University of Salzburg

Automated Assessment in Massive Open Online Courses

Seminar aus Informatik

Authors:
Siegmar Alber (0720046)
Luca Debiasi (0720045)

16.07.2013

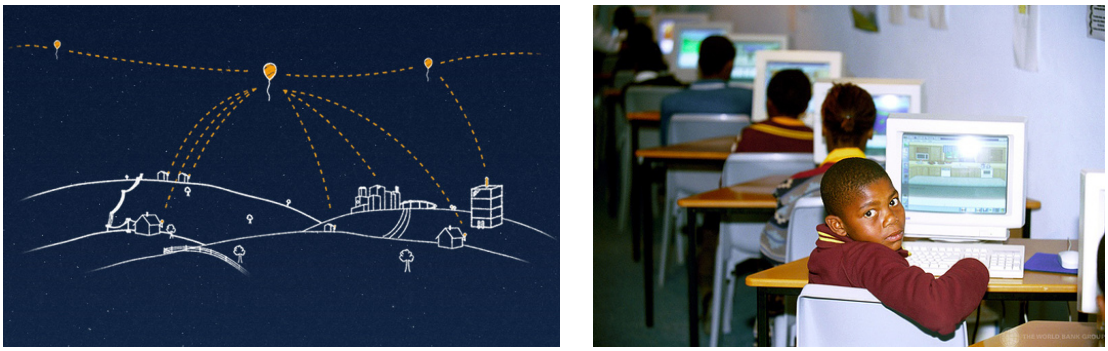
Contents

1 Introduction

Every human has the right to education, as noted in the Universal Declaration of Human Rights. Education is also the key to a successful society and forms the base of all the things we do. We act and behave based on our knowledge, gathered through observation, assimilation and instructions.

While the the first two can also be performed without the help of a third person, the latter one implies an educated person as instructor. The instructor also needs to have a better knowledge than the student in the field he is instructing, so that the student is able to expand his know-how. Usually the number of instructors is inferior to the number of apprentices, especially in developing countries where it can be difficult for everyone to get a proper education, especially higher education. There are often also spatial challenges in bringing instructors and students together because of weak infrastructure or just the geographical distance, even in well developed countries. This is where technology comes into play.

The internet is a great possibility to communicate with other people all over the world and is pushing into developing countries more and more, e.g. with Project Loon from Google (figure 1). So why not use it to bring instructors and apprentices together and provide high-quality education to everyone?



(a) Project Loon (Google).

(b) Students using computer at school.

Figure 1: Ways to provide high quality education in developing countries. Project Loon is a network of balloons traveling on the edge of space, designed to connect people in rural and remote areas, help fill coverage gaps, and bring people back online after disasters. (`http://www.google.com/loon/`).

Massive Open Online Courses (MOOCs) are a platform for students to participate in courses via internet. Because these courses are open (i. e. can be accessed from anywhere in the world) it is possible that a huge number of students follow the same course. Well known providers of such platforms include Coursera, edX and Udacity.

A big challenge that needs to be addressed for MOOCs to work is automated assessment of student work: Because of the large number of students participating in a single course it is unlikely for a teacher to manually evaluate each single assignment.

Additional issues include giving students detailed feedback to help them recognizing weaknesses and getting better. The assignment process itself, i. e. how to transfer the students work to the evaluator or teacher, needs to be considered, too.

In this paper we will concentrate on the automated assessment process:

What type of documents can be evaluated automatically?

Which techniques exist and how do they work?

What are the benefits and problems of automated assessment?

What will the future of MOOCs be like with automated assessment?

We will give answers to this questions and present an overview on this challenging topic.

2 Concept for Automated assessments

As briefly noted in the previous section, automated assessments are needed to handle MOOCs with a proper effort and therefore in an economical way. Courses with over $50,000$ students (Sahami et al. [19]) lead to new challenges in the organization of the course and evaluation and marking of the assessments. Of course one could just hire lots and lots of tutors, which help to solve this problems, but this would generate huge expenses that have to be covered. Automated systems are a great possibility to address this problems, but a lot of challenges have to be mastered before these systems can be on par with a human performing this tasks.

The main tasks of an automated assessment system is to simply evaluate assignments and tests created by the instructor and accomplished by the students. The output of the system should support the instructor to evaluate the level of knowledge of each student and help him to determine an appropriate grading for the student. All this should be realized with as few as possible work for the instructor.

This evaluation of the level of knowledge should not only take into consideration the correctness and functionality of the solution from the student, but should also include other aspects as quality of the work done, design, form and terminology respectively phrasing. It should also consider the case that for certain assignments there is not only one correct solution and the endeavor of the student in elaborating the solution.

The revising of assignments and tests is just one part of the whole assessment process and beyond that there are other time-consuming tasks that have to be performed in this process. Such tasks are:

- Compile tests and assignments
- Distribute this tests and assignments to the students
- Collection of the accomplishments from the students
- Give feedback to the students on their solutions
- Detection of plagiarisms
- Grading of assignments

An extended and sophisticated automated assessment system should also support the instructor in performing the tasks mentioned above and also help performing administrative tasks regarding the course, e.g. as to determine who is participating to the course.

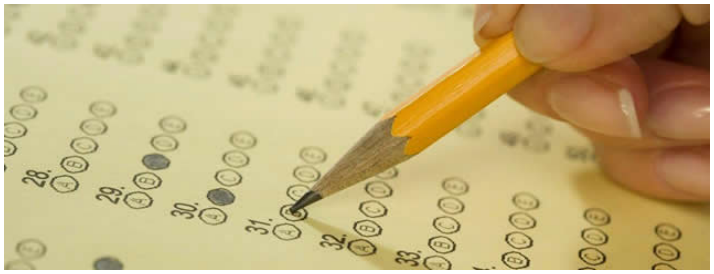
In the following section we are having a closer look at how the evaluation of the level of knowledge is performed for different kinds of assignments.

3 Document types to assess

This chapter focuses on different ways of how the level of knowledge can be expressed by the students. Obviously the presented types do not cover all possibilities of how knowledge can be expressed, e.g. oral exams are not considered.

In this section we concentrate on written documents and give an overview of the current and past research activities regarding different document types.

3.1 Selection tasks



(a) Multiple choice test.

Penguins

Fill in all the gaps, then press "Check" to check your answers.

adaptations **allows** **exclusively** **flightless** **harsh** **ingesting**

Penguins are a group of _____ birds that live almost _____ in the southern hemisphere. Th
the largest and the Little Blue Penguin being the smallest. Some penguins are able to _____ temperatures
help them _____ in the _____, icy climate such as waterproof feathers and extra _____ of f
water without _____ the salt.

Check

(b) Cloze about penguins.

Figure 2: Examples for selection tasks.

The basic document type that intuitively come into someones mind when thinking about automated assessment are multiple choice tests. They are wide spread and often used for tests that many people are willing to take, e.g. driving license tests, because they can be easily assessed. Usually they consist of one or more question and each question has different predefined answers to choose from. The correct answer (or answers, depending if it a single choice test or multiple choice one) to the question is among this possible answers together with incorrect ones. The student only has to check the box near the answer he thinks to be correct. Since the number of correct answers is prede-

fined and the marking of boxes is easy to read even for machines this kind of tests can be evaluated automatically since many years.

A cloze is a type of text where some words are substituted by blanks. This blanks have to be filled in by the students with a word to complete the sentence. Usually there is a set of words from which the student can choose the word from that he wants to fill into the blank. Since the correct word to fill into each blank is predefined it is also trivial to asses this type of document.

It can be stated that the assessment of these two types of selection tasks is trivial and easy to implement since the answers a student can give to the questions are usually predefined. Such form of assessment provides only limited insight into students knowledge [14]. The flaw with predefined answers is that it can not be determined if the student chose the correct answer based on his knowledge or if it was just a fluke.

3.2 Text

A text can be seen as a linguistic manifestation of a communication between two subjects, in our case the instructor and the student. The student responds to a question in a written form, with structured sequences of words treating the questions topic.

There are many possibilities how to respond correctly to a question in form of a text and even more how to respond incorrectly. So text can be regarded as a rather complex expression of knowledge. It is not clear yet which features distinguish a good from a bad text, but assessment of texts should not only take into consideration the content and spelling, but also phrasing, argumentation, structure and in some cases it should adhere to specific patterns.

As noted by Valenti et al. [24] automated assessment could lead to the discovery of features that are different for good and bad writing. This is also a chance to establish a guideline for humans to objectively grade writings, because the assessment and grading of the same text, e.g. an essay, performed by different human assessors varies because of subjective peculiarities of the human assessors [24][17]. Essays are best when assessing the learning outcome of students, so the majority of the research in assessment of text focuses on essays [9][24][23].

When looking at automated text assessment there are three major categories to distinguish: assessment of content only, assessment of style only, assessment of style and content. In the next paragraphs we will present an example for each category.

An example for an automated assessment system focusing exclusively on the content is the Intelligent Essay Assessor (IEA). As described by Valenti et al. [24] it is based on a technique called Latent Semantic Analysis that was originally designed for indexing documents and text retrieval (Deerwester et al.[7]). LSA represents documents and their word content in a large two-dimensional matrix semantic space (Williams, R. [25]). Us-

ing a matrix algebra technique known as Singular Value Decomposition (SVD), new relationships between words and documents are uncovered, and existing relationship are modified to more accurately represent their true significance.

To grade an essay, a matrix for the essay document is built, and then transformed by the SVD technique to approximately reproduce the matrix using the reduced dimensional matrices built for the essay topic domain semantic space. (The semantic space typically consists of human graded essays). Cosine correlation is used to measure the similarity of the reduced dimensional space constructed from a "model answer", such as an instructional text taken from a course text or an essay prepared by the class tutor, against a student essay. Valenti et al. [24] affirm that LSA makes no use of word order since the authors claim that this is not the most important factor for grabbing the sense of a passage. It also requires large amounts of data to construct a suitable matrix representation of word use/occurrence, and due to the size of the matrices involved computations are very cumbersome.

Looking at the assessment of text-style only we present the Project Essay Grade (PEG). As noted by Valenti et al. [24] PEG is one of the earliest and longest-lived implementations of automated essay grading. It was developed by Page and others (Hearst [10]) and primarily relies on style analysis of surface linguistic features of a block of text. Thus, an essay is predominantly graded on the basis of writing quality, taking no account of content.

The design approach for PEG is based on the concept of "proxes", i.e. computer approximations or measures of so called trins, intrinsic variables of interest within the essay (what a human grader would look for but the computer can't directly measure) to simulate human rater grading. Proxes include: essay length (as the amount of words) to represent the trin of fluency; counts of prepositions, relative pronouns and other parts of speech, as an indicator of complexity of sentence structure; variation in word length to indicate diction (because less common words are often longer). Proxes are calculated from a set of training essays and are then transformed and used in a standard multiple regression along with the given human grades for the training essay to calculate the regression coefficients. These regression coefficients represent the best approximation to human grades when obtained according to proxes. Then, they are used with the proxes obtained from unmarked essays to produce expected grades. PEG purely relies on a statistical approach based on the assumption that the quality of essays is reflected by the measurable proxes.

No Natural Language Processing (NLP) technique is used and lexical content is not taken in account. PEG also requires training, in the form of assessing a number of previously manually marked essays for proxes, in order to evaluate the regression coefficients, which in turn enables the marking of new essays.

As a combination of content and style assessment Valenti et al. [24] present the Paperless School free-text Marking Engine (PS-ME). PS-ME is designed as an integrated component of a Web-based Learning Management System (Mason & Grove-Stephenson [16]) and is still under development. Due to its processing requirements, the PS-ME does not grade essays in real-time.

This system applies Natural Language Processing (NLP) techniques to assess student essays in order to reveal their level of competencies as for knowledge, understanding and evaluation. The student essay is submitted to the server, together with information about the task in order to identify the correct master texts for comparison. Each task is defined via a number of master texts that are relevant to the question to be answered. An interesting issue is raised by the existence of 'negative' master texts containing a set of false statement composed using typical student mistakes and misconception. The essay to be rated is compared against each relevant master text to derive a number of parameters reflecting the knowledge and understanding exhibited by the student. The ability to evaluate parameter is calculated through a linguistic analysis as described above. When multiple master texts are involved in the comparison, each result from an individual comparison gets a weighting, which could be negative in the case of a master text containing common misconceptions. The weights are derived during the initial training phase. The individual parameters computed during the analysis phase are then combined in a numerical expression to yield the assignment's grade. The parameters are also used to select specific comments from a comment bank relevant to the task. With a fine-grained set-up it is possible to provide the student with formative feedback regarding his/hers performance in different areas within a given subject.

The output from the marking process is then passed back to the client for presentation to the teacher. This includes details on sections of essay that are particularly good or bad in relation to the knowledge, understanding and evaluation factors.

3.3 Mathematical formulas

Student work in mathematical or engineering courses almost always contain mathematical formulas. This section gives an overview about how to automatically assess such mathematical formulas. Information for this section is taken from [18].

Formulas, when compared to text (see previous section), are well structured and can be interpreted reasonably unambiguously. However even though formulas can be understood precisely, they can be written down very different: E. g. one can use other symbols or rearange terms in other ways. The formula shown in figure 3 demonstrates this in a humoristic way. To assess a formula, its mathematical correctness and syntax needs to be verified.

One method to extract and verify formulas from engineering assignments is described in [18]: First all formulas in the document are required to be in MathML (or need to be converted to it). MathML is a XML format to represent mathematical terms

$$\ln\left(\lim_{c\to\infty}\left(1+\frac{\left(\left(x^T\right)^{-1}-\left(x^{-1}\right)^T\right)!}{c}\right)^c\right)+\sin^2 a+\cos^2 a=\sum_{n=0}^{\infty}\frac{\cosh s\cdot\sqrt{1-\tanh^2 s}}{2^n}$$

Figure 3: Alternative representation of $1+1=2$.

and formulas. From this representation a math tree is created. Operators are transformed to root nodes, their operands to child nodes. This tree is used as input vector for a support vector machine (SVM), a supervised learning approach for artificial neural networks. This SVM decides (after being trained) if the formula is correct or not.

3.4 Programming assignments

Programming assignment is the document type the most research is done for, probably because research in this field is mostly done in computer science faculties. Research already started in 1973 (e. g. [1]) or maybe even earlier. Current exaples include [3], [4], [8], [11], [12], [15], [21] and [26].

There are multiple characteristics that influence the grade of some programming assignment. Besides correct syntax and semantic of the code and proper behavior of the program, its code quality and usage of patterns can be used as additional metrics, too. Probably there are even more possible features to check on.

Some of this features are relatively simple to verify. E. g. the programs syntax is probalby correct if the program compiles. Others are more difficult, like the usage of patterns. This is one of the reasons there are so many different systems. Another reason is that many systems only work for one or two specific programming languages.

Systems like GAME[3] or Sakai[21] do a simple code analysis (number of variables, amount of comments, occurrence of loops) to evaluate the code quality. Because checking the output of the program for different inputs makes up a major part of the evaluation result, these systems are, to a high degree, independent of the programming language in use.

Other systems like AutoGrader[11] or Infandango[12] require the student to implement an interface and then then does unit tests on the students implementation. Additional to that it does a static code analysis to find potential programming problems and reports them as feedback to the students.

The system AnalyseC[26] tries another approach: It builds an AST (abstract syntax tree) from the students program, performs normative transformations on it and compares it with the teachers reference implementation. Using this technique the programs structure can be evaluated.

3.5 Other approaches

There exist multiple other systems for automated evaluation of students knowledge. One of them concentrates on helping students learning to use some software. The system in [22] describes automated assessment of spreadsheet concepts. The students needs to create formulas and manipulate cells and the system verifies them.

Another example is VM (virtual machine) system administration. In [2] a system is explained that verifies the correct setup and functionality of a virtual machine. This is realized using shell scripts running in the VM under test. Using this method various parameters can be checked, like for example disk size, memory configuration or network connection.

Especially for children the system SAVE[20] was developed to automatically test their scientific knowledge. The system presents a simulated world to the student asking him to solve a specific problem. The student can control e. g. the temperature, pressure or humidity to change the simulated worlds state which causes the solution of the problem. The system observes the students behavior while solving the problem and is hence able to mark the students knowledge.

The system described in [18] combines the evaluation of mathematical formulas, as described in section 3.3, with assessment of text, as in section 3.2, in order to allow marking of whole engineering assignments.

As a attempt to generically solve the automated marking problem, ODALA[5] tries an ontology driven approach. A knowledge network, called ontology, is created by the teacher with all facts and relations connecting them. The system then extract the facts from the students work and tries to match them with the teachers reference.

Last but not least there are multiple methods and systems to support teachers with the assignment and marking process as a whole. As an example the BOSS[13] system helps with the administrative tasks (the assignment itself, marking, giving feedback to students) letting the marking system itself interchangeable. Thus this system can be used to manage any type of student work.

4 Discussion and outlook

While online educational resources have existed for decades, the past year has seen a tremendous acceleration in the adoption and potential disruption afforded by recent online education initiatives. In the Fall of 2011 the concept of Massive Open Online Courses (MOOCs) more broadly entered the public consciousness with a group of three Stanford CS courses collectively enrolling over 200,000 students, and gaining national press coverage from the New York Times and other outlets [19].

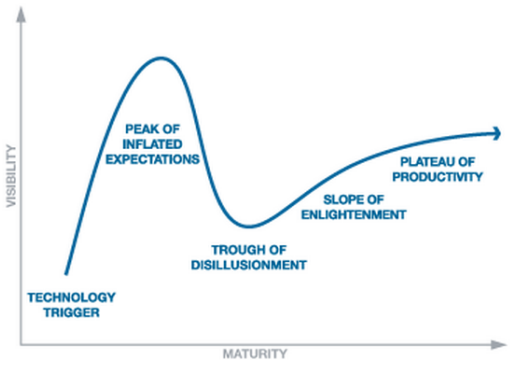


Figure 4: A technology's life cycle.

Mehran Sahami [19] says, that beyond the publicity, however, the availability of such online materials has resulted in universities in the US, China, India, and South America using these videos as part of teaching their own introductory programming courses. Of note is the fact that these online materials generated such a strong response without any of the affordances of the most recent generation of MOOCs (enrollment, quizzes and assessments, statements of accomplishment, etc.). Based on these experiences, we believe that online education has the potential to be transformative in CS education, regardless of whether appropriate business models or methods for verifiable assessment or certification are developed for the most recent wave of MOOCs .

Although the initial set of MOOCs focused on relatively straightforward means for evaluation, such as multiple-choice quizzes or short-answer questions, richer evaluation models measuring student engagement more fully with the material soon emerged. In a computing context, such evaluations include assessing students' programs and assessing larger student projects. While mechanisms such as testing suites can be used to measure aspects of a program's functionality, such tests are not applicable in all contexts, such as with interactive applications. Peer assessments have also been proposed as a means for providing human assessments at scale.

Cooper and Sahami [6] affirm that perhaps the most widely discussed challenge in online education is that of validating original work and preventing (or at least detecting)

plagiarism. It has been reported that plagiarism is a potentially significant problem in online courses.

Another important component of MOOCs is whether and how they provide some form of certification to students. For example, the University of Washington has offered to give college credit for some of its courses taken through Coursera for students who pay a fee and complete additional assessments. Thus, models for the certification of online work certainly exist. The extent to which such certifications are recognized by others, especially employers, will certainly impact how MOOCs are viewed relative to more traditional courses [6].

Mark Guzidal [19] states that the greatest danger of MOOCs is seeing them as panaceas, as a technological form that will work across the spectrum of learners ("from K to gray") for all subjects. No educational technology has yet been successful across such a broad spectrum. The scale of MOOCs can lead us to believe that it is working broadly, but most MOOC offerings have had an 80-90% non-completion rate. Most MOOC teachers have spoken of the impassioned missives they've received, but even at several hundred, it means that we have feedback from only a fraction of a percent of the enrollees. We have little evidence that MOOCs are successful for "K" or "gray."

Fred G. Martin [19] observed that delivering content is the easy part. Without our role in probing students understandings, encouraging them to discuss and share, and mentoring them in applying their nascent knowing new situations, a MOOC will only produce "knowing about" - not "knowing how to do.", he says.

Cooper and Sahami [6] think that the massive scale of MOOCs provides the opportunity to collect unprecedented volumes of data on students' interactions with learning systems. As a result, it becomes possible to use machine learning to gain insight on and potentially personalize human learning. MOOCs also have the potential to present information to students using many different pedagogic approaches, allowing each student to select a particular desired approach, or even making such suggestions to the student. This could positively affect the completition rate of MOOCs.

From the statements cited above we can see that MOOCs offer great opportunities and that online education has the ability to change the traditional education model. If issues and lessons learned are shared among each other in the MOOC community the near future could bring a lot of benefits to the global education sector.

5 Conclusion

In this paper we have understood the need for automated assessment in MOOCs. We have seen multiple approaches for automatic assessment, evaluation and marking for many different types of documents. In this section we will summarize the benefits and problems all this systems add up to.

One obvious benefit is the savings of time and cost for evaluating a huge number of student assignments. Because the system always works the same this results in a more objective marking of student work. As the system is able to process more work in shorter time it allows more students to follow a single course. For the same reason a faster, more precise and more frequent feedback to the student is possible. All this is independent of physical location and allows a broad audience to participate. With the time saved for the marking process, teachers are able to prepare better lecture notes and thus provide a better quality. Another good thing about all the research done in this area is that the results can also be used in other areas, e. g. the interpretation of free text can be used in other areas requiring natural language processing.

Although the systems can process many assignments in short time, this is a problem, too: Often a manual verification for the automated marking is necessary and the work related to this is easily overlooked. Another problem is that currently every kind of document needs a different marking system and each exercise a teacher gives to its students needs additional work to teach the system what a correct solution should look like. Also there are many problems a automated system is not easily able to check, like for example finding its way through a menu of a graphical system or understanding free from texts.

A Figures

List of Figures

B References

References

[1] Norman M. Aaronson. Agsicp: An automated grading system for the instruction of cobol programming. In *Proceedings of the ACM annual conference*, ACM '73, pages 428.1–429, New York, NY, USA, 1973. ACM.

[2] Lewis Baumstark and Edwin Rudolph. Automated online grading for virtual machine-based systems administration courses. In *Proceeding of the 44th ACM technical symposium on Computer science education*, SIGCSE '13, pages 477–482, New York, NY, USA, 2013. ACM.

[3] M. Blumenstein, S. Green, A. Nguyen, and V. Muthukkumarasamy. Game: a generic automated marking environment for programming assessment. In *Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on*, volume 1, pages 212–216 Vol.1, 2004.

[4] Michael Blumenstein, Steven Green, Ann Nguyen, and Vallipuram Muthukkumarasamy. An experimental analysis of game: a generic automated marking environment. In *Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education*, ITiCSE '04, pages 67–71, New York, NY, USA, 2004. ACM.

[5] Farida Bouarab-Dahmani, Malik Si-Mohammed, Catherine Comparot, and Pierre-Jean Charrel. Learners automated evaluation with the odala approach. In *Proceedings of the 2009 ACM symposium on Applied Computing*, SAC '09, pages 98–103, New York, NY, USA, 2009. ACM.

[6] Steve Cooper and Mehran Sahami. Reflections on stanford's moocs. *Commun. ACM*, 56(2):28–30, February 2013.

[7] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 41(6):391–407, 1990.

[8] Christopher Douce, David Livingstone, and James Orwell. Automatic test-based assessment of programming: A review. *J. Educ. Resour. Comput.*, 5(3), September 2005.

[9] Mingyu Feng, Neil T. Heffernan, and Kenneth R. Koedinger. Addressing the testing challenge with a web-based e-assessment system that tutors as it assesses. In Les Carr, David De Roure, Arun Iyengar, Carole A. Goble, and Michael Dahlin, editors, *WWW*, pages 307–316. ACM, 2006.

[10] Marti A. Hearst. The debate on automated essay grading. *Intelligent Systems and Their Applications*, 15(5):22–37, 2000.

[11] Michael T. Helmick. Interface-based programming assignments and automatic grading of java programs. In *Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education*, ITiCSE '07, pages 63–67, New York, NY, USA, 2007. ACM.

[12] Michael J. Hull, Daniel Powell, and Ewan Klein. Infandango: automated grading for student programming. In *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education*, ITiCSE '11, pages 330–330, New York, NY, USA, 2011. ACM.

[13] Mike Joy, Nathan Griffiths, and Russell Boyatt. The boss online submission and assessment system. *J. Educ. Resour. Comput.*, 5(3), September 2005.

[14] Richard Klein, Angelo Kyrilov, and Mayya Tokman. Automated assessment of short free-text responses in computer science using latent semantic analysis. In *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education*, ITiCSE '11, pages 158–162, New York, NY, USA, 2011. ACM.

[15] S. Lewis and P. Davies. Automated peer-assisted assessment of programming skills. In *Information Technology: Research and Education, 2004. ITRE 2004. 2nd International Conference on*, pages 84–86, 2004.

[16] O. Mason and I. Grove-Stephenson. Automated free text marking with paperless school. In *Proceedings of the 6th International Computer Assisted Assessment Conference*, 2002.

[17] Norma C. Ming and Vivienne Ming. Automated predictive assessment from unstructured student writing. In *DATA ANALYTICS 2012 : The First International Conference on Data Analytics*, page 57 to 60. XPS, 2012.

[18] J.T.-S. Quah, Luo-Ren Lim, H. Budi, and Kim-Teng Lua. Towards automated assessment of engineering assignments. In *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*, pages 2588–2595, 2009.

[19] Mehran Sahami, Mark Guzdial, Fred G. Martin, and Nick Parlante. The revolution will be televised: perspectives on massive open online education. In *Proceeding of the 44th ACM technical symposium on Computer science education*, SIGCSE '13, pages 457–458, New York, NY, USA, 2013. ACM.

[20] Avirup Sil, Diane Jass Ketelhut, Angela Shelton, and Alexander Yates. Automatic grading of scientific inquiry. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 22–32, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[21] Hussein Suleman. Automatic marking with sakai. In *Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology*, SAICSIT '08, pages 229–236, New York, NY, USA, 2008. ACM.

[22] Peter Summons, Jo Coldwell, Christine Bruff, and Frans Henskens. Automated assessment and marking of spreadsheet concepts. In *Proceedings of the 2nd Australasian conference on Computer science education*, ACSE '97, pages 178–184, New York, NY, USA, 1996. ACM.

[23] Pete Thomas, Debra Haley, Anne deRoeck, and Marian Petre. e-assessment using latent semantic analysis in the computer science domain: a pilot study. In *Proceedings of the Workshop on eLearning for Computational Linguistics and Computational Linguistics for eLearning*, eLearn '04, pages 38–44, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[24] Salvatore Valenti, Francesca Neri, and Ro Cucchiarelli. An overview of current research on automated essay grading. *Journal of Information Technology Education*, 2:2003, 2003.

[25] R. Williams. Automated Essay Grading: An evaluation of 4 conceptual models. 2001.

[26] Weimin Wu, Guangqiang Li, Yinai Sun, Jing Wang, and Tianwu Lai. Analysec: A framework for assessing students' programs at structural and semantic level. In *Control and Automation, 2007. ICCA 2007. IEEE International Conference on*, pages 742–747, 2007.