A Research Prototype for an Automated Essay Scoring Application in .LRN

Fridolin Wild*), Robert Koblischke*), Gustaf Neumann*)

*) *Vienna University of Economics, Institute for Information Systems and New Media, Vienna, Austria*

Abstract. Automated Assessment has a long tradition in technology-enhanced learning, however, in the recent years with a refreshing interest in evaluating free texts with respect to their content, i.e. the semantic of the text. While .LRN does not yet provide an automated essay scoring module, there is a plethora of (commercial) tools available in the market, many of them based on latent semantic analysis. With the essay scoring application (ESA), we present a research prototype for automatically evaluating the content of unstructured texts which invites interested practitioners and scientists to experiment with it.

1. Automated Essay Scoring

Computer-assisted assessment has a long tradition in education. While early experiments on grading free text responses – run during the Project Essay Grade (PEG) by Page [1] already in 1966 – were carried out on punched cards and had predominantly been syntactical in nature, research today focuses on emulating humans' semantic understanding, backed up by hitherto unknown computing power. Semantic understanding refers to judging the meaning of written essays against them being an exhaustive and satisfying response to an issued question.

Free text assessment, however, is an inherently subjective process when carried out manually. Typically, facilitators read, analyse, and interpret the latent structure in the essays to be scored, with an ever-increasing rate of errors the more essays they have to process and with subjective differences grounding in varying interpretations. Choosing representative and trained correctors (which usually is not the case in higher education) can circumvent this only to a limited amount. Empirical evidence for this can be found in the usually rather low inter-rater correlation of two human assessors which typically floats around .6 to .8 (see e.g. [2]). Backing up the human assessment process with automated essay scoring meachanisms is an option to increase both effectiveness and efficiency in the assessment process.

Today a plethora of commercial tools for meaning oriented computer-assisted assessment is available: PEG, e-Rater, Criterion, IntelliMetric, Intelligent Essay Assessor (IEA), Assistant for Preparing Exams (apex), Summary Street, AutoTutor, and many others. Many of them are based on a method called 'latent semantic analysis' (LSA), originally invented by Landauer et al. [3] in the late eighties of the 20th century which is still up to today developed to maturity by a worldwide research community. Landauer and colleagues were able to show that by combining the vector model of a document-term-space with singular value decomposition (SVD, a two mode factor analysis), grade ranges similar to those awarded by human graders can be produced. An open source implementation for the statistical language and environment R has been released by Wild in 2005 [4]. More background information on the working prin-

ciple of LSA can be found in [5] and [6]. With this contribution we intend to provide a first prototype for a .LRN essay scoring application based on open-ACS.

2. Prototype Essay Scoring Application

In a typical LSA process, first a document-term matrix has to be constructed from a given text base of *n* documents containing *m* terms. We call this document-term matrix 'textmatrix', as its *m* rows represent the terms and the *n* columns the documents thus denoting the frequency how often a certain term appears in a document in its cells. The underlying text base, also called corpus, usually consists out of generic background documents, e.g., a collection of newspaper articles, and those domain-specific documents that – together and redundantly – contain the information necessary to answer a particular question. Compiling this corpus of generic background documents and domain-specific documents is often a work-intensive iterative process until satisfying results are yielded.

When constructing the textmatrix from the corpus, several pre-processing measures can be taken: highly frequent (thus judged less important) stopwords such as 'he', 'is', and 'as' can be eliminated, terms can be reduced to their word-stems so that 'goes' would become 'go', a controlled vocabulary can be used, term weighting schemes such as inverse document frequencies (IDF) can be applied, etc. The textmatrix created while applying certain preprocessing methods then is resolved with a singular value decomposition into the partial matrices, one of them being a diagonal matrix containing the singular values in descending order. By truncating this diagonal matrix to a small number of singular values, an annealed textmatrix can be reconstructed which shows a desired loss of information compared to the (same-format) input textmatrix. By using the left-sided matrix and the diagonal matrix, additional documents can be folded into this so called latent-semantic space, which is a lot less computationally expensive compared to calculating the singular value decomposition.

Within this new, lossy textmatrix, distance measures between terms and documents (already existing and the ones folded-in ex post) can be applied to evaluate the content of an essay. One simple scoring rule, for example, is to assign the score to an essay directly proportional to the cosine correlation between the vector representing the essay and the vector representing a best-practice solution (the 'gold-standard' scoring rule). Many other rules have been developed; however, further innovation is seeked.

In the prototype of the essay scoring application, a set of components encapsulates the depicted complex process: the open-ACS module, the essay scoring application, is responsible for the graphical user interface which is separated into a 'feedback' user interface to be exposed to the learner and into a 'administration' user interface to be operated by a facilitator.

The administration part offers facilities to manage questions, text corpora to be used to compose a latent-semantic space, and add-on meta-data such as classification information or scores that had been assigned to example essays by manually evaluating them. Administrators can upload (file by file or packaged in a zip-file) and manage these corpora, create new questions, decide upon the latent-semantic space in which the evaluation will be performed, and create and assign scoring rules which will perform the evaluation. A scheduler relies on cron jobs to shift the computationally costly calculation of a latent-semantic space to a separate thread to be carried out in suspected idle times of the system. The resulting space is stored in the database again. Thereby, the bridge PLR allows to write stored procedures for postgreSQL in R. For the calculation of the latent-semantic spaces, the R routines of the *lsa* package are re-used, see [4].

The feedback part offers guidance routines to select the assignment for which an essay has to be written and subsequently forms to enter the essay text and evaluate it. Therefore, the essay is folded into an existing latent-semantic space: first, tsearch-2 is used to convert the essay into the required vector format. Then, stored procedures written in R are used to fold this vector into the existing space. Third, the scoring rule (written in R) assigned to the question is read from the database and is parsed and evaluated. Thereby, a set of variables is provided via the interface to allow the scoring rule to access the data required (e.g., the reconstructed textmatrix or the essay vector). The scoring rule returns html code formatting the desired response message that is to be displayed to the user.

3. Outlook & Future Work

In the current form, the essay scoring application is merely a research prototype which is used for experimentation. One of the problems with using LSA based scoring algorithms is that according to the state of the art it is difficult to set up corpora that work reliably well and that the calibration process to adjust the influencing parameters is currently still not as simple as it needs to be to have the administration operated by regular facilitators without a background in latent semantic analysis. From the experiments in our group we have seen that there is a lot of room for improvement and innovation regarding the scoring rules. With the scoring application we provide an easy to use interface that allows for convenient experiment with new scoring rules. Additionally, we see the need of packaging or sharing services that allow to transfer tuned latent-semantic spaces and their associated assignments from one system to another to allow for easy sharing and re-use.

References

[1] Page, E. (1966). The imminence of grading essays by computer. Phi Delta Kappan, 47, 238-243.

[2] Landauer, T., Psotka, J. (2000): Simulating Text Understanding for Educational Applications with Latent Semantic Analysis: Introduction to LSA, In: Interactive Learning Environments, vol. 8, no. 2, pp. 73-86(14)

[3] Deerwester, S., Dumais, S., Furnas, G., Landauer, T., Harshman, R. (1990): Indexing by Latent Semantic Analysis, In: Journal of the American Society for Information Science, 41(6), pp. 391-407

[4] Wild, F. (2006): lsa: Latent Semantic Analysis, R package version 0.58.

[5] Wild, F. and Stahl, C. (2006): Investigating Unstructured Texts with Latent Semantic Analysis, In: (Eds.): …., Springer, Berlin

[6] Berry, M., Dumais, S., O'Brien, G. (1995): Using Linear Algebra for Intelligent Information Retrieval, In: SIAM Review, Vol. 37(4), pp. 573-595