Australasian Journal of
Educational Technology

2009, 25(2), 268-291

Improving critical thinking using web based argument mapping exercises with automated feedback

Sam Butchart, Daniella Forster
Monash University

Ian Gold
McGill University

John Bigelow, Kevin Korb, Graham Oppy
Monash University

Alexandra Serrenti
National University of Singapore

In this paper we describe a simple software system that allows students to practise their critical thinking skills by constructing argument maps of natural language arguments. As the students construct their maps of an argument, the system provides automatic, real time feedback on their progress. We outline the background and theoretical framework that led to the development of the system and then give a detailed example of how a student would work through a particular argument mapping exercise using the software. We then describe how the system was used in a single semester undergraduate critical thinking course. We evaluated the course using a standardised critical thinking test and measured an improvement in critical thinking skills of 0.45 standard deviations from pre-test to post-test; a modest, but encouraging result for a single semester course. We compare these results to those obtained in a number of other critical thinking courses, incorporating a variety of teaching methods. We conclude the paper with some comments on the limitations of the system and ways in which it might be improved and extended.

1. Background

A goal of higher education acknowledged by many educators is to train students to be *critical thinkers*. We would like our students to be able to think critically about the increasing amounts of information they are presented with from a variety of sources. Graduates should be skilled in evaluating claims, policies, decisions and arguments, not only in academic contexts but also in everyday life and the work place.

A necessary component of the ability to think critically about a claim, policy or decision is the ability to assess the *evidence* or *reasons* which might count for or against it. For that reason, the ability (and disposition) to analyse, construct and evaluate *arguments* is often considered to be a core component of critical thinking. (see Ennis, 1987 for a detailed analysis of the skills and dispositions commonly supposed to be involved in critical thinking).

Is it possible to improve students' ability to analyse and evaluate arguments? Some evidence suggests that this it is possible, and that what is required is extensive practice with appropriate feedback (see section 1.2). This raises a practical problem however. Higher education in Australia, as elsewhere, has become a mass education system; more people than ever before are now entering the universities (Australian Bureau of Statistics, 2009). At the same time, funding to universities has not kept pace with this increase in student numbers. There is pressure to teach more and more students with fewer and fewer resources. Gone are the days, if ever they existed, when students could be given much one on one personal tutoring.

Yet, if students are to improve their argument analysis skills they need a large amount of practice, with useful and timely feedback on their efforts. In these circumstances, some kind of automated feedback would clearly be very useful. How can this been done, when the task is as complex as identifying the structure of an argument? In this paper we describe a modest first step in the direction of providing useful automated feedback on students' attempts to map out the structure of arguments.

1.1 Computer assisted argument mapping software

Many universities in Australia and elsewhere now offer undergraduate courses in critical thinking (also known as *critical reasoning* and *informal logic*). Typically, the focus is on the analysis and evaluation of real arguments taken from articles, books, opinion pieces, editorials, letters to the editor and so on. Students are taught to identify the components of an argument (premises, main conclusion, intermediate conclusions, unstated premises), analyse their structure (how the components fit together) and evaluate them using a variety of informal and semi-formal methods.

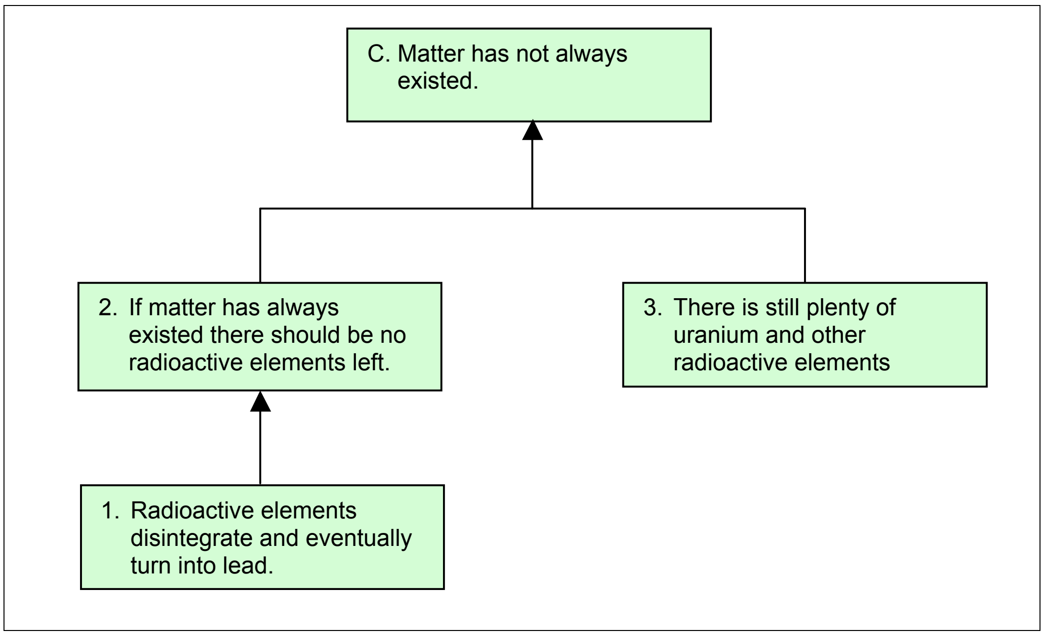

C. Matter has not always existed.
2. If matter has always existed there should be no radioactive elements left.
3. There is still plenty of uranium and other radioactive elements
1. Radioactive elements disintegrate and eventually turn into lead.

Figure 1: An example argument map

A tool that is often used in such courses is the *argument map*. An argument map is a graphical representation of the logical structure of an argument – the ways in which premises, intermediate steps and the final conclusion all fit together. Consider, for example, the argument (from Fisher, 2001, p. 42) in Figure 1. In this ans similar diagrams that follow, boxes represent the premises and conclusions and arrows indicate logical support. So in this case, premise 1 supports the intermediate conclusion 2, which in conjunction with premise 3 (shown by the line connecting 2 and 3) supports the main conclusion C.

Such diagrams have been used in critical thinking courses and textbooks since at least the 1950s (for example; Beardsley, 1950; Toulmin, 1958; Scriven, 1976; Fisher, 1988; LeBlanc, 1998). Students are given an argumentative text to analyse and must create an argument map to represent its structure. This may sound like a trivial task, but it is not. Many students initially find it very difficult. It is in fact no easy task, when faced with even a short passage of ordinary argumentative text, to 'extract out' the core components of the argument. Even when students are able to do this, they can find it hard to see how the components fit together. Asking them to produce an argument map can provide the teacher with valuable insights into the student's 'mental model' of the argument in question.

A recent innovation has been the use of computer software such as *Araucaria*, *Athena* and *Reason!able*, which help students create and manipulate argument map diagrams (Reed & Rowe, 2006; Rolf & Magnusson, 2002; Austhink, 2006; see Kirschner, Buckingham & Carr, 2002 and Harrell 2005 for an overview and comparison of these argument mapping software applications). Such software allows students to easily create, manipulate and share complex argument maps – something which is not so easy using pencil and paper methods. Text can be typed into boxes and edited, supporting premises can be added, deleted or moved around and so on. In some systems, evaluations of the argument (assessments of the truth of premises and strength of inferential connections) can also be incorporated into the argument map. The results can be saved, printed out, shared online or pasted into a word processing document.

1.2 The quality practice hypothesis

Assuming that critical thinking is a skill that can be taught and improved, how can argument mapping software help achieve that goal? One view about how critical thinking skills can be improved is represented by the *quality practice hypothesis* (van Gelder, 2001; van Gelder, Bissett & Cumming, 2004; van Gelder, 2005). According to this theory, acquiring expertise in critical thinking, as in other areas, requires large amounts of *deliberate practice*. The concept of deliberate practice is based on research in cognitive science on how expertise is acquired in a variety of cognitive domains (see Erricsson & Charness, 1994 and references cited in van Gelder, Bissett & Cumming, 2004).

Deliberate practice must be *motivated* (the students should be deliberately practising in order to improve their skills), *guided* (the students should have access to help about what to do next), *scaffolded* (in the early stages, it should be impossible for the students to make certain kinds of mistake) and *graduated* (exercises gradually increase in difficulty and complexity). In addition, for practice to be effective, sufficient *feedback* must be provided – students should have some way of knowing whether they are doing the right thing or not. The quality practice hypothesis states that critical thinking practice must be deliberate in this sense to have any chance of substantially improving students' performance (van Gelder, Bissett & Cumming, 2004, p.143).

The use of computer assisted argument mapping exercises can help support extensive deliberate practice without expensive one on one tutoring. Students can be provided with a sequence of exercises of increasing difficulty in which they have to create a map of an argument. Computer software can support the creation of these argument maps in a way that is both guided and scaffolded. Scaffolding can be provided by building certain constraints into the kind of diagram that can be produced (for example, every argument must have one and only one main conclusion). Guidance can be provided by context sensitive help – when the students select a component of the argument map they are constructing, the system might provide some advice as to what to do next (see van Gelder, 2001).

Using this approach to teaching critical thinking, van Gelder and others at the University of Melbourne have achieved impressive results, using the *Reason!able* argument mapping software (since superseded by *Rationale*, see Austhink 2006). Over a single semester, 12 week course, they have recorded significant improvements in critical thinking, as measured by a standardised multiple choice test (the *California Critical Thinking Skills Test*). Effect sizes for the mean gain from pre-test to post-test range from 0.8 to 0.89 standard deviations (van Gelder, Bissett & Cumming, 2004; see also section 5.4).

1.3 The problem of feedback

A significant problem remains however – that of providing appropriate *feedback* to the student. Marking and grading argument maps produced by students is a time consuming process and requires a fair amount of expertise. In class, tutors can provide feedback to students on whether their argument maps are correct or not. With large classes of course this can be difficult – there may not be enough time for tutors to give every student the feedback they need. One approach (adopted by van Gelder) is to provide model answers, so that students can assess themselves. Students compare their maps of a given argument to completed maps provided by the instructors. One problem with this is that students may not be able to work out *why* their answer is wrong and the model answer correct. Another problem arises from the fact that there is often more than one correct way to map a given argument. That being so, students may not be able to tell when a difference between their map and the model answer is an *important* difference.

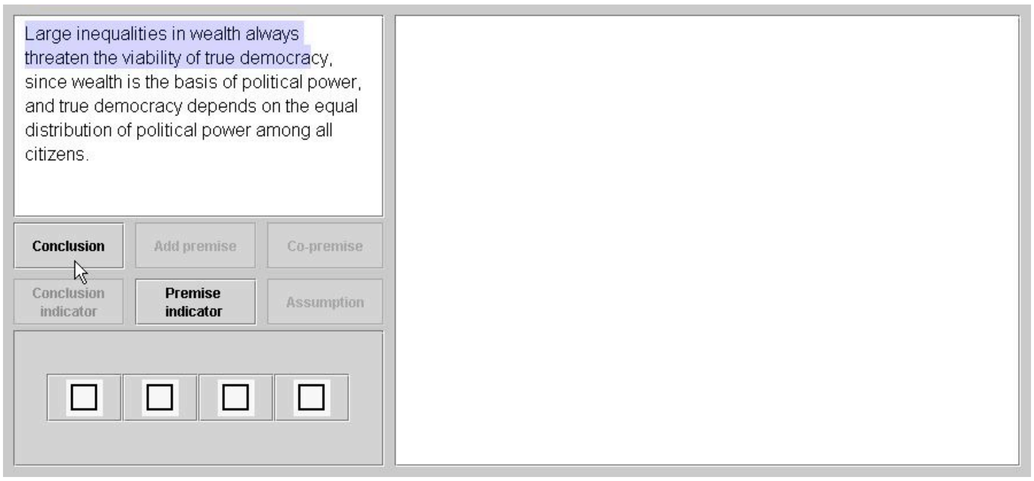
Can good quality feedback be automated? If it could, this would represent a substantial advance in the use of argument mapping software, by allowing for deliberate practice in a way that does not require a very low student-tutor ratio or hours of difficult marking.

In what follows, we describe one solution to this problem – a simple argument mapping software system which is able to automatically provide instant feedback to the students as they construct a map of a given argument. A prototype of the system was implemented in 2004 and incorporated into the Monash University critical thinking course. The system was implemented as a *Java Applet* (a program that can be included in a web page) so that the argument mapping exercises could be provided to students online. We used *WebCT* for this purpose, but the system can be used to embed argument mapping exercises in any web page (see section 3, 'Implementation' for further details).

We will first describe the system, giving an example of how a student would work through one argument mapping exercise. We will then report on how the system was used in the critical thinking course and the results from a pre- and post-test study using a standardised critical thinking test. We then compare these results with those obtained in a number of comparable critical thinking courses, taught with and without argument mapping software. Finally, we comment on some of the limitations of the system described here and some possibilities for future enhancements.

2. An example automated argument mapping exercise

Figure 2 shows a simple example of an automated argument mapping exercise in progress. The window in the top left hand corner contains the text of a simple argument (taken from LSAC 2002). The student's task is to construct a map of the argument, using the mouse to select the appropriate segment of text and then clicking on the buttons below.

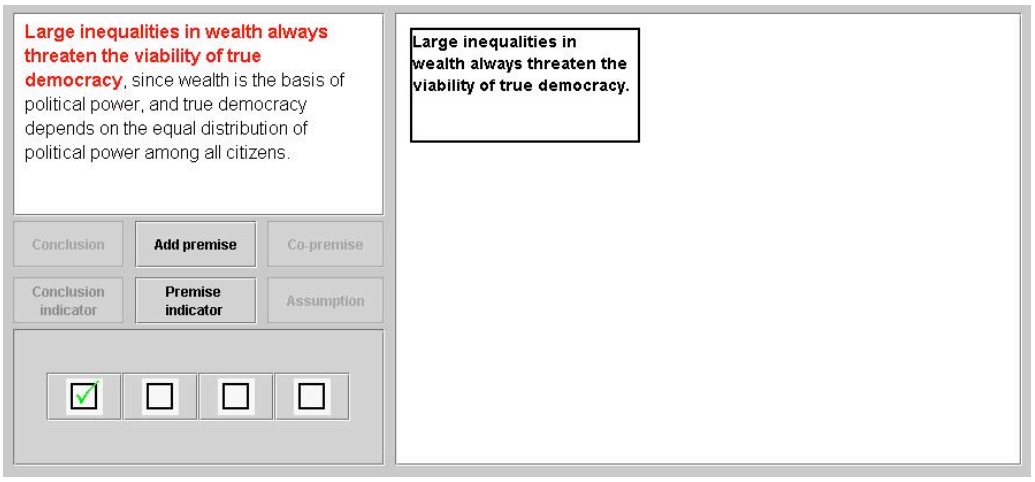


Large inequalities in wealth always threaten the viability of true democracy, since wealth is the basis of political power, and true democracy depends on the equal distribution of political power among all citizens.

Figure 2: Student selects the main conclusion

In Figure 2, the student has selected the segment of text that they take to represent the conclusion of the argument. The student then clicks on the button labelled 'Conclusion' to indicate their choice. The result is shown Figure 3. A green tick has appeared in the progress pane, so that the student knows they have correctly identified the conclusion. A box representing the conclusion appears in the argument map pane and the text of the conclusion is highlighted in red.

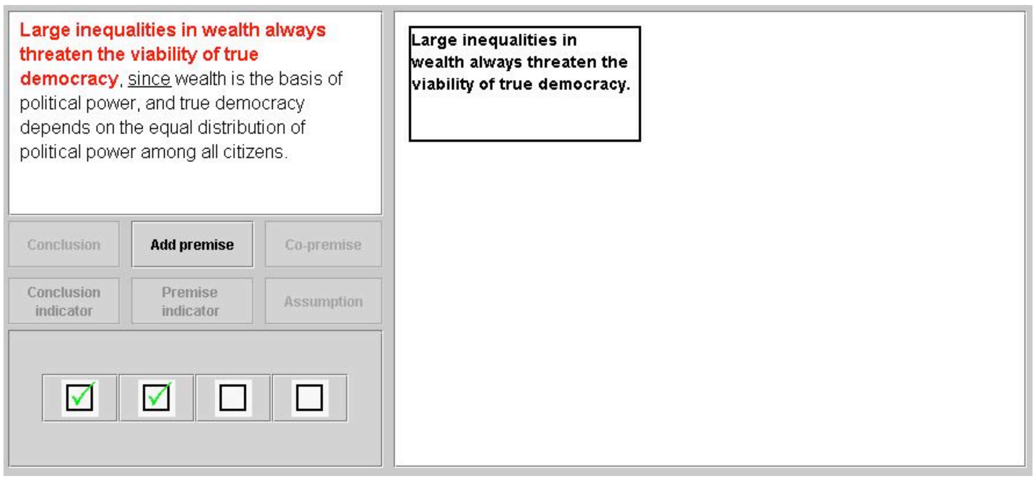
This illustrates the mechanism for providing automated feedback to the student. The system knows which segment of the text represents each component of the argument, so it can instantly inform the student whether they have correctly identified that component or not. To avoid unnecessary frustration, the system allows for a fair amount of leeway in the selection of text. Notice, for example, that in Figure 2, the student has not selected all of the word 'democracy'. The system automatically expands selections to the nearest word. If the student had only selected up to the word 'true' however, this would have been marked as incorrect – a premise or conclusion must always be a complete sentence.


Large inequalities in wealth always threaten the viability of true democracy, since wealth is the basis of political power, and true democracy depends on the equal distribution of political power among all citizens.

Large inequalities in wealth always threaten the viability of true democracy.

Figure 3: Conclusion correctly identified

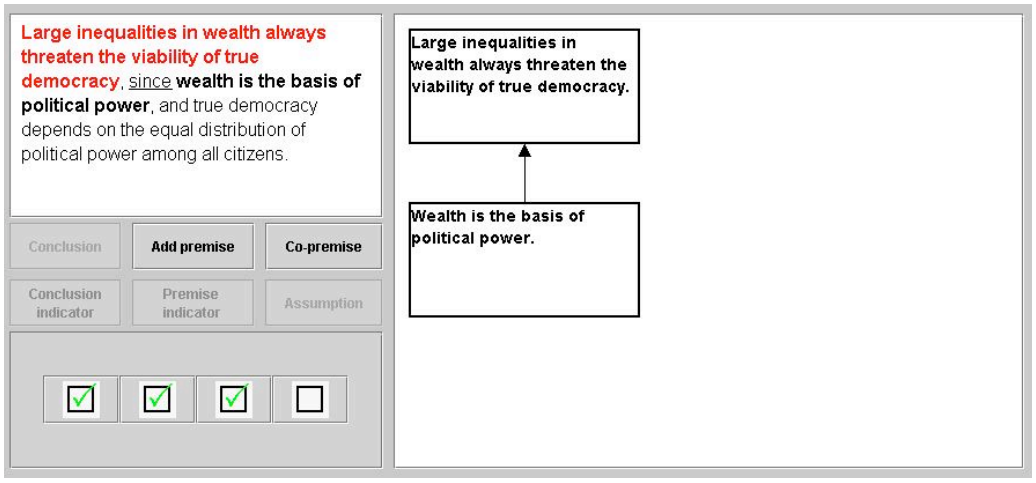
Notice also that the 'Conclusion' button has now been disabled, since every argument has only one main conclusion. The previously disabled 'Add Premise' button is now enabled. This is one mechanism used to provide scaffolding and guidance – by selectively enabling and disabling buttons the student is guided as to what to do next and prevented from making mistakes.


Large inequalities in wealth always threaten the viability of true democracy, since wealth is the basis of political power, and true democracy depends on the equal distribution of political power among all citizens.

Large inequalities in wealth always threaten the viability of true democracy.

Figure 4: A premise indicator identified

The next step for the student is to identify the premises supporting the main conclusion of the argument. In Figure 4, the student has correctly identified the word 'since' as a *premise indicator*. Again, a green tick has appeared in the box to the right of the first tick, to indicate that this identification is correct (the word is then also underlined). The identification of the premise indicator provides a clue that that the text immediately following is a premise. The use of premise and conclusion indicators (words like 'since', 'because', 'therefore', 'hence', 'it follows that' and so on) to help students identify the components of an argument is a standard part of most critical thinking textbooks and courses (e.g. Fisher, 2001, pp. 22-32; LeBlanc, 1998, pp. 2-3).

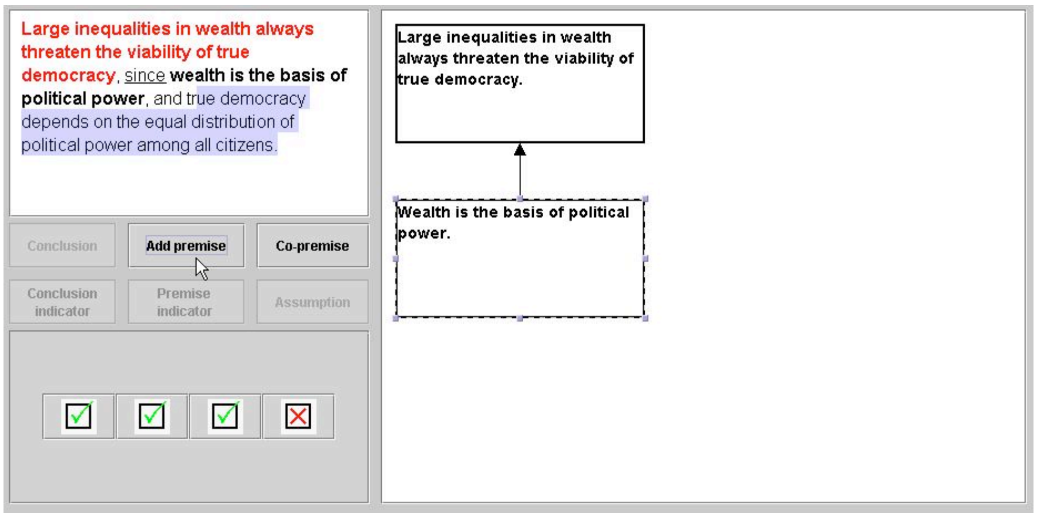

Large inequalities in wealth always threaten the viability of true democracy, since wealth is the basis of political power, and true democracy depends on the equal distribution of political power among all citizens.

Conclusion
Add premise
Co-premise
Conclusion indicator
Premise indicator
Assumption
Large inequalities in wealth always threaten the viability of true democracy.
Wealth is the basis of political power.

Figure 5: One premise correctly identified

In Figure 5, the student has correctly identified one of the premises, by selecting the appropriate text and clicking on the 'Add premise' button. A tick appears to indicate that this is correct and a box representing that premise is added to the argument map.

The fact that one box has yet to be ticked off tells the student that there is one more component of this argument to identify. Since only two buttons remain enabled, the student knows that the remaining item is either a co-premise or a supporting premise.

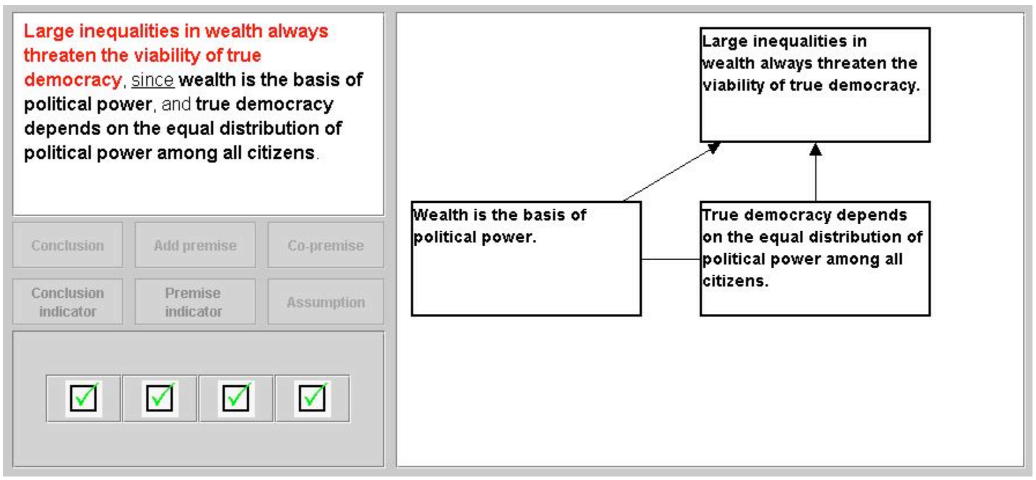

Large inequalities in wealth always threaten the viability of true democracy, since wealth is the basis of political power, and true democracy depends on the equal distribution of political power among all citizens.

Conclusion
Add premise
Co-premise
Conclusion indicator
Premise indicator
Assumption
Large inequalities in wealth always threaten the viability of true democracy.
Wealth is the basis of political power.

Figure 6: Final premise misidentified as supporting premise

In Figure 6 the student has selected the appropriate segment of text and clicked on the 'Add premise' button. The function of this button is to add the selected text as a *supporting premise* under the currently selected box in the argument map pane. In this case, the selected box is the premise 'Wealth is the basis of political power'. This choice is incorrect however. The selected text does not support that premise, but rather acts as a co-premise supporting the main conclusion. So a red cross appears in the final box, to indicate that the student has made a mistake.

Finally, in Figure 7, the student has correctly identified the selected text as a co-premise, by clicking on the 'co-premise' button. The co-premise is added to the argument map and a green tick appears in the final box, informing the student that this exercise is completed, proceed to the next one.


Large inequalities in wealth always threaten the viability of true democracy, since wealth is the basis of political power, and true democracy depends on the equal distribution of political power among all citizens.

Large inequalities in wealth always threaten the viability of true democracy.

Wealth is the basis of political power.

True democracy depends on the equal distribution of political power among all citizens.

Figure 7: The argument fully mapped

It is worth noting that there is a fair amount of flexibility in the order in which components of the argument can be identified. In this example, the student could have selected the text 'true democracy depends on the equal distribution of power ...' first and then added the co-premise 'wealth is the basis of political power'. However, the system does enforce a 'top down' method of working – claims must always be identified before any premise that supports them. This may actually be an advantage however, since it is one way of providing scaffolding and guidance to the student.

The system can handle maps of any degree of complexity, so that exercises can become more complex as students progress. The distinction between linked premises (co-premises) and independent or 'convergent' premises can also be represented. An additional feature is the ability to incorporate unstated premises ('assumptions') into the argument map. This is done by clicking on the 'Assumption' button and selecting the assumption from a multiple choice list.

It is also possible to include a multiple choice question as part of the exercise. Students must first create a map of the argument and then answer a question about it. For example, the question might ask the student to identify a flaw in the argument. Again, instant feedback is given on whether the answer is correct. Further details and screen shots are given on the web page http://www.arts.monash.edu.au/philosophy /research/thinking/04webmap.php

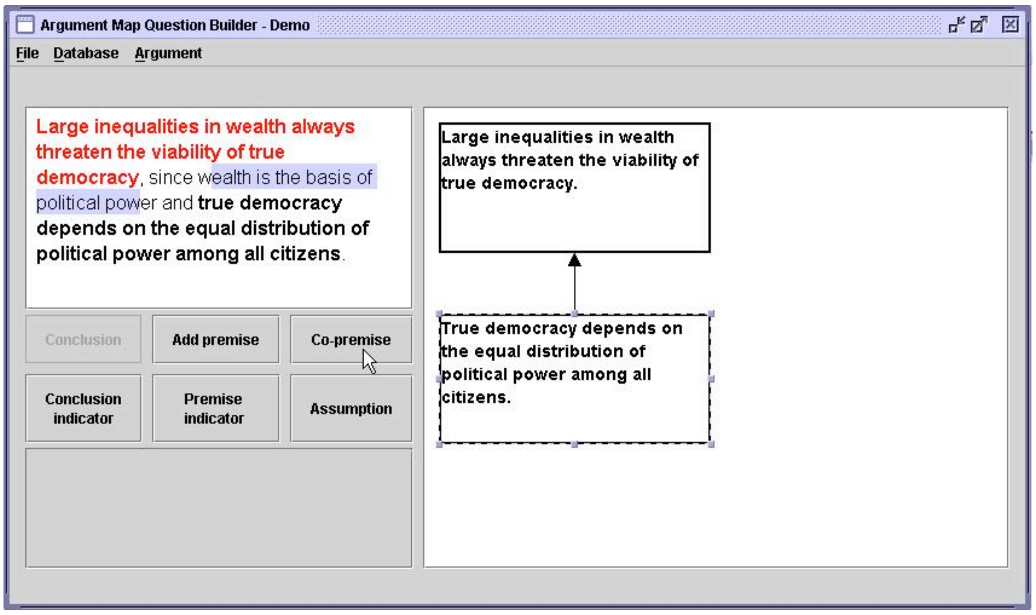
The interested reader might care to try out some sample exercises using the system. These are available at http://www.arts.monash.edu.au/philosophy/research/think ing/argumentmaps/

3. Implementation

The system was implemented in the Java programming language by one of the authors. Each argument mapping exercise runs inside a web page as a *Java applet*. An applet is a Java program embedded in a web page. When a user accesses a web page containing an applet, the program is automatically downloaded from the server and then runs locally, inside the user's web browser (see Sun Microsystems, 2008). Since Java is a platform independent programming language, the very same program will run on any supported operating system and in any web browser that supports the Java plugin. We used the standard Java compiler and libraries (Java Development Kit version 1.4.1) which are available free from the Sun Microsystems website. We also made use of *JGraph*, an open source graph drawing Java library (JGraph Ltd, 2001).

Each argument mapping exercise consists of a web page incorporating an instance of the argument mapping applet. A few lines of HTML are all that is necessary to include an applet in a web page. The text and structure of the argument is also embedded in the HTML for the web page and is passed as an input parameter to the applet.

How does a lecturer who wants to create an argument mapping exercise create these web pages? For this purpose, we implemented a separate Java application (the *Question Builder*) which makes use of exactly the same basic user interface. A lecturer wishing to create an argument mapping exercise begins by typing (or pasting in) the text of the argument (Figure 8). Then, by highlighting segments of the text and clicking on the appropriate buttons, the lecturer specifies which segments of the text are the conclusion, premises and indicators.



Large inequalities in wealth always threaten the viability of true democracy, since wealth is the basis of political power and true democracy depends on the equal distribution of political power among all citizens.

Large inequalities in wealth always threaten the viability of true democracy.

True democracy depends on the equal distribution of political power among all citizens.

Figure 8: The *Question Builder* application

The lecturer can create any number of these argument maps and save them as a single database file. Once an exercise has been completed, the lecturer simply selects 'Export to HTML' and the application automatically creates all the HTML files (one for each argument map in the database) which can then be uploaded to a web server or provided directly to students. An exercise consisting of 8-10 argument maps typically takes no more than a few hours to create in this way.

The system we have described was implemented as an experimental prototype, as part of a project to evaluate and compare techniques for teaching critical thinking (see section 5). For that reason, there is no open source distribution and the system is not currently available to the general public. We have some plans for future development (see section 6) and a version of the system may then become publicly (and we anticipate, freely) available in the near future.

4. Using the system in a critical thinking course

In first semester 2004, we incorporated these argument mapping exercises into a single semester critical thinking course, run by the School of Philosophy and Bioethics at Monash University (PHL1030 *Thinking: Analysing Arguments*).

4.1 Overview of the course

The unit is a single semester course, with 12 weeks of instruction, consisting of one 60 minute lecture and one 2-hour tutorial session each week. The course falls into two main parts. First, there is a section on argument *analysis* (identifying conclusions, premises, unstated assumptions and structure of arguments). This is followed by a section on argument *evaluation* (evaluating arguments, criticising arguments, identifying fallacies). A study guide (*The Elements of Argument*) was written by one of the authors to go with the course. This was a short textbook covering the basics of argument analysis and evaluation.

Assessment took the form of eight homework exercises. There was no final exam. Each homework exercise consisted of two sections; a set of LSAT multiple choice logical reasoning and reading comprehension questions, followed by written analysis and evaluation of a short argumentative text. Tutorials for the course took place in computer labs. Students spent the first half hour of each tutorial working on argument mapping exercises using the software. The remaining tutorial time was spent on further practice at analysing and evaluating example arguments. For this purpose, students were required to read one fairly short passage (essay or book extract) each week. The passages were on a wide variety of topics: philosophy, evolutionary theory, psychology, law, politics and so on. There were four tutorial groups, taught by two different tutors. Class sizes ranged from 12 to 17 students. For a week by week outline of the course, see Table 1.

4.2 Argument mapping exercises

Ten sets of exercises, consisting of 5-10 arguments for analysis were provided. These were made available on the course website, provided by *WebCT*. Students worked at their own pace and none of the exercises were graded. Students worked on the exercises by themselves, rather than in pairs or groups, but students were not prevented from discussing their work with their neighbour. On average, approximately 30-40 minutes each week were spent working on these exercises. The tutor was present to offer help if required. Since all the exercises were available online, students were able to complete the exercises at home if they did not finish them in class. Many students took advantage of this opportunity.

Table 1 lists the type of exercise used in each tutorial for the course. Note that the first exercise did not involve any argument mapping. In this exercise, students were given a collection of passages and had to say which ones were arguments. The pre-test and post-test are explained in the following section.

Table 1: Course outline and argument mapping exercises used

Week	Lecture topic	Tutorial exercise
1	Why study argument?	Pre-test
2	Introduction to reason and argument	1. Identify arguments Distinguish arguments from non-arguments. 10 multiple choice questions (no argument mapping exercises).
3	Argument analysis 1	2. Identify conclusions Identify the main conclusion of the argument. 12 argument mapping exercises.
4	Argument analysis 2	3. Map simple arguments Create a map of the argument. 11 exercises.
5	Argument evaluation: Truth, justification	4. Map complex arguments Create a map of the argument. 14 exercises.
6	Argument evaluation: Clarity, relevance, strength	5. Argument structure Map the argument and identify the role played by a particular statement or the argumentative strategy used. 14 exercises with multiple choice questions.
7	Criticism: Objections and replies	6. Argument structure As above. 14 exercises with multiple choice questions.
8	Criticism: Assumptions	7. Assumptions Map the argument, selecting the correct assumption from a list. 8 exercises.
9	Fallacies (ambiguity)	8. Fallacies of ambiguity Map the given argument then answer the question to identify the flaw. 8 exercises with multiple choice questions.
10	Fallacies (relevance)	9. Fallacies of relevance Map the given argument then answer the question to identify the flaw. 8 exercises with multiple choice questions.
11	Fallacies (truth, ambiguity)	10. Fallacies of truth Map the given argument then answer the question to identify the flaw. 8 exercises with multiple choice questions.
12	Fallacies (strength)	11. Fallacies of strength Map the given argument then answer the question to identify the flaw. 8 exercises with multiple choice questions.
13	Reason and happiness	Post-test

5. Gains in critical thinking skills

As part of a three-year project to evaluate and compare different methods of teaching critical thinking, students enrolled in this course were pre- and post-tested using a standardised test of critical thinking ability, the *California Critical Thinking Skills Test*. This is a commonly used test in critical thinking research (see section 5.4). It is a timed (45 minute) multiple choice test, coming in two equivalent forms: A and B. Each form consists of 34 items which test students' ability to clarify the meaning of claims, analyse and evaluate arguments, and draw correct conclusions from given information (Facione & Facione, 1992).

5.1 Sample and procedures

The final sample consisted of 43 undergraduate students (16 men and 27 women). Ages ranged from 17 (1 student) to 55 (1 student). The mean age was 21.5 years, while the mode was 18 years (30.2%). All students taking the course were required to complete the pre- and post-test. They were informed about the purpose of the study and asked to sign a consent form giving permission for their test scores to be used. Test scores did not count towards the students' final grade.

Out of an initial enrolment of 63 students, 52 (82.5%) consented to be part of the study. Of the students who consented 50 (96%) completed the pre-test and 43 (83%) also completed the post-test. Of the 9 students who failed to complete both tests, 7 completed the pre-test but not the post-test and 2 completed neither test. When calculating gain scores, only students who completed both pre- and post-test were included.

Students completed the CCTST during the first half of the scheduled 2-hour tutorials for the course. The pre-test was completed in the first tutorial (week 1) and the post-test in the final tutorial (week 13). The tests were completed under examination conditions, as outlined in the test manual. Students were not informed of their test scores until after the end of the course. All students were given form A of the CCTST for the pre-test and form B for the post-test.

5.2 Teaching methods

In the following semesters, we used the same procedures described in section 5.1 to evaluate and compare several other methods of teaching critical thinking. As far as possible, the course structure and content were kept the same, while new techniques and exercises were introduced in the 2-hour tutorials or (in the case of the *Peer Instruction* method) in the lectures. Table 2 shows the teaching methods we investigated in each semester.

Table 2: Teaching methods investigated

Semester	Method	Description
1 (2004)	Web based argument mapping exercises.	Online argument mapping exercises with automated feedback.
2 (2004)	Standard version.	A 'control' version of the course, taught without using any special techniques or exercises.
1 (2005)	Argument mapping exercises using *Reason!able*.	Argument mapping exercises using the *Reason!able* software were used in tutorials. (No automated feedback).
2 (2005)	Actively open minded thinking (AOMT) exercises.	A variety of strategies aimed at increasing AOMT were incorporated into the course.
1 (2006)	Peer Instruction.	Peer Instruction was used in the lectures for the course.

Actively open minded thinking (AOMT) can be defined as the ability and disposition to avoid what is sometimes called 'myside bias' or 'confirmation bias'. It is "the willingness to search actively for evidence against one's favoured beliefs, plans or goals and to weigh such evidence fairly when it is available" (Baron, 2002, p. 1; see also Baron, 1991, 1994).

In this version of the course, we attempted to incorporate a variety of exercises and teaching strategies aimed at improving students' levels of AOMT. These included teaching students about some of the empirical evidence for myside bias and the evidence that AOMT reduces bias and improves thinking; exercises that focus on the ability of students to find alternative explanations or counter-evidence for a given claim; and exercises in 'considering the opposite', in which students must criticise arguments in support of their own position on the topic under discussion and suggest evidence or arguments against their position (see Lord, Lepper & Preston, 1984; Budesheim & Lundquist, 1999).

We also investigated a general purpose pedagogical strategy, which might be expected to improve student performance on critical thinking tasks. This was *Peer Instruction* (PI). PI is a simple technique that can be incorporated into lectures. At various points during the lecture, the instructor stops and asks a multiple choice question that all students must respond to using flash cards or an audience response system (we used flash cards). If most students have the right answer, the lecturer confirms it and moves on. If there is a mixture of right and wrong answers, the lecturer asks the students to spend 1-2 minutes trying to convince a neighbour that their answer is correct. Students then 're-vote' on the answer. Typically, more students have the right answer after these peer discussions – students who have just mastered the material are able to explain it effectively to their peers (see Mazur, 1997; Butchart, Handfield & Restall, forthcoming).

Further details of the teaching methods and more information about the study can be found at the project website (Butchart et al. 2006):
http://www.arts.monash.edu.au/philosophy/research/thinking/

5.3 Results

The results are shown in Table 3 and Figure 9. A standardised effect size for gains in critical thinking ability was calculated by dividing the mean gain in raw CCTST score by an estimated population pre-test standard deviation of 4.45 (this is the value used in other studies using the CCTST, such as van Gelder et al., 2004). We also calculated a 'normalised' gain score for each student (Hake, 1998). This is the difference between pre- and post-test score expressed as a percentage of how many points each student could have earned (or lost).

Table 3: Improvement in critical thinking scores on the CCTST

Teaching method		N	Effect size	95% conf. interval	Mean gain (%)	S.D. of gain	95% conf. interval
1.	Argument mapping exercises with automated feedback	43	0.45	± 0.29	13.70	21.08	± 7.39
2.	Standard course	65	0.19	± 0.20	7.85	22.36	± 5.54
3.	Argument mapping exercises with no automated feedback	41	0.22	± 0.26	7.10	20.27	± 6.39
4.	AOMT exercises	49	0.14	± 0.74	6.63	23.93	± 6.88
5.	Peer Instruction	40	0.40	± 0.25	17.23	26.64	± 8.5

When the automated argument mapping exercises described in the present article were used, students showed an improvement in critical thinking ability of 0.45 standard deviations. Since the 95% confidence interval for this improvement does not overlap zero, the result is statistically significant at the 0.05 level. The mean percentage improvement was 13.7%. By contrast, the standard version of the course (in which no special teaching methods were used) resulted in average gain of just 0.19 standard deviations, while argument mapping exercises with no automated feedback resulted in a comparable gain of 0.22 standard deviations.

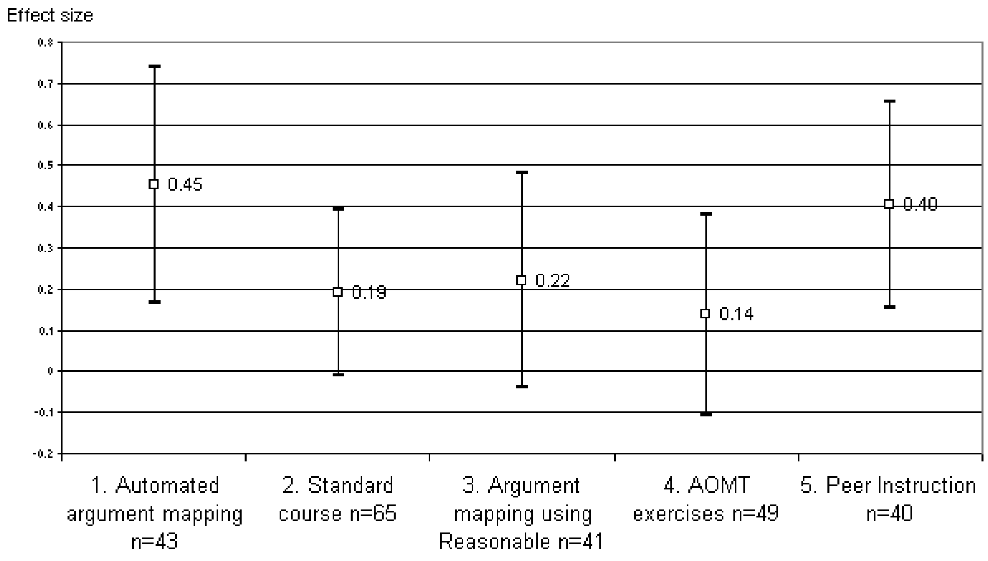


Figure 9: Gains in critical thinking scores on the CCTST

AOMT exercises did not seem to have much of an effect on students' critical thinking skills either; here the average gain on the CCTST was 0.14 standard deviations. We think it is worth pointing out that since AOMT exercises focus mainly on critical thinking *dispositions* rather than specific skills it is perhaps not too surprising that we discovered little 'transfer effect' in terms of improved test scores on the CCTST.

Interestingly, the only other teaching method to have a statistically significant effect was *Peer Instruction*, where the gain was 0.4 standard deviations, significant at the 0.,05 level. This suggests that simply improving the standard of teaching and lecturing in a critical thinking course (by incorporating *Peer Instruction* in the lectures for example) can be just as effective at improving student performance as critical thinking specific techniques such as computer assisted argument mapping exercises.

5.4 Comparison with other studies

In the semester in which we used automated argument mapping exercises, we measured an improvement of 0.45 standard deviations in critical thinking skills. Just how much of this improvement is due to the exercises themselves is of course unclear - we discuss that issue further in section 5.6. In the present section, we want to put the result in perspective by comparing it to other studies.

Many studies have been carried out attempting to determine the impact of a university education on students' critical thinking skills. In their review of the research literature, Pascarella and Terenzini estimate that the total effect of the first three years of college is 0.55 standard deviations (Pascarella & Terenzini, 2005, p. 205). This is their estimate of the amount of change in critical thinking skills which is uniquely attributable to exposure to university education. Hence, the critical thinking gains measured during our 12 week course are close to that which could be expected to result from three years of undergraduate education.

To get an idea of how our results compare with other single semester critical thinking courses, Figure 10 shows results from a number of studies, all of which used the CCTST as a pre- and post-test measure of critical thinking ability. Study 1 is typical of the amount one might expect students to improve over a single semester of university study, without any specific critical thinking instruction. These are the results from 90 students enrolled in a single semester introductory philosophy course at California State University at Fullerton (Facione, 1990). The mean gain for these students was 0.14 standard deviations on the CCTST. Based on a meta-analysis of studies of gains in critical thinking at university (Alvarez, 2007), van Gelder estimates that without explicit instruction in critical thinking, students typically improve by approximately 0.1 of a standard deviation over a single semester (van Gelder, 2007, p. 29).

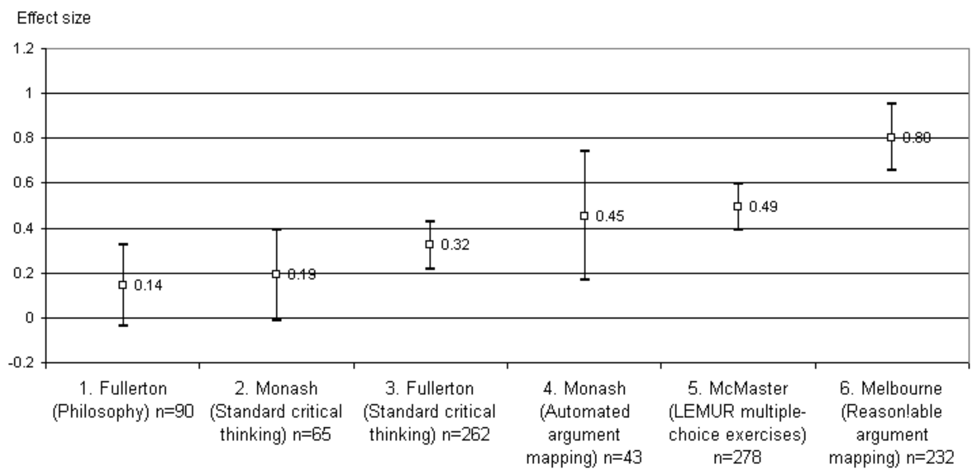


Figure 10: Gains in critical thinking ability for single semester critical thinking courses.
Bars show 95% confidence intervals. 1, 3: Facione (1990);
5: Hitchcock (2004); 6: van Gelder et al. (2004).

At California State University at Fullerton, a single semester critical thinking course lead to a gain of 0.32 standard deviations (Study 3, Facione, 1990). At McMaster University, Hitchcock measured a similar gain to ours of 0.49 standard deviations for a 13-week course using the LEMUR software (Study 5, Hitchcock, 2004). The LEMUR software accompanies LeBlanc's textbook (Le Blanc, 1998) and incorporates numerous multiple choice quizzes and exercises. The software also includes some simple argument mapping exercises in which students can drag component sentences of a text into a pre-structured argument map diagram (see Hitchcock, 2004, pp. 186-187).

Using these and other studies incorporated into the Alvarez (2007) meta-analysis, van Gelder estimates that the average gain for students enrolled in a single semester critical thinking course is 0.3 standard deviations (van Gelder 2007, p. 29). The gains in critical thinking obtained using extensive practice with the *Reason!able* argument mapping software are significantly higher, at 0.8 standard deviations (Study 6, van Gelder, Bissett & Cumming, 2004).

Our results therefore sit somewhere in the middle of the range. The effect size of 0.45 SD is more than four times greater than the expected gain of 0.1 SD for a single semester of university with no critical thinking instruction. It is only marginally higher than the average gain of 0.3 SD for students taking a single semester critical thinking course. But it is about half the size of the largest known effect of 0.8 SD obtained using the van Gelder group's argument mapping approach.

5.5 Feedback from students and tutors

We did not carry out any formal evaluation of students' opinions about the use of the argument mapping software. Instead, we report here some informal feedback from students and the tutors.

The tutors reported that student response to the system was not uniform. Many students liked the argument mapping exercises and found them useful. This group of students appreciated the exercises because they targeted specific skills and gave immediate feedback when they went wrong. These students also liked the scaffolding provided by the system, in particular, the way it forced them to analyse the arguments in a systematic fashion; starting with the conclusion and then working down through the premises.

For another group of students however, this last feature was actually a disadvantage. These students sometimes found it frustrating to have to work through the analysis of the argument in specific steps. They wanted to work piecemeal at different sections of the argument at one go. These students were encouraged to use pen and paper before using the software, and this solution seemed to work quite well. Another group of students who did not respond as positively were older students and students who were not as proficient or comfortable with computers in general. These students were sometimes handicapped by their anxiety about using the software and this affected their ability to learn from the exercises.

Some students worked on the exercises by themselves and found that the feedback helped them to work out problems on their own. Other students preferred working on the exercises in pairs and talking it through. Students also enjoyed discussing the exercises as a class after completing the exercises and this was found to be an important follow up activity.

Although we did not use a questionnaire to evaluate the software specifically, students did complete a standard evaluation of the course as a whole. This was a 14 item Likert style questionnaire, completed anonymously. 50 students out of the final enrolment of 60 completed this questionnaire (a response rate of 83%). None of these items were particularly pertinent to the use of the argument software. Overall satisfaction for the course was somewhat below average at 78%, compared to an average of 82% for the following semesters. We suspect this has to do more with various administrative problems with the course (changes in due dates for homework exercises for example), rather than the argument mapping exercises themselves (many students complained about these changes on their questionnaires).

The questionnaire also contained a 'General comments' section. 34 (68%) of the students who completed the questionnaire wrote additional comments. Of these, a small proportion (7 students) mentioned the argument mapping software specifically. Three students said they did not find the argument mapping exercises useful. The other four comments were less about the software itself and more about the necessity of holding the tutorials in computer labs. The lab we used had the computers set up in long rows and these students complained that this isolated students from each other and impeded participation and discussion in the classroom. This was a problem also noted by the tutors.

Despite these problems, many students took up the opportunity of completing the online argument mapping exercises from home, suggesting that these students at least found the exercises to be valuable.

5.6 Discussion

Although we found a statistically significant improvement in students' critical thinking test scores, caution is clearly required in interpreting this result. It is unclear how much of that improvement was due to the use of the automated argument mapping exercises and how much due to other aspects of the course or to factors not directly related to teaching, such as student maturation.

A comparison with the studies described in section 5.4 is suggestive however. Firstly, the improvement in critical thinking scores of 0.45 standard deviations is more than 4 times greater than the 0.1 estimate for the change in critical thinking scores to be expected during a single semester of university with no specific instruction in critical thinking. This suggests that the improvement we measured was at least partly due to the instruction students received, rather than simply to maturation. Furthermore, the improvement is more than twice as large as that obtained the following semester, teaching the same course without the automated argument mapping exercises (0.19 standard deviations), suggesting that some of the improvement is in fact due to the argument mapping exercises. These comparisons are no more than suggestive however; due to the small sample sizes involved, no statistically reliable inference can be drawn from these comparisons.

Nonetheless, the gains in critical thinking for the course as a whole are encouraging and compare favourably to gains obtained in similar studies. The result certainly seems *consistent* with the quality practice hypothesis, even if it does not strongly support it. Students using the system certainly got a lot more guided and scaffolded practice than students the following semester and the students who had more practice showed a statistically significant improvement in critical thinking scores, while the latter group did not. Again however, no reliable inference to the conclusion that the practice caused the improvement can be drawn.

The mean gain in critical thinking scores obtained using our automated argument mapping system is significantly lower than that obtained using the *Reason!able* argument mapping software at the University of Melbourne – a difference of 0.35 standard deviations (95% confidence interval = 0.026, 0.675). How can this difference be explained? We can do no more here than offer some possible explanations.

Firstly, the *Reason!able* software is much more flexible than our system in several ways. Most obviously, it allows students to put any text at all into argument map boxes, rather than requiring them to select a segment of text. It also allows students to edit the structure of their maps as they build them, by moving and dragging boxes into new positions. Students can also build up their argument maps in any order they like; they can start with the conclusion and work down to the premises, or they can start with the premises and build up to the conclusion. This extra flexibility may well make a big difference to how much students can learn from argument mapping exercises. Finally, the software allows students to incorporate evaluations of premises and inferences into their argument maps.

Secondly, there are additional factors, not directly related to the software itself, which may account for the difference between the Melbourne results and our own. One difference is that at Melbourne, students work together in groups of two on their argument maps. In the present study, students typically worked on the argument mapping exercises on their own (although some students sometimes worked in pairs). It may be that the collaboration and peer interaction that become possible when students work together provides a significant boost to the power of the argument mapping method.

Finally, the sheer number of hours of practice with argument mapping exercises is somewhat greater for the Melbourne university course. Students at Melbourne spent (on average) a total of 10 hours on argument mapping using the *Reason!able* software over a 12-week period (van Gelder, Bissett & Cumming, 2004, p. 147, Table 1). The Monash students, on the other hand, spent a total of 5-7 hours working on automated argument mapping exercises over the same period (30-40 minutes on argument mapping exercises for each of ten tutorials). It seems quite possible then that the difference between the Melbourne University result and the Monash result is due to the amount of quality practice provided. Indeed, van Gelder's study found a positive correlation ($r = 0.31$) between the number of hours of practice (as measured and logged by the software itself) and gains in critical thinking as measured by the CCTST (van Gelder, Bissett & Cumming, 2004, pp. 147-8).

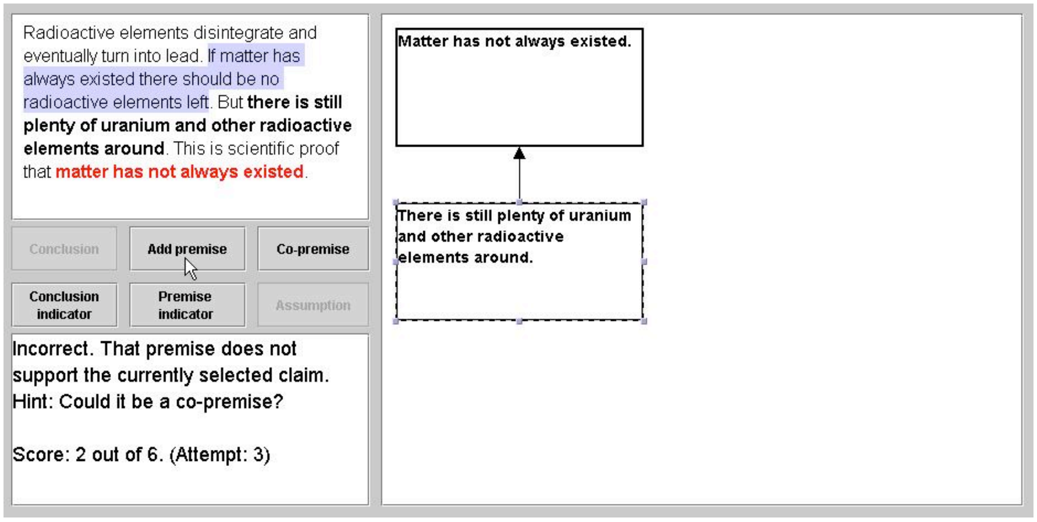
6. Limitations and future development

We conclude with brief comments on the limitations of the system and possibilities for future extensions and improvements.

6.1 Making the feedback more informative

One obvious way in which the system could be improved would be to make the feedback to the student more informative than a simple 'correct' or 'incorrect'. In particular, since the system has a complete representation of the correct argument map, it would be possible to distinguish between various kinds of common mistake that students make when constructing argument maps. For example, students will often make the mistake of placing a claim *underneath* a given premise, when the claim should be a co-premise. The opposite kind of mistake – representing a claim as a co-premise, when it should be a supporting premise – is also possible. All these sorts of mistakes and others could be captured by the system quite easily and a more informative message delivered to the student. One way to do this is shown in Figure 11.

Instead of a sequence of ticks and crosses, we now have a text pane which displays information to the students on their progress with the argument map. In this example, the student has correctly identified the main conclusion and one of the premises. But the student has then misidentified the claim 'If matter has always existed there should be no radioactive elements left' as providing support for that premise. The system has identified the mistake and displays an appropriate message and a hint.


Radioactive elements disintegrate and eventually turn into lead. If matter has always existed there should be no radioactive elements left. But **there is still plenty of uranium and other radioactive elements around**. This is scientific proof that **matter has not always existed**.

Incorrect. That premise does not support the currently selected claim. Hint: Could it be a co-premise?

Score: 2 out of 6. (Attempt: 3)

Matter has not always existed.

There is still plenty of uranium and other radioactive elements around.

Figure 11: More informative feedback

In addition to giving more informative feedback about the student's mistakes, this text pane could also be used to display context sensitive help about what to do next as the student constructs an argument map. There could also be an option to turn off the hints and the help once students feel they no longer needed it. In this way, a further aspect of quality practice would be incorporated – gradually removing scaffolding as the student becomes more expert.

6.2 More flexibility in constructing maps

The current system enforces a fairly strict ordering on the way students construct argument maps. The conclusion must be identified first, followed by premises that directly support it, followed by premises that support those premises, and so on. Sometimes however, it is more natural to construct an argument map from the bottom up, starting with the premises and building up to the main conclusion. The system could be made more flexible by allowing students to construct their maps in either of these two ways.

Indeed, as already noted, one group of students found this aspect of the 'scaffolding' provided by the software frustrating. They wanted to work on different parts of the argument in their own order, and then put the final argument together after that. The system could quite easily be modified to allow for this. In conformity with the idea of gradually removing the scaffolding however, it might make sense to start students off with exercises in which they must work in a systematic, pre-defined order and only later allow students to turn off this restriction, if they choose.

6.3 Objections

In addition to premises that *support* a claim, argument mapping systems often include the concept of an *objection* to a claim. This is especially useful when mapping a complex debate, consisting of arguments, counter-arguments, rebuttals, and so on. It would be straightforward to add objections to argument maps in the current system. An 'Add objection' button could be used to add an objection, rather than a supporting premise under the currently selected node in the argument map pane. Objections could then be distinguished from supporting premises in the argument map using different colours or by adding a simple prefix to the text.

6.4 Evaluations

Another limitation of the system described here is that it does not allow students to include evaluations of arguments directly in their maps (instead, we included multiple choice questions asking students to identify flaws in the argument). There are two aspects to argument evaluation; evaluation of premises and evaluation of inferences. Both aspects can be incorporated into argument maps. One way to do this is to ask students to rate premises and inferences on a scale. For premises, the scale might range from 'definitely false' to 'probably false' to 'probably true' to 'definitely true'. For inferences, the scale might range from 'definitely does not support' to 'weak support' to 'strong support' to 'conclusive support'. These judgements could then be shown in the argument map diagram itself, by attaching labels to the arrows or boxes.

This capability could be added to the system described here by adding 'evaluate premise' and 'evaluate inference' buttons. The student selects a box in the argument map pane and then clicks on the 'evaluate premise' button. A dialog box then appears from which the student can select one of range of possible evaluations from a fixed scale. If appropriate, the system can then provide feedback on the student's selection. Evaluating inferences would work in a similar way.

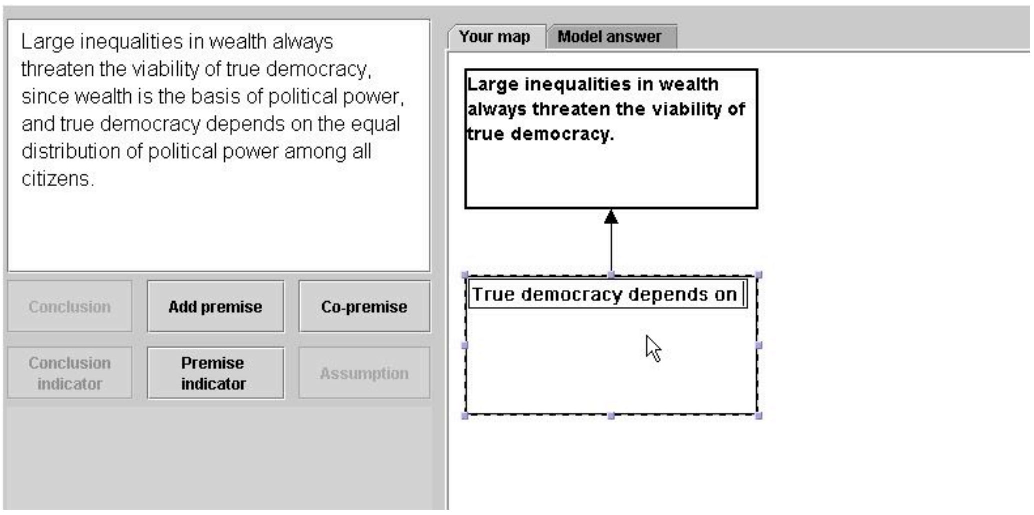
The system might also enforce rules relating to the internal consistency of students' evaluations. One such rule might be that a conclusion cannot be any more acceptable than the premises used to support it. This rule would be violated if, for example, a student evaluated a conclusion as 'definitely true', while the premise that supports that conclusion was evaluated as only 'probably true'. The system could automatically pick up on mistakes like this, and provide the student with appropriate feedback.

6.5 Throwing away the scaffolding: Free form maps with a model answer

Finally, we will mention one further limitation and possible extension of the system. Building automated feedback into the system in the way we have done places some constraints on the text of arguments that can be mapped. Firstly, conclusions and premises must be represented by complete sentences that can be selected as a continuous block of text using the mouse. Secondly, the system does not give students practice at an important skill of argument analysis; that of *paraphrasing* claims made in an argumentative text.

One way in which the system might be made more flexible then is to allow exercises where the automated feedback is turned off. Instead of selecting a block of text for the conclusion of the argument, the student clicks on the 'Conclusion' button and an empty box would appear in the argument map. The student then types the text of the conclusion into the empty box. Adding premises and assumptions would work in the same way. This would allow students to paraphrase conclusions and premises that appear in the text, ensuring that they are represented by complete sentences. Instead of selecting unstated assumptions from a multiple choice list, students would be able to enter them directly onto their maps.

Feedback could then be provided by means of the traditional 'model answer'. One way this could be incorporated into the current system is shown in Figure 12. Here the student is typing in the text of a premise of an argument. When they have finished constructing their map they click on the 'Model answer' tab, which shows a completed model map for the argument. The student can then compare their own map to the model. The system might even keep track of how many elements (conclusion, premises and assumptions) the student has added to their map and only allow the model answer to be viewed when a sufficient number of elements have been added.


Large inequalities in wealth always threaten the viability of true democracy, since wealth is the basis of political power, and true democracy depends on the equal distribution of political power among all citizens.

Large inequalities in wealth always threaten the viability of true democracy.

True democracy depends on

Figure 12: A free form argument map under construction.
Clicking on the 'Model answer' tab displays a completed map for the argument.

The idea is that the system could operate in two different modes. In one mode, automated feedback is turned on and the student receives instant feedback on their progress as they construct their map. In the second mode, automated feedback is turned off and students construct their maps by typing directly into the boxes, rather than selecting text. Feedback is then provided by allowing students to compare their map to a model answer.

Automated feedback and 'free form' argument mapping exercises could then both be incorporated into a critical thinking course. Early in the course, automated feedback is turned on, providing students with scaffolding and feedback. As the course progresses, the argument mapping exercises should become gradually more complex. Gradually, the scaffolding is removed by turning off the automated feedback and using more free form exercises. Here again, quality practice would be further supported by allowing scaffolding to be removed as the student becomes more expert.

7. Summary

We have described a simple system for creating argument mapping exercises incorporating automated feedback. A prototype of the system was implemented and used in a single semester undergraduate critical thinking course at Monash University. Students spent approximately 30-40 minutes each week working on argument mapping exercises using the system. The results were encouraging; students showed an average improvement of 0.45 standard deviations on a standardised test of critical thinking over the course of the semester. These results compare favourably to those obtained in other studies of critical thinking courses and are significantly higher than the expected gain of 0.1 standard deviations expected to result from a single semester of university without critical thinking instruction. Although the system is clearly limited in various ways and certainly no substitute for good classroom teaching, it does show great potential for supporting and enhancing quality deliberate practice in an important critical thinking skill.

Acknowledgments

The research described here was funded by an Australian Research Council Linkage Project grant, (LP0346935; Effective Pedagogy for Improving Critical Thinking) and by a grant from the Australian Government Department of Education, Science and Training (DEST). The research was carried out in collaboration with DEST and the Australian Council for Educational Research (ACER).

References

Alvarez, C. M. (2007). *Does philosophy improve critical thinking skills?* MA Thesis, Department of Philosophy, University of Melbourne.

Austhink Pty Ltd (2006). *Rationale*. Argument mapping computer software. Melbourne: Austhink Pty Ltd. http://www.austhink.com/

Australian Bureau of Statistics (2009). *Measuring Australia's progress: Summary indicators, 2009.* http://www.abs.gov.au/AUSSTATS/abs@.nsf/Latestproducts/1383.0.55.001Main%20Features502009?opendocument&tabname=Summary&prodno=1383.0.55.001&issue=2009&num=&view=

Baron, J. (1991). Beliefs about thinking. In J. Baron, J. F. Voss, D. N. Perkins & J. W. Segal (Eds.), *Informal reasoning and education*. Hillsdale, N.J: L. Erlbaum Associates.

Baron, J. (1994). *Thinking and deciding*. Cambridge, New York: Cambridge University Press.

Baron, J. (2002). Actively open-minded thinking. Unpublished manuscript, personal communication.

Beardsley, M. C. (1950). *Practical logic*. Englewood Cliffs, NJ: Prentice Hall.

Budesheim, T. L. & Lundquist, A. R. (1999). Consider the opposite: Opening minds through in-class debates on course-related controversies. *Teaching of Psychology*, 26(2), 106-110.

Butchart, S., Gold, I., Bigelow, J, Korb, K, Oppy, G. (2006). *The Monash Critical Thinking Study*. Project website. Monash University. http://www.arts.monash.edu.au/philosophy/research/thinking/

Butchart, S., Handfield, T. & Restall, G. (2009). Teaching philosophy, logic and critical thinking using peer instruction. *Teaching Philosophy*, 32(1).

Ericsson, K.A. & Charness, N. (1994). Expert performance. *American Psychologist*, 49, 725-747.

Ennis, R. H. (1987). A taxonomy of critical thinking dispositions and abilities. In J. Baron & R. Sternberg (Eds.), *Teaching thinking skills: Theory and practice* (pp. 9-26) New York: Freeman.

Facione, P. A. (1990). *The California Critical Thinking Skills Test. College level. Technical report #1. Experimental validation and content validity*. Millbrae: California Academic Press.

Facione, P. A. & N. C. Facione (1992). *The California Critical Thinking Skills Test and manual.* Millbrae, CA: California Academic Press. http://www.insightassessment.com/

Fisher, A. (1988). *The logic of real arguments*. Cambridge: Cambridge University Press.

Fisher, A. (2001). *Critical thinking: An introduction*. Cambridge: Cambridge University Press.

Hake, R. (1998). Interactive engagement vs. traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses. *American Journal of Physics*, 66, 64-74.

Harrell, M. (2005). Using argument diagramming software in the classroom. *Teaching Philosophy*, 28, 163-177.

Hatcher, D. L. (1999). Why critical thinking should be combined with written composition. *Informal Logic,* 19(2-3), 171-183.

Hatcher, D. L. (2001). Why Percy can't think: A response to Bailin. *Informal Logic*, 21(2), 171-181.

Hitchcock, D. (2004). The effectiveness of computer-assisted instruction in critical thinking. *Informal Logic*, 24(3), 183-217.

JGraph (2001). *JGraph*. The Java open source graph drawing component. [computer software]. Northampton, UK: JGraph Ltd. http://www.jgraph.com/jgraph.html

Kirschner, P. J., Buckingham Shum, S. J. & Carr, C. S. (Eds.) (2002). *Visualizing argumentation: Software tools for collaborative and educational sense-making*. London: Springer-Verlag.

LeBlanc, J. (1998). *Thinking clearly: A guide to critical reasoning*. New York: W.W. Norton.

Lord, C. G., Lepper, M. R. & Preston, E. (1984). Considering the opposite: A corrective strategy for social judgement. *Journal of Personality and Social Psychology*, 47(6), 1231-1243.

LSAC (2002). *The Official LSAT PrepTest* (*nos. 7-38*). Newton, PA: Law School Admission Council (LSAC).

Mazur, E. (1997). *Peer instruction: A user's manual*. Saddle River: Prentice Hall.

McKeachie, W., Pintrich, P., Lin, Y. & Smith, D. (1986). *Teaching and learning in the college classroom*: *A review of the research literature*. Ann Arbor: University of Michigan, National Centre for Research to Improve Post-Secondary Teaching and Learning.

Pascarella, E. T. & Terenzini, P. T. (2005). *How college affects students Volume 2*: *A third decade of research*. San Francisco, Oxford: Jossey-Bass.

Reed, C. & Rowe, G. (2006). *Araucaria*. Argument mapping computer software. School of Computing, University of Dundee. http://araucaria.computing.dundee.ac.uk/ [verified 3 May 2009]

Rolf, B. & Magnusson, C. (2002). *Athena*. Argument mapping computer software. [verified 3 May 2009] http://www.athenasoft.org/

Scriven, M. (1976). *Reasoning*. New York: McGraw-Hill.

Sun Microsystems (2008). The Java Tutorials. Lesson: Applets. [Online Java tutorials]. Sun Microsystems Inc. http://java.sun.com/docs/books/tutorial/deployment/applet/

Toulmin, S. E. (1958). *The uses of argument*. Cambridge: Cambridge University Press.

van Gelder, T. (2001). How to improve critical thinking using educational technology. In *Meeting at the crossroads: Proceedings ASCILITE Melbourne 2001*. http://www.ascilite.org.au/conferences/melbourne01/pdf/papers/vangeldert.pdf

van Gelder, T., Bissett M. & Cumming G. (2004). Cultivating expertise in informal reasoning. *Canadian Journal of Experimental Psychology*, 58(2), 142-152. http://www.philosophy.unimelb.edu.au/reason/papers/CJEP_van_Gelder.pdf

van Gelder, T. (2005). Teaching critical thinking: Some lessons from cognitive science. *College Teaching*, 35(1), 41-46. http://www.philosophy.unimelb.edu.au/reason/papers/Teaching_CT_Lessons.pdf

van Gelder, T. (2007). The rationale for *Rationale*^{TM}. *Law, Probability and Risk*, 6, 23-42. [verified 3 May 2009] http://lpr.oxfordjournals.org/cgi/reprint/6/1-4/23.pdf?ijkey=GFHrAQMNkJ09woP

Sam Butchart, John Bigelow, Graham Oppy
School of Philosophy and Bioethics
Monash University, Clayton Victoria 3800, Australia
Email: Sam.Butchart@arts.monash.edu.au, John.Bigelow@arts.monash.edu.au, Graham.Oppy@arts.monash.edu.au

Daniella Forster, School of Education
Faculty of Education and Arts, The University of Newcastle
Email: Daniella.Forster@newcastle.edu.au

Alexandra Serrenti, Department of Philosophy
National University of Singapore, Faculty of Arts and Social Sciences
3 Arts Link, Singapore 117570. Email: phisam@nus.edu.sg

Kevin Korb, School of Information Technology
Monash University, Clayton, Victoria 3800, Australia
Email: Kevin.Korb@infotech.monash.edu.au

Ian Gold, Department of Philosophy, McGill University
Leacock Building, 855 Sherbrooke St. W., Montreal, Quebec H3A 2T7, Canada
Email: Ian.Gold@McGill.ca