A Consideration of the Use of Plagiarism Tools for Automated Student Assessment

Olaf Hallan Graven and Lachlan Mhor MacKinnon, *Member, IEEE*

***Abstract*—In this paper, the authors evaluate the flexibility and richness of two well-established text analysis plagiarism tools, through a consideration of the use of plagiarism detection software as a mechanism for the automated assessment of student-created narrative in a virtual learning environment (VLE). The authors are currently engaged in a project creating a prototype VLE, using technologies for multilevel and multiplayer games, based on the inherent support such an environment would provide for constructivist learning, engagement, and contextual socialization. Progress between levels in the VLE will be based on the creation, by the student, of a narrative linking together a number of conceptual elements obtained through game-play at that level. Support for the narrative creation process will help the student to contextualize the conceptual elements, providing the necessary linking elements or themes to enable the student to produce a coherent description of their understanding of the concepts. A particular challenge in such environments is the need for fast, real-time feedback to students to maintain the level of engagement and to support the game-play metaphor. Additionally, the student must be able to make as many attempts to progress as they need and it will be their decision when and how often to submit for assessment. Since the student narrative will be in a textual form and can therefore be related to a sample solution narrative, generated by the author of the level within the learning environment, the idea of using plagiarism detection software as the mechanism for automated comparison and assessment was considered appropriate for investigation. While the limitation of such tools would appear to be that they are seeking direct copies of text elements, the authors wanted to investigate whether they offered sufficient richness and fuzziness to detect common conceptually-linked texts. The initial decision was to experiment with text-analytic tools, since they are both widely used and readily available. The tools chosen were TurnItIn, a commercial tool provided to the U.K. higher education community by the U.K. Joint Information Systems Committee (JISC), and VALT/VAST, a set of tools created at the Centre for Interactive Systems Engineering at London South Bank University, London, U.K., the workings of which are based on recognized and well-published research. An experiment using a small group of students in a traditional assessment situation was carried out, and is described in detail. The rationale for this approach was that there is not yet a fully working prototype of the VLE in which to carry out such an experiment, but that the conditions necessary to test the hypothesis that plagiarism tools could be utilized for such a purpose could be replicated sufficiently to make such an experiment viable. The results of the experiment demonstrated neither a correlation between the sample solution and student solutions, nor any correlation between the individual student solutions, proving the null hypothesis. This result demonstrates that these tools are not useful for the development of automated assessment within the VLE, and the authors are now giving consideration to the use of lexical analysis/tokenizer and other tools. However, it also suggests that these text-analytic plagiarism tools are too firmly focused on direct copy, which does raise the question of whether or not they offer enough richness and fuzziness to detect a sophisticated plagiarism attempt using, for example, text replacement tools. An ongoing close relationship between research in automated assessment and plagiarism detection is also proposed, to achieve mutual benefit.**

***Index Terms*—Automated assessment, formative and summative assessment, games-based virtual learning environment, plagiarism detection tools, student-created-narrative-based assessment.**

Manuscript received January 1, 2007; revised April 23, 2007.

The authors are with the Department of Technology, Buskerud University College, Kongsberg 3601, Norway (e-mail: Olaf.Hallan.Graven@hibu.no; lachlan.mackinnon@hibu.no).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TE.2007.914940

I. Introduction

THE project described here focuses on the creation of a games-based virtual learning environment (VLE), using tools developed for multilevel and multiplayer games. The system design uses a games-based model, which provides inherent support for constructivist learning in a higher education environment. The gaming format also profits from the known advantages of narrative from oral traditions, and fits with the younger generation's interest in current trends in the entertainment industry. The decision to adopt a game-based approach to the development of e-learning follows a survey of current state-of-the-art support for lifelong learning that was carried out by the authors [1], a key element of that investigation being the identification of different techniques to improve learning and retention through engagement of the learner.

In the design for the game-based learning environment the subject area is divided into topics and subtopics, which are then modelled as levels within the game. The different levels will contain multiple quests, each representing some contextualized learning material for the student. The aim of the game is to complete all the levels, and progress within the game is controlled by assessments, which are focused on the learning that has taken place in completing quests on that particular level. The basis for the game model, within each level, is the completion of a number of these quests, using the quest format from existing computer games models [2]. The quest format is built on the simple principle of setting a task for the players and then rewarding them on the successful completion of that task. If a player does not succeed then they have the option to go back and try again until they are successful. Successful completion of each of these quests will provide the students with conceptual information related to the topic of study, and the quest itself will also provide some contextual information linked to the area of study. Once a student has completed enough quests, s/he is in a position to consider the creation of a narrative linking the concepts they have obtained within the context identified. At this point the environment will provide narrative construction support. This support takes the form of helping the student to take the conceptual elements and link them in a structure that uses contextual information that may have been either acquired from the quests or generated from the student's own understanding of the way that these concepts interrelate. The narrative construction mechanism will help the student to create their own narrative as the explicatory description for this particular level. When the student feels s/he has gained enough from this help facility, and considers him/herself to be ready, s/he can submit to an assessment. This assessment requires the student to complete an individual narrative, without support, to describe how the conceptual information gathered from the quests can be related within the context of the subject of study. The student can enter the narrative construction area once s/he has completed enough quests, this threshold being determined by the author of the level. As already indicated, the model operates on the basis of assessment when the student is ready, rather than at an arbitrary time set by others (i.e., end of term, week x of semester, etc.), although there may well be some terminal deadline if the learning is associated with a course of study.

The authors have previously published [3] a complete conceptual design, and some parts of the logical and physical design for such an environment. One of the challenges in using any games-based environment is that the feedback to the students on the assessments must be fast and in real-time to maintain flow [4]. *Flow* is a high state of engagement in the current task/game-play, achieved when the challenge undertaken is neither too easy nor too hard. Since this is a multilevel, multiplayer environment, many assessment attempts may be submitted at any given time. This, coupled with the need for fast response to a student narrative, which will be in a textual form, predicates the need for an automated assessment system.

The particular focus for this study is therefore the consideration of an automated assessment system for student-created narrative. Since the student narrative will be in a textual form, and can be related to a sample solution narrative generated by the author of the level within the learning environment, the idea of using plagiarism detection software as the mechanism for comparing similarities between student-created narrative and the sample solution narrative, was identified as an appropriate area for study.

II. Learning and Assessment in a Games-Based VLE

The game format is inherently constructivist and does offer a way to permit multiple attempts at solving one task, constructed in such a way as to increase the amount of help available on each rerun. Thus, students can be offered the opportunity to retry the same tasks multiple times, if necessary, and these retries can either be immediate or subsequent to attempting other quests. Different students can be offered different support during the learning phase, dependent on their actual needs. When a student reaches the final assessment phase of each learning level, all scaffolding and support will be removed, to allow them independently to demonstrate their understanding of the topic area.

To decide when a student has reached this final phase, the system has to be able to assess automatically a narrative (free text) produced by a student. This assessment is intended to test the student's ability to produce a coherent narrative, demonstrating an understanding of the topic under consideration. The student producing a narrative that uses the concepts and terms they have acquired during game play and that links these concepts in an appropriate or correct way, demonstrates this understanding. The textual glue that connects the concepts and the use of the concepts appropriately or correctly represents the key element of the assessment. The students create the narrative as they progress though the learning material, and when they feel that they are ready to attempt an assessment they progress to the narrative room; here they will be presented with an interactive learning scenario on how to create a narrative with given concepts.

The use of narrative assessment is pedagogically regarded as a good mechanism for the demonstration of understanding. The use of essay or discursive text to describe understanding has its foundations in the Greek and Roman traditions. A modern description, discussion and history of writing as assessment can be found in the work of Hamp–Lyons [5]. One major drawback of the traditional essay-style assessment is that while this assessment is based on the students constructing and developing their learning over a period of time, the mechanism used to demonstrate that this learning has taken place usually requires the reproduction of that learning over a much shorter, and often highly pressurized, period of time. One outcome of this may be that the assessment is focused on students' ability to produce an essay over a short period of time, under pressure, and does not reflect their true understanding of the subject area. This is exacerbated by the fact that the assessment is performed after the learning has taken place and is not integrated into the learning process. In the assessment process based on student-created narrative, proposed as part of the VLE, the creation of the narrative by the student is seen as an integral part of the learning process. One of the key concepts for the system is that the creation and development of a narrative can aid students in developing their understanding. In addition, the resulting narrative can be used to demonstrate their understanding of the topic area and thereby prove that learning has taken place. Thus, the narrative construction process is fundamental both to aiding and assessing learning. Hence, the assessment model places no limits on the number of attempts at assessment or the time taken for an assessment by a student. Therefore, students can use the feedback from the assessment of their constructed narrative in a formative fashion if that is their preferred learning strategy.

Educators generally have been consistent in arguing that higher education should be about encouraging students to take a deep approach to their studies, i.e., they should learn the material and not just memorize it to recite in exams. One of the major challenges is that teaching via a deep approach does not necessarily lead to the desired learning outcomes. Assessment, as so clearly shown by both Biggs [6] and Bould [7], drives the learning process and overrides practically every other aspect of curriculum design, with the result that students will be guided more by the assessment itself than by teaching or other forms of input. In order to achieve the best levels of learning and most effective form of assessment, that assessment should be designed into the system and should be an integral part of the learning process. The importance of having the assessment as an integral part of the process, and not something added just at the end, is well documented [6], [8].

Assessment is generally agreed to take two forms, formative and summative. Formative assessment is designed to aid the learning process, and is a mechanism to encourage student reflection by means of synthesis of their existing understanding and then reflection on the feedback they receive on that understanding; Summative assessment evaluates and provides a measure of a student's knowledge, skills or understanding in a discrete topic area. As already mentioned, it is important that summative assessment is included as an ongoing part of the course, and not solely at the end. This point is clarified in the definition provided by the American Association for Higher Education, November 1995: ***"Assessment is an ongoing process aimed at understanding and improving student learning."*** The evaluation of courses must contain both formative and summative assessment.

III. The Use of Plagiarism Detection Tools for Automated Assessment

To achieve the goal of automated assessment the system has to be able to compare two or more narratives, one or more sample solutions and one text produced by a student. That comparison should provide a score that indicates if the student has demonstrated understanding of the topic area, to what level, and if there are elements within the topic area of which the student has not demonstrated understanding. While this does not represent a requirement to determine plagiarism, since the system will track student activity throughout the process and can therefore document that the work presented is authentic to the student, the actual process of comparison is identical to the task required of plagiarism software. Since the manual assessment of a narrative is a time-consuming task, and students immersed in an environment, such as the proposed game-based environment, require fast feedback, there is a clear need to include mechanisms for automated assessment in the VLE. Various forms of automatic assessment have been applied with varying success over the years [9], [10]. Assessments using multiple choice questions, point and click interfaces, diagrams or mathematical expressions are examples of automated assessment that have been successful; automated assessment of discursive or essay solutions, however, has not been either as straightforward or as successful. Research involving the use of computers in assessment started in the early 1960s, and has more recently addressed the automatic grading of essays [11]–[18]. Of particular interest to this research is that some of the methods developed in that earlier research have in more recent times been used as the basis for some plagiarism detection tools. Given this background and the need to develop automated assessment and fast feedback for the proposed VLE, this research considers the possibility of grading student work using tools developed for plagiarism detection. To test the usefulness of plagiarism detection tools for the comparison of a student-created narrative and a sample solution, appropriate plagiarism detection software had to be identified. Since the need is for a comparison of natural text, plagiarism detection tools that are available for such textual analysis were considered. Two sets of tools, in wide use and offering the opportunity for comparative analysis, were chosen: TurnItIn, a commercial tool provided to the U.K. higher education community by the Joint Information Systems Committee (JISC), and VALT/VAST, a set of tools created at the Centre for Interactive Systems Engineering at London South Bank University, London, U.K., whose function is based on recognized and well-published research [19]–[22].

An experiment was then designed to test the hypothesis that these tools would be able to detect similarities between student-created narrative and a sample solution, to a degree that would allow their use for automated assessment. A traditional exam-based scenario was chosen, the rationale for this approach being that although there is not yet a fully working prototype with which to carry out such an experiment, the conditions necessary to test the hypothesis could be replicated sufficiently to make such an experiment viable.

IV. The Experiment

A. The Exam Setup

The exam chosen was an end-of-module exam for a module on "C++ programming" with a group of 13 second-year undergraduate computing students. The exam was a traditional written exam, four hours long, open book, consisting of multiple questions, all of which the students had to answer. The section of the exam that was set up for this trial was marked as 30% of the total available.

The scenario of a traditional exam brings with it the usual set of problems with this type of assessment: the time of the assessment is fixed, there is time pressure due to the fixed length of the exam, the exam situation is abnormal and uncomfortable for the students, and the help available to the students is static and passive. The students were given questions that required them to generate narrative-based answers, and they were provided with conceptual elements, which they were expected to link into these narrative-based answers.

The three primary differences that distinguish this exam situation for learners from that in the proposed VLE are the following.

- The help is static and passive. Help information is provided to the students but it is not interactive as it would be in the VLE.
- The assessment is at a fixed time. The envisaged system will provide an assessment when ready for the students.
- The exam situation is very different to, and far more stressful and uncomfortable than, the working environment for the student. In contrast, the assessment inside the learning environment will be similar to that of the learning environment itself. This major issue is somewhat mitigated by the exam environment being already familiar to the students, if not much loved by them, as an assessment environment.

An additional minor difference is that the exam is paper based. Apart from these differences it can be argued that the conditions are comparable to those envisaged inside the learning environment. The amount of time available for the exam was deemed to be a less significant factor as most students handed in the completed exam before the four hour time limit was up: one student used the full four hours and three others left during the last 15 minutes of the exam, the rest took less than three hours. The exam was also set as an open-book exam, making the conditions as close as possible to the conditions supported in the VLE model. The style of the questions used for this exam was slightly different from traditional essay-style questions, but this difference was kept to a minimum and should have no significant effect on the usefulness of the plagiarism detection tools in comparing the texts thus generated.

A set of questions were then designed, all of which were set, and answered, in Norwegian, the native language for the students. Here are the questions.

1) Explain how to make/construct a class in C++.
2) Explain how to create and delete objects/instances in C++.
3) Explain how to control access to the content of objects during runtime.
4) What does it mean when a method is marked virtual?
5) Give some major differences between Java and C++.

Before being given the questions the students were given an explanation of what was expected of them, this explanation being naturally static and noninteractive, as was commented upon previously. The students were also given the following list of concepts that were to be linked/used in the created narratives:

Access, Binding, catch, Class, Compile, Constructor, Default, delete, Destructor, Dynamic, Exception, Function, Generic, Heap, Inheritance, Java, Member data, Method, Namespace, new, Object, Operator, Overloading, Package, private, protected, public, Pure virtual, Stack, static, Static, Template, throw, try, Type, Variable, virtual.

B. Example of a Sample Solution

The sample solutions were of varying length, approximately a half-page long, and typed into a standard word processor. A sample solution to question one, *"Explain how to make/construct a class in C++"* (in English translation) follows. When the concepts given to the students are used they are bolded for emphasis.

Classes in C++ are made by programming them in source code. The usual process is to create a **class** declaration and a separate definition of the methods. The definitions of the **methods** are usually placed in a separate file. The **class** definition is placed in a header file with a.h extension and the definitions of the methods in a corresponding .cpp file. The **class** declaration can also contain complete **methods:** these are referred to as inline methods. The **class** declaration in C++ is a type declaration and begins with the word **class** followed by a block and is ended with a;.

A **class** usually has both a **constructor** and a **destructor**. The **constructor** is a **method** that is run during the creation of **objects** of the **class** type. The **destructor** is run when a **object** of the **class** type is deleted from memory.

Inside the **class** declaration the programmer can declare constants, **member data** and member **functions** or **methods**. These **member data** and **methods** can be **static** or **dynamically** declared. **Static member data** belongs to the **class** whereas **dynamic member data** belongs to **objects**/instances of the **class**. **Member data** that is **static** can be used by both **static** and **dynamic methods**. **Dynamic member data** can only be used by **dynamic methods. Static methods** can only work with **static member data**.

Classes can **inherit** from other **classes** and **structures**. When **inheriting** from **structures** the **member data** is **inherited**. When **inheriting** from **classes** both **member** data and **methods** are **inherited**. When using **inheritance** the programmer can add new **methods**. When using **inheritance** the programmer can remove/hide **methods**. When using **inheritance** the programmer can alter **methods**. So when using **inheritance** the programmer can add, alter and remove **methods** that are **inherited**.

One thing to note is the attempt by the academic who wrote the sample solution to cover multiple solutions attempted by the students. This effort is particularly noticeable in the last paragraph, which is the alterations of methods in connection with inheritance.

TABLE I
RESULTS FROM TRADITIONAL MARKING OF EXAM, GIVING THE NUMBER OF STUDENTS ACHIEVING THE VARIOUS SCORES

Score	Class in C++	Create object	Access to data	Virtual method	Java vs. C++
0	0	0	1	3	1
1	1	2	2	2	1
2	5	3	2	5	4
3	5	4	3	2	1
4	1	2	1	0	3
5	1	0	0	0	3
6	0	1	2	1	0

C. Result of the Exam

The texts produced by the students were then marked following a traditional process, the texts being marked anonymously by both the module lecturer and an external second marker. The marks for each student were from 0 to 6 for each question marked. The number of students achieving the various scores is shown in Table I.

The results from the two markers following the traditional marking process for the students were similar, and have been combined in the results shown in Table I and Fig. 1. As can be seen from the data there were very few students that managed to achieve a perfect answer on any of the questions. Some of the students did not attempt some questions and these are the ones with zero score. The data shows some differences in the grouping of the scores in the questions, but nothing out of the ordinary for such a small group. Most of the available scores (0–6) were in use for all of the questions.

V. AUTOMATED ASSESSMENT

The automated assessment was carried out in two stages: 1) stage one was the generation of an electronic version from the paper-based student answers in the exam, which also included some data cleansing, is described in the following; and 2) the second stage was the actual use of the plagiarism tools.

All of the answers generated by the students were typed into a word processor to generate an electronic version. The data was cleansed by removing any obvious errors that would have been ignored or condoned by a human marker in the marking process,

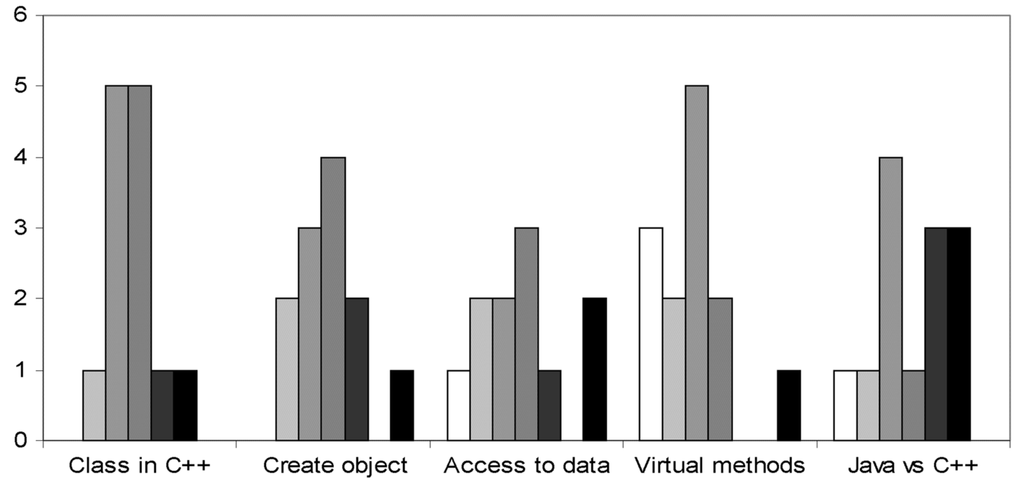


Fig. 1. Results from traditional marking of exam, results from Table I given as graphs.

Fig. 2. A comparison of the solution to Answer 1 and the students' responses in the exam.

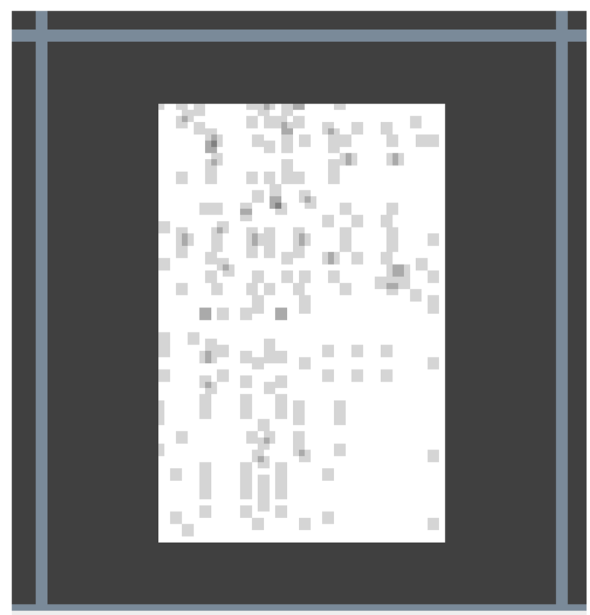


Fig. 3. A comparison of the solution to Answer 1 and one student's response in the exam.

and included creating uniform spelling and removing spelling errors. No other changes were made; no words were changed, nor were any sentences rearranged. The generated files, together with the sample solution, then provided the input for the plagiarism detection tools. Two identical sets of files containing the electronic text were then produced, both a Microsoft Word and an ASCII text version.

As already indicated, the hypothesis to be tested is that these text-analytic plagiarism tools will demonstrate similarities between the sample solution generated by the academic and the student-generated solution, as well as within the student-generated solutions themselves. The closer the correlation between the generated sample solution and the student solution, the higher the score. The null hypothesis is that the tools will not be able to demonstrate any correlation between the sample solution generated by the academic and the student-generated solution, nor within the student generated solutions themselves.

A total of 67 files were generated (five sample solution files and 62 student answers). The reason there were only 62 student answers was that some students had left a blank answer for some of the questions.

A. Result From VALT/VAST

The first tools selected for the experiment were the VALT and VAST tools available from the Centre for Interactive Systems Engineering at London South Bank University.

The visualization front end of VAST shows a graphical representation of multiple results from the VALT comparison tool. The visualization will show a black diagonal line if the documents are identical and several broken and possible fuzzy diagonal lines if there are parts that are similar. The results are shown as follows:

1) comparing sample solution to students' solutions in Question 1 (Fig. 2);
2) comparing sample solution to one particular student solution to Question 1 (Fig. 3);
3) comparing all student solutions to each other for Question 1 (Fig. 4).

The results for the other questions followed the same pattern seen in Fig. 4 when other answers are compared.

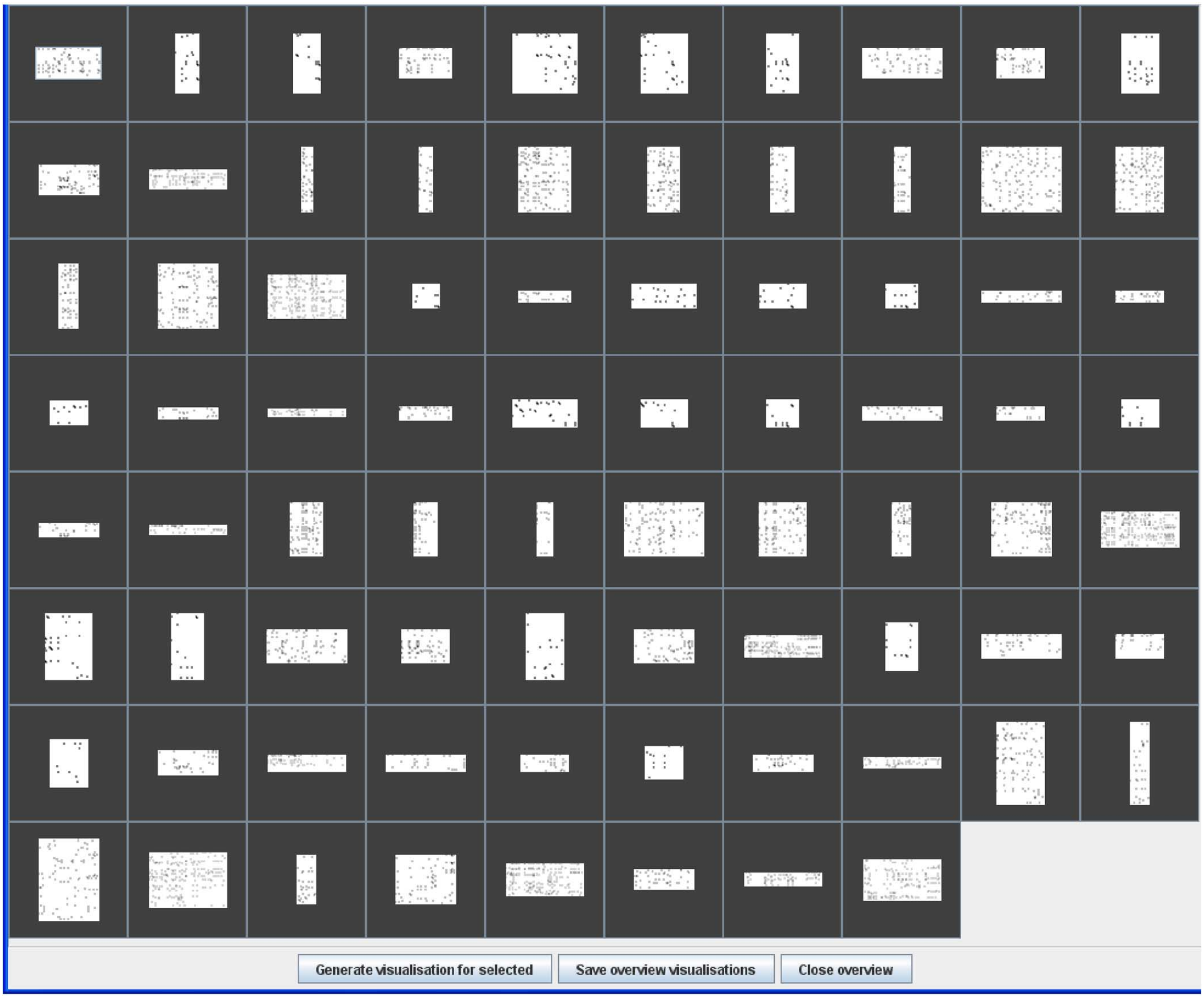


Fig. 4. A comparison of all the answers to Question 1 against each other.

A comparison was made of all the answers to Question 1 against each other, to find if the tools were able to detect any correlation between any individual student answers.

The comparisons using the VALT tools are all visualized in VAST. The pictures in Figs. 2–4 clearly show that no correlations were detected, either between a sample solution and the student answers or within the student answers themselves. The same answers marked by the human markers achieved a score between 1–5 marks out of 6, with no answer being awarded full or no marks. The same results were found for the other questions in the test, with no correlations being detected for any of the questions. When performing a final comparison with all student answers for all questions simultaneously, the results were the same, with no correlations being found.

The conclusion drawn is that the results from the use of the VALT/VAST toolset demonstrated no correlations, supporting the null hypothesis.

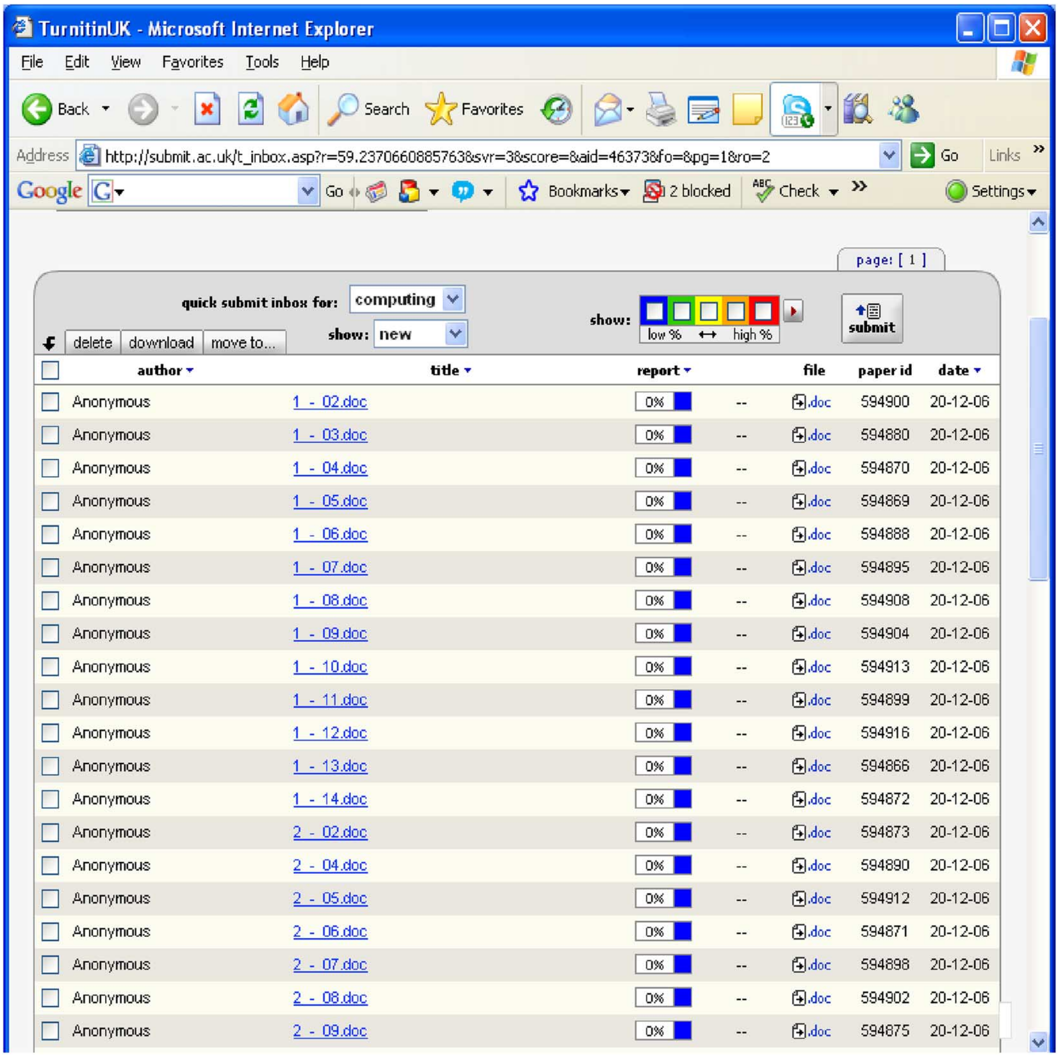
B. *Result From TurnItIn*

The second tool used for the experiment was TurnItIn, a commercial tool provided to the U.K. higher education community by the U.K. JISC. This tool requires login and then upload of the text files to be compared, in a Microsoft Word format or similar. The tool provides no mechanism to compare one sample solution to multiple student answers, instead providing an all-with-all comparison as a standard function. Therefore, this function was used for the experimental comparison.

The student answers had already been entered into Microsoft Word files using a common naming format: *question number dash student number*; 1—solution.doc, 1—02.doc 1—03.doc and so on for question 1; 2-02.doc 2-03.doc and so on for the rest. All of these files were uploaded into TurnItIn; at this stage four of the answers were rejected on the ground that they contained less than 100 characters.

Some of the output from TurnItIn is given in Fig. 5. The information shown is as follows:

- author—this was anonymous in all cases;
- filename—question number dash student number.doc;
- comparison score—this was 0% in nearly all cases except for two trivial cases shown in the following;
- file type—doc in all cases;



	author	title	report		file	paper id	date
	Anonymous	1 - 02.doc	0%	--	.doc	594900	20-12-06
	Anonymous	1 - 03.doc	0%	--	.doc	594880	20-12-06
	Anonymous	1 - 04.doc	0%	--	.doc	594870	20-12-06
	Anonymous	1 - 05.doc	0%	--	.doc	594869	20-12-06
	Anonymous	1 - 06.doc	0%	--	.doc	594888	20-12-06
	Anonymous	1 - 07.doc	0%	--	.doc	594895	20-12-06
	Anonymous	1 - 08.doc	0%	--	.doc	594908	20-12-06
	Anonymous	1 - 09.doc	0%	--	.doc	594904	20-12-06
	Anonymous	1 - 10.doc	0%	--	.doc	594913	20-12-06
	Anonymous	1 - 11.doc	0%	--	.doc	594899	20-12-06
	Anonymous	1 - 12.doc	0%	--	.doc	594916	20-12-06
	Anonymous	1 - 13.doc	0%	--	.doc	594866	20-12-06
	Anonymous	1 - 14.doc	0%	--	.doc	594872	20-12-06
	Anonymous	2 - 02.doc	0%	--	.doc	594873	20-12-06
	Anonymous	2 - 04.doc	0%	--	.doc	594890	20-12-06
	Anonymous	2 - 05.doc	0%	--	.doc	594912	20-12-06
	Anonymous	2 - 06.doc	0%	--	.doc	594871	20-12-06
	Anonymous	2 - 07.doc	0%	--	.doc	594898	20-12-06
	Anonymous	2 - 08.doc	0%	--	.doc	594902	20-12-06
	Anonymous	2 - 09.doc	0%	--	.doc	594875	20-12-06

Fig. 5. A comparison of the sample solution to Question 1 and the student solutions.

- paper id—internal for TurnItIn;
- date submitted—20 December 2006 in all cases.

As shown in Fig. 5, the main result from TurnItIn was in essence the same as that for the VALT/VAST tools, i.e., no correlations were found. TurnItIn did find some trivial correlation as follows.

Student X:

C++ does not have a **thread**
running in the background **doing garbage collection**.

Student Y:

C++ does not have a **Virtual Machine**
running in the background.

VI. Conclusion

The results detailed previously for the experiment show that no correlations were found with the VALT/VAST tools and only a trivial correlation was found with TurnItIn. This holds both between sample solution and student solutions, and within the individual student solutions. On the basis of these findings the null hypothesis is proven.

Since this is only a small sample set and the answers generated by the students were relatively short, no statistical significance can be claimed for these outcomes. However, the fact that there are no correlations between sample solutions and student solutions nor any correlation within the individual student solutions, suggests that these particular text-analysis plagiarism tools are too firmly focused on direct copy, and do not provide enough richness and fuzziness to meet the needs of automated assessment within the proposed VLE. Clearly, in practice within the VLE, where the students select the time of assessment, take as long as they need, and can have multiple attempts, longer narratives might well be generated. While this might potentially be thought to be valuable in providing the plagiarism tools with more text to analyze, given the fact that the student solutions in the experiment are all of a similar size, style, and shape and the tools did not find any correlation between them suggests that, even given these longer texts to analyze, these tools would be of little value for automated assessment. However, there are other plagiarism detection tools to consider, and experiments with lexical analysis/tokenizer tools, such as those used for plagiarism detection in source code analysis, proximity analysis tools, and other existing automated essay marking tools, are now being undertaken.

The fact that the tools did not find any correlations, even though the student solutions are all of a similar size and shape, and are based on the same concepts and keywords, also raises a question about the tools themselves. Are these text analysis tools too firmly focused on direct copy? Specifically, do they offer enough richness and fuzziness to discover a sophisticated plagiarism attempt, where a text is copied and then some words replaced and altered without altering the content, for example, using text replacement tools to change terms and obfuscate a plagiarized text? This question is not one that this research will attempt to address, but it is of particular importance, given the known prevalence of plagiarism and the outcomes of this experiment. Human inspection of the solutions generated by the students demonstrated considerable commonality between them, and in the marking process the comparisons with the sample solution also identified strong similarities, yet these went undetected by the plagiarism tools. Could it be argued that the tools are successful at detecting unsophisticated, naïve, and blatant forms of plagiarism, but incapable of dealing with sophisticated, covert, or even relatively simple substitutional forms of plagiarism? This research offers no current answers or solutions to this question, which remains an open issue for consideration and further research by the plagiarism community. However, in the longer term, a close relationship between research in automated assessment and plagiarism detection could offer considerable benefits to both areas.

References

[1] O. H. Graven and L. M. MacKinnon, "A survey of current state of the art support for lifelong learning," in *Proc. 6th Int. Conf. Information Technology Based Higher Education and Training*, Santo Domingo, Dominican Republic, Jul. 2005, pp. F2C/19–F2C/25.

[2] E. Aarseth, "Quest games as post-narrative discourse," in *Narrative Across Media*. Baltimore, MD: The Johns Hopkins Univ. Press, 2002.

[3] O. H. Graven and L. M. MacKinnon, "Exploitation of games and virtual environments for e-learning," in *Proc. 7th Int. Conf. Information Technology Based Higher Education and Training*, Ultimo, Australia, 2006, pp. 409–421.

[4] M. Csikszentmihalyi, *Flow: The Psychology of Optimal Experience*. New York: Harper Perennial, 1990.

[5] L. Hamp-Lyons, "The scope of writing assessment," *Assess. Writing*, vol. 8, pp. 5–16, 2002.

[6] J. B. Biggs, *Teaching for Quality Learning at University: What the Student Does*, 2nd ed. Philadelphia, PA: Open Univ. Press, 2003, Society for Research into Higher Education.

[7] D. Boud, "Assessment and the promotion of academic values," *Studies High. Educ.*, vol. 15, pp. 101–111, 1990.

[8] D. Boud, "Assessment and learning: Contradictory or complementary?," in *Assessment for Learning in Higher Education*. London, U.K.: Kogan, 1995, pp. 35–48.

[9] E. Foxley, C. Higgins, P. Symeonidis, and A. Tsintsifas, "The coursemaster automated assessment system—A next generation ceilidh," presented at the Computer Assisted Assessment Workshop, Warwick, U.K., 2001.

[10] E. Foxley, C. Higgins, and A. Tsintsifas, "The ceilidh system: A general overview," presented at the 2nd Annu. Computer Assisted Assessment Conf., Loughborough,, U.K., 1998.

[11] P. W. Foltz, D. Laham, and T. K. Landauer, "Automated essay scoring: Applications to educational technology," presented at the World Conf. Educational Multimedia, Hypermedia and Telecommunications, Seattle, WA, Jun. 1999.

[12] D. Kanejiya, A. Kumary, and S. Prasad, "Automatic evaluation of students answers using syntactically Enhanced LSA," presented at the Human Language Technology/North American Chapter of the Association for Computational Linguistics Workshop on Building Natural Language Processing, Edmonton, AB, Canada, 2003.

[13] T. K. Landauer, D. Laham, and P. W. Foltz, "The intelligent essay assessor," *IEEE Intell. Syst.*, vol. 15, no. 5, pp. 27–31, 2000.

[14] L. S. Larkey, "Automatic essay grading using text categorization techniques," presented at the Int. Conf. Research and Development in Information Retrieval, Melbourne, Australia, 1998.

[15] D. Lonsdale and D. Strong-Krause, "Automated rating of ESL essays," presented at the Human Language Technology/North American Chapter of the Association for Computational Linguistics Workshop on Building Natural Language Processing, Edmonton, AB, Canada, 2003.

[16] M. D. Shermis, H. R. Mzumara, J. Olson, and S. Harrington, "On-line grading of student essays: PEG goes on the World Wide Web," *Assess. Eval. High. Educ.*, vol. 26, no. 3, pp. 247–259, Jun. 2001.

[17] R. M. Duwairi, "A framework for the computerized assessment of university student essays," *Comput. Human Behav.*, vol. 22, no. 3, pp. 381–388, May 2006.

[18] T. Mitchell, N. Aldridge, W. Williamson, and P. Broomhead, "Computer based testing of medical knowledge," presented at the 7th Computer Assisted Assessment Conf., Loughborough, U.K., 2003.

[19] F. Culwin and T. Lancaster, "Plagiarism issues for higher education," *Vine*, vol. 123, pp. 36–41, 2001.

[20] F. Culwin and T. Lancaster, "Visualising intra-corpal plagiarism," presented at the 5th Int. Conf. Information Visualisation, London, U.K., 2001.

[21] T. Lancaster, "Efficient and effective plagiarism detection," Ph.D. disseration, London South Bank Univ., London, U.K., 2003.

[22] F. Culwin and T. Lancaster, "Towards an error free plagiarism detection process," presented at the Innovation and Technology in Computer Science Education Conf., Canterbury, U.K., Jun. 2001.

Olaf Hallan Graven received the Bachelor's degree in computer engineering from Kongsberg Ingeniørhøskole, Kongsberg, Norway, and the B.Eng. and M.S. degrees from the University of Strathclyde, Glasgow, Scotland, U.K., in 1991, 1993, and 2005, respectively.

He is working towards the Ph.D. degree in games-based virtual learning environments at Buskerud University College, Kongsberg, Norway. He was a Lecturer in Computing from 1993 to 2000, served as Head of Department for the Department of Computing from 2000 to 2003, and has been a Research Fellow in the Department of Technology since 2003 at Buskerud University College. His areas of interest include programming languages, software engineering, engineering education, e-learning, computer games development, and simulations.

Lachlan Mhor MacKinnon (M'01) received the B.Sc. Honours degree in computer science and the Ph.D. degree in intelligent query manipulation for heterogeneous databases from Heriot-Watt University, Edinburgh, U.K. in 1992 and 1998, respectively.

He had a 15-year career in the United Kingdom in health and social services, obtaining qualifications in nursing and social work, before moving into computer science in 1988. He is Professor of Information and Knowledge Engineering and Head of the School of Computing and Creative Technologies at the University of Abertay Dundee, Dundee, U.K., and Professor of Multimedia Technology at Buskerud University College, Kongsberg, Norway. His research interests include information and knowledge engineering, smart systems, creative technologies, eHealth and eLearning, and computer security.

Dr. MacKinnon is a member of the U.K. national committee of the Council of Professors and Heads of Computing (CPHC) and Chair of the Information Group of CPHC. He is Chair of the executive committee of the British National Conference on Databases, and a member of the U.K. national committee of the British Human Computer Interaction Group. He is a Fellow of the British Computer Society, a Chartered Information Technology Professional, a member of the Association for Computing Machinery, Arts Computation Engineering, and the IEEE Standards working groups.