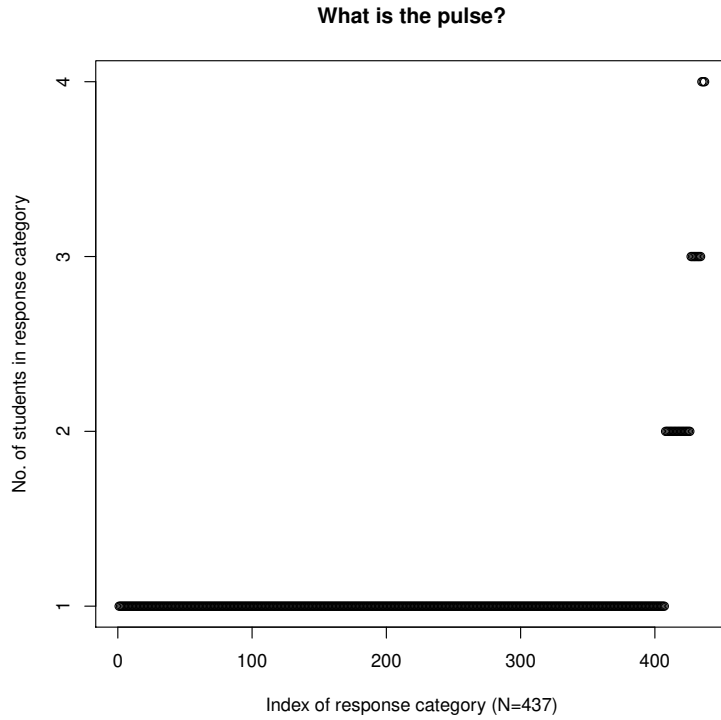


Figure 5.3: Frequency of unique textual responses.

down into one. This insight provides the starting point for creating classes of response for ‘What is the pulse?’ and is described in the next section.

#### 5.4.1.3 Selecting classes

A review of the top responses reveals that, ‘difference between systolic and diastolic pressure’ is semantically equivalent to at least 7 other responses. Adding the counts for each of these shows that 20 students out of 72 (27%) in this sub-sample felt that the pulse had something to do with the difference in pressures between systole and diastole.

Proceeding in this manner through the list of responses produces the set of response classes detailed in Table 5.6. The *Response Class* column contains the labels assigned to each class and as far as possible these attempt to encapsulate the meaning expressed in each class. For example, *pulse-pressure* is used to describe any response where the primary meaning is the difference between systolic and diastolic pressures. The label, *heart-rate* encapsulates responses which primarily refer to the heart beating or the pulse rate, and so on. The one exception is the label *not-classified* which is simply

Table 5.5: Unique Responses &amp; No. of Respondents

| Unique Response  | No. of respondents |
|--|--------------------|
| difference between systolic and diastolic pressure                   | 4                  |
| heart beat   | 4                  |
| ?  | 3                  |
| blood flow   | 3                  |
| d  | 3                  |
| pulse  | 3                  |
| the difference between diastolic and systolic pressures in the aorta | 3                  |
| the difference between systolic and diastolic pressure               | 3                  |
| the heartbeat  | 3                  |
| the number of beats per minute                                       | 3                  |
| arterial blood pressure  | 2                  |
| beats per minute   | 2                  |
| blood pumping  | 2                  |
| blood pumping through arteries                                       | 2                  |
| difference between systolic pressure and diastolic pressure          | 2                  |
| difference in systolic and diastolic pressure                        | 2                  |
| flow of blood  | 2                  |
| heart contractions   | 2                  |
| i don t know   | 2                  |
| not sure   | 2                  |
| pressure   | 2                  |
| s  | 2                  |
| the cardiac output   | 2                  |
| the closing of the valves in the heart                               | 2                  |
| the difference between systole and diastole                          | 2                  |
| the difference between systolic and diastolic pressures in the aorta | 2                  |
| the difference between the diastolic and systolic pressure           | 2                  |
| the heart beating  | 2                  |
| [space character]  | 2                  |

used as a catch-all for responses which are not used to train the classifier; single letter and single word responses and random characters are assigned to this label. It is worth

noting that in the top 29 responses (that is, >1 student per class) there is no class assigned for a correct response since there was no correct response shared by more than 1 student. In other words, this was a hard question for students to answer.

Table 5.6: Initial ‘What is the pulse?’ response classes

| Response Class        | No. Responses in Class | Total Students in Class |
|-----------------------|------------------------|-------------------------|
| <i>pulse-pressure</i> | 8                      | 20                      |
| <i>heart-rate</i>     | 8                      | 23                      |
| <i>blood-flow</i>     | 4                      | 9                       |
| <i>blood-pressure</i> | 2                      | 4                       |
| <i>dont-know</i>      | 3                      | 7                       |
| <i>cardiac-output</i> | 1                      | 2                       |
| <i>cardiac-valve</i>  | 1                      | 2                       |
| <i>non-class</i>      | 2                      | 5                       |
| <b>Totals</b>         | 29                     | 72                      |

The next step involves using the class labels as a starting point, manually going through the remaining student responses and assigning labels to them. However, the first difficulty with the process arises when a student response does not seem to fit a pre-existing label. Does each new response deserve its own label? How close to an existing label is close enough? Given that the derived classes will be used to train a classifier it is clear that too many labels for the size of training set will hardly result in useful classification. Furthermore, if it is intended that each class receives a response in the dialogue, an excess of classes will produce a rapid explosion in dialogue script size and complexity.

In the end, the process of assigning student responses to classes was very much one of judgement and interpretation. Neither the time nor resource was available to undertake an in-depth analysis of the by-hand classification process so a simple least effort for maximum return ‘greedy’ heuristic was employed instead. The heuristic adopted can be described as follows:

1. Starting from the labels in Table 4.1, for each new student response see if it can readily be assigned to an existing label.
2. If it can, assign the label. If it cannot then ask whether it demonstrates

- an important new misconception.
3. If it does reveal an important misconception, assign it a new label.
  4. If it does not, see which of the existing classes it is closest to and ask whether it can be fitted into an existing class without renaming the class.
  5. If yes, assign the response to that class. If no, rename the class to reflect the inclusion of a broader range of responses.
  6. As responses are assigned to classes, periodically check how many response classes there are. If there are too many (> 4 or 5), review whether some classes can be collapsed together.

The choice of no more than 4 or 5 classes is based only on an intuition that this should be manageable for a) the classifier given the size of the training data set and b) the script so that it does not become unwieldy. Whether this intuition is valid is put to the test when it comes to building the classifier and finalising the script. After working through the heuristic above for all student responses (N=480) the final classification scheme shown in Table 5.7 was adopted only after several iterations of feature selection and bench-testing demonstrated that the classification produced reasonable results on unseen data (described below).

There are a total of 5 classes in Table 5.7, not including the generic classes (dont-know, question and dont-understand) which are discussed in Section 4. A number of the classes from the original classification scheme shown in Table 5.6 have been collapsed into a single class called '*incomplete*'. The original class, *heart-rate* has been extended to a broader class, *self-ref* which describes any response that seems to repeat or refer to the previous question in the script, 'How would you check someone's heart rate?' or which simply states a figure for a pulse rate. *cardiac-valve* and *arterial-movt* both reflect answers which revealed infrequent but important misconceptions relating to the pulse. The first, that the pulse is somehow produced by closure of heart valves and the second, that the pulse is produced by the ability of arteries to expand and contract.

Employing the heuristic requires some familiarity with what students might mean by their responses; in other words people doing the classification (markers) should either be teachers on the course or have some familiarity with the subject matter either as a teacher or former student. It is also important to consider whether the classification scheme adopted is reliable; if two markers assign given classes to a set of student responses, will their classifications be reasonably close?

Table 5.7: Final ‘What is the pulse?’ response classes

| <b>Response Class</b> | <b>No. Responses in Class</b> |
|-----------------------|-------------------------------|
| <i>correct</i>        | 36                            |
| <i>incomplete</i>     | 142                           |
| <i>self-ref</i>       | 124                           |
| <i>cardiac-valve</i>  | 20                            |
| <i>arterial-movt</i>  | 18                            |
| <i>dont-know</i>      | 17                            |
| <i>question</i>       | 3                             |
| <i>non-class</i>      | 120                           |
| <b>Totals</b>         | 480                           |

From a summative assessment perspective, a systematic and repeatable approach to creating a classification scheme is desirable. Nevertheless, in many real-life formative assessment situations and in tutorials involving real people the degree of reliability achieved in practice is likely to be rather lower than the levels of inter-rater reliability frequently reported in research studies which involve content analysis. The reason for this is that researchers working in a shared context tend to establish shared meanings with the effect that reliability measures can become inflated (see for example Stemler, 2001, p.141).

For the purpose of this study classification reliability is less of an issue since our interest is primarily the performance of the classifier compared to a knowledgeable human rater. Nonetheless, inter-rater reliability (IRR) between 2 raters was checked for the two open question case studies. One was the author and the other was a member of the HUBS teaching staff who had not been involved in developing or discussing the classification of responses. Ideally, it would have been useful to check IRR for all questions but this was not possible in the context of the current project, so IRR was checked only on the hardest problems: the two open question case studies. The classification scheme devised above was provided to both raters with a very brief explanatory email of the classification scheme and a sample of 100 student responses to the question. An un-weighted inter-rater reliability measure was calculated using the R statistical software package (R Core Team, 2012). The resulting Cohen’s Kappa score of 0.21 indicates fair agreement between raters according to the Landis and Koch (1977) benchmark.

However, in practice what this means is that agreement between the raters was only a little better than chance alone (score = 0). Whether this indicates that the training data for the classifier provides very little information, or that the second annotator had not adequately understood the classification scheme being applied, is an open question which arises again in Section 5.4.2.3.

In keeping with existing practice in the HUBS course it is envisaged that if the system is adopted by HUBS, or any other course, the training set classification scheme should be developed and the process for its use agreed upon by all teaching staff involved in the course. It is anticipated that this would yield a somewhat higher inter-rater reliability measure.

#### 5.4.1.4 An initial revision of the classifier: adjustment of feature sets and entropy threshold

The classification of student responses described in 5.4.1.3 provides a label for each student response in the dialogue. The next step is to decide which are the most appropriate features of student responses for use in training a multilabel classifier.

Following the procedure detailed in Chapter 4, the best featureset combination, when tested on unseen data, turned out to be word stems in combination with the exclusion of stopwords, with an overall accuracy on unseen data of 60%. The confusion matrix is shown in Figure 5.4.

|                | Arterial Movt | Correct | Incomplete | Self-ref | Valves | Not classified |
|----------------|---------------|---------|------------|----------|--------|----------------|
| Arterial-movt  |               |         |            |          |        |                |
| Correct        |               |         | 1          |          |        | 1              |
| Incomplete     | 1             |         | 7          |          |        | 1              |
| Self-ref       |               |         |            | 2        |        | 6              |
| Valves         |               |         |            |          | 1      |                |
| Not classified |               |         |            |          |        |                |

*Row = Reference, Column = Classifier, NC = not classified*

Figure 5.4: Confusion Matrix,  $e = 1.0$

After examining the confusion matrix, along with the probability distributions of label likelihood for each of the unseen data samples, an entropy value of 1.5 was chosen in order to reduce the number of failed classifications (NC). Increasing the entropy value to 1.5 had the effect of increasing the accuracy for the *self-ref* label and did not

increase the number of incorrect classifications. The overall accuracy increased to 75% (See Figure 5.5).

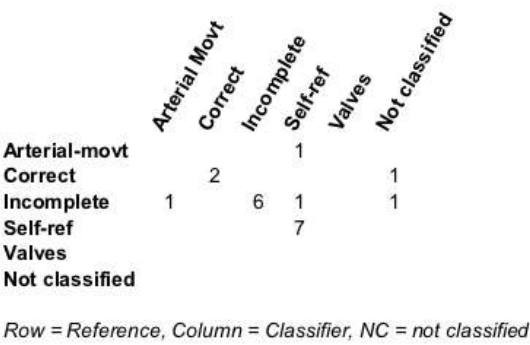


Figure 5.5: Confusion Matrix, e = 1.5

The final step before accepting this classifier involved running a series of manual checks to ensure that:

1. A series of random characters and nonsense sentences are not classified,
2. Question, dont-know and dont-understand responses are not classified,
3. A random series of correct answers, in addition to the scripted model response, are positively classified as correct
4. A random statement for each remaining class is ‘mostly’ correctly classified.

At this point it became clear that correct responses were seldom classified correctly and this even included the model response. From an end-user or student perspective this is not good as discussed at the beginning of this section.

#### 5.4.1.5 A second revision of the classifier: high-information words

A potential strategy for improving classifier performance on the training set involves restricting the featureset to high-information words. The general approach involves filtering out words that provide little or no class-specific information. This can be thought of as a more sophisticated version of filtering out noise by eliminating stop words. Pseudocode for the algorithm is given in Figure 5.6 below.

```

Count the total number of words in the training set -> total_words
Count how many responses in the training set -> total_responses

For each labelled response in the training set count how many responses
have this label -> count_label_i

For each word in each response, count how many times the word appears:
in the class -> count_word_in_label_i
in the training set -> count_word_in_total-words
word_freq -> count_word_in_total_words/total_words
word_freq_in_class -> count_word_in_label_i/count_label_i

If p(word_freq_in_class) > p(word_freq): -- score for example using
    -- Chi-Square test
add word to the set of high frequency words

```

Figure 5.6: High-Information Word Function Pseudocode

Python code which provides exactly this function (Perkins (2010), page 187) was added to the classifier testbed and the resulting set of high information words (N=152) were extracted as features from the training set.

When testing the classifier using high-information words as features, the model correct answer was now classified correctly; however many other potential correct answers still were not. Examination of the training set pointed to the likely source of the problem; there were insufficient correct answers in the training set relative to the *incomplete* class and there was little distinction between high-information words in both these classes. In response to this, the size of the training set was increased by adding an additional 417 training items (note that these were drawn from a mix of first year students, 2nd year students and teaching staff responses which were collected during system development and pre-release tests). The resulting confusion matrix is shown in Figure 5.7.

The average accuracy of this classifier on unseen data was still only 75%. While this figure compares favourably with a baseline of only 44% accuracy, which is obtained by assigning the most common label to all inputs, there are clearly further improvements to be made; whether these are around the choice of class labels, choice of featuresets or

|                | Arterial Movt | Correct | Incomplete | Self-ref | Valves | Not classified |
|----------------|---------------|---------|------------|----------|--------|----------------|
| Arterial-movt  |               |         |            |          |        |                |
| Correct        |               | 1       | 1          |          |        |                |
| Incomplete     |               |         | 5          |          |        | 1              |
| Self-ref       |               |         | 1          | 7        |        | 1              |
| Valves         |               |         |            |          | 2      |                |
| Not classified |               |         |            |          |        |                |

Row = Reference, Column = Classifier, NC = not classified

Figure 5.7: Final Confusion Matrix,  $e = 1.5$

with increased amounts of training data from a stable cohort remains to be seen. At the very least, it seems there are limits to how well you can classify an open question.

## 5.4.2 Case study: Can you describe what is meant by contractility?

‘Can you describe what is meant by contractility?’ is another open question and is asked immediately following the assertion, ‘Inotropic state is a term that is sometimes used to describe the contractility of the heart.’ In this way a technical term, inotropic state, is introduced and a more approachable synonym, contractility, is offered to help students to explain the term. Nevertheless, the notion of contractility itself can be a little elusive. What exactly does it mean?

### 5.4.2.1 Objective of the question

The goal of the question is to encourage students to think about contractility in physiological terms. Contractility is the intrinsic ability of heart muscle to generate pressure or force so that it can fulfill its primary function as a pump and maintain cardiac output. Coming to grips with cardiac contractility is fundamental to understanding the role of sympathetic stimulation and also some drugs which affect contractility.

The model answer which is provided in the script is, ‘The force generated by the heart muscle during contraction’. The intention here was to focus on the intrinsic ability of the heart to contract and generate a force rather than just a simple restatement of contractility illustrated by the following student responses:

The amount of contraction the heart can produce.

How well the heart contracts.

The idea that cardiac contractility is independent of other factors which may also increase the force of contraction, such as ventricular filling, is not dealt with directly in this question.

#### 5.4.2.2 Training dataset description

The same approach described above in Section 5.4.1.2 is used to get a picture of the overall training data set. The top 18 unique textual responses are listed in Table 5.8 and responses plotted against the number of students providing a given response is illustrated in Figure 5.8. The distribution of unique responses is very similar to the previous open question, ‘What is the pulse?’

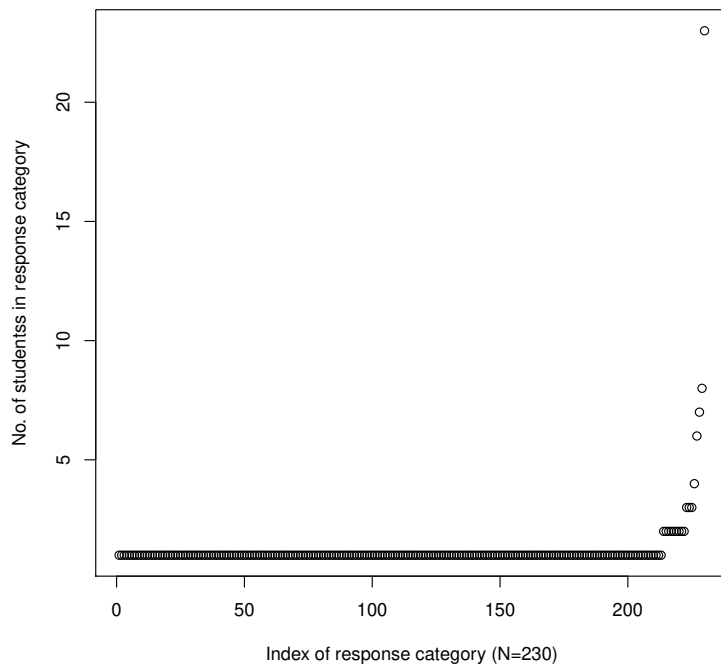


Figure 5.8: Frequency of responses by category.

### 5.4.2.3 Selecting classes

The first obvious feature of student responses is that only 24/288 or around 8% of students answered the question as a binary one rather than an open question. ‘Can

Table 5.8: Unique Responses &amp; No. of Respondents

| Unique Response   | No. of respondents |
|---|--------------------|
| no  | 23                 |
| ability to contract   | 8                  |
| the ability of the heart to contract                              | 7                  |
| how much the heart can contract                                   | 4                  |
| contraction of the heart  | 3                  |
| how much it can contract  | 3                  |
| how much the heart contracts                                      | 3                  |
| ?   | 2                  |
| ability of muscle cell to contract or shorten to produce movement | 2                  |
| how hard the heart contracts                                      | 2                  |
| how much it can contract ?  | 2                  |
| how well it contracts   | 2                  |
| s   | 2                  |
| strength of contraction   | 2                  |
| the force with which the heart contracts                          | 2                  |
| the strength of contraction                                       | 2                  |
| yes   | 1                  |

you describe what is meant by contractility’ can be answered with a simple yes (I can describe contractility) or no (I cannot). This is a good example of conversational implicature introduced in Section 3.2.2. Students seem willing to buy in to the formative purpose of the tutorial which is to try to explain contractility if they can. For this reason, a binary negative response was interpreted as equivalent to, ‘I don’t know how to answer your question’ and all ‘no’ responses were assigned to the generic *dont-know* class (see Table 5.9).

Another feature compared with ‘What is the pulse?’ is that there were more students responding in identical ways. Nevertheless, while the overall distribution pattern is similar it was still indicative of wide variation in the nature of responses to different open questions. In other words, some open questions appear to be more open than others.

Proceeding in the same manner for dealing with responses to the question, ‘What is the pulse?’ and using the heuristic described in Section 5.4.1.3, a set of response classes

was produced (Table 5.9). Given the relatively poor results obtained when using more classes for the question, ‘What is the pulse?’ a decision was taken with this question to put all partially correct answers into a single class called *incomplete*.

Table 5.9: ‘Can you describe what is meant by contractility?’ re-sponse classes

| Response Class    | No. Responses in Class |
|-------------------|------------------------|
| <i>correct</i>    | 63                     |
| <i>incomplete</i> | 130                    |
| <i>dont-know</i>  | 33                     |
| <i>non-class</i>  | 62                     |
| <b>Totals</b>     | 288                    |

Inter-rater reliability between two raters was checked for this question too. Again, the classification scheme for responses to the question was provided to both raters with a very brief explanatory email of the classification scheme and a sample of 100 student responses to the question. An unweighted inter-rater reliability measure was calculated using the R statistical software package (R Core Team, 2012). The resulting Cohen’s Kappa score of 0.32 indicates fair agreement between raters according to the Landis and Koch (1977) benchmark. This is a small improvement on the previous case-study question. Nevertheless, in practical terms, it is still relatively low agreement. One possible reason for this is because insufficient information relating to categorisation was shared between the raters. It also suggests that classifier performance is unlikely to be very good if the categories assigned by humans are vague and subject to wide interpretation.

#### 5.4.2.4 Feature selection, bench-testing and script revisions

As in the first open question case-study, the classification of student responses provides a label for each student response in the dialogue. The next step is to decide which are the most appropriate features of student responses for use in training a multilabel classifier. This time however, there are only two custom classes to deal with and only one generic class, *dont-know*. In theory, this should make the job of the classifier a little easier.

Testing on unseen data, it turned out that the best featureset combination for this

classifier was word stems and no stopwords, the overall accuracy was 90% and the confusion matrix is given below in Fig 5.9:

|                | Correct | Incomplete | Not classified |
|----------------|---------|------------|----------------|
| Correct        | 3       |            |                |
| Incomplete     | 2       | 15         |                |
| Not classified |         |            |                |

Row = Reference, Column = Classifier

Figure 5.9: Contractility Confusion Matrix,  $e = 1.0$

As with the previous case-study, a series of manual checks was run to ensure that responses which should be passed to the generic classifiers are not classified and that correct responses were correctly classified. This was the case. However from the confusion matrix it was also clear that some incomplete responses were being incorrectly classified as correct. In this situation, it is preferable to fail classification rather than to incorrectly classify if possible. Examination of the entropy values showed in both cases that they were quite high ( $E = 0.95$  and  $E = 0.90$ ) with  $e$  set at 1.0 so the classifier was trained again, this time setting  $e = 0.8$ . The accuracy, again on unseen data, remained at 90% and the resulting confusion matrix is shown in Figure 5.10.

In practice, there is probably little difference between these two versions (there is still one incomplete response being classified as correct) nevertheless, the classifier with  $e = 0.8$  was the one selected to go into the experimental system.

### 5.4.3 Open Question Summary

Given that there was only one opportunity in the year to run the experiment to coincide with cardiovascular coursework, it was decided to use the ‘*What is the pulse?*’ and ‘*Describe what is meant by contractility*’ classifiers ‘as-is’ and see what happened in practice. While the overall accuracies of these classifiers (75% and 90%) in principle seems not too bad, this is likely to be far from the case if you happen to be the 1 in 4, or 1 in 10 students being misunderstood! The script would not allow more than two

|                       |                |                   |                       |
|-----------------------|----------------|-------------------|-----------------------|
|                       | <b>Correct</b> | <b>Incomplete</b> | <b>Not classified</b> |
| <b>Correct</b>        | 8              |                   | 1                     |
| <b>Incomplete</b>     | 1              | 10                |                       |
| <b>Not classified</b> |                |                   |                       |

*Row = Reference, Column = Classifier*

Figure 5.10: Contractility Confusion Matrix,  $e = 0.8$ .

attempts at the same question before moving on and then reframed the question as a binary one. The hope was that this would go some way towards reducing student frustration level while not completely giving up on eliciting a student response. Equally, it was hoped that the provision of appropriate and relevant feedback to students whose responses were classified correctly would result in benefits.

If nothing else, both open question case-studies illustrate some of the difficulties associated with classifying open questions and describe some strategies for reducing the impact of incorrect classification in the dialogue.

The next section provides a complete summary of the performance, on held-out data, of all classifiers which are used in the final system.

#### 5.4.4 Classifier bench evaluation results

The student response data which was used to assess the accuracy of all classifiers prior to release of the system to students is drawn from the same pool of data which was used to train the classifiers and make the initial assessments of their performance. However, as described in Section 4.5.3, data used for the evaluations and summarised in this section, were held-out from the data used to train the classifiers. That is, the evaluation is on data which was previously unseen by the classifiers but from the same student cohort. Evaluation results for each of the case study classifiers, using data from a completely new student cohort and collected during in-class evaluation, is presented in Chapter 7.

Performance data for each multilabel classifier used in the main experimental eval-

uation is summarised in Table 5.10. The most common feature used to train these classifiers was word stems with stop words removed and the choice of case-study classifiers presented in this chapter reflects this. Other features used, depending on the dialogue context, included text length, first word or words and high information words. Accuracy for these 27 classifiers ranged from 0.75 to 1.00 with the mean value at 0.93. The number of labels ranges from 2 (15 classifiers) to 7 (1 classifier), and the remaining 11 classifiers have between 3 and 5 labels. Overall, fewer labels tends to support higher accuracy on unseen data but this is not the case where there are a wide range of ways of expressing the same meaning assigned to a single label. For example, the question represented by the name, *vein-why* ‘Can you explain why you cannot feel a pulse in someone’s vein?’, has only 2 labels but relatively low accuracy scores on both test data and previously unseen data. The reason for this is that the question for this dialogue context is open-ended. There were two main classes of student response. The first was a correct response, the second was a plausible response but omits the main reason why a pulse cannot be felt. It is clear that within both classes there are multiple ways of expressing either meaning. These issues and some options for improving classifier performance are discussed in more detail in Chapter 8.

Performance data for each of the three generic binary classifiers is summarised in Table 5.11. The best results in terms of accuracy, precision and recall were obtained with the *question* classifier. The training data for each of these classifiers is drawn from every dialogue context where a question, ‘I don’t know’ or ‘I don’t understand the question’ type response was entered by students. This provided a good range of responses to train the *question* classifier and the *dont-know* classifier. However, there was only one labelled training example for ‘I don’t understand the question’. The accuracy, precision and recall for the *dont-understand* classifier was therefore highly suspect. To try to improve the situation, the training set was weighted with 10 additional handcrafted *dont-understand* responses. When retrained using this hand-crafted set, the accuracy dropped slightly to 0.90, the precision dropped to 0.33 and the recall at 0.89, remained high.

Finally, performance data for the three multi-binary classifiers is summarised in Table 5.12. As described in detail in Chapter 5, each of these classifiers returns a list of labels which apply to a given response. In addition to accuracy, precision and recall for the individual binary classifiers, MASI distance is used as an overall performance metric (0 indicates complete agreement between the label lists in the reference and test sets and 1 indicates no overlap).

Table 5.10: Multilabel classifier metrics (refer Appendix A.4 for an index of classifier names and questions)

| <i>Name</i>        | <i>Labels</i> | <i>Test Set</i> |            | <i>Unseen</i>   |
|--------------------|---------------|-----------------|------------|-----------------|
|                    |               | $\mu$           | $\sigma^2$ | <i>Accuracy</i> |
| baro-parasymp      | 2             | 0.98            | 0.019      | 1.0             |
| baro-resp-inc      | 3             | 0.99            | 0.017      | 1.0             |
| baro-resp-inc-symp | 2             | 0.99            | 0.019      | 1.0             |
| baro-response      | 3             | 0.99            | 0.009      | 1.0             |
| bp-syst            | 2             | 0.97            | 0.026      | 1.0             |
| circ-design        | 2             | 0.99            | 0.012      | 1.0             |
| circ-pulm          | 2             | 0.99            | 0.005      | 1.0             |
| circ-sys           | 4             | 0.96            | 0.017      | 1.0             |
| feedback-loop      | 3             | 0.87            | 0.158      | 1.0             |
| vein               | 4             | 0.96            | 0.022      | 1.0             |
| venous-valves      | 2             | 0.99            | 0.013      | 1.0             |
| work-heart         | 2             | 0.99            | 0.012      | 1.0             |
| yes-no             | 2             | 0.99            | 0.006      | 1.0             |
| baro-resp-inc-para | 2             | 0.96            | 0.033      | 0.95            |
| bp-diast           | 2             | 0.96            | 0.012      | 0.95            |
| baro-symp          | 2             | 0.88            | 0.178      | 0.95            |
| parasymp-dec       | 2             | 0.90            | 0.071      | 0.95            |
| same-flow          | 2             | 0.97            | 0.037      | 0.95            |
| adult-sv           | 5             | 0.89            | 0.048      | 0.90            |
| inotropic-state    | 2             | 0.92            | 0.022      | 0.90            |
| adult-co           | 4             | 0.92            | 0.040      | 0.85            |
| artery-wall        | 7             | 0.89            | 0.050      | 0.85            |
| baro-loc           | 4             | 0.82            | 0.105      | 0.85            |
| check-hr           | 3             | 0.93            | 0.030      | 0.85            |
| vein-why           | 2             | 0.90            | 0.041      | 0.85            |
| what-pulse         | 5             | 0.78            | 0.048      | 0.75            |
| bp-map             | 3             | 0.73            | 0.117      | 0.75            |

Table 5.11: Generic binary classifier metrics

| <i>Name</i>                           | <i>Test Set</i> |            | <i>Unseen</i>   |                  |               |
|---------------------------------------|-----------------|------------|-----------------|------------------|---------------|
|                                       | $\mu$           | $\sigma^2$ | <i>Accuracy</i> | <i>Precision</i> | <i>Recall</i> |
| question                              | 0.87            | 0.17       | 0.95            | 0.67             | 0.94          |
| dont-know                             | 0.82            | 0.30       | 0.90            | 0.89             | 0.33          |
| dont-understand                       | 0.99            | 0.006      | 1.0             | 1.00             | 1.00          |
| dont-understand<br>(with handcrafted) | 0.95            | 0.006      | 0.90            | 0.33             | 0.89          |

Table 5.12: Multi-binary classifier metrics (refer Appendix A.4 for an index of classifier names and questions)

| <i>Name</i>                    | <i>Test Set</i> |            | <i>Unseen</i>    |               |              |
|--------------------------------|-----------------|------------|------------------|---------------|--------------|
|                                | $\mu$           | $\sigma^2$ | <i>Precision</i> | <i>Recall</i> | <i>MA SI</i> |
| tpr                            | 0.83            | 0.14       | 0.70             | 0.88          |              |
| sv                             | 0.87            | 0.06       | 0.83             | 0.91          |              |
| hr                             | 0.88            | 0.07       | 0.64             | 0.88          |              |
| <b>ans-vars/symp-<br/>vars</b> |                 |            |                  |               | <b>0.039</b> |
| mass                           | 0.81            | 0.24       | 1.00             | 0.67          |              |
| symp                           | 0.88            | 0.08       | 0.50             | 1.00          |              |
| vfill                          | 0.74            | 0.16       | 0.71             | 0.83          |              |
| <b>inotropic factors</b>       |                 |            |                  |               | <b>0.23</b>  |

## 5.5 Summary

This chapter has presented five representative case studies drawn from the three different types of question posed in the dialogue script, namely binary, multi-part and open questions. In addition, bench test results on held-out data for all classifiers which were deployed in the final system, multilabel, binary and multi-binary, were summarised. It is clear from these case studies and bench test results that there are significant issues still to address, especially when classifying open questions. However, given a mean accuracy of 0.93 for the multilabel classifiers, a similar level of accuracy for the

binary classifiers and what seemed like reasonable performance for the multi-binary classifiers in terms of MASI distance (closer to 0 than to 1), the decision was taken to proceed with in-class evaluation. Early in-class evaluation results for each of the case-study classifiers are given in Chapter 7. The implications for further work and possible strategies to address the problems associated with classifying open questions are discussed in Chapter 8.

# Chapter 6

## An in-class evaluation of the tutorial system: description and evaluation methods

### 6.1 Introduction

This chapter describes the in-class evaluation conducted with student volunteers from HUBS 192 during a three week period in 2012. It is important to be clear about the intended goals for the evaluation of any educational intervention. Shute and Regian (1993) particularly highlight this in the case of intelligent tutoring systems. The goals of evaluating this dialogue system are twofold:

- Firstly, to evaluate the tutorial dialogue system (tutor) performance in terms of a) its ability to recognise and respond appropriately to student input, and b) the student experience of using the two different versions of the tutor.
- Secondly, to formally test a set of hypotheses involving student use of free-text and menu-based versions of the tutor.

The choice of in-class evaluation is consistent with the goals of evaluating the tutor. As described in Chapter 4, the design and development of the dialogue script was informed not only by the demands of the curriculum and practical constraints related to a large first year course, but significantly by student responses to initial dialogue questions. It makes sense therefore to evaluate the system in terms of student learning outcomes and student experiences of the system in a real class setting. As Woolf

(2008, p.189) points out, in-class evaluations also increase the opportunity to validate the results for a broader range of teaching and learning situations when compared to laboratory studies.

A side-effect of the in-class evaluation is that the existing corpus of student responses to cardiovascular homeostasis dialogue questions is substantially increased. This provides an additional pool of material for further training of classifiers as well as a resource for further NLP and educational research.

This chapter begins by presenting some background to the in-class evaluation of the new tutorial dialogue system, including obtaining ethical approval, in Section 6.2. Section 6.3 introduces the methods for evaluation of the text classification component of the system as well as the method for evaluating student experiences of the system. The methodology for conducting an experiment to compare free-text and menu-based versions of the system is covered in Section 6.4 including the preparation of pre- and post tests. Procedures for data collection and management are outlined in Section 6.4.2 and the final section, 6.4.3 covers the procedures for marking pre- and post-tests following student use of the new system.

## **6.2 Background and Ethics approval**

Extensive student input was required for the design and development of the tutorial dialogue system right through to conducting the in-class evaluation. For this reason ethics approval for the research project as a whole was obtained at the outset.

Students from the 1st year Health Sciences course ( $N \approx 1500$  in 2012) were asked to volunteer for the in-class evaluation. The first year Health Sciences course is a prerequisite for all professional health science programmes, such as Medicine, Dentistry, Pharmacy, and Physiotherapy. Entry into these programmes is highly competitive and is dependent, amongst other things, on students achieving excellent grades in their 1st year courses. None of the students from this cohort had previously been exposed to the dialogue system.

The researcher and the lecturer for the cardiovascular physiology section of the course introduced students to the evaluation and the dialogue system during the last lecture on the cardiovascular system. Because of the large number of students, the lecture was repeated once on the same day and both lecture sessions were simultaneously streamed by video-link to students in additional lecture theatres. A recording of the lecture was also available for students to access online from the following day. A

summary of the key points from the lecture presentation was incorporated into lecture handouts and these are included in Appendix C.1. The only incentive offered to students was the suggestion that taking part in the study would give them an opportunity to practice and develop their understanding of the cardiovascular section of the course by answering a series of questions related to the lectures they had received during the preceeding two weeks.

The course coordinator, lecturers and teaching fellows for the cardiovascular section of the course were all consulted regarding the evaluation design and in particular the experimental design required to compare the two versions of the tutor. The classifier evaluation, student experience evaluation and the experimental design are described in detail in the next section, 6.3.

## **6.3 Tutor Evaluation**

### **6.3.1 Classifier evaluation**

One way of evaluating the free-text version of the tutorial dialogue system is in terms of the performance of its classifiers, using the same measures (accuracy, precision and recall) as were used to evaluate them during system development. While the measures are the same, the inputs to the classifier are all unseen: the responses of the new cohort of students in the in-class evaluation are truly novel, and provide a good test of the classifiers' learning. The results of this evaluation are presented in Section 7.2.1.

In the menu-based version of the tutor 'classifying student responses' is trivial. A simple check to ensure that menu-items listed in the dialogue script are those presented to students and that the options chosen by students are correctly recorded is all that is needed. This check was completed as part of the dialogue system implementation.

In the free-text version, classifier accuracy, precision and recall on previously unseen data provide such a measure for multi-label and binary classifiers. As described in Chapter 5, Section 5.3.1.4, MASI distance, in addition to precision and recall is used for multi-binary classifiers.

### **6.3.2 Student experience evaluation**

A key component of evaluating an ITS which is intended to find practical utility in a classroom should involve developing an understanding of how students experience the ITS (for example, see Woolf, 2008, p.198). In addition to the opportunity to extract

student usage data, which was logged in the course of the in-class experiment and which is described in Section 6.4.2, a student perception questionnaire was sent to all students who used either version of the tutorial dialogue system during the experiment (refer Section 6.4).

Timing of the questionnaire administration was designed to coincide with the overall course evaluation run towards the end of the year and several weeks after students had actually used the system. This was done to avoid overburdening students who had already volunteered to take part in the experiment during a particularly busy period. The evaluation was conducted through the University Evaluation Research and Development Unit (ERDU) in the Higher Education Development Centre and in accordance with University policy, individual student responses were treated in confidence. The researcher could not therefore match individual student evaluation responses to individual tutor sessions; however individual responses by experimental group (free-text or menu-based) could be extracted from data supplied by the evaluations office. Finally, all student emails and other informal feedback received during the experimental period was recorded and saved for later analysis. The results of all these analyses are presented in Section 7.2.2.

## **6.4 Version Comparisons: Free-text vs Menu-based**

The in-class evaluation was also designed to explicitly compare two versions of the tutorial system, free text and menu-based, using an experimental design. The educational impacts of each of these systems were compared to a control condition, in which the students used no tutorial dialogue system at all, are of interest. This section describes the specific hypotheses tested, and the methods used.

### **6.4.1 Experimental Design**

Student volunteers from the class were randomly assigned to one of three groups. Each group was given a pre-test on the material covered in the cardiovascular section of their course, then either a tutorial intervention, or nothing, then a post-test covering the topics taught in the tutorial. Students in each group were therefore assigned to one of the following experimental conditions:

- A free-text condition where students complete a pre-test, then the free-text version of the tutorial dialogue, and conclude with an immediate post-test;

- a menu-based condition where they complete a pre-test, then the menu-selection version of the tutorial dialogue, followed by an immediate post-test, or
- a control condition where they simply complete pre- and post-tests, in that order.

Given the findings from the literature reported in Chapter 2, the specific hypotheses tested were:

- A.** Either tutorial intervention, free-text or menu-based, results in better post-test performance than no intervention (the control group);
- B.** Free-text input results in better post-test performance overall than MCQ, because construction of a textual response from scratch requires first, recall of the relevant material and second, active processing of this material. Construction of responses should therefore promote retention and/or understanding better than simply selecting from pre-constructed options;
- C.** Free-text tutorials lead to increased performance particularly on short-answer questions because of a practise or testing effect which was introduced in Section 2.2.7;
- D.** Menu-based tutorials lead to increased performance particularly on MCQ questions, also because of a practise or testing effect.

While this design is very similar to a randomised controlled trial, because students are self-assigning into the experimental group as a whole, it cannot strictly be considered a random assignment to experimental conditions from the sample population. In other words, selection bias is a potential threat to internal validity in this case. However, because student volunteers are randomly assigned to an experimental group this should not impact on the experimental conditions they are assigned to.

Performance in each condition was evaluated by:

1. Normalised score on an immediate post-test (conducted straight after the intervention or the pre-test for the control group) minus normalised score on pre-test. The immediate post-test comprised 7 MCQs and 7 short answer questions and along with their model answers, is available in Appendix D.
2. Normalised score on a delayed post-test comprising 3 MCQs, short-answer questions and a mini-essay question from the cardiovascular section of the final examination for the course. Questions and model answers are detailed in Appendix D.3.

Following the introduction to the experiment given to students during their last lecture on the cardiovascular system and described in Section 6.2, a URL or web-page link for the experiment was provided to all students. Prior to logging in via the University authentication system, students were also able to read the background to the study and experiment (the same material provided in the lecture handouts. See Appendix C.1). The URL was available to students at any time during a three-week period which began immediately following the lecture during which the experiment was introduced. The three-week period coincided with the laboratory and self-study periods assigned to the cardiovascular system and ended on the day of a summative multiple-choice terms test designed to examine student understanding of the cardiovascular section of the course.

Student volunteers were told that different styles of questions, short-answer and MCQ, might be used in different combinations and that not all students would receive the same style of questions. They were also told to allow 20-40 minutes to complete the questions. They could answer the questions by logging in to an online system at anytime during a three-week period which ran concurrently with their normal laboratory and self-paced study sessions on the cardiovascular system.

The pre- and immediate post-tests in each case consisted of equal numbers of MCQ and short-answers (3+3 for the pre-test and 7+7 for the post-test). The pre-test directly reflected material taught in the lectures which students had just received and the post-test reflected material explicitly covered in the tutorial dialogues. A senior teaching fellow for HUBS 192 developed the initial pre- and post-test questions. These were then discussed with one of the course lecturers and the researcher and the final questions agreed by all.

The delayed post-test questions were designed by the HUBS teaching staff. Questions suitable for use in the delayed post-test analysis were selected by the researcher on the basis that material covering these questions had been included in the tutorial dialogue.

## **6.4.2 Data Collection and Management**

All student interactions with the system in each experimental condition were recorded and logged to a relational database which was implemented using the PostgreSQL database system. Writing data to the database was managed by the tutorial dialogue web application. Specific information recorded (or calculated) in the database was:

- student identity,
- time and date of participation
- duration of participation in the experiment
- student responses to pre-test and immediate post-test questions
- student responses to each dialogue context (free-text and menu-based conditions only)

At the end of the experimental period only data from completed sessions (i.e. all of pre-test, post-test and tutorial condition completed) were extracted from the PostgreSQL database for analysis.

In addition, a marked up transcript for each student dialogue session in NPSCChat format was written out to an XML file and labelled with a timestamp and a user id by the Dialogue Manager module of the tutorial server. In this way, the entire collection of dialogue session transcripts was in a ready-to-use format for additional NLP tasks or further research as required.

### **6.4.3 Question Marking**

#### **6.4.3.1 Pre-test and immediate post-test**

A representative sample of pre- and post-tests from at least 30 students who completed the experiment would be used to check for inter-rater reliability between the researcher and a senior teaching fellow from the HUBS teaching staff. Both raters would mark short answers according to the agreed model answer schedule (See Appendix D) developed prior to running the experiment. The researcher would then proceed to mark all pre- and immediate post-test short-answer questions, assuming an acceptable inter-rater reliability. Multiple-choice questions were scored automatically and no inter-rater reliability measure was required.

#### **6.4.3.2 Delayed post-test**

Delayed post-test questions were selected from the final course examination. All questions were marked independently of the researcher and in accordance with University procedures for final examinations. The researcher selected MCQ, short-answer and

mini-essay questions from the final examination which were directly relevant to material covered in the dialogue system. Unique student identity was used to match delayed post-test data to individuals who had taken part in the experiment.

#### **6.4.4 Summary**

This chapter has presented the background and methods used in the in-class evaluation of the new tutorial dialogue system, including the evaluation of the classifiers, student experience evaluation and the experiment to compare two versions of the tutorial dialogue system. Chapter 7 provides a detailed description of the results obtained.

# Chapter 7

## Results of the in-class evaluation

### 7.1 Introduction

This chapter summarises dialogue system performance data and the results from an in-class evaluation conducted with student volunteers during a three week period, immediately following cardiovascular system lectures. The purpose of presenting this data is to provide evaluative evidence of the tutorial dialogue system overall, in addition to conducting an experiment which compares performance between three groups of students who used the free-text version of the system, the menu-based version, and no system at all (the control group).

Section 7.2, covers overall tutor performance including classifier performance and student evaluation. Section 7.3 details the results of the experiment. The chapter concludes with Section 7.4, which provides a brief summary and discussion of the evaluation results.

### 7.2 Overall Tutor Performance

Evaluation data for both the menu-based and free-text versions of the dialogue system are summarised in this section. Classifier performance data only applies to the free-text version but student completion data, student evaluation data and unsolicited feedback data apply to both versions. In each case, where appropriate, data are summarised by version. The next section discusses preliminary classifier performance data from the in-class experiment.

## 7.2.1 In-class classifier performance data

This section reports only on the performance of case study classifiers. Performance data from all classifiers used in the system are not yet available.

### 7.2.1.1 Case study classifier performance data

In-class performance data from the evaluation experiment for each of the case-study classifiers is summarised below in Table 7.1 and Table 7.2. In each case a sample of 100 student responses was marked by a human marker and compared to the classifier results.

It is clear from this that on the open questions, classifier accuracy dropped substantially with in-class use. It is important to point out however that the inter-rater reliability scores for both the open questions was low and therefore it is perhaps not too surprising that the accuracy dropped in the in-class evaluation. (Refer Appendix A.4 for an index of classifier names and questions).

Table 7.1: In-class Case-study Classifier Metrics - Open and Binary Questions

| <i>Name</i>     | <i>Labels</i> | <i>Bench Test Accuracy</i> | <i>In-class Accuracy</i> |
|-----------------|---------------|----------------------------|--------------------------|
| what-pulse      | 5             | 0.75                       | 0.61                     |
| inotropic-state | 2             | 0.90                       | 0.65                     |
| vein            | 4             | 1.0                        | 0.97                     |

The binary question, *vein* which utilised a multilabel classifier, performed reasonably well with an overall accuracy in the evaluation of 0.97. It was clear from inspection of the sample taken that the few classification issues that arose related to responses where students had elaborated on their *yes* or *no* response. Precision for the *correct-explain* label (4 samples) was 1.0 and recall was 0.5. Precision for the *part-explain* label (1 sample) was 0 and recall was 0.

The multi-part question, *inotropic-factors* which utilised a multi-binary classifier, like the two open question classifiers, performed relatively poorly compared to benchmarking.

The average MASI distance (described in 5.3.1.4 where 0 indicates complete agreement between the label lists in the reference and test sets and 1 indicates no overlap)

Table 7.2: In-Class Case-study Classifier Metrics - Multi-part Questions

| <i>Name</i>              | <i>Bench test</i> |               |             | <i>In-class</i>  |               |             |
|--------------------------|-------------------|---------------|-------------|------------------|---------------|-------------|
|                          | <i>Precision</i>  | <i>Recall</i> | <i>MASI</i> | <i>Precision</i> | <i>Recall</i> | <i>MASI</i> |
| mass                     | 1.00              | 0.67          |             | 0.44             | 0.67          |             |
| symp                     | 0.50              | 1.00          |             | 0.46             | 0.55          |             |
| vfill                    | 0.71              | 0.83          |             | 0.80             | 0.38          |             |
| <b>inotropic factors</b> |                   |               | <b>0.23</b> |                  |               | <b>0.68</b> |

was much higher than during bench-testing, 0.68 compared to 0.23. It is also interesting to note that the precision and recall values for *mass* and *vfill* reversed in the in-class situation compared to bench-testing. What these figures do not reflect, however, was the relatively large number of student responses that were correctly *not classified* (54 out of 68, or 79% when compared with a human marker); that is, situations where none of the three possible labels was applied and this agreed with a human marker.

#### 7.2.1.2 In-class classifier performance summary and discussion

While bench-testing of classifiers yielded promising results across all three types of classifier question (open, binary and multi-part), preliminary evaluation of the four case-study classifiers used in-class suggests a dramatic performance reduction for all but the binary question classifiers. Accuracy for the two open question classifiers dropped from 0.75 and 0.9 on bench-testing to 0.61 and 0.65 in-class respectively. By contrast accuracy on a binary question classifier dropped from 1.0 on bench testing to 0.97 in-class. The multi-part question, multi-binary classifier also fared poorly, increasing from a MASl distance of 0.23 on bench-testing to 0.68 in-class. There are a number of possible reasons for the drop in performance. For the open questions, the already high degree of variability in student responses evident during bench-testing is likely exacerbated in-class: any subtle change in emphasis from teaching staff will very likely result in a drop in classifier performance. In addition, while the curriculum remained essentially the same between the collection of training data and its application in the tutorial dialogue system there was one change of lecturer during this time and this would have the potential to introduce new language, new expressions and new emphasis for the students, which, in turn would lead to reduced classifier performance;

the classifier had simply been trained on different data. The reduced performance of the multi-part question classifier was likely due to limited training data for some parts of the question leading to a potential class-imbalance problem which was discussed in Section 5.3.1.3.

There are a number of potential strategies for improving classifier performance in in-class settings in future and these are explored in Chapter 8. The next section describes the qualitative evaluation of students' experiences of the new tutor during the in-class experiment.

## **7.2.2 Student experiences**

This section reports the number of tutorial completions, as a proxy for user acceptability, and a combination of student perception questionnaires and unsolicited feedback from both students and a member of the teaching staff. Taken together, these reports provide a qualitative assessment of the usability of the dialogue system.

### **7.2.2.1 Student completions**

During the three week period in which it was available 720 students logged into the experimental system. Of these, 578 students completed the session through to the end of the immediate post-test. Distribution of completions was initially relatively even suggesting little or no sampling bias effects across conditions (McDonald *et al.*, 2012). However, at completion of the study all data marked by the system as complete was checked. Following this process, 47 student sessions were removed from the analysis because either a) web browser issues created problems with correct rendering of MCQs and some students were unable to complete MCQs or b) a few sessions were recorded incorrectly as complete by the system where the web browser timeout period of 30 minutes had been exceeded.

The final number of completions included in the analysis therefore was 531. The number of completions in the control condition was the highest (205), followed by the menu condition (177) with the fewest completions in the free-text condition (149). Sessions removed from the analysis came from each experimental condition. Seven were removed from the control group, 24 from the free-text condition and 16 from the menu-based condition. To check whether the test conditions themselves influenced completion, a Chi-square test for the analysis of categorical data was conducted where the expected distribution between the three conditions was even (that is approximately

177/531 or about a third of students in each group). The Chi-square test was significant at the 1% level ( $\chi^2 = 8.86$ ). It seems likely that the variation in completions reflects some factors related to student acceptability or willingness to complete each condition. The control group had higher than expected completions, possibly due to the fact that this condition took substantially less time to complete. The free-text group was lower than expected, perhaps due in part to issues related to understanding of student responses or because it took a little longer to complete than the menu-based condition.

### 7.2.2.2 Student evaluations

A summary of the student questionnaire data is provided in Appendix C.2. Recall that the evaluation questionnaire was sent to all students who logged on to the experimental system who were in either the free-text condition or menu-based condition (N=456). The evaluation was not sent to students in the control condition since they did not use the tutorial system.

A total of 105 responses were received (23%). Of these responses, 47 were from students who had been assigned the free-text tutorial and 58 were from students assigned to the menu-based condition. This response rate is consistent with large class evaluation response rates processed by the Evaluation Research and Development Unit (ERDU) of the Higher Education Development Centre at the University of Otago but at the lower end of the range (typically 20-30%). Written comments from three students noted that the questionnaire may have been better conducted soon after exposure to the system, rather than several weeks later.

The approach taken here in reporting the results of the questionnaire is consistent with usual practice by ERDU. That is, in reporting the results of the questionnaire the focus is on the frequency distributions of responses to 5-ratings Likert scale questions. The term ‘overall positive rating’ refers to the total percentage of 1s and 2s for individual questions. The standard report for the questionnaire which was issued by the ERDU is provided in Appendix C.2. Note that interpolated median scores are not used for formal processes but are reported in the University course questionnaire reports and therefore are reported in the questionnaire report generated for this study. To provide a context for interpreting the results of the questionnaire it is worth noting that 85% of standard teaching evaluations (5 core questions) conducted at the University of Otago have an overall positive rating (% 1s and 2s) of 70% or higher.

The most striking feature of the evaluation is that 94% of all those who responded

indicated that they would recommend the tutorial to other students and this percentage was identical for students who completed free-text or menu-based tutorials. This feedback is consistent with the 80% completion rate of those who participated in the experiment (see 7.2.2.1), the 78% overall positive rating of the tutorial as an aid to learning and the 73% overall positive rating of the tutorial as a revision tool (refer Appendix C.2).

By contrast, there was only weak agreement from students to question four (I felt like the tutor could understand my answers: strongly agree - disagree) (53%). This suggests that factors other than some inherent understanding-like affordance in the tutor were responsible for it being perceived by students as a useful tool to recommend to others.

Responses to question five (I found the tutor's questions: very easy - very difficult) suggest that generally students found the level of the tutorial about right and possibly on the easier rather than the harder side.

When the responses are analysed by tutorial condition this overall pattern remains consistent but there is some variability ( $\pm 10\%$ ) in the overall positive response scores on most of the questions (Refer Table 7.3). There are two exceptions to this. First, perhaps unsurprisingly, on question four, where 68% of respondents in the menu-based condition felt like the tutor could understand their answers compared with only 42% in the free-text condition. Second, on question 5, 'I found the tutor's questions easy/difficult', where there was only a 3% difference between the scores. Again, this is not unexpected since the tutorial content of the two conditions was identical. Given that this is a basic opinion survey comprising untested questions it is hard to read too much into these numbers other than to say that qualitatively there is little difference in student perceptions of the tutorials between the two conditions except in terms of feeling that the tutor 'understood' their answers, where the menu-based system seems to have an edge.

The student evaluation included 38 written comments. Eight responses related to reasons for non-completion. Three of these cited technical issues and two suggested either the tutorial was too long or that the student had insufficient time to devote to it. One student noted that they did not find the tutorial helpful and one felt that the 'Tutor' did not properly understand their answers.

There were 30 general comments. These were predominantly complimentary and/or positive about the tutorial (19). Five found the 'Tutor' frustrating or felt their responses were poorly understood. Other key themes from student suggestions and

Table 7.3: Overall positive ratings (%1s&amp;2s) by condition

| <i>Likert Scale Question</i>   | <i>Experimental condition</i> |                   |
|--|-------------------------------|-------------------|
|  | <b>Free-text</b>              | <b>Menu-based</b> |
| 1. I would rate the online tutor as an aid for learning: (very useful = 1 to of little use = 5)  | 72%                           | 83%               |
| 2. I learned new things about cardiovascular homeostasis from the online tutor: (strongly agree = 1 to strongly disagree = 5)  | 61%                           | 50%               |
| 3. I felt the tutor helped me to understand things about cardiovascular homeostasis that we had covered in the course: (strongly agree = 1 to strongly disagree = 5) | 78%                           | 70%               |
| 4. I felt like the tutor could understand my answers: (strongly agree = 1 to strongly disagree = 5)  | 42%                           | 68%               |
| 5. I found the tutor's questions: (very easy = 1 to very difficult = 5)  | 27%                           | 24%               |
| 6. Overall, in helping me to revise my understanding of cardiovascular homeostasis I found the tutorial: (very effective = 1 to very ineffective = 5)                | 68%                           | 77%               |

comments included: Supporting media (e.g. video) would be helpful (2), technical issues (2), more questions and/or more depth to questions (2), tutor questions hard to understand (2), tutorial patronising (1), abbreviations not explained (1), tutorial too long/lack of time (1). In addition, three students commented on the timing of the evaluation questionnaire as noted previously.

### 7.2.2.3 Unsolicited feedback

During the experiment unsolicited feedback was received from 6 students. These comments are all brief and are given verbatim, below. The comments provided useful validation of the feedback solicited via the student evaluation questionnaire and most

refer to web browser issues. All students received responses to their questions. Student 3 and student 6 were in the control group and did not receive any tutorial; however both these comments demonstrate that at least some students were discussing the tutorial with friends.

Student 1: "I am writing about your Cardiovascular Homeostasis Tutorial page. I am unable to type any answers into the boxes under the questions."

Student 2: "This was very helpful :) Thanks so much. If only every topic had one of these."

Student 3: "I found the tutorial useful but I could only enter answers for about half of the questions. Also, I seem to have only been able to do a short version of the tutorial with no feedback given at the time whereas a friend got an interactive tutorial with feedback given at the time in a dialogue box. Was I supposed to have been able to do this same tutorial or is it only available for some people?"

Student 4: "Your tutorial had really good questions in it and was set out really well so it would have been great for revision but it did not let me see any of the options for the multi choice questions so i could not answer the questions."

Student 5: "thanks :)"

Student 6: "All my friends found the tutorial very helpful as did i however the tutorial bit didn't work on my computer which was a shame i could only do the opening and closing questions i am not sure as to why"

Unsolicited feedback was also received from a member of the teaching staff who had not been involved in system development:

Teacher 1: "I am a teaching fellow in Physiology and am teaching in the HUBS labs this term. For interest I completed the CVS online tutorial. If my participation confounds your results in any way I am happy to provide my data so you can remove it. I found it laid out very well and only one or two questions that were a little ambiguous. It emphasised important details and helped reinforce concepts that the students often find difficult."

The next section describes and tests the main experimental hypotheses where the free-text and menu-based versions of the system are compared with a control and with each other.

### 7.3 Version Comparisons: Free-text vs Menu-based

Data from the experiment to test which version of the dialogue system would yield the largest gains in student performance are presented in this section. Section 7.3.1 compares student results on the immediate post-test. Section 7.3.2 compares student results from relevant sections of the final examination. Section 7.3.3 discusses a selection of post-hoc analyses including the time taken to complete the tutorial, timing of the experiment, student cohort and question type.

As discussed in Chapter 6, the main hypotheses being tested are:

- A. Either tutorial intervention, free-text or menu-based, results in better post-test performance than no intervention (the control group);
- B. Free-text input results in better post-test performance overall than MCQ, because students recalling and constructing their own response should promote retention and/or understanding better than simply selecting from pre-constructed options;
- C. Free-text tutorials lead to increased performance on short-answer questions because of a practise or testing effect;
- D. Menu-based tutorials lead to increased performance on MCQ questions, also because of a practise or testing effect.

The number of students who completed the session through to the end of the immediate post-test was 578. Of these, 531 these are included in the analysis with 47 excluded for reasons noted in Section 7.2.2.1. The highest number of completions was obtained in the control group (205) and the lowest in the free-text group (149). As noted in Section 7.2.2.1, a Chi-square test for the analysis of categorical data was significant at the 1% level ( $\chi^2 = 8.86$ ). However, given that the variation in completions likely reflects student acceptability no corrections have been made for sample bias.

Short-answer sections were checked for inter-rater reliability on pre- and post-test questions for a sample of 30 students using Cohen's kappa for 2 raters. The value of kappa ranges from -1 to 1, where -1 is complete disagreement between two markers, 0 is the extent of agreement one would get by chance and 1 is complete agreement. With reference to the Landis and Koch (1977) benchmark which provides a practical guide for interpreting the value, a Cohen's kappa of 0.93 ( $p=0$ ) in this case confirmed very high agreement between the two markers.

### 7.3.1 Immediate post-test

Table 7.4 summarises the descriptive statistics across the three experimental conditions for the pre-test and immediate post-test. Across all three conditions students performed well in the pre-test with mean normalised scores ranging from 0.82-0.84. It can be seen from Table 7.4 that in the post-test student scores dropped across all three conditions. The post-test questions were deliberately designed to be harder than the pre-test questions, so a drop in score was expected as a main effect. However, the mean scores on immediate post-test were higher in both the tutorial conditions compared to the control (0.76 and 0.77 c.f. 0.70).

Table 7.4: Descriptive Statistics

|                            | <b>Control</b><br>n=205 |           | <b>Free-text</b><br>n=149 |           | <b>Menu-based</b><br>n=177 |           |
|----------------------------|-------------------------|-----------|---------------------------|-----------|----------------------------|-----------|
|                            | <i>mean</i>             | <i>sd</i> | <i>mean</i>               | <i>sd</i> | <i>mean</i>                | <i>sd</i> |
| <b>Pre-test</b>            | 0.82                    | 0.17      | 0.84                      | 0.13      | 0.83                       | 0.15      |
| <b>Immediate Post-test</b> | 0.70                    | 0.20      | 0.77                      | 0.16      | 0.76                       | 0.17      |

The dependent variable to test the first hypothesis (A) was taken as the difference between pre- and post-test performance for each student with the pre-test result serving as a common baseline in each case. The differences between pre- and post-test scores were normally distributed (refer Figure 7.1) which allowed the use of parametric tests to see if there were differences between the means in each condition. A between-subjects ANOVA gave an F value of 3.73 and a post-hoc Tukey multiple comparison of means at 95% confidence level showed a significant difference when compared with the control for the menu-based tutorial condition ( $p=0.039$ ) but just outside significance for the free-text condition ( $p=0.076$ ). This result is discussed further in Section 7.3.3.1 in light of post-hoc analysis related to the timing of the tutorial intervention.

There was no support for the second hypothesis (B), that free-text input results in better post-test performance overall than menu-based input; comparison between the mean scores for free-text condition and menu-based condition was not significant ( $p=0.987$ ). Given this result, it was perhaps not surprising that there was also no support for the third hypothesis (C), that free-text tutorials improve scores on free-text questions in the immediate post-test ( $p=0.901$ ). And, consistent with this result, there was no support for the fourth hypothesis (D), that multiple-choice questions

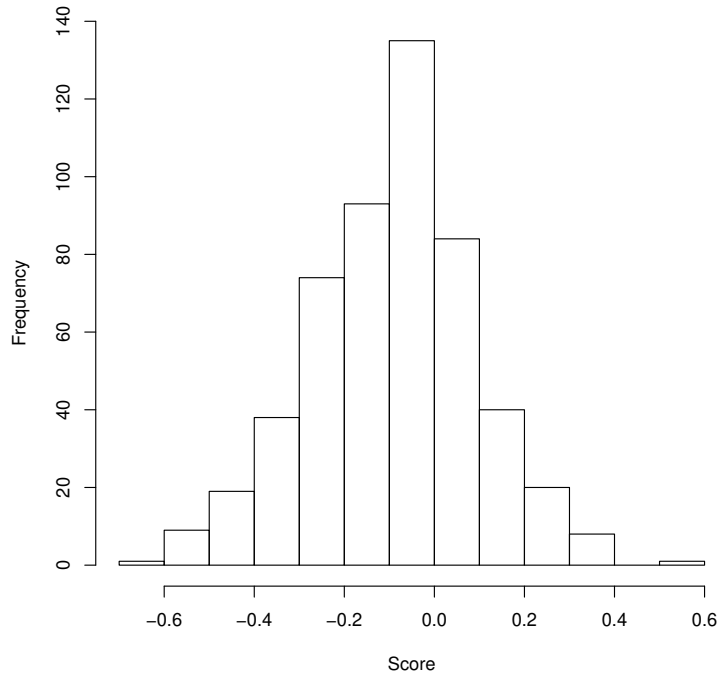


Figure 7.1: Distribution of normalised pre-test minus normalised post-test scores.

improve immediate post-test performance on the MCQs ( $p=0.558$ ).

As a further check, an ANCOVA was conducted, using pretest scores as the covariate, condition as the independent variable and post-test score as the dependent variable, to see whether greater statistical power would yield a different result. Overall, it did not change the result for hypothesis (B). There was no significant difference between the menu-based and free-text conditions ( $p=0.99$ ) although both menu-based ( $p=0.004$ ) and free-text ( $p=0.005$ ) conditions were significantly different from the control group. In other words it provided stronger support for hypothesis (A)

### 7.3.2 Delayed post-test

The HUBS final examination comprised three main sections, multi-choice, short-answer and mini-essay, and was taken by students some 10 weeks following the experimental intervention. Examination questions used for the delayed post-test analysis and their model answers are listed in Appendices D.3. Student answers to relevant questions for the cardiovascular section of the course were extracted from each section and the

results normalised. In the same way as the immediate post-test analysis above, the pre-test was used as a baseline for mean comparisons in order to attain approximately normal distributions for analysis. Table 7.5 summarises the descriptive statistics for each condition in each section of the final examination.

Nine students who took part in the experiment did not complete the final examination. This reduced the number of participants in the experiment from 531 to 522. Five students were lost from the control condition and two each from the menu-based and free-text conditions.

Table 7.5: Delayed Post-Test Descriptive Statistics

|                     | <b>Control</b><br>n=200 |           | <b>Free-text</b><br>n=147 |           | <b>Menu-based</b><br>n=175 |           |
|---------------------|-------------------------|-----------|---------------------------|-----------|----------------------------|-----------|
|                     | <i>mean</i>             | <i>sd</i> | <i>mean</i>               | <i>sd</i> | <i>mean</i>                | <i>sd</i> |
| <b>MCQ</b>          | 0.87                    | 0.17      | 0.86                      | 0.18      | 0.87                       | 0.18      |
| <b>Short-answer</b> | 0.75                    | 0.18      | 0.77                      | 0.17      | 0.76                       | 0.19      |
| <b>Mini-essay</b>   | 0.74                    | 0.26      | 0.74                      | 0.25      | 0.75                       | 0.25      |

Again, the dependent variable to test the first hypothesis (A), this time with reference to the delayed post-test, was taken as the difference between pre- and post-test performance for each student with the pre-test result serving as a common baseline. The differences between pre-test and delayed post-test scores were approximately normally distributed (refer Figure 7.2).

Looking first at the overall score (combining the three sections of the delayed post-test), a between-subjects ANOVA gave an F value of 0.41. A post-hoc Tukey multiple comparison of means at 95% confidence level showed no significant difference when compared with the control for either the menu-based tutorial condition ( $p=0.99$ ) or for the free-text condition ( $p=0.66$ ). Again, as a further check, an ANCOVA was conducted using pretest scores as the covariate, condition as the independent variable and delayed post-test score as the dependent variable. There was no significant difference between the control, menu-based or free-text conditions ( $F=0.17$ ,  $p=0.84$ ). Analysis of each section of the delayed post-test similarly revealed no significant differences between-subjects in each condition. These results are summarised in Table 7.6.

From the descriptive statistics it appears that there is a within-subjects difference in the means for all three conditions between MCQ results (0.86 - 0.87) and short-answer

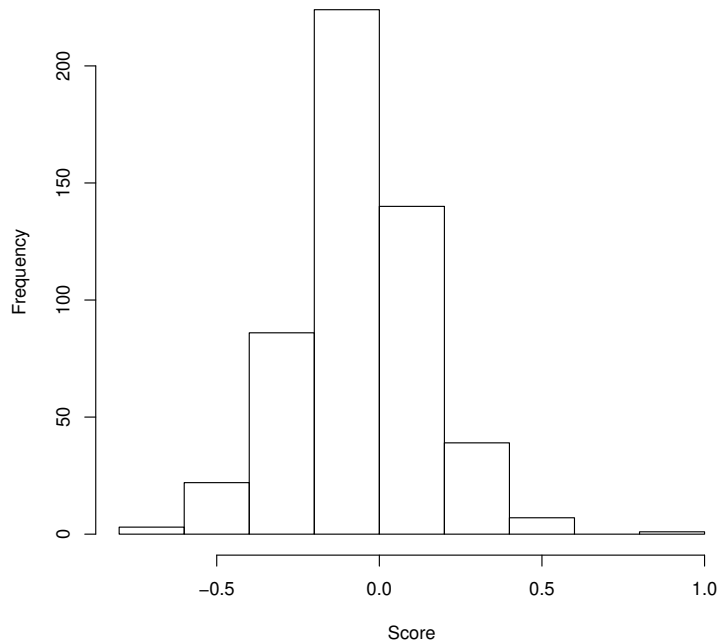


Figure 7.2: Distribution of normalised pre-test minus normalised post-test scores.

Table 7.6: Delayed Post-Test Between-Subjects ANOVA

|                     | ANOVA          |          |
|---------------------|----------------|----------|
|                     | <i>F-score</i> | <i>p</i> |
| <b>MCQ</b>          | 0.34           | 0.71     |
| <b>Short-answer</b> | 0.10           | 0.90     |
| <b>Mini-essay</b>   | 0.57           | 0.55     |

or mini-essay results ( $\approx 10$  percentage points lower). Paired t-tests with the Bonferroni correction confirm this ( $p \ll 0.01$ ). This result is not unexpected and is consistent with anecdotal comments from teaching staff that students generally perform better on MCQs in the final exam than they do on short-answer and mini-essay questions.

In summary, the first hypothesis (A), that either tutorial condition confers an advantage over the control on a delayed post-test is not supported. Given that the tutorial intervention reflected only a very small part of the overall course and indeed only one

part of the cardiovascular section of the course, and given the long delay between intervention and the final exam, this result is perhaps not too surprising.

### 7.3.3 Post-hoc data analysis

#### 7.3.3.1 Timing of tutorial

Interim results for this experiment were reported in McDonald *et al.* (2012) when data from 338 students were available. At that time, a between-subjects ANOVA showed a significant difference between the control condition and both the free-text tutorial condition ( $p=0.03$ ) and the menu-based tutorial condition ( $p=0.01$ ) with an F value of 4.95 and a post-hoc Tukey multiple comparison of means at the 95% confidence level. It seemed odd that an initial strongly significant difference between both tutorial conditions and the control condition should appear to be diluted once all data was available. One possible hypothesis to explain this is that students were ‘cramming’ for the terms test which ran at the end of the experimental period. This would result in the effect of the tutorial being diluted as students taking the tutorial closer to the test are likely to be doing far more additional study than students taking the tutorial earlier on. To investigate this further an analysis of scores by timing of the tutorial was undertaken.

Figure 7.3 shows the number of students who elected to take part in the experiment on each of the days that it was available. As described in Chapter 6 the experiment started at the end of the lecture series on the cardiovascular system, coincided with a period of laboratory work and self-directed study assignments and ended on the day of a terms test (11th August) which covered all the material to which students had been exposed. There are two features in the histogram that are important. The first occurs on the 1st August when there is an increase in the number of students taking part. This is most likely explained by additional notice of the experiment being posted on the course LMS on that day, at the suggestion of one of the course Teaching Fellows. The suggestion was made in order to encourage greater uptake of the tutorial by students. The second major increase occurs just 2 days before the terms test and the conclusion of the experiment itself. It is noteworthy that the results reported in McDonald *et al.* (2012) included all data except data from the last 2 days before the terms test (9th and 10th August). Five students took part in the experiment on the same day as the terms test (11th August).

When data from the last two days were analysed separately a between-subjects

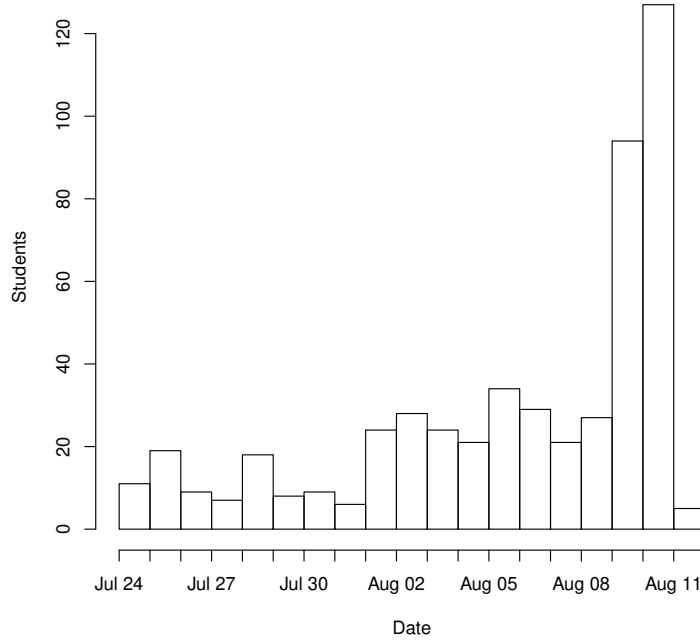


Figure 7.3: Distribution of student participation in experiment by date.

ANOVA giving an  $F$  value of 0.40 and a post-hoc Tukey multiple comparison of means at 95% confidence level showed no significant difference when compared with the control for either the free-text tutorial condition ( $p=0.65$ ) or the menu-based tutorial condition ( $p=0.84$ ). Consistent with all other results there was also no significant difference between the free-text and menu-based conditions ( $p=0.93$ ).

This supports the ‘cramming’ effect posited above. That is, normalised scores in the control group increase on average as students study towards the terms test and the effect of completing either tutorial when combined with intensive study confers no additional advantage. This may also represent a ceiling-effect in terms of student performance on the immediate post-test. Linear regression analysis of scores in each condition over time appears to confirm this. Figure 7.4 plots interpolated scores in each experimental condition by time. The slope of the black line, representing the control condition, increases over time and  $p < 0.05$  for both slope and intercept. The green and red lines have negligible slope throughout and are almost identical. The three lines intersect at the end of the experimental period by which time there is no significant

difference between scores in any condition.

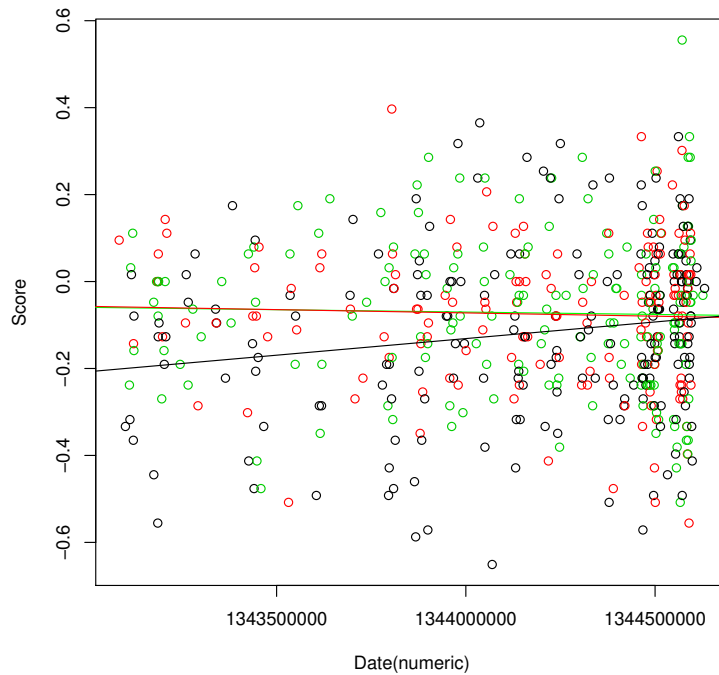


Figure 7.4: Linear model of normalised scores over time by condition.  
control=black, menu-based=green, free-text=red

### 7.3.3.2 Implication for overall results

In terms of addressing the first hypothesis, whether the use of either free-text or menu-based versions of the new system confers an advantage on student performance, the answer is a qualified yes. There are highly significant differences between both conditions and the control when the tutorial is taken earlier in the three week period leading up to a course terms test. This effect is diluted by the large number of students who are preparing for their terms test and complete the tutorial in the last couple of days. This demonstrates that evaluation of the tutorial intervention is sensitive to the timing of summative events in the course. This is a useful finding in terms of the practical issues to consider when conducting in-class evaluations. However, timing has no effect on the remaining hypotheses where the two versions of the system are explicitly compared. This remains a null result; it makes no difference to student performance on an immediate post-test whether the tutorial system used is free-text or menu-based.

### 7.3.3.3 Time taken for tutorial

Prior to the experiment it seemed reasonable to assume that any differences between the free-text condition and menu-based condition might be accounted for by time-on-task (refer to Chapter 2). For example, a student may take longer to think through and write a response than to select from a series of options. The extra time spent thinking about and generating the answer may in itself account for differences. Given that there was no difference between the free-text and menu-based conditions this seems unlikely. Nevertheless, it was worth reviewing just how long students took to complete the task/s in each condition and to see whether there was a relationship between performance in the post-test and time spent on the tutorial.

The time taken by students across all tutorial conditions is summarised in Figure 7.5. Note that the time on task for all 205 students in the control condition is zero since they took only the pre and post-tests and did not spend any time on the tutorial task: this is represented by the first bar at zero on the histogram. Almost 90% of students completed either the menu-based or free-text tutorial in less than 2000 seconds ( $\approx 30$  minutes).

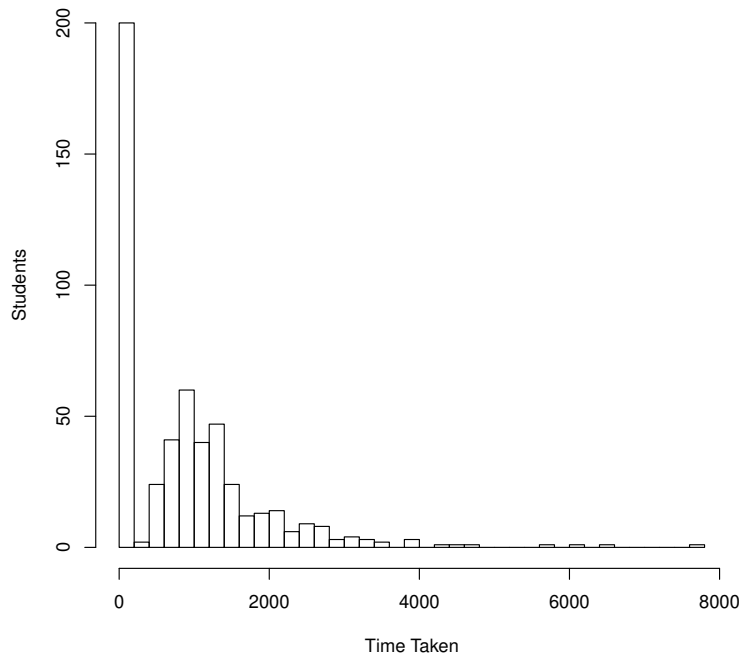


Figure 7.5: Distribution of Time Spent on Tutorial Task

The average time taken for both tutorial conditions is summarised over the range of immediate post-test marks in Table 7.7. The average time on task in the free-text condition ( $\approx 27$  minutes) was unsurprisingly longer than the average time on task in the menu-based condition ( $\approx 20$  minutes). Students in the top range of immediate post-test marks (above the third quartile) spent less time on average on the tutorial ( $\approx 17$  minutes for menu-based and  $\approx 23$  minutes for free-text) than those students who scored below the first quartile ( $\approx 22$  minutes for menu-based and  $\approx 35$  minutes for free-text).

Table 7.7: Summary of immediate post-test score ranges by condition

|   | <b>Tutorial Condition</b>  |                            |
|---|----------------------------|----------------------------|
|   | <i>Menu-based</i>          | <i>Free-text</i>           |
| <b>Mean time-on-task (secs.)</b>                                | 1211 ( $\approx 20$ mins.) | 1668 ( $\approx 27$ mins.) |
| <b>Q1 (Norm. post-test score diff. <math>&lt; -0.14</math>)</b> | 1337 ( $\approx 22$ mins.) | 2080 ( $\approx 35$ mins.) |
| <b>Q3 (Norm. post-test score diff. <math>&gt; 0.06</math>)</b>  | 1032 ( $\approx 17$ mins.) | 1391 ( $\approx 23$ mins.) |

Linear regression analysis (Figure 7.6) confirms that scores decrease with time spent in both conditions with a somewhat steeper fall-off in scores in the free-text condition ( $F = 9.818$ ,  $p = 0.002$ ) than in the menu-based condition ( $F = 3.822$ ,  $p = 0.05$ ) with the lines intersecting at around the 30 minute mark ( $\approx 2000$  secs). On this basis it appears that students who spend less than 30 minutes perform slightly better in the free-text condition and students who spend more than 30 minutes perform slightly better in the menu-based condition.

### 7.3.3.4 Student cohort

Final examination results for all students who took part in the experiment were provided to the researcher. Additional analysis questions were explored using this information and are important in terms of both the internal and external validity of the study. First, did the students who volunteered for this experiment perform better in general than students who did not? Second, was there a correlation between the immediate post-test results and the final examination results? (This does not indicate causality but should provide evidence of consistency between the two assessments for any given student.) Third, did students who took part in the experiment earlier,

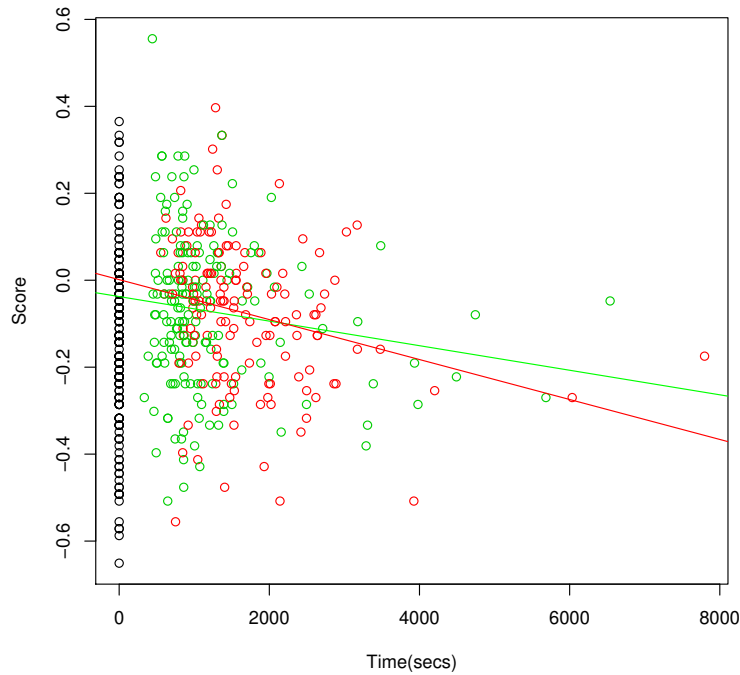


Figure 7.6: Linear model of normalised baseline scores over time on task by condition.

control=black, menu-based=green, free-text=red

perhaps indicating higher motivation, do better in the final examination?

To answer the first question the final examination score distributions for the experimental group and those of the rest of the class were compared. Final grades were available for 1441 students. The distribution of grades is illustrated in Figure 7.7.

Marks ranged from 14-99, the mean grade was 70 and the median was 72. Grades for students who volunteered for, and completed the experiment, and who completed the final exam (N=522) are illustrated in Figure 7.8.

Grades in this group also ranged from 14-99; however the mean grade was 78 and the median grade was 82 with the quartiles skewed slightly to the right as well (Q1=69 c.f. Q1=59 and Q3=90 c.f. Q3=85). This supports the idea that students who volunteered for the experiment were perhaps generally higher performing but the similar range of grades and distribution pattern is good evidence that students of varying ability volunteered for the experiment.

To see whether there was a correlation between the immediate post-test and final

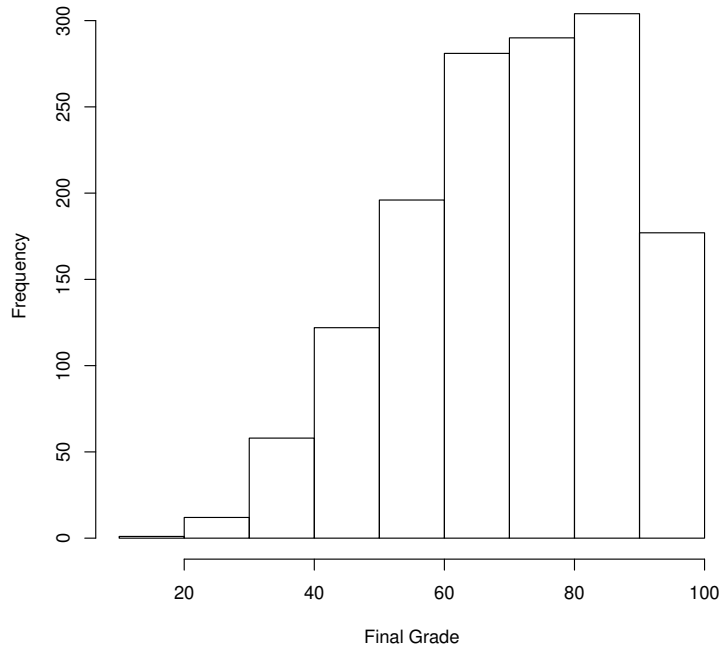


Figure 7.7: Distribution of final grades for the HUBS class.

examination questions, a scatter plot and regression analysis using normalised baseline scores from the immediate-post test and the cardiovascular system (CVS) component of the final exam are shown in Figure 7.9. There is a strong positive correlation between the immediate post-test and final exam questions ( $p \ll 0.01$ ) for both the slope and intercept suggesting that the two performance assessments are consistent with each other.

To examine the last question which uses early participation in the experiment as a proxy for motivation, a scatter plot was used again (refer Figure 7.10.) This plot compares the date of experiment participation with normalised baseline score on the CVS component of the final examination. In this case there is only a weak negative correlation (i.e. score decreases with later participation in the experiment), with date of participation in the experiment accounting for less than 2% of the variance ( $R\text{-squared} = 0.014$ ), although the relationship is significant ( $p=0.006$ ). It is difficult to draw any conclusions about motivation from this result.

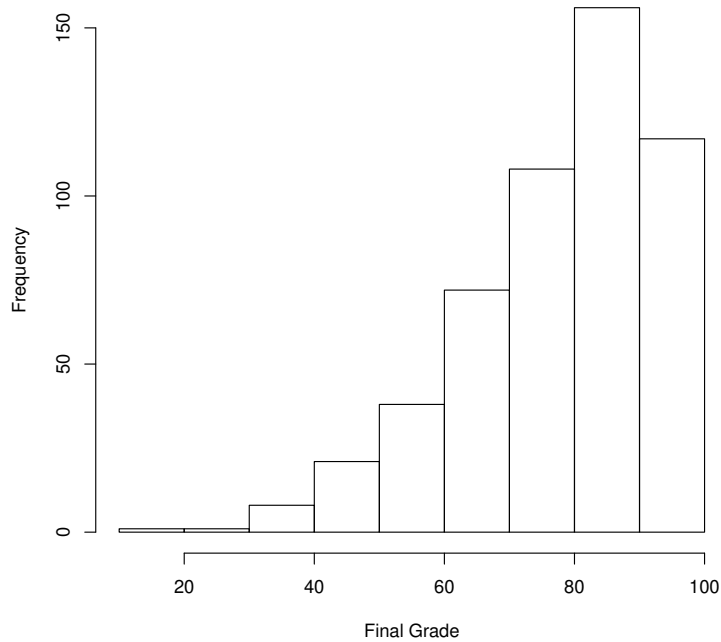


Figure 7.8: Distribution of final grades for experiment volunteers.

### 7.3.3.5 Immediate post-test question type

During marking of the immediate post-test questions it became clear that students found some questions easier than others and some were better understood than others. This is typical of some of the many issues encountered in writing good assessment questions (for example, see Miller, Imrie, and Cox, 1998). Within the HUBS course itself, a measure of question difficulty and understanding is recorded for all examination questions each year which helps to inform assessment development in subsequent years. Of interest here is whether there is variation across the three experimental conditions. For example, does completing the free-text tutorial condition confer an advantage for answering a particular question? In order to answer this question chi-square tests examining score versus condition were run for each question in the immediate post-test. The immediate post-test questions, where chi-square test results were significant at either the 1% or 5% level, are summarised in Table 7.8.

It is worth noting that no clear difference was found between the menu-based and free-text conditions for any specific question. Nevertheless, as listed in Table 7.8 there are six specific post-test questions where completing either tutorial appears to confer

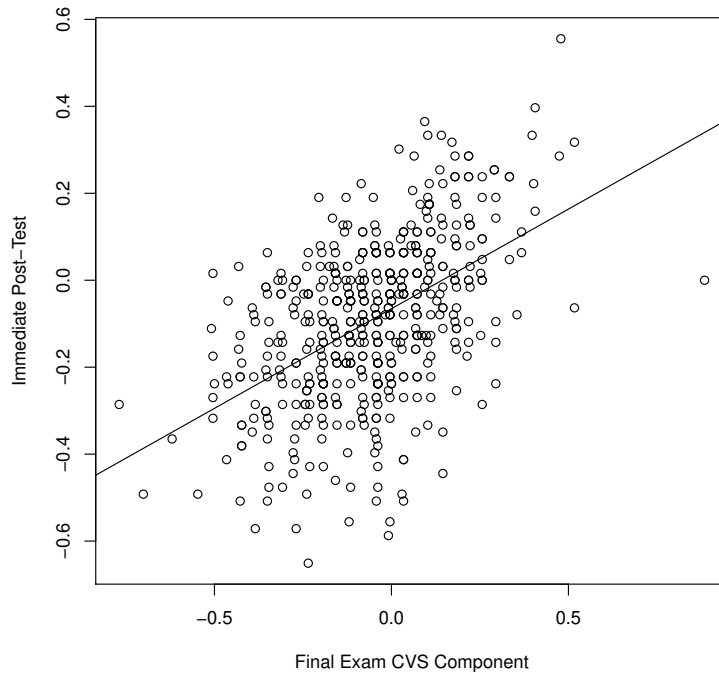


Figure 7.9: Scatter plot and regression line for score correlation: Final exam vs immediate post-test.

a benefit over the control condition. Note that the additional degrees of freedom in questions 9, 10 and 11 relate to the number of score categories; in this case, 0, 1 or 2 marks as opposed to either 0 or 1 mark for questions 2, 3 and 6. Three of these six post-test questions are MCQs and three are free-text. In terms of linkage to specific dialogue contexts in the tutorial, the same six questions are also fairly evenly distributed between dialogue contexts where relevant classifiers performed well in bench-testing and where they performed less well.

The next and final section of this chapter summarises the overall evaluation results.

## 7.4 Overall summary of the evaluation

In spite of a substantial drop in performance for the classifiers used in the free-text version of the system, it is interesting to note that a large number of students still chose to complete the free-text version of the tutorial (149). Although significantly lower than the number of students who chose to complete the menu-based version of

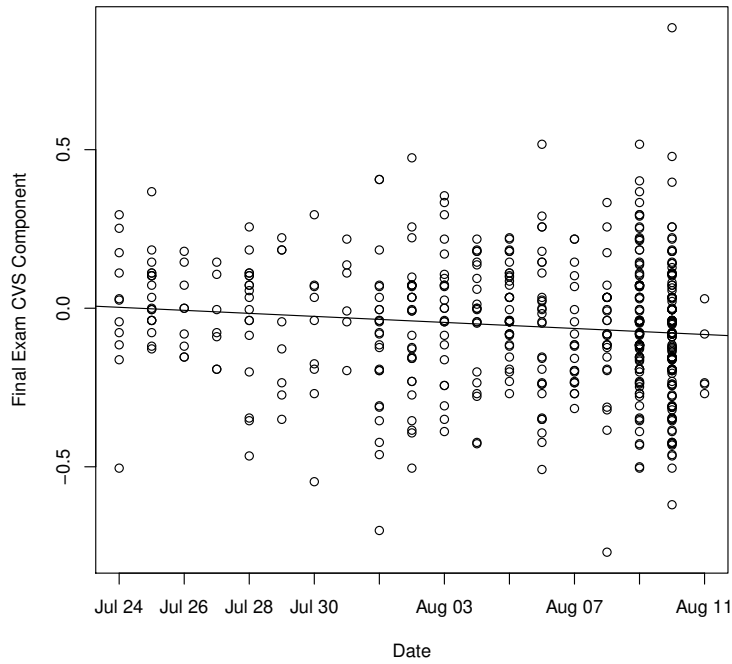


Figure 7.10: Scatter plot and regression line for final score correlation with date of experiment participation.

the tutorial (177), where incorrect classification is not an issue, both groups showed lower completions than the control group (205). This suggests that classification issues alone are not sufficient to account for differences in tutorial completion and other factors such as overall duration of the tutorial condition may play a role. This is also supported by the fact that among students who responded to the system evaluation questionnaire there was little qualitative difference in perceptions between those who used the free-text version and those who used the menu-based version. A key finding from the questionnaire was that 98% of all respondents indicated they would recommend the system (free-text or menu-based) to other students.

In terms of hypothesis A, whether the use of either free-text or menu-based versions of the new system confers an advantage on immediate post-test performance, the answer is yes, provided the tutorial intervention is carefully timed. There were highly significant differences between both conditions and the control when students took the tutorial earlier in the three week period leading up to course mid-term test and this is suggestive of a ceiling effect. Thus, there is support for the first hypothesis with a

Table 7.8: Significant Chi-square test results

|   | $\chi^2$ | $df$ | $p$         |
|---|----------|------|-------------|
| Q2. A decrease in blood pressure would result in: [Select]  | 9.47     | 2    | 0.008       |
| Q3. The term used to describe blood pressure at its highest point is: [select]                          | 15.61    | 2    | 0.0004      |
| Q6. In response to an increase in blood pressure: [select]  | 24.17    | 2    | $\approx 0$ |
| Q9 The flow in Mr X's pulmonary circulation is 4.9L/min. What is his Cardiac output?                    | 10.823   | 4    | 0.028       |
| Q10 What is the specific arrangement of blood flow between different organs of the body?                | 22.41    | 4    | 0.0001      |
| Q11. The measurement of somebodys pulse rate is a good indication of what variable of cardiac activity? | 15.18    | 4    | 0.004       |

small caveat related to timing of the intervention. However, there was no difference between any of the conditions on the delayed post-test. Given that the tutorial intervention reflected only a very small part of the overall course and only one part of the cardiovascular section of the course; and given the long delay between intervention and the final exam, and finally given the ceiling effect noted on immediate post-test performance, this is not unexpected. Overall, these results are consistent with the findings of similar previous studies (Corbett *et al.*, 2006; Aleven *et al.*, 2004) although this study was unable to confirm any effect on delayed post-test.

None of the remaining hypotheses (B-D), which related to performance advantages for either free-text or menu-based versions of the system, was supported.

The next chapter, Chapter 8, summarises the project as a whole, discusses the implications and limitations of this evaluation and outlines some opportunities for further work.

# Chapter 8

## Discussion

### 8.1 Introduction

The first goal of this study was to design, build and evaluate a pragmatic tutorial dialogue system for the cardiovascular section of the HUBS 192 course. Two versions of the new system were required in order to investigate the second goal, namely, to determine whether the opportunity to practise writing answers to short-answer questions and receiving detailed automated feedback would result in performance gains. These two goals were combined into an overall system evaluation which aimed to assess the performance of the new system in a real class setting and to formally test a series of hypotheses which followed from the second goal. This final chapter briefly summarises the rationale for, and features of, the new tutorial dialogue system and discusses the implications of the results of the in-class evaluation. Some of the limitations of this study as a whole are described along with opportunities for further work. Finally the specific contributions of the thesis as a whole are delineated and the chapter concludes with some closing remarks.

### 8.2 Rationale for the new tutorial dialogue system

Intelligent Tutoring Systems, including tutorial dialogue systems, rarely find utility in real class settings at present (Reeves and Hedberg, 2003; Shute and Zapata-Rivera, 2010). The primary goal of this research was to build a new tutorial dialogue system and to address some of the reasons for the lack of utility in classroom settings which were discussed in Chapter 2:

- The tutorial dialogue system should be both responsive and practical in a real

class setting.

- System development should, from the outset, be informed by student responses and teacher feedback and a mechanism for incorporating ongoing student feedback (that is, student responses to the questions posed) should be designed into the system itself.
- There should be no explicit student model other than what is required to avoid circular questioning and repeated turns. The reason for this is twofold: first, student models are hard to create and second, they have been resisted by educators on both practical and theoretical grounds. The representation of expert or tutor knowledge is **compiled** into a script which is created by the tutorial author.
- Given the difficulty and expense of authoring, or customising ITS for specific contexts, a generic structure for creating tutorial dialogues should be designed into the new system in order that it can be readily adapted for use in different contexts.

As described in Chapters 4 and 5, the tutorial dialogue system which was designed and implemented as part of this study was indeed largely informed by student responses as well as by teacher feedback. This was made possible largely as a consequence of adopting an empirical, surface-based approach to the natural language understanding part of the system. In addition the representation of tutor knowledge was compiled into a pre-authored script and the development of the script itself required no fine-grained analysis of the task domain beyond the detailed examination of representative student responses. An existing model of human dialogue informed the design of the dialogue script structure which is entirely independent of the domain covered by the tutorial itself. In theory, this should support the development of tutorials in alternative domains in the future. In order to determine whether the system created was responsive and practical in a real class setting, an in-class evaluation was performed with two versions of the system, one which supported free-text input and the other where students selected their preferred response from a menu of options, and this was described in detail in Chapters 6 and 7. The evaluation is summarised and discussed in the next Section 8.3.

## 8.3 Implications of the in-class evaluation

In many large classes, as the teaching staff who provided the original impetus for this study attested, it can be all but impossible to find opportunities to provide individualised formative feedback to short-answer questions. Did the evaluation of the new tutorial dialogue system offer some hope for addressing this issue?

The evaluation results demonstrated that there is a benefit to, and an appreciation from, students for the two versions of the tutorial dialogue system, both free-text and menu-based. There were highly significant differences between the tutorial conditions and the control condition when students took the tutorial earlier in the three week period leading up to course mid-term test. As described in section 7.3.3.1, the difference in performance between the conditions decreases as students study towards a terms test, and is suggestive of a ceiling effect in relation to the material being studied. In addition, a large number of students chose to take part in the evaluation, a relatively large number completed the tutorial condition (especially when compared to the much lower completions achieved during initial prototyping) and there was an overall positive response from students who responded to a student experience questionnaire which sought specific feedback on the tutorial systems. This result, especially coming from an in-class evaluation with a very large class, suggests that the provision of automated formative feedback to questions in a tutorial dialogue setting is a viable option for large classes generally. Nevertheless, there are still a number of issues and areas to address before a tutorial dialogue system like the one presented here would become commonplace in a real classroom setting, and these are discussed in Section 8.4

It also remains to be seen whether there is a discernible difference between the provision of feedback to written responses and the provision of feedback to options selected from a menu. This is an interesting and important distinction and the results of this evaluation found no difference between the two. However, there are caveats to accepting this result at face value and again, these are discussed in the next section, 8.4. Clearly it is much easier to set up a menu-based system. In the main, this is because there is no need to deal with classifying text which is an active and as yet, far from resolved research area in computational linguistics. But to simply resort to a menu-based system misses the point. The menu-based system is also a tutorial dialogue system in its own right and as described in Section 4.4.3.3 and Section 4.4.4.1, differs only from the free-text system through altering a single line of code in the dialogue manager and referring to menu-options in the script which are identical to the class

labels available to the classifiers in the free-text system. In other words, the menu-based system relied on a ‘training set’ of free-text student responses in order to generate the options from which students selected their response. Furthermore, these options were presented in the context of a coherent tutorial dialogue as opposed to being presented simply as a series of discrete questions. Again, there is more work to do to delineate the subtle effects of these different types of intervention.

Nevertheless, the tutorial dialogue system presented has been shown to be a viable and robust platform for exploring these types of issues in a real-class setting. It is worth mentioning that apart from some relatively minor browser-compatibility issues (these were noted in Section 7.2.2.1), the system itself stood up very well to large numbers of students logging in and completing the tutorial conditions. There were no significant technical issues reported during the in-class evaluation and this is perhaps testament to the simplicity of the design and implementation of the system. That said, the extensive prototyping and piloting that was undertaken with students and teachers during system development and which was described in Section 4.3, also played a role in ensuring that the in-class evaluation ran smoothly.

As already noted, there are a number of limitations to the in-class evaluation and these are discussed further in the next section.

## **8.4 Limitations of this study and some opportunities for further work**

The in-class evaluation provided evidence of the system’s acceptability to students and its impact in terms of student performance gains. Whether there is a benefit for teaching staff remains to be established. It is likely that staff who teach large classes will be supportive of any system which benefits their students, but there are limits to the cost of systems which both teachers and teaching institutions are prepared to bear. Teaching staff in this study, apart from consulting on the nature and content of the tutorial dialogue and pre and post-test questions, had little to do with the practical task of authoring the dialogue. As noted in Chapter 2, the problem of authoring for ITS is well known and if the difficulty or expense of authoring is too great this is likely to become a significant barrier to wider adoption, even if there are demonstrable benefits to students. This aspect of the system has yet to be evaluated but there are good reasons to be cautiously optimistic. The tutorial dialogue structure is completely independent of the domain of the tutorial. There is nothing in the dialogue structure

which ties it to, in this case, cardiovascular homeostasis. The dialogue structure follows a simplified model of a normal human-human conversation, with forward-looking and backward-looking elements, within which are embedded dialogue acts. The dialogue acts defined so far, are certainly consistent with a directed tutorial dialogue but this is not to say that additional acts cannot be added. What is missing at present is a usable computer or web-based interface in order to facilitate authoring the dialogue or specific parts of it. Developing and evaluating such an interface will be an important next step.

What can be said in relation to teaching staff and this research, is that there was sufficient interest in the study from an enthusiastic and supportive teaching staff to see it through to completion; it simply would not have been possible to do this research without the support and assistance of the teaching team. Furthermore, a request for the system to be made available to students again, in the year following the study reported in this thesis, has been made (personal communication) and some of the teaching team are keen to take part in ongoing research and development. This bodes well and indeed is a prerequisite for the prospect of in-class adoption at some point in the future.

Post-hoc examination of the immediate post-test questions, which were prepared in consultation with teaching staff, revealed that they arguably only tested the surface recall of facts, even though some were open-ended questions. This is a well documented problem both globally (Miller *et al.*, 1998) and locally (Walker, Spronken-Smith, Bond, McDonald, Reynolds, and McMartin, 2010) and future work will need to address this. One possible explanation for the lack of difference in performance between students who completed the free-text tutorial and those who completed the menu-based version is that post-test questions were not discriminating sufficiently between surface recall of facts and deeper understanding. Another possible reason for the lack of difference between the two versions of the system is the poor performance of the system classifiers in-class relative to bench-tests. Since student performance is the same when the free-text tutorial is known to be performing in a sub-optimal fashion, does this mean that greater student learning gains may be achieved if the classification issues can be resolved and system performance increased? Again, this is an opportunity for further work. Of course, it is possible that there really is no difference between the two systems on immediate post-test; however on the basis of the present study it is not possible to conclude this, even though this result is consistent with two other similar studies (Corbett *et al.*, 2006; Aleven *et al.*, 2004).

Similarly, this study was unable to show any benefit for either system for student

performance on the delayed post-test, which comprised the relevant sections of the final examination for the course and which was taken several weeks after the experimental intervention. The rationale for using parts of the examination as the delayed post-test was that if there was an effect from a relatively small intervention after such a period of time the effect was likely to have been large. In addition, utilising the final exam as a post-test was a way to ensure that there was minimal imposition on both staff and students at a very busy time of year. Given the overall lack of effect, the study was unable to reproduce the effects found in Corbett *et al.* (2006) which favoured the free-text condition for longer-term retention. The lesson learned in terms of the effect of timing when experimental interventions are conducted in-class suggest that a specially designed delayed post-test should be administered well before students enter their study period leading up to the final examination in order to see whether there is an effect.

Despite the limitations, this study makes a number of specific contributions to research in the field of tutorial dialogue systems and these are described in the next section, 8.5

## 8.5 Main contributions of this thesis

While the empiricist approach to the design of human-machine dialogue systems is not new and is arguably becoming increasingly popular (Manning and Schütze, 1999), the pragmatic tutorial dialogue system design presented here is new in the classroom context and has been inspired by the desire to create a system which addresses some of the practical and, especially for educators, theoretical objections which have resulted in mainstream education largely ignoring ITS to this point. In keeping with more conventional ITS, the new system provides individualised and automated feedback to student responses. However it does this through the combination of a directed tutorial script, which is based on a contemporary and well understood model of human dialogue (Core and Allen, 1997), and by maintaining a highly simplified model of the current dialogue state. There is no student, teaching or expert model other than that implied by the script author in writing questions, model answers, hints and feedback, and other than that implied by the students who interact with the system in writing responses, revised responses, statements and questions. This sets the system apart from many current ITS which aim to analyse and model, to a very fine level of detail, student, tutor and sometimes, expert actions. In short, the system has been designed to demand no

more from educators or students than is already demanded of them in a contemporary educational setting: asking and answering questions and learning through engaging in an ongoing dialogue.

The evaluation described in this study confirms that the system can be deployed in a large-class setting and, at least in the context of first-year health sciences undergraduate courses, is likely to find acceptance with students in addition to having a positive impact on their performance.

In addition to the direct benefits of the research there are two additional benefits. First is the development of a stable server-web client platform for the further study of tutorial dialogues using a design research approach (Van den Akker, 1999). It was a relatively simple matter to create two versions of the system, one for free-text input and the other to provide menu-based options and this idea can be readily extended. For example, the same system could easily be used to evaluate a number of possible alternative classifier options. The second benefit is the automated creation of a large corpus of student-machine dialogues. This can be used in a wide range of ways by both educators and system designers for further research and of course can easily be expanded as the system is deployed in new contexts.

Finally, this study is a small contribution towards getting automated tutorial dialogue systems back on the educational agenda, at least in the Australasian context, and any system which has been demonstrated to work in practice and which can be provided at reasonable cost should at least be on the agenda. Much has changed in terms of technology and its application in classroom settings since the 1980s and 1990s. The rise of MOOCs noted in the introduction is a very recent change, and any research which addresses the provision of individualised feedback will be relevant in this area. The often quoted issues of the difficulty and expense of creating ITS may also be changing; while there are still plenty of practical obstacles, in terms of development some things have got much easier. For example, it is doubtful whether 20 to 30 years ago, an individual PhD researcher could have designed, built, deployed and evaluated a new ITS or tutorial dialogue system in a large class setting. Today, with advanced web technologies, a profusion of open source NLP tools and other development tools, networked classrooms and homes, and students and teachers who are familiar with on-line educational supports and services, this thesis itself is evidence that it is possible. It is also humbling to realise what an extraordinary achievement, back in the 1970s, Jaime Carbonell's *Scholar* really was.

## 8.6 Some closing remarks

There is clearly much work still to do, especially in improving the text classification part of the new system and in making it possible for teaching staff to create dialogues. Nevertheless, a firm base from which to develop has been established and there is enthusiasm for ongoing use and development from the teaching team of at least one very large class at my own institution. Hopefully, there will be more. There is strong appeal for a system which (ideally) requires no more from teachers than to pose questions and provide feedback, hints and ask further questions, and no more from students than to provide feedback, hints and pose further questions. No educational technology should demand more from educators or from students than is already demanded from them in the normal course of their teaching and study.

For me, the most exciting part of the project has been learning much more about, and realising the potential of, tutorial dialogue; between humans and between humans and machines. Beginning to understand the subtle and complex interactions that occur when we have a conversation seems to hold some promise for understanding aspects of how we learn, both as a teacher and as a student. Indeed, much of the appeal of tutorial dialogue, is that the roles of student and teacher are so often reversed and reversed again, and again. In many ways, this project too has felt like a dialogue; a dialogue between students, teachers, my supervisors, me and a very limited little machine, and like a good dialogue, I hope it has something to offer.

# References

- Aleven, V., McLaren, B. M., Sewall, J., and Koedinger, K. R. (2009). A new paradigm for intelligent tutoring systems: Example-tracing tutors. *International Journal of Artificial Intelligence in Education*, 19(2), 105–154.
- Aleven, V., Ogan, A., Popescu, O., Torrey, C., and Koedinger, K. (2004). Evaluating the effectiveness of a tutorial dialogue system for self-explanation. In *Intelligent Tutoring Systems*, 443–454. Springer.
- Allen, J., Ferguson, G., and Stent, A. (2001). An architecture for more realistic conversational systems. In *Proceedings of the 6th international conference on Intelligent user interfaces*, 1–8. ACM.
- Anderson, J. R., Conrad, F. G., and Corbett, A. T. (1989). Skill acquisition and the LISP tutor. *Cognitive Science*, 13(4), 467–505.
- Austin, J. L. (1975). *How to do things with words*, Volume 88. Cambridge, MA: Harvard University Press.
- Bangert-Drowns, R. L., Kulik, C.-L. C., Kulik, J. A., and Morgan, M. (1991). The instructional effect of feedback in test-like events. *Review of educational research*, 61(2), 213–238.
- Benjamin, L. (1988). A history of teaching machines. *American Psychologist*, 43(9), 703–712.
- Biggs, J. (1999). *Teaching for Quality Learning at University : What the Student Does* (1st ed.). Buckingham: Society for Research into Higher Education : Open University Press.
- Bird, S. (2006). NLTK: the natural language toolkit. In N. Calzolari, C. Cardie, and P. Isabelle (Eds.), *Proceedings of the COLING/ACL on Interactive presentation*

- sessions, COLING-ACL '06, Stroudsburg, PA, USA, 69–72. Association for Computational Linguistics.
- Bird, S., Klein, E., and Loper, E. (2009). *Natural language processing with Python*. Sebastopol, CA: O'Reilly Media.
- Black, P. and William, D. (2004). The formative purpose: assessment must first promote learning. *Yearbook of the National Society for the Study of Education*, 103(2), 20–50.
- Bloom, B. S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13(6), 4–16.
- Bobrow, D. G., Kaplan, R. M., Kay, M., Norman, D. A., Thompson, H., and Winograd, T. (1977). GUS, a frame-driven dialog system. *Artificial intelligence*, 8(2), 155–173.
- Brown, J. S., Collins, A., and Duguid, P. (1989). Situated cognition and the culture of learning. *Educational researcher*, 18(1), 32–42.
- Butcher, P. G. and Jordan, S. E. (2010). A comparison of human and computer marking of short free-text student responses. *Computers & Education*, 55(2), 489–499.
- Carbonell, J. R. (1970). AI in CAI: An artificial-intelligence approach to computer-assisted instruction. *Man-Machine Systems, IEEE Transactions on*, 11(4), 190–202.
- Chi, M., VanLehn, K., and Litman, D. (2010). Do micro-level tutorial decisions matter: Applying reinforcement learning to induce pedagogical tutorial tactics. In *Intelligent Tutoring Systems*, 224–234. Springer.
- Chi, M. T. H. (2009). Active-Constructive-Interactive: a conceptual framework for differentiating learning activities. *Topics in Cognitive Science*, 1(1), 73–105.
- Chi, M. T. H., Roy, M., and Hausmann, R. G. M. (2008). Observing tutorial dialogues collaboratively: insights about human tutoring effectiveness from vicarious learning. *Cognitive Science: A multidisciplinary journal*, 32(2), 301–341.
- Chomsky, N. (1986). *Knowledge of language: Its nature, origin, and use*. New York: Praeger Publishers.
- Clark, A. and Lappin, S. (2010). *Linguistic Nativism and the Poverty of the Stimulus*. Boston, MA: Wiley-Blackwell.

- Clark, H. H. and Schaefer, E. F. (1989). Contributing to discourse. *Cognitive Science*, 13(2), 259–294.
- Coates, H., James, R., and Baldwin, G. (2005). A critical examination of the effects of learning management systems on university teaching and learning. *Tertiary Education & Management*, 11(1), 19–36.
- Cohen, P. A., Kulik, J. A., and Kulik, C.-L. C. (1982). Educational outcomes of tutoring: A meta-analysis of findings. *American educational research journal*, 19(2), 237–248.
- Cohen, P. R. and Perrault, C. R. (1979). Elements of a plan-based theory of speech acts. *Cognitive science*, 3(3), 177–212.
- Collins, A. and Brown, J. S. (1988). The computer as a tool for learning through reflection. In H. Mandl and A. Lesgold (Eds.), *Learning Issues for Intelligent Tutoring Systems*, 1–18. Springer.
- Collins, A., Brown, J. S., and Newman, S. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L. Resnick (Ed.), *Knowing, learning, and instruction: Essays in honor of Robert Glaser*, Volume 18, 32–42. NJ: Lawrence Erlbaum Hillsdale.
- Corballis, M. (2006). History of cognitive psychology in New Zealand. In C. Fletcher-Finn and G. Haberman (Eds.), *Cognition and Language: Perspectives from New Zealand* (1st ed.), 1–10. Bowen Hills, Qld.: Australian Academic Press.
- Corbett, A., Wagner, A., Lesgold, S., Ulrich, H., and Stevens, S. (2006). The impact on learning of generating vs. selecting descriptions in analyzing algebra example solutions. In S. Barab, K. Hay, and D. Hickey (Eds.), *Proceedings of the 7th international conference on Learning sciences*, ICLS '06, 99–105. International Society of the Learning Sciences.
- Core, M. G. and Allen, J. F. (1997). Coding Dialogs with the DAMSL Annotation Scheme. In D. Traum (Ed.), *Working Notes of the AAAI Fall Symposium on Communicative Action in Humans and Machines*, Cambridge, MA, 28–35.
- Crain, S. and Pietroski, P. (2001). Nature, nurture and universal grammar. *Linguistics and Philosophy*, 24(2), 139–186.

- Crooks, T. J. (1988). The impact of classroom evaluation practices on students. *Review of Educational Research*, 58(4), pp. 438–481.
- Evens, M. and Michael, J. (2006). *One-on-One Tutoring by Humans and Computers*. NJ: Lawrence Erlbaum Associates.
- Falchikov, N. (2001). *Learning together: Peer tutoring in higher education*. London: Routledge Falmer.
- Fillmore, C. J. (1968). The case for case. In E. Bach and R. Harms (Eds.), *Universals in Linguistic Theory*. New York: Holt, Rinehart and Winston.
- Frederiksen, N. (1984). The real test bias: influences of testing on teaching and learning. *American Psychologist*, 39(3), 193–202.
- Gay, L. R. (1980). The Comparative Effects of Multiple-choice Versus Short-answer Tests on Retention. *Journal of Educational Measurement*, 17(1), 45–50.
- Ginzburg, J. and Fernández, R. (2010). Computational Models of Dialogue. In A. Clark, C. Fox, and S. Lappin (Eds.), *The Handbook of Computational Linguistics and Natural Language Processing*, Volume 57 of *Blackwell Handbooks in Linguistics*, 429. Wiley-Blackwell.
- Gipps, C. V. (2005). What is the role for ICTbased assessment in universities? *Studies in Higher Education*, 30(2), 171–180.
- Graesser, A. C., Hu, X., Susarla, S., Harter, D., Person, N., Louwerse, M., Olde, B., et al. (2001). AutoTutor: An intelligent tutor and conversational tutoring scaffold. In *10th ICAI in Education*, 47–49.
- Grice, H. P. (1978). Some further notes on logic and conversation. In P. Cole (Ed.), *Pragmatics: Syntax and Semantics*, Volume 9. Academic Press.
- Grosz, B. J. and Sidner, C. L. (1986). Attention, intentions, and the structure of discourse. *Computational linguistics*, 12(3), 175–204.
- Hattie, J. (2009). *Visible Learning: A synthesis of over 800 meta-analyses relating to achievement*. New York: Routledge.
- Hattie, J. and Timperley, H. (2007). The power of feedback. *Review of educational research*, 77(1), 81–112.

- Hobbs, J. R. (1979). Coherence and coreference. *Cognitive science*, 3(1), 67–90.
- Japkowicz, N. (2000). The class imbalance problem: significance and strategies. In *Proceedings of the 2000 International Conference on Artificial Intelligence (ICAI2000)*, Volume 1, Las Vegas, 111–117.
- Jordan, P. (2007). Tools for authoring a dialogue agent that participates in learning studies. In R. Luckin, K. Koedinger, and J. Greer (Eds.), *Artificial Intelligence in Education (AIED 07)*, 43–50.
- Jurafsky, D. and Martin, J. (2009). *Speech and language Processing*. New Jersey: Prentice Hall.
- Kumar, R., Rosé, C. P., Wang, Y.-C., Joshi, M., and Robinson, A. (2007). Tutorial dialogue as adaptive collaborative learning support. In R. Luckin, K. Koedinger, and J. Greer (Eds.), *Frontiers in artificial intelligence and applications*, Volume 158, 383–390. Virginia: IOS Press.
- Lajoie, S. P. and Derry, S. J. (1993). *Computers as cognitive tools*, Volume 1. New York and London: Routledge.
- Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 33(1), 159–174.
- Laurillard, D. (1988). The pedagogical limitations of generative student models. *Instructional Science*, 17(3), 235–250.
- Laurillard, D. (2002). *Rethinking University Teaching* (2nd ed.). London: Routledge Falmer.
- Leacock, C. and Chodorow, M. (2003). C-rater: Automated scoring of short-answer questions. *Computers and the Humanities*, 37(4), 389–405.
- Lesgold, A., Lajoie, S., Bunzo, M., and Eggan, G. (1992). SHERLOCK: A Coached Practice Environment for an Electronics Troubleshooting Job. In J. Larkin and R. Chabay (Eds.), *Computer-Assisted Instruction and Intelligent Tutoring Systems*, 201238. Hillsdale, N.J.: Lawrence Erlbaum.
- Lipnevich, A. A. and Smith, J. K. (2009). I really need feedback to learn: students perspectives on the effectiveness of the differential feedback messages. *Educational Assessment, Evaluation and Accountability*, 21(4), 347–367.

- Mann, W. C. and Thompson, S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text - Interdisciplinary Journal for the Study of Discourse.*, 8(3), 243–281.
- Manning, C. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press.
- Marton, F. and Saljo, R. (1976). On qualitative differences in learning: I Outcome and process. *British Journal of Educational Psychology*, 46(1), 4–11.
- McDaniel, M. A., Anderson, J. L., Derbish, M. H., and Morrisette, N. (2007). Testing the testing effect in the classroom. *European Journal of Cognitive Psychology*, 19(4-5), 494–513.
- McDonald, J., Knott, A., and Zeng, R. (2012). Free-text input vs menu selection: exploring the difference with a tutorial dialogue system. In P. Cook and S. Nowson (Eds.), *Proceedings of the Australasian Language Technology Association Workshop 2012*, Volume 149, Dunedin, 97–105.
- Merrill, D. C., Reiser, B. J., Ranney, M., and Trafton, J. G. (1992). Effective tutoring techniques: A comparison of human tutors and intelligent tutoring systems. *The Journal of the Learning Sciences*, 2(3), 277–305.
- Miller, A., Imrie, B., and Cox, K. (1998). *Student Assessment in Higher education: A handbook for Assessing Performance*. London: Kogan Page.
- Milne, A. A. (1928). *The House at Pooh Corner*. London: Methuen.
- Mitrovic, A. (2012). Fifteen years of constraint-based tutors: what we have achieved and where we are going. *User Modeling and User-Adapted Interaction*, 22(1-2), 39–72.
- Mitrovic, A., Martin, B., Suraweera, P., Zakharov, K., Milik, N., Holland, J., and Mcguigan, N. (2009). ASPIRE: an authoring system and deployment environment for constraint-based tutors. *International Journal of Artificial Intelligence in Education*, 19(2), 155–188.
- Murray, T. (1999). Authoring Intelligent Tutoring Systems: An Analysis of State of the Art. *International Journal of Artificial Intelligence in Education*, 10, 98–129.

- Nielsen, R. D., Ward, W., and Martin, J. H. (2008). Soft computing in intelligent tutoring systems and educational assessment. In B. Prasad (Ed.), *Soft Computing Applications in Business*, 201–230. Berlin Heidelberg: Springer.
- Norvig, P. (2013). How to write a spelling corrector. <http://norvig.com/spell-correct.html>. Online; accessed 20-June-2013.
- Ohlsson, S. (1994). Constraint-based student modeling. *NATO ASI Series F Computer and Systems Sciences*, 125, 167–167.
- Olney, A. M., Graesser, A. C., and Person, N. K. (2010). Tutorial dialog in natural language. In R. Nkambou, J. Bourdeau, and R. Mizoguchi (Eds.), *Advances in Intelligent Tutoring Systems*, 181–206. Springer.
- Passonneau, R. (2006). Measuring agreement on set-valued items (MASI) for semantic and pragmatic annotation. In N. Calzolari (Ed.), *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*, Genoa, Italy, 831–836.
- Pea, R. D. (2004). The social and technological dimensions of scaffolding and related theoretical concepts for learning, education, and human activity. *The Journal of the Learning Sciences*, 13(3), 423–451.
- Perkins, J. (2010). *Python Text Processing with NLTK 2.0 Cookbook*. Birmingham, UK: Packt Publishing Ltd.
- Perrault, C. R. and Allen, J. F. (1980). A plan-based analysis of indirect speech acts. *Computational Linguistics*, 6(3-4), 167–182.
- Pinker, S. (1984). *Language learnability and language learning*. Cambridge, Massachusetts: Harvard University Press.
- Pon-Barry, H., Clark, B., Schultz, K., Bratt, E. O., and Peters, S. (2004). Contextualizing learning in a reflective conversational tutor. In Kinshuk, C.-K. Looi, E. Sutinen, D. Sampson, I. Aedo, L. Uden, and E. Kähkönen (Eds.), *Proceedings. IEEE International Conference on Advanced Learning Technologies, 2004.*, Joensuu, Finland, 236–240. IEEE.
- Pressey, S. (1926). A simple apparatus which gives tests and scores - and teaches. *School and Society*, 23(585), 373–376.

- Pullum, G. K. (1996). Learnability, hyperlearning, and the poverty of the stimulus. In J. Johnson, J. Juge, and J. Moxley (Eds.), *Proceedings of the 22nd Annual Meeting: General Session and Parasession on the Role of Learnability in Grammatical Theory*, Berkley, 498–513. Berkeley Linguistics Society.
- R Core Team (2012). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0.
- Ramsden, P. (1992). *Learning to teach in higher education* (1st ed.). London: Routledge.
- Ramsden, P. (2003). *Learning to teach in higher education* (2nd ed.). London: Routledge Falmer.
- Reeves, T. C. (2006). Design research from a technology perspective. In J. V. den Akker, K. Gravemeijer, S. McKenney, and N. Nieveen (Eds.), *Educational design research*, 52–66. Oxon and New York: Routledge.
- Reeves, T. C. and Hedberg, J. G. (2003). *Interactive learning systems evaluation*. Englewood Cliffs, New Jersey: Educational Technology Publications.
- Rodriguez, M. C. (2003). Construct equivalence of multiple-choice and constructed-response items: a random effects synthesis of correlations. *Journal of Educational Measurement*, 40(2), 163–184.
- Rosé, C. P., Roque, A., Bhembé, D., and Vanlehn, K. (2003). A hybrid text classification approach for analysis of student essays. In *Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing-Volume 2*, 68–75. Association for Computational Linguistics.
- Sacks, H., Schegloff, E. A., and Jefferson, G. (1974). A simplest systematics for the organization of turn-taking for conversation. *Language*, 50(4), 696–735.
- Scardamalia, M., Bereiter, C., McLean, R. S., Swallow, J., and Woodruff, E. (1989). Computer-supported intentional learning environments. *Journal of educational computing research*, 5(1), 51–68.
- Schegloff, E. A. (1968). Sequencing in Conversational Openings. *American anthropologist*, 70(6), 1075–1095.
- Searle, J. R. (1975). Indirect speech acts. *Syntax and semantics*, 3, 59–82.

- Searle, J. R. (1976). A classification of illocutionary acts. *Language in society*, 5(01), 1–23.
- Self, J. A. (1990). Bypassing the intractable problem of student modelling. In C. Frasson and G. Gauauthier (Eds.), *Intelligent tutoring systems: At the crossroads of artificial intelligence and education.*, 107–123. Ablex.
- Shute, V. and Psotka, J. (1994). Intelligent Tutoring Systems: Past, Present and Future. Technical report, Human Resources Directorate, Manpower and Personnel Research Division, Brooks Air Force Base.
- Shute, V. J. (2008). Focus on formative feedback. *Review of educational research*, 78(1), 153–189.
- Shute, V. J. and Regian, J. W. (1993). Principles for evaluating intelligent tutoring systems. *Journal of Artificial Intelligence in Education*, 4(2-3), 245–271.
- Shute, V. J. and Zapata-Rivera, D. (2010). Educational measurement and intelligent systems. In P. Peterson, E. L. Baker, and B. McGaw (Eds.), *International Encyclopedia of Education*. Oxford, UK: Elsevier Publishers.
- Skinner, B. F. (1957). *Verbal behavior*. New York: Appleton-Century-Crofts.
- Skinner, B. F. (1958). Teaching machines. *Science*, 128(3330), 969–977.
- Stalnaker, R. C. (1978). Assertion. In P. Cole (Ed.), *Pragmatics: Syntax and Semantics*, Volume 9, 315–332. Academic Press.
- Stemler, S. (2001). An overview of content analysis. *Practical Assessment, Research and Evaluation*, 17(7), 137–146.
- Tamim, R. M., Bernard, R. M., Borokhovski, E., Abrami, P. C., and Schmid, R. F. (2011). What forty years of research says about the impact of technology on learning. A second-order meta-analysis and validation study. *Review of Educational Research*, 81(1), 4–28.
- Topping, K. J. (1996). The effectiveness of peer tutoring in further and higher education: A typology and review of the literature. *Higher education*, 32(3), 321–345.
- Traum, D. R. and Larsson, S. (2003). The information state approach to dialogue management. In J. Kuppevelt and S. R. W (Eds.), *Current and new directions in discourse and dialogue*, 325–353. Dordrecht: Springer.

- Van den Akker, J. (1999). Principles and methods of development research. In J. van den Akker, R. M. Branch, K. Gustafson, N. Nieveen, and T. Plomp (Eds.), *Design approaches and tools in education and training*, 1–14. The Netherlands: Kluwer Academic Publishers.
- VanLehn, K. (2011). The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems. *Educational Psychologist*, 46(4), 197–221.
- VanLehn, K., Graesser, A. C., Jackson, G. T., Jordan, P., Olney, A., and Rosé, C. P. (2007). When are tutorial dialogues more effective than reading? *Cognitive Science*, 31(1), 3–62.
- VanLehn, K., Jordan, P., and Litman, D. (2007). Developing pedagogically effective tutorial dialogue tactics: Experiments and a testbed. In M. Eskenazi (Ed.), *Proceedings of SLATE Workshop on Speech and Language Technology in Education ISCA Tutorial and Research Workshop*, 17–20.
- VanLehn, K., Jordan, P. W., Rosé, C. P., Bhembé, D., Böttner, M., Gaydos, A., Makatchev, M., Pappuswamy, U., Ringenberg, M., Roque, A., *et al.* (2002). The architecture of Why2-Atlas: A coach for qualitative physics essay writing. In S. Cerri, G. Gouarderes, and F. Paraguacu (Eds.), *Proceedings of the Sixth International Conference on Intelligent Tutoring Systems*, Berlin, 158–167. Springer-Verlag.
- Walker, R., Spronken-Smith, R., Bond, C., McDonald, F., Reynolds, J., and McMartin, A. (2010). The impact of curriculum change on health sciences first year students approaches to learning. *Instructional Science*, 38(6), 707–722.
- Weizenbaum, J. (1966). ELIZAa computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1), 36–45.
- Wenger, E. (1987). *Artificial intelligence and tutoring systems : computational and cognitive approaches to the communication of knowledge*. Los Altos, California: Morgan Kaufmann.
- Wittgenstein, L. (1968). *Philosophical investigations*. Oxford: Basil Blackwell.
- Woolf, B. P. (2008). *Building Intelligent Interactive Tutors*. Morgan Kaufman.

# Appendix A

## Dialogue script and dialogue manager source

### A.1 Homeostasis script XML file

Please visit the following web link in order to access the XML source file for the dialogue script: <https://app.box.com/s/nj8u49haauj797eaj9va>

### A.2 Tutorial dialogue system XML style file

Please visit the following web link in order to access the XSD source file for the dialogue script: <https://app.box.com/s/dobvdosc8hyrimqz6qeu>

### A.3 Dialogue manager source code

Please visit the following web link in order to access the .py source file for the dialogue manager: <https://app.box.com/s/c2yblnwkgrrmddvz5p64e>

## A.4 Index of classifiers and associated questions

Table A.1: Question, classifier label and classifier type

| Question   | Classifier label   | Classifier type |
|--|--------------------|-----------------|
| Does parasympathetic activity increase or decrease?  | baro-parasymp      | multilabel      |
| If the baroreceptors detect an increase in BP will they trigger a sympathetic response, a parasympathetic response or both?                          | baro-resp-inc      |                 |
| What is the name given to the BP at its highest point?   | bp-syst            |                 |
| How is systemic blood flow arranged between different organs?  | circ-design        |                 |
| The right side of the heart pumps blood to which circulation?  | circ-pulm          |                 |
| The left side of the heart pumps blood to where?   | circ-sys           |                 |
| What is meant by a 'negative feedback loop'?   | feedback-loop      |                 |
| Can you feel a pulse in someone's vein?  | vein               |                 |
| What feature of veins prevents blood from flowing backwards, away from the heart?  | venous-valves      |                 |
| Assuming nothing else changes, if TPR increases, do you think the heart would have to do more or less work to maintain the same cardiac output (CO). | work-heart         |                 |
| -  | yes-no*            |                 |
| What is the effect of an increase in parasympathetic stimulation on HR?  | baro-resp-inc-para |                 |
| What is the name given to describe blood pressure at its lowest point?   | bp-diast           |                 |
| Will sympathetic activity be increased or decreased?   | baro-symp          |                 |
| What is the main effect of a reduction in parasympathetic activity?  | parasymp-dec       |                 |
| Is the flow, or cardiac output, the same in the systemic and pulmonary circulations?   | same-flow          |                 |
| Please calculate the stroke volume for an average adult.   | adult-sv           |                 |
| Can you describe what is meant by contractility?   | inotropic-state    |                 |
| Please calculate the CO of a normal adult rounded to the nearest litres/minute.  | adult-co           |                 |
| What feature of artery walls allows us to feel the pulse?  | artery-wall        |                 |
| Where in the body would you put baroreceptors?   | baro-loc           |                 |
| How would you check someone's HR?  | check-hr           |                 |
| Can you explain why you cannot feel a pulse in someone's vein?   | vein-why           |                 |
| What is the pulse?   | what-pulse         |                 |
| How is MABP estimated in clinical practice?  | bp-map             |                 |
| What are the three cardiovascular variables controlled by the ANS?   | ans-vars           | multi-binary    |
| An increase in sympathetic activity will result in an increase in which three variables?   | symp-vars          |                 |
| Can you think of the three main factors which affect cardiac contractility?  | inotropic-factors  |                 |
| -  | question           | binary          |
| -  | dont-know          |                 |
| -  | dont-understand    |                 |

\*Note: the same yes/no classifier is used for a number of yes/no questions.

# Appendix B

## Term definitions for Chapter 5: Classifier and script design case-studies

### B.1 Question categories

- **Binary:** A question which requires a particular response and the response is either present or not, or takes one of two possible values. For example, 'Has the goat eaten my tree?' would typically elicit a yes or no response, although it may also elicit a clarification question such as, 'Which tree is your tree?'.
- **Multi-part:** A question which requires several components in a response. For example, 'Describe three signs of irritation in a goat' is a multi-part question.
- **Open:** A question which requires an expression of opinion, development of ideas, justifying a statement or applying a principle. There will be a potentially wide range of responses.

It is important to note that each of these question categories does not preclude any response. In other words, while a question may be designed to elicit a particular kind of response, this does not guarantee that this is the response that will be received.

### B.2 Classifier types

- **Binary:** A classifier which detects the presence of a particular feature set in the input text. If the feature set is present then the text is classified. If the feature

set is not present then the text is not classified.

- **Multilabel:** A classifier which assigns input text to two or more predefined classes or categories. If the text is not assigned to any class then it is not classified.
- **Multi-binary:** A classifier which is a series of binary classifiers used together to determine the presence or absence of a set of responses in the input text. This approach can provide performance gains over using a multilabel classifier where the training set is relatively small. (Refer Section 4.5.2 for detailed description.)

## B.3 Training dataset description and class selection

- **Unique responses:** Refers to the particular responses given to a question. Several respondents may give the same particular response. It is the response which is unique in the context of the question even though several respondents may give the same response.
- **Class:** Refers to a group or category of unique responses. These groups are designed by hand for any given question and generally contain text which, while different in form, shares a common meaning.
- **Label:** The specific name given to a particular class.

# Appendix C

## Information for Students and Evaluation Summary

### C.1 Information for students

#### Cardiovascular Homeostasis Tutorial



Go to:

<http://harambe.otago.ac.nz/dialogue-mb/>

Click 'Start' to start the tutorial

Login with Uni username and password

Takes 20-30 minutes to complete

Short-answer and MCQ style questions – with feedback

Really helpful if you complete all questions! :)

You can only do this tutorial once BUT happy to make tutorial available after the experimental period (24<sup>th</sup> July - 10<sup>th</sup> August) as well if students would like this for revision.

Questions/Feedback e-mail [jenny.mcdonald@otago.ac.nz](mailto:jenny.mcdonald@otago.ac.nz)

Figure C.1: Information for students - Page 1

# Cardiovascular Homeostasis Tutorial

## **Project Background and Ethics Information.**

This web site is part of a PhD research project which aims to investigate the use of different pedagogical strategies by natural language intelligent tutoring systems. Currently the site is in prototype form and is made available on an "as-is" basis. It has been designed to assist with aspects of cardiovascular coursework for HUBS 192 but no reliance should be placed on it being available at any particular time during semester 2 or during the examination period.

All interactions with this system are logged but all contributions will be anonymised prior to use in any research outputs. By using this system you understand that interactions you have with the system will be recorded and may be used or referred to, in anonymous form, in published or unpublished research outputs. Category B Ethical approval for this project has been obtained at departmental level.

Any feedback on the system by e-mail is very welcome. All comments should be e-mailed to [jenny.mcdonald@otago.ac.nz](mailto:jenny.mcdonald@otago.ac.nz).

The prototype natural language tutorial system and cardiovascular homeostasis tutorial dialogue has been developed by Jenny McDonald, Educational Media, HEDC as part of her PhD research.

The web application to make tutorials available to Otago Students has been developed by Richard Zeng in Educational Media, HEDC and web interface design by Ayelet Cohen, Educational Media, HEDC.

This project is supervised Dr Alistair Knott, Computer Science and Dr Sarah Stein, HEDC. We are most grateful to Dr Pamela Jordan from the Learning Research and Development Centre at the University of Pittsburgh, USA., for her support and permission to use their TuTalk system in early student trials.

We are very grateful for the support and encouragement of the HUBS teaching staff, in particular, Dr Zoe Ashley, Dr Rebecca Bird, Dr Ruth Napper, Prof. Fiona McDonald and Assoc. Prof. John Reynolds, and to Dr Simon Green and Dr Greg Jones for permission to review their lecture notes on the cardiovascular system to support this tutorial.

Figure C.2: Information for students - Page 2

## C.2 Student evaluation summary

Higher Education Development Centre On-Line Evaluation Questionnaire

Online Cardiovascular Homeostasis Tutorial Evaluation

456 - Students

105 - Responses

23 - % Class

Q.ID: 20121047

Date of Survey: 12 Oct 2012

Note: For comments questions, numbers assigned to respondents, such as 1) or 3), will be missing if they have made no comment

|   |   |                 |     |     |     |    |    |               |     |        |
|---|---|-----------------|-----|-----|-----|----|----|---------------|-----|--------|
| 1 | I would rate the online tutor as an aid for learning: | Very useful     | 1   | 2   | 3   | 4  | 5  | Of little use | NIL | Median |
|   |   | Number          | 40  | 42  | 18  | 3  | 2  |               | 0   | 1.8    |
|   |   | Distribution 1  | 38% | 40% | 17% | 3% | 2% |               | 0%  |        |
|   |   | Distribution 2* |     | 78% | 17% | 5% |    |               |     |        |

|   |  |                 |     |     |     |     |    |                   |     |        |
|---|--|-----------------|-----|-----|-----|-----|----|-------------------|-----|--------|
| 2 | I learned new things about cardiovascular homeostasis from the online tutor: | Strongly agree  | 1   | 2   | 3   | 4   | 5  | Strongly disagree | NIL | Median |
|   |  | Number          | 17  | 41  | 35  | 9   | 3  |                   | 0   | 2.4    |
|   |  | Distribution 1  | 16% | 39% | 33% | 9%  | 3% |                   | 0%  |        |
|   |  | Distribution 2* |     | 55% | 33% | 11% |    |                   |     |        |

|   |   |                 |     |     |     |    |    |                   |     |        |
|---|---|-----------------|-----|-----|-----|----|----|-------------------|-----|--------|
| 3 | I felt the tutor helped me to understand things about cardiovascular homeostasis that we had covered in the course: | Strongly agree  | 1   | 2   | 3   | 4  | 5  | Strongly disagree | NIL | Median |
|   |   | Number          | 28  | 49  | 25  | 2  | 0  |                   | 1   | 2.0    |
|   |   | Distribution 1  | 27% | 47% | 24% | 2% | 0% |                   | 1%  |        |
|   |   | Distribution 2* |     | 73% | 24% | 2% |    |                   |     |        |

|   |  |                 |     |     |     |     |    |                   |     |        |
|---|--|-----------------|-----|-----|-----|-----|----|-------------------|-----|--------|
| 4 | I felt like the tutor could understand my answers: | Strongly agree  | 1   | 2   | 3   | 4   | 5  | Strongly disagree | NIL | Median |
|   |  | Number          | 21  | 39  | 29  | 12  | 4  |                   | 0   | 2.3    |
|   |  | Distribution 1  | 20% | 37% | 28% | 11% | 4% |                   | 0%  |        |
|   |  | Distribution 2* |     | 57% | 28% | 15% |    |                   |     |        |

|   |                                |                 |    |     |     |    |    |                |     |        |
|---|--------------------------------|-----------------|----|-----|-----|----|----|----------------|-----|--------|
| 5 | I found the tutor's questions: | Very easy       | 1  | 2   | 3   | 4  | 5  | Very difficult | NIL | Median |
|   |                                | Number          | 4  | 23  | 69  | 9  | 0  |                | 0   | 2.9    |
|   |                                | Distribution 1  | 4% | 22% | 66% | 9% | 0% |                | 0%  |        |
|   |                                | Distribution 2* |    | 26% | 66% | 9% |    |                |     |        |

|   |   |                 |     |     |     |    |    |                  |     |        |
|---|---|-----------------|-----|-----|-----|----|----|------------------|-----|--------|
| 6 | Overall, in helping me to revise my understanding of cardiovascular homeostasis I found the tutorial: | Very effective  | 1   | 2   | 3   | 4  | 5  | Very ineffective | NIL | Median |
|   |   | Number          | 31  | 46  | 22  | 4  | 2  |                  | 0   | 2.0    |
|   |   | Distribution 1  | 30% | 44% | 21% | 4% | 2% |                  | 0%  |        |
|   |   | Distribution 2* |     | 73% | 21% | 6% |    |                  |     |        |

|   |   |                 |     |      |  |  |  |  |  |     |        |
|---|---|-----------------|-----|------|--|--|--|--|--|-----|--------|
| 7 | I would recommend the tutorial to other students: | Yes             | Yes | No   |  |  |  |  |  | NIL | Median |
|   |   | Number          | 99  | 6    |  |  |  |  |  | 0   | 1.0    |
|   |   | Distribution 1  | 94% | 6%   |  |  |  |  |  | 0%  |        |
|   |   | Distribution 2* |     | 100% |  |  |  |  |  |     |        |

|   |   |  |  |  |  |  |  |  |  |  |
|---|---|--|--|--|--|--|--|--|--|--|
| 8 | If you did not complete the tutorial can you say why? For example was it because of technical issues or because you didn't find the tutorial helpful or some other reason?  |  |  |  |  |  |  |  |  |  |
|   | 2) Didn't find the utotial all that helpful.  |  |  |  |  |  |  |  |  |  |
|   | 4) The tutorial is non-stop. Essentially, I didn't know how long it took. Otherwise I am very satisfied with this program.  |  |  |  |  |  |  |  |  |  |
|   | 46) I heard a lot of positive comments about the tutorial so I wanted to do it. It took a while before I got it to work. I found it very helpfull but then it got stuck half way through. I did try several times to start it again but unsuccessful. I was dissapointed about the technical problem. |  |  |  |  |  |  |  |  |  |
|   | 48) Needs to be a bit longer to have an impact and the tutor often didn't understand my answers. was good for seeing how much I understood from the course though   |  |  |  |  |  |  |  |  |  |
|   | 55) I didnt complete it because a page refreshed and it took me back to the start.  |  |  |  |  |  |  |  |  |  |

H.E.D.C.  
University of Otago

\*Distribution 2 shows the responses as %(1&2), %(3) and %(4&5).  
The "Median" calculation is an interpolated median.

Page 1  
Printed: 29/10/2012

Figure C.3: Questionnaire Summary - Page 1

Higher Education Development Centre On-Line Evaluation Questionnaire

Online Cardiovascular Homeostasis Tutorial Evaluation

456 - Students  
105 - Responses  
23 - % Class

Q.ID: 20121047  
Date of Survey: 12 Oct 2012

- 
- 69) I couldn't complete the tutorial due to technical issues- there would be no answer options shown, though the buttons to select an answer were there, but since i couldn't see potential answers I stopped after the first slide.
- 92) I completed the tutorial
- 98) ran out of study time
- 9 Any additional comments or feedback?
- 2) It used abbreviations and didn't explain them, in a test this is understandable but in a tutorial it would help further leaning to have the abbreviations explained.
- 4) I hope in addition to words of explanation from the tutor, diagrams, demonstrations and videos related to the questions tutor was asking could be used as an additional effective tool of helping explanation. Thanks :)
- 5) ggreat tutorial thankyou very much
- 9) It was quite long, maybe split into parts for people to be able to do bits at a time
- 11) Would be great to have access to this tutor for revision purposes!!
- 12) Comments were kind of patronizing, pretty frustrating
- 13) Hope to see more tutes like these for other topics
- 15) There were a couple of questions that were difficult to understand what was being asked of me - maybe incorporated a way of asking to rephrase the question? The addition of picture/animations would also be helpful in learning as well as probably assist in understanding what is being asked.
- 19) Very good and useful programme
- 21) Maybe do the survey/review a little earlier next time! I can hardly remember doing it!
- 22) The tutor's questions were not always as leading as I am sure they were meant to be, which I found quite confusing, so eventually gave up with some answers and just typed randomly until it told me the answer and let me move on. Also, there were some cases where I had half the answer right, but the tutor told me I was wrong overall, which made me second-guess the whole answer, when I should have only been correcting half of it. Perhaps a system could be put in place where it also points out which parts you got right, or it's very confusing when trying to get it right after that.
- 33) I would suggest that this tutor programme be available for the other modules of HUBS 191 and 192 as well. I believe they will help us tremendously.
- 34) Pretty good recognition to be honest. Was frustrating when the tutor did not understand, but this was only in some questions where it felt like the tutor wanted the answer in one specific way
- 35) The tutor was a bit frustrating at times. Overall this was really helpful though, especially in pointing out bits of the module that I perhaps didn't know as well as I should. Thanks :)
- 39) There were some technical issues with several questions but overall it was useful
- 46) Would be nice if you could make it available this week to refresh the cardiovascular system before the exam.=)
- 55) I wish the tutorial was still available for exam preparation.
- 56) It would have been good to be able to do it again, for example now when we are studying for the final exam :)
- 59) cant remember, it was too long ago.
- 62) It's a good idea but would be more useful if we could use it more than once and if there was a range of the most important concepts within the cardiovascular system as different tutorials
- 63) Can be longer or more in depth but good starting basis - thanks!
- 69) It's a good idea- access to extra resources by the department is always great
- 71) Resources like this are really helpful, it gave me an idea of things I had to study for which I wasn't aware that my knowledge was lacking in to begin with. Would be great to have these for all the topic in hubs!
- 72) The conversation part was having a lot of difficulties loading and I had to refresh the page from start again many times. However, it could be because the internet at the place I live is quite slow.
- 75) I did find that the tutor sometimes didn't get my answers properly, apart from that it worked ok.
- 78) I thought it was pretty good, Well done.
- 83) Wish we had this across all topics! Thank you
- 89) Great learning aid!
- 90) found it pretty cool how it explained things as you went and speciefc the parts of the question that you got right and didn't get right so that you didn't have to re write all the parts you did know and waste time
- 91) i thought it was very good, although from memory i found 1 or 2 of the questions hard to understand what kind of answer it was looking for

H.E.D.C.  
University of Otago

\*Distribution 2 shows the responses as %(1&2), %(3) and %(4&5).  
The "Median" calculation is an interpolated median.

Page 2  
Printed: 29/10/2012

Figure C.4: Questionnaire Summary - Page 2

Higher Education Development Centre On-Line Evaluation Questionnaire

Online Cardiovascular Homeostasis Tutorial Evaluation

456 - Students  
105 - Responses  
23 - % Class

Q.ID: 20121047  
Date of Survey: 12 Oct 2012

- 
- 92) I like how the GLMs enable more than one attempt at answering the questions. As a tutorial it would be good if students could pause the tutorial and start it again from the last question they completed at a later time/date. It would also be good if the tutorial was available for multiple attempts, as I wanted to redo the tutorial to make sure I had learned the content and could answer the questions easier in a following attempt.
- 93) a bit frustrating with replies not being understood
- 98) no
- 99) It would be good if there were more questions that are more comprehensive.

Processed by: *AG (Allen Goodchild)*

Figure C.5: Questionnaire Summary - Page 3

# Appendix D

## Pre-test and Post-test Questions and Model Answers

### D.1 Pre-test and model answers (in bold)

1. Which of the following events occurs in the ventricles during systole?
  - A. All four heart valves close
  - B. **The AV valves close and the semilunar valves open**
  - C. The AV valves open and the semi lunar valves close
  - D. All four heart valves open
2. The normal pacemaker of the heart is:
  - A. **The SA node**
  - B. the AV node
  - C. the ventricles
  - D. the Purkinje fibres
3. The layer of the heart responsible for contraction is the:
  - A. pericardium
  - B. endocardium
  - C. epicardium
  - D. **myocardium**

4. What are the two structures within the circulatory system that enable blood to flow in a one-way circuit?

**Heart valves and venous valves**

5. Which ventricle of the heart is the apex associated with?

**Left**

6. Cardiac Output can be calculated using Heart Rate and what other variable?

**Stroke Volume**

## **D.2 Immediate post-test and model answers (in bold)**

1. The pressure in arteries and veins is:

A. equal

B. higher in veins

C. **higher in arteries**

D. not detectable in veins

2. A decrease in blood pressure would result in:

A. **an increase in heart rate**

B. a decrease in heart rate

C. a decrease in stroke volume

D. a decrease in sympathetic activity

3. The term used to describe blood pressure at its highest point is:

A. diastolic pressure

B. **systolic pressure**

C. cardiac output

D. MABP

4. The MABP of a person with a blood pressure of 130/100 would be:

A. 105

B. **110**

- C. 115
  - D. 120
5. If the CO of a person drops by  $1/4$ , to  $3/4$  of its original level what must happen to their TPR in order to restore their MABP to its original level? (Remember  $MABP = CO \times TPR$ )
    - A. **TPR must increase by  $1/3$**
    - B. TPR remains the same
    - C. TPR must increase by  $1/4$
    - D. TPR must double
  6. In response to an increase in blood pressure:
    - A. parasympathetic activity decreases
    - B. sympathetic activity increases
    - C. **parasympathetic activity increases and sympathetic activity decreases**
    - D. sympathetic activity increases and parasympathetic activity decreases.
  7. A response via a negative feedback loop aims to:
    - A. eliminate receptor activity
    - B. increase receptor activity
    - C. increase the change in a variable in the same direction
    - D. **stabilise a variable**
  8. The pulmonary circuit receives blood from which side of the heart? **Right**
  9. The flow in Mr X's pulmonary circulation is 4.9L/min. What is his Cardiac output? **It is the same. 4.9L/min**
  10. What is the specific arrangement of blood flow between different organs of the body? **Blood flow between body organs is arranged in parallel**
  11. The measurement of somebodys pulse rate is a good indication of what variable of cardiac activity: Heart rate

12. Ms Y has a heart rate of 70 beats/min at rest. During exercise her cardiac output increases from 5Litres/min at rest to 6Litres/min. If her stroke volume remains the same please calculate her heart rate during exercise. **84 beats/min**
13. What is the term used to describe the difference between systolic and diastolic pressures in the aorta? **Pulse pressure**
14. The force generated by the heart when it contracts is affected by which part of the autonomic nervous system (ANS)? **The sympathetic NS**

### **D.3 Delayed post-test: selected CVS questions from HUBS final examination and model answers (in bold)**

1. Blood vessels that carry blood towards the heart are called:
  - A. **veins.**
  - B. arteries.
  - C. arterioles.
  - D. lymph vessels.
  - E. aorta.
2. An increased rate of firing in the vagus nerve would result in:
  - A. an increase in heart rate.
  - B. a decrease in contractility.
  - C. **a decrease in heart rate.**
  - D. an increase in contractility.
  - E. a decrease in heart rate and contractility.
3. With respect to the ventricles of the heart, which of the following statements is CORRECT?
  - A. The power of the left and right ventricles is similar.
  - B. **The power of the right ventricle is lower than that of the left ventricle.**

- C. Blood flow from the right and left ventricles is different.
- D. The pressure generated by the right and left ventricle contraction is the same.
- E. The power of the left ventricle is lower than that of the right ventricle.
4. During the marathon Annas cardiac output increases through large changes in ...and ...**heart rate (HR), stroke volume (SV)**
  5. In order to maintain mean arterial blood pressure homeostatically there must have been a large decrease in ...**total peripheral resistance (TPR)**
  6. During the marathon the metabolic demands of the skeletal muscle and heart will have resulted in ...of ...in these regions to increase blood flow to these areas.  
**vasodilation, arterioles**
  7. At the finish line there would have been a(n) ...in venous return as the skeletal muscle pump would have stopped working. This would result in severe ...and therefore reduced cardiac output, resulting in venous ...**decrease, pooling, hypotension**
  8. Describe changes in ventricular volume and pressure that occur during a single cardiac cycle. Include a brief description of the structures involved in creating one-way blood flow around the heart, and their positioning during the cardiac cycle.

**Model answer:** 4 valves: between the atria and ventricles (AV) and leaving the right (pulmonary) and left (aortic) ventricles. Valves structure bi- or tri-cuspid.

5 phases:

- Atrial systole: AV valves open, Aortic and pulmonary valves closed. Atria contract small increase in ventricular volume.
- Isovolumetric ventricular contraction: All valves are closed. No change in volume of blood in ventricles. Large increase in pressure as ventricle contraction decreases chamber size.
- Ejection: Pressure in ventricle exceeds the aortic pressure aortic and pulmonary valves open (AV stay closed). Blood leaves the ventricle decrease in ventricular volume.

- Isovolumetric ventricular relaxation: All valves closed. No change in volume but large decrease in ventricular pressure as ventricles relax and increase chamber size.
- Passive ventricular filling: AV valves open, increase in blood volume in ventricles with small increase in pressure.