Full Paper – Peer reviewed

Analytical assessment rubrics to facilitate semi-automated Essay grading and feedback provision

Andreas Weinberger (andreas.weinberger@alumni.tugraz.at)
IICM, Graz University of Technology,

Heinz Dreher (h.dreher@curtin.edu.au)
School of Information Systems, Curtin University

Mohammad Al-Smadi (msmadi@iicm.tu-graz.ac.at)
IICM, Graz University of Technology

Christian Guetl (Christian.Guetl@iicm.tu-graz.ac.at)
IICM, Graz University of Technology & Curtin University

Assessment is an essential part of the learning process, both in formative learning settings and traditional summative assessment. Both types are challenging, as it can be difficult to ensure consistency, reliability and absence of bias. In formative assessment the problem of workload and timely results is even greater, as the task is carried out more frequently. Information technology is able to assist teachers in these challenges to various degrees depending on the type of test items. The essay test item, besides the well-known application to train language skills and to acquire foreign languages, is widely used to test higher order thinking skills and therefore it can be applied in a great variety of subject domains at different educational levels. Evaluating essays is a time-consuming task hence supporting technologies can deliver great advantages. In this paper we introduce a semi-automated approach to essay grading based on analytical assessment rubrics, the use of which facilitate feedback provision. A prototype system is described in terms of requirements derived from the authors' own experience and the published literature, a workflow model. Reflection of the development experience and user feedback informs further development of the system.

Keywords: automated essay grading, analytical assessment rubrics, formative assessment Go to Program

Conference Themes: ❶ Standards ❸ Practical solutions

Introduction

Educational processes have significantly changed from being repetitive, mechanized learning to more active learning with understanding in which assessment and feedback has to be an integrated part of the learning process (Bransford, Brown & Cocking, 2000). Despite the given importance of assessment and feedback, it is challenging to provide results in a timely manner especially in subjective question items such as essays. The labour intensive nature of assessing essays provides strong motivation for the creation of technology-based systems to support the work of human assessors. Whilst essays are typically used to test higher order thinking skills and knowledge levels, the large student numbers characteristic of many undergraduate university and college courses has increased the assessment workload and thus overall time spent on grading. Additionally, it has decreased the time available for individual feedback (Carter et al., 2003).

As the greater numbers of essay-type assessment artefacts are created and submitted in digital format, the opportunity and demand for electronic assessment gives impetus for a variety of approaches (Bull & McKenna, 2004). However, existing essay analysis systems specific to essays mostly offer little fine-grained flexibility in criteria when automated evaluation features are incorporated. This stems from the fact that the statistical algorithms being employed for the semantic analysis derive their power from large samples, and are therefore restricted to large class size courses. This situation had led us to initiate research on an alternate solution for essay grading in which we hope to be flexible in the assessment criteria (using a rubric based approach) and also to be applicable to low numbers of students at the classroom level. Teachers should be actively supported in evaluating students' submissions to gain time advantages, and to have more possibilities to give feedback to individual students. We have designed a configurable rubric-based system for assessment and feedback provision, with a scalable and configurable set of modules for semi-automatic essay analysis based on flexible criteria specified in

the rubric. Clearly, this system aims to support teachers by decreasing their marking workload thus making more time to dedicate to formative assessment practices.

Background and related work

The most widespread approach to automated essay grading is by employing statistical methods such as Latent Semantic Indexing (Weinberger, 2011) to analyse the semantic content of submitted essays and comparing the derived mathematical model (usually a vector space model) with a previously derived and defined 'standard'. Once such a system is set-up, literally thousands of essays can be graded fully automatically within just a few hours of computer processor time. The set-up time is however such that unless thousands, or at least many hundreds of essays are involved, the cost is not justified. In addition, these methods do not permit fine-grained and flexible assessment criteria setting, for example via a rubric or parameter-set. These factors suggest an obvious new avenue of research to investigate the feasibility of using assessment rubrics in a semi-automated (or fully automated) tool to assess essay-type assignment submissions (Shortis & Burrows 2009).

Rubrics, in the sense we are using the term here, are predefined evaluation schemes either defined specifically for an assignment task or more generally for a class of similar tasks, necessarily neglecting specific content of a single task. Achievement levels must be clear indicated in conceptual terms, rather by single-term descriptions. The expected performance to reach the respective level on the assessment scale can include short examples to clarify the definition. Different features to be evaluated can be either summed up in a holistic rubric or be evaluated separately in an associated analytical rubric. Moskal (2000) has maintained that the subjectivity in essay grading becomes more objective due to the predefined evaluation scheme comprising a rubric. Common criteria in essay writing are grammar, usage (word choice), spelling, style, organisation (structure), discourse, content (knowledge, topics, ideas) and conventions (citation style, usage of figures, etc.) – these, together with the required knowledge domain semantic content can all be specified in our rubric-based approach.

Focusing on the grading of essays with rubrics, a few computer-assisted approaches are available. Writing Roadmap 2.0 is a tool designed for language training using a holistic rubric in which automatic results can be overridden by teachers (Rich & Wang, 2010). The system is limited as it cannot be used outside the language-training context and teachers cannot correct the additional analytical results. Other rubric-based systems such as iRubric (www.rcampus.com/indexrubric.cfm) only provide tools to design rubrics and electronic grade handling but are not specific to essays and therefore offer no supporting essay analysis or other automatic features.

In the related domain of fully automated essay grading a small number of systems have come to prominence - Project Essay Grade, Intelligent Essay Assessor, MarkIT™, IntelliMetric, E-Rater and e-Examiner for example. Some of these, including derived works such as Criterion, are mainly used in the context of language instruction. As the application of essay-style testing in higher education is broader, the content and concepts to be assessed in the submitted text are more varied than those for assessing basic writing skills making solutions that neglect the content unusable. MarkIT™ is an explicitly content centred approach employing semantic analysis but still needs around one hundred human scored essays in the training phase (Williams, 2006). Similarly, approaches based on Latent Semantic Analysis evaluate the content of essays to determine the similarity between documents (Landauer et al., 1998) as applied in the Intelligent Essay Assessor; it also requires 100-200 pre-graded essays (Palmer et al., 2002).

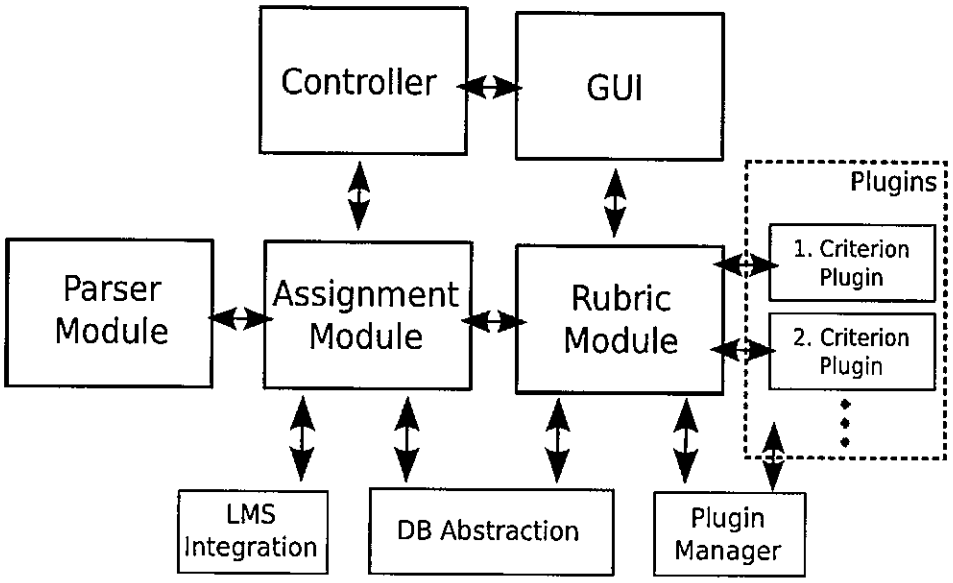
The aim of this project is to utilise a rubric in which the criteria can be flexibly defined by teachers to assess student essay submissions at classroom level. The lower student numbers allow teachers to review each submission and decide about the grade. The system should support the teachers by providing grade suggestions based on an easy to extend set of criterion-modules, and provide analysis features to support teachers in giving feedback to students.

Requirements and architecture

From our past experience and a review of other approaches we have defined the following requirements:

2. Flexible rubric construction and configuration where the number and level of criteria can be selected and specified by teachers. This flexibility should also provide for the use of the system at different educational levels.
13. Analytical assessment inherently contains feedback for students and also matches with the flexible criteria selection.
14. Interactive on-screen assessment and analysis tools to enable teachers to give in depth feedback, especially in formative settings.
15. Analytical assessment is time consuming for humans, raising the need for automated evaluation, or automated support and analytical tools for use by teachers.
16. The assessment/grading workflow should be consistent across multiple grading sessions allowing teachers to easily resume their work.
17. Content related criteria may require that rubrics are specific to certain assignments and to permit the linkage of rubrics to assignments and to classes. As this data is often available in existing systems (e.g. university-wide Learning Management System), along with the student to class relations, integration into these systems is desired resulting in minimal internal storage of administrative data in the system.
18. The system must cater for simultaneous users and should provide a multiple-grader mode where the workload is either split between multiple persons, or multiple persons grade the same essays.
19. Language independence - the system should be designed to be usable for multiple languages, although we will concentrate on English in the first implementation.

The flexible rubric approach is reflected in the architectural design that uses software module plug-in technology for the rubric implementation. Fig. 1 outlines the overall architectural design with the different modules to facilitate prototype integration into existing Learning Management Systems and to provide a service-based solution. Therefore the assignment module is mainly used to represent the rubric-to-assignment and assignment-to-class (and students) relations. As depicted in Fig. 1 the assignment module would utilize a LMS-Integration module to receive the actual student and assignment data and export student grades and feedback. A key aspect of these modules is to provide an identifier for each student that is internally used to link student submissions and grades. All student submissions are internally stored in the original format and as parsed version, suitable for further processing through the parser module. The DB-Abstraction module is used to support different database back-end systems as may be featured at individual user sites. A database approach was chosen for storage as it allows simultaneous run-time clients thus supporting the multiple user requirement. The necessary user account management can be provided by an extension of the LMS-Integration module.



GUI – Graphical User Interface; LMS – Learning Management System; DB – DataBase

Figure 1: High-level Architecture Overview

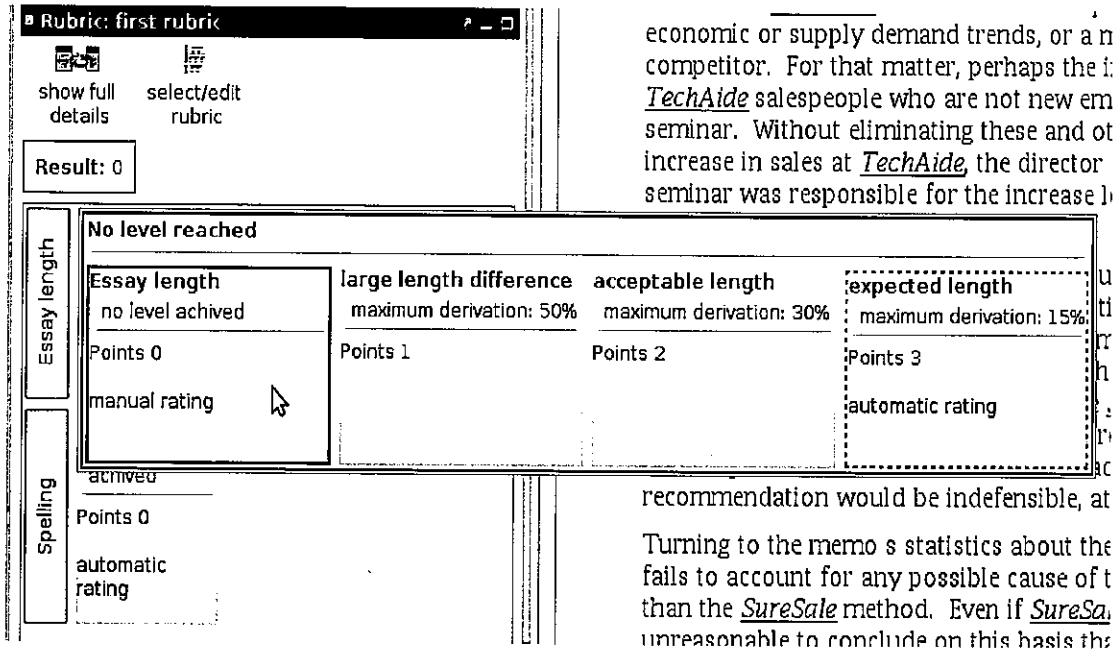
All available criteria are implemented as separate plug-ins managed by the Plug-in Manager module (Fig. 1). The Rubric Module contains a rubric representation constructed from the selected criteria therefore enabling the evaluation of essays through each criterion implementation. Criteria operate independently from each other enabling parallel evaluation. Each criterion should provide essay analysis supporting teachers in giving feedback and may provide an automatic rating as well. Plug-in criteria are therefore split into the basic necessary processing implementation and an optional additional Graphical User Interface implementation. The GUI module is completely independent from the business model or logic model and therefore can be easily replaced with different implementations. As criteria are implemented as plug-ins it is possible to support additional essay languages with different sets of plug-ins or optionally a criterion plug-in can support multiple languages.

Prototype development

Quite obviously, when building a prototype some decisions which may later turn out to be in need of change may be taken. We have been very mindful however of the need to provide flexibility in the implementation in order to facilitate 2nd prototype construction; here follow some design choices we made, with reasons for those choices. The prototype system was implemented as a standalone native Java application, as with Java Web Start through the Java Network Launch Protocol the possibility to deploy and update the application on the Internet is given. Essays are parsed into an internal XML (eXtensible Markup Language) representation optimised for common text processing tasks and document display. In addition, the Parsing Module performs Part-of-Speech tagging, word stemming and possibly baseform derivation (e.g. noun singular form). Part-of-Speech tagging is provided through the log-linear Stanford Part-of-SpeechTagger (Toutanova & Manning, 2000) and stemming is performed with the Porter2 algorithm (Porter, 2002). The prototype uses an SQL database to store the essays and analysed data that allows different or multiple clients to operate simultaneously. An abstraction layer allows the use of different SQL database back-ends and the prototype supports either a hosted MySQL database or an internal local database provided through Apache Derby. This allows for faster testing of the prototype on local machines while preserving the option of adapting the system into an existing infrastructure. The Java Plug-in Framework (Lazarou, 2007) was used as a simplified approach as compared to Open Service Gateway initiative (Hall & Cervantes, 2004) frameworks, to support the implementation of Criterion Plugin modules.

By locating and using existing code as published in software libraries were able to deliver a superior user interface experience than if we developed all our own software, the native SWING Java API (Application Programming Interface) is a good example. A core concept used is to support different views that can be configured by the user as found in many integrated development environments. The Infonode Docking Window software libraries were used to implement this approach allowing users to adapt their workbench layout to personal preferences.

The various Criterion Plugin implementations, for example a spellchecker are based on published, open source software libraries. A full list of the used libraries and licenses can be found in Weinberger (2011, Appendix J). During the development, a unit testing approach was used to verify sub-system operation. The GUI was been tested using interface heuristics during the development cycle (Nielson, 1993, p. 115-155).



economic or supply demand trends, or a n
competitor. For that matter, perhaps the i
TechAide salespeople who are not new em
seminar. Without eliminating these and ot
increase in sales at *TechAide*, the director
seminar was responsible for the increase l
recommendation would be indefensible, at

Turning to the memo s statistics about the
fails to account for any possible cause of t
than the *SureSale* method. Even if *SureSa*
unreasonable to conclude on this basis tha

Figure 2: Criterion Rating

User workflow

The typical workflow of a user in the system, after the assignment has been specified, consists of the following steps:

1. The rubric is defined by selecting and configuring criteria for the evaluation. One or more example essay solutions can be added to the rubric to support the criteria configuration.
2. The rubric is applied to the example solutions to check if automatic results are as expected. [Optional step]
3. After the submission deadline automatic evaluation for submitted essays is triggered. This only occurs if criteria providing automatic evaluation features have been selected for the rubric. In this step, ratings for each criterion capable of automatic evaluation are calculated.
4. Teachers use the provided analysis and evaluation data from the rubric criteria to review each essay, provide further feedback for students through additional comments, and possibly to correct automatic ratings (see Fig. 2) by inserting a manual rating. Automatic ratings are corrected in two ways: either by overriding the result in the rubric based analysis, which is possible for every criterion, or by providing some system retraining input by the user, specific for a criterion, followed of course by a re-run (see next step).
5. Another automatic evaluation run is done as appropriate to re-evaluate essays where the teacher has not overridden automatic results. This step only applies if some criteria received retraining input through the user in step 4. [Conditional step]
6. When all essays have been reviewed by a human expert (the teacher), the student results can be published.

The semi-automatic approach is illustrated in step four, in which teachers override automatic ratings is depicted in Fig. 2. The criteria are displayed in a condensed view only showing the actual reached level. When teachers want to correct an automatic rating or provide a manual rating for criteria without automatic evaluation, a click on the criterion pops-up all the defined levels and the correct level can be quickly selected. Beside the direct manual override of automatic ratings criteria can implement an automatic re-evaluation feature.

As described in the User Workflow section, example, or model, essay solutions can be used in the preparation phase to configure the rubric (step two). How the provided example essays are used depends on the criteria implementations. For example the spell-checker criterion uses these essays to discover domain specific vocabulary not found in the default dictionary. In the criterion configuration teachers can review the list of unknown words and mark correct words before the actual evaluation. This minimizes the rate of false positive matches when the submitted student essays are automatically evaluated making the evaluation outcome or rating

for the criterion more reliable. Fig. 3 shows the detail pane of the spell-checker criterion that offers to mark errors found in student essays as correct. This invalidates all automatic ratings by the spell-checker criterion for the current assignment. The system will trigger an re-evaluation for all students essays with the updated criterion configuration so that teachers have to mark a word as correct only once during the grading process but the results will be updated for all essays. This feature can be provided by any criterion implementation if it allows teachers to supply correction data while reviewing a single student essay.

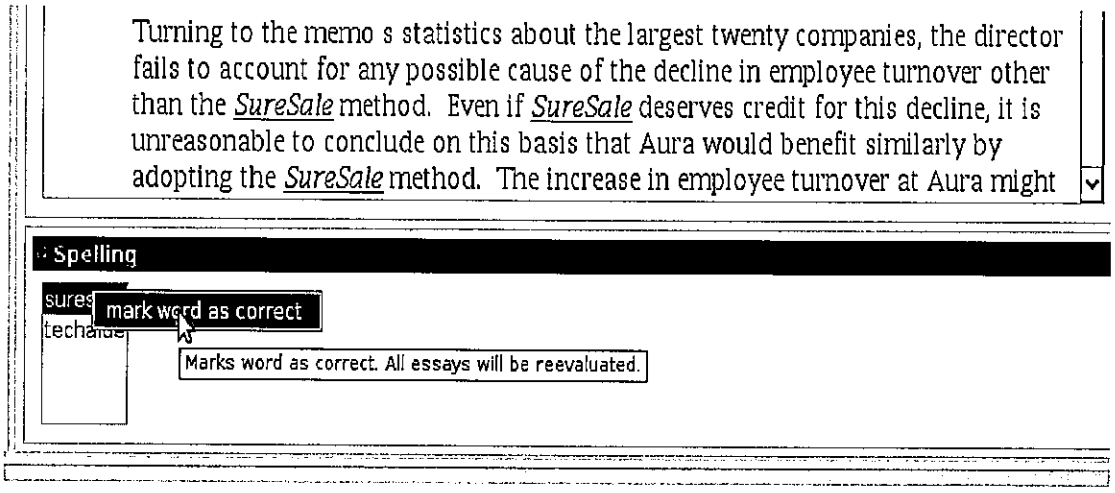


Figure 3: Automatic Re-evaluation Feature

Graphical user interface & usability review

Achieving good user interface can be a challenge especially if there are not enough resources or time available to perform large-scale tests with users. Graphical User Interface heuristics are effectively used during system development. Additionally, the review by an experienced software developer not involved in the project delivered valuable input in adapting the user interface. Finally a test with two different user types was done – the technical specialist on the one hand, and the educator or pedagogue on the other. This partly validated the Graphical User Interface design but also pointed to further improvement possibilities.

The performed think aloud user tests (Lewis & Rieman, 1994) showed that users could confidently use the system after one training session aside from the configuration of criteria for automatic evaluation. As this is specific to each criterion it is much harder to provide a consistent user interface experience, especially where different teacher-developers implement criteria. The different test user background demonstrated how the user knowledge influences the usability of a software system in the sense of how much help and explanation must be provided. Due to their teaching experience teachers naturally understand why criteria need to be weighted, even if they were not familiar with the concept of rubrics, as we have implemented them. On the other hand, software developers were conversant with plug-in based design and therefore were able to use the different criteria more efficiently.

While the plug-in based approach proved to be an efficient solution at the technology level, the Graphical User Interface tests revealed that the approach needs to be hidden from teachers to improve their usability experience. Teachers should only need to concern themselves with which criteria they use for a rubric and how to configure them properly. The concept of using example essay solutions to ease the configuration of criteria was problematic for most users. The usage therefore must be better documented in the program as well as in training material provided to teachers.

Prototype development review

The software reusability feature of modern Integrated Development Environments (IDE) was used extensively during the development cycle as the incorporation of assessment criteria as specified in the rubrics proved to be challenging. As we sat together with the teachers who developed the rubrics and reviewed the prototype, new aspects were discovered that needed to be addressed. Whilst use of existing software libraries was helpful, choosing which libraries were suitable and ensuring their flawless operation can still consume much. The

experience underlined the importance of spending enough time evaluating software libraries properly, which includes deploying them in actual code trials. The requirement that multiple essay languages should be supported can be challenging, as many software libraries are natural language specific.

Where a system is adaptable, and changes to operational parameters are made in the interests of superior performance, the issue of the ability to re-compute earlier evaluations arises. After the results have been published to students any automatic rating must persist in the record. This is easily achieved with the storage of results in the database. Nonetheless, this does not address the problem that it must be possible to understand and explain how a certain automatic rating was calculated. For this reason the configuration of each criterion is stored as part of the rubric configuration for each assignment. Additionally the original plug-in implementation must be still available, which is a challenge when plug-ins are updated. Unsolved at the moment is the problem of updates of plug-in specific datasets as for example used in dictionaries. If these datasets are updated the old version must be still available to be able to recalculate previous automatic ratings. As may be seen, some compromise may have to be made regarding these issues.

Summary

The approach to grade essays with flexible analytical rubrics in a semi-automatic system has been successful based on the trials and testing as reviewed in the previous two sections, and provides an alternative and support mechanism for manual grading or assessment of this style of assignment. To refine and further establish the utility of this approach we need to develop some case studies with willing teachers and students. To achieve this end, we need the co-operation of the Learning Management System administrators and technicians to help implement the LMS Integration module (Fig. 1).

References

Bransford, J.D., Brown, A.L., & Cocking; R.R. (Eds.) (2000) *How People Learn: Brain, Mind, Experience, and School. Expanded Edition*, Washington DC: National Academies Press. {book ISBN 978-0309070362, http://www.nap.edu/openbook.php?isbn=0309070368&page=24}

Bull, J., & McKenna, C. (2004). *Blueprint for computer-assisted assessment.* New York: Routledge Falmer. {ISBN 0415287030}

Carter, J., Ala-Mutka, K., Fuller, U., Dick, M., English, J., Fone, W., et al. (2003) How shall we assess this? In *ITICSE-WGR '03: Working group reports from ITICSE on innovation and technology in computer science education* (p. 107-123). New York, USA: ACM. {conference proceedings, http://doi.acm.org/10.1145/960875.960539}

Hall, R. S., & Cervantes, H. (2004) An OSGi implementation and experience report In *1st IEEE Consumer Communications and Networking Conference: Consumer Networking: Closing the Digital Divide: Proceedings*, 2004 (p. 394-399). NJ, USA: IEEE. {conference proceedings, http://dx.doi.org/10.1109/CCNC.2004.1286894}

Landauer, T. K., Foltz, P. W., & Laham, D. (1998) Introduction to latent semantic analysis. *Discourse Processes, 25* , p. 259-284. Available from http://lsa.colorado .edu/papers/dp1.LSAintro.pdf

Lazarou, J. (2007, June). *Plugin Ready Java Application.* Retrieved 1.7.2011 from http://www.alef1.org/jean/jpf/

Lewis, C., & Rieman, J. (1994) *Task-centred user interface design: A practical introduction.* E-Book published at ftp://ftp.cs.colorado.edu/pub/cs/distribs/clewis/HCI-Design-Book, copy retrieved 12.4.2011 from http://hcibib.org/tcuid/tcuid.pdf

Moskal, B. M. (2000) Scoring rubrics: What, when and how? *Practical Assessment, Research & Evaluation, 7*(4). Available from http://pareonline.net/getvn.asp?v=7&n=3 {online journal article, ISSN 15317714}

Nielson, J. (1993) *Usability engineering.* San Diego, CA: Academic Press.

Palmer, J., Williams, R., & Dreher, H. (2002) Automated essay grading systems applied to a first year university subject: how can we do it better? In *InSITE 2002 Proceedings of the informing science and it education conference*, Cork, Ireland (p. 1221-1229). Santa Rosa, California: Informing Science Institute. Available from http://proceedings.informingscience.org/IS2002Proceedings/ papers/Palme026Autom.pdf {conference proceedings ISBN 0-7803-8145-9 http://dx.doi.org/10.1109/CCNC.2004.1286894}

Porter, M. F. (2002) *The English (porter2) stemming algorithm.* Retrieved 14.6.2011 from http://snowball.tartarus.org/algorithms/english/stemmer.html.

Rich, C., & Wang, Y. (2010) Online formative assessment using automated essay scoring technology in China and U.S. - two case studies. In *2010 2nd International Conference on Education Technology and Computer (ICETC)*, (Vol. 3, p. 524 -528). {conference proceedings, doi 10.1109/ICETC.2010.5529485}

Shortis, M., & Burrows, S. (2009) A review of the status of online, semi- automated marking and feedback systems. In J. Milton, C. Hall, J. Lang, G. Allan, & M. Nomikoudis (Eds.), *ATN Assessment Conference 2009: Assessment in different dimensions.* , (p. 302-313). Melbourne, Australia: Learning and Teaching Unit, RMIT University.
http://emedia.rmit.edu.au/atnassessment09/sites/emedia.rmit.edu.au.atnassessment09/files/ATNA09_Conference_Proceedings.pdf {{conference proceedings, ISBN 9780646524214

Toutanova, K., & Manning, C. D. (2000) Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora*: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics - Volume 13 (p. 63-70). Stroudsburg, PA, USA: Association for Computational Linguistics. {conference proceedings http://dx.doi.org/10.3115/1117794.1117802 }

Weinberger, A. (2011) *Semi-Automatic Essay Assessment based on a flexible Rubric*, (Master's thesis, Graz University of Technology)

Williams, R. (2006). The power of normalised word vectors for automatically grading essays. *Issues in Informing Science and Information Technology*, 3, 721-729 {journal article Available from http://informingscience.org/proceedings/InSITE2006/IISITWill155.pdf}