Free-Text Assessment in a Virtual Campus

Philippe DESSUS, Benoît LEMAIRE, & Angélique VERNIER

Proc. CAPS'2000. Paris : Europia, 13-14 déc.

Laboratoire des Sciences de l'Éducation
Bât. SHM, 1251 av. Centrale
BP 47, Université Pierre-Mendès-France
38040 Grenoble Cedex 9, France
Ph. (+33) 4 76 82 57 09 or (+33) 4 76 82 58 78 Fax. (+33) 4 76 82 78 11
{Philippe.Dessus; Benoit.Lemaire}@upmf-grenoble.fr
avernier_fr@yahoo.fr

ABSTRACT: *Current web-based learning environments (such as* WebCT *or* LearningSpace*) are better at presenting the content than assessing the learning. Indeed, these environments provide sophisticated access to learning content by means of images, videos, sounds, hypertexts, glossaries, etc. However, their main weakness lies in the assessment part. Either the tests are based on rigid technologies such as quiz or multiple-choice questions (MCQ), or a teacher is required to manage forums or asynchronous e-mail exchanges. In this article we present* Apex*, a module of a web-based learning environment which is able to assess student knowledge based on the content of free texts. The goal of this module is not just to grade student productions but rather to engage students in an iterative process of writing/assessing at a distance. In addition to the preparation of their courses, teachers are required to partition them, to design exam questions, and to select the parts of the course that cover these questions. All courses, questions and student essays are managed by a database which is part of* Apex*. The assessment relies on Latent Semantic Analysis, a tool which is used to represent the meaning of words as vectors in a high-dimensional space. By comparing an essay and the text of a given course on a semantic basis, our system can measure how well the essay matches the course. Various assessments are presented to the student regarding the content, the outline and the coherence of the essay. These assessments provide a more authentic feedback than those*

currently provided by MCQ in virtual campuses.

KEY-WORDS*: Computer-Assisted Assessment, Automatic Essay Grading, Latent Semantic Analysis, Virtual Campus*

RÉSUMÉ *: Les environnements d'apprentissage à distance tels que* WebCT *ou* LearningSpace *sont plutôt destinés à communiquer un contenu qu'à évaluer l'apprentissage. Leurs fonctionnalités de présentation d'un contenu par image, vidéos, sons, hypertextes, glossaires, etc. sont bien plus riches que leurs fonctionnalités d'évaluation : les tests qu'ils proposent sont en général basés sur des QCM, ou bien ils nécessitent de l'enseignant un important travail d'évaluation en ligne, par forum ou courrier électronique interposés. Nous présentons ici* Apex*, un logiciel pouvant s'intégrer dans un environnement d'enseignement à distance et permettant l'évaluation du contenu de copies d'étudiants. Le but d'*Apex *n'est d'ailleurs pas seulement de noter ces copies, mais surtout de mettre à la disposition des étudiants un environnement interactif où ils peuvent écrire, soumettre leur copie à évaluation, puis réviser leur texte selon les indications du logiciel. Le travail de l'enseignant est ici de proposer un cours en ligne, de le hiérarchiser, et de proposer éventuellement des questions d'examen types. La procédure d'évaluation utilise la méthode* Latent Semantic Analysis *(Analyse de la sémantique latente), un outil permettant de représenter la sémantique des mots à l'aide de vecteurs dans un espace de très grandes dimensions.* Apex *mesure la façon dont une copie rend compte du cours en comparant leurs vecteurs respectifs. Trois types d'évaluation sont proposés à l'étudiant : à propos du contenu, du plan et de la cohérence interparagraphes. Ces évaluations sont plus pertinentes que les corrections des questionnaires à choix multiples actuellement présents dans les environnements d'apprentissage à distance.*

MOTS-CLÉS *: Évaluation assistée par ordinateur, Correction automatique de copies, Analyse de la sémantique latente, Campus virtuel*

1. Introduction

Current web-based learning environments (such as *WebCT* or *LearningSpace*) are better at presenting the content than assessing the learning. Indeed, these environments provide sophisticated access to learning content by means of images, videos, sounds, hypertexts, glossaries, etc. However, their main weakness lies in the assessment part. Either the tests are based on rigid technologies such as quiz or multiple-choice questions, or a teacher is required to manage forums or asynchronous e-mail exchanges. Vigilante [VIG 99] measured the amount of time teachers spend to manage typical on-line courses with intensive discussion/collaboration areas. He showed that 25 % of on-line time is spent in grading written assignments. This time-consuming task could be reduced by the development of automated grading systems. These systems are mainly based on the following three techniques:

1. multiple-choice exams;
2. free-text assessment based on surface features;
3. free-text assessment based on course content.

1. The first one just requires appropriate tools to design and develop multiple-choice questions [BLA et al 98; STO 99; VAN 98]. However, the teacher has to work hard to write out appropriate items which should be close enough to the right answer but still wrong.

2. Concerning the second point, the earliest attempt to grade free texts was done by Page [PAG 66, cited by CO 97; WRE 93] with *PEG* (*Project Essay Grade*). Page selected a set of surface features of a text, called "proxes": i.e. length of essay in words, number of commas, number of connectives, average word length, etc. Correlations between already graded essays and these surface measures were computed. Some of them were: average word length (r = 0.51), number of commas (r = 0.34), length of essay (r = 0.32). A global score was then computed by weighting each of these features by means of a regression. Further experiments by Page [PAG 94] showed high correlations (around 0.8) between computer and human scores. In a similar way, Burstein and his colleagues [BUR et al 98; WH 99] are developing *e-rater*. This software relies on around sixty surface text features—close to Page's proxes—to compute essay grades. These features are based on a guide for scoring essays. Each of these features relies on a general characteristic of the text: syntactic, rhetorical or topical content. The syntactic parsing of sentences is processed by a third-party software, then a second program identifies the rhetorical structure of the essay. It relies on words or sentences that contain rhetorical arguments like: "in conclusion…", "in summary…", "perhaps", etc. Third, a content score is computed: the word pattern of the essay is first translated into a representative vector, which is then compared with the vectors of manually graded essays. Correlations around 0.8 between *e-rater* and two human raters following the scoring procedure were found. These

results are impressive and the scoring method straightforward. However, this approach does not take into account the semantic content of the essays: an irrelevant essay could be given a high grade.

3. This drawback is precisely addressed by the third technique. The teacher has only to provide the text of the course, which is then automatically processed by a computer. Grades are computed by comparing the content of the course with the student free texts. The core of most of these softwares is LSA (Latent Semantic Analysis), a technique originally developed for Information Retrieval purposes, although concurrent methods are also developed [LAR 98; MW 98].

The paper is organized as follows: first, we will present LSA, second we will review the various intelligent assessors or tutors that are based on LSA, then we will describe our tool, *Apex*.

2. Description of LSA

In order to perform a semantic matching between the student essay and the course, LSA relies on large corpus of texts to build a semantic high-dimensional space containing all words and texts, by means of a factor analysis [DEE 90; LAN et al 98]. Basically, the semantics of a word is determined from all of the contexts (namely paragraphs) in which that word occurs. For instance, the word *bike* occurs generally in the context of *handlebars*, *pedal*, *ride*, etc. Therefore, if a word like *bicycle* occurs in a similar context, the two words will be considered close to each other from a semantic point of view. Their corresponding vectors in the semantic space will be also close to each other.

This semantic space is built by considering the number of occurrences of each word in each piece of text (basically paragraphs). For instance, with 300 paragraphs and a total of 2,000 words, we get a 300 x 2,000 matrix. Each word is then represented by a 300-dimensional vector and each paragraph by a 2,000-dimensional vector. So far, it is just straightforward occurrence processing. The power of LSA, however, lies in the reduction of these dimensions, and in so doing induces semantic similarities between words or paragraphs. All vectors are reduced by a method close to eigenvector decomposition to, for instance, 100 dimensions. The matrix X is decomposed as a unique product of three matrices: $X = T_0 S_0 D'_0$ such that T_0 and D_0 have orthonormal columns and S_0 is diagonal. This method is called singular value decomposition. Then only the 100 columns of T_0 and D_0 corresponding to the 100 largest values of S_0 are kept, to obtain T, S, and D. The reduced matrix $\bar{X}$ such that: $\bar{X} = TSD'$ permits all words and paragraphs to be represented as 100-dimensional vectors. It is this reduction which is the heart of the method because it allows the representation of the meaning of words, by means of the context in which they occur. If the number of dimensions is too small, too much information is lost. If it is too big, not

enough dependencies are drawn between vectors. Dumais [DUM 91] empirically showed that a size of 100 to 300 gives the best results in the domain of language.

This method is quite robust: a word could be considered semantically close to another one although they do not co-occur in texts. In the same way, two documents could be considered similar although they share no words. An interesting feature of this method is that the semantic information is derived only from the co-occurrence of words in a large corpus of texts. There is no need to code semantic knowledge by means of a semantic network or logic formulas.

3. LSA-based free-text assessors

In a seminal study [FOL 96], subjects were asked to write essays from a number of texts. These essays were then ranked by human graders. Their task was to assess the adequacy between the essay and the texts. In parallel, LSA was "trained" with the texts and the essays were ranked according to the semantic proximity between each of them and the texts. LSA results compare favorably again with the human results. In this vein, several systems were designed by various researchers.

3.1. Intelligent Essay Assessor (IEA)

Intelligent Essay Assessor [FOL et al 99] is first "trained" on several texts related to the domain. The student essay is then compared with pre-graded essays by means of two kinds of scores:

— the *holistic score*, which returns the score of the closest pre-graded essay;
— the *gold standard*, which returns the LSA proximity between the student essay and a standard essay.

An experiment using the holistic score was performed on 188 essays on biology. A correlation of 0.80 was shown between *IEA* grades and human graders. One main interest of *IEA* is that the student can submit the essay again and again in order to improve the score. Foltz et al. note that the average grade increases from 85/100 (first submission) to 92/100. However, one problem is that the teacher needs to grade standard essays beforehand.

3.2. Summary Street

Summary Street and *State of Essence* [KIN et al. to appear; SD to appear] are also built on top of LSA. Both systems help students to write good summaries. First of all, the student is provided with general advice on how to write a summary: select the most important information, find two or three ideas,

substitute a general term for lists of items, not include trivial information, etc. Then the student selects a topic, reads the text and writes out a summary. LSA procedures are then applied to give a holistic grade to the summary. Prompts about the length of the essay or the spelling are also delivered.

3.3. Select-a-Kibitzer

Wiemer-Hastings and Graesser [WG to appear] have recently built *Select-a-Kibitzer*, an agent-based computer tool that assesses student compositions. Each agent is responsible for providing a kind of advice: coherence, purpose, topic, and overall quality. It is worth noting that these assessments are quite similar to those provided by the previous applications. However, this software emphasizes the negotiated construction of the text by associating each agent to a character.

Now we will present *Apex*. It differs from *IEA* in that it does not rely on pre-graded essays but rather on various semantic comparisons with the course. It also differs from *Summary Street* and *Select-a-Kibitzer* since it takes into account the structure of the course to grade an essay.

4. Apex

Apex is a web-based learning environment which manages student productions, assessments, and courses. Once connected to the system, the student selects a topic (a part of a course or a question) that he or she wishes to work on. The student then types a text about this topic into a text editor. At any time, he or she can get a three-part evaluation of the essay. After reading any of the three assessments, the student goes back to the essay, modifies the text and submits it again. Once connected to the system, the teacher can either add a course, view student assessments, or create exam questions.

All courses, student texts and exam questions are represented in a database, which is managed by an administrator. Figure 1 presents the architecture of *Apex*. The system runs under Linux and uses three tools: *Apache server* 1.3.12, *PHP* 4.0.0 and *MySQL* 3.22.32. In order to provide the user with dynamic pages, *Apache* runs *PHP* scripts. *PHP* is a server-side HTML-embedded scripting language. In particular, these scripts permit the communication with *Apex assessor*, which in turn runs *LSA* routines. The third tool is a database which allows us to better control the application. *MySQL* is necessary for designing an application managing several students and teachers.

Insert figure 1 about here

Figure 1. Architecture of Apex

Table 1 displays the allocation of responsibilities between teacher, students and *Apex*, after the Crossley and Green [CG 90] framework.

Table 1. Apex *allocation of responsibilities table.*

TEACHER	STUDENT	ADMINISTRATOR	*APEX*
Connect to the system	Connect to the system	Connect to the system	Verify logins and passwords
Maintain *Apex* database Course level Add course Modify course **Mark up** Delete course Exam level Add question and related parts of course Delete question	Select course Select topic Select question	Maintain *Apex* database Teacher level Add teacher Delete teacher Student level Add student Delete student	Allow database accesses about courses and students
	Create or revise essay Delete essay		Provide text processing window
	Read course		Display course
View student activity from database Student essay Student assessments	Get assessment Content-based Outline-based Coherence-based		**Compute assessments** **Content-based** **Outline-based** **Coherence-based** Write assessments into database
Adjust assessment parameters			Save parameters

Now we will detail the two tasks in bold face type: marking up the course by the teacher and providing assessments by *Apex*.

4.1. Marking up the course

The course has to be marked up by the teacher to give it a two-level structure. This task is much less time-consuming than preparing a MCQ test. The text has to be divided into topics, and each topic divided into notions. Basically, the structure of the file corresponds to an outline view of a word processor, so it is quite straightforward to mark up such a document. First level titles should begin with `#T` (for Topic) whereas second level titles should begin with `#N` (for Notion). Only second level titles are followed by a paragraph. A notion can belong to several topics. However, to avoid redundancies, the text of such a notion is written only once. To specify such a cross-reference, the notion title should begin with `#S`. Using the same mechanism, the teacher could specify questions by indicating the relevant parts of the course. Figure 2 gives an example of such a marking-up: the notion title `Introduction to the solar system` is defined in the topic `The solar system` and referred to in the topic `Eclipses`. The response to the question `Describe the different types of`

eclipses? needs to cover two notions of the course: Sun eclipses and Moon eclipses.

```
###################### Text of the course ################

#T The solar system
   #N Introduction to the solar system
      The solar system is composed of....
      ................
   #N The sun
      The sun is a star which is....
      ................
   #N The planets
      There are 9 planets in the solar system: ....
      ................

#T Eclipses
   #S Introduction to the solar system
   #N Sun eclipses
      When the moon is between the earth and the sun,....
      ..................
   #N Moon eclipses
      When the earth is between the sun and the moon,....
      ..................

###################### Additional questions ##############

#T Describe the different types of eclipses.
   #S Sun eclipses
   #S Moon eclipses
```

Figure 2. Marked-up text of a course to be processed by Apex

4.2. Providing assessments by *Apex assessor*

In this section, we will detail how *Apex* can use the previous structure to grade a student essay; however it can also provide detailed assessments on the content, the outline and the coherence of a student essay. The only information *Apex* needs to process a student essay is the text of the course. It is worth noting that we added a very large set of French texts (290,000 words) to the course, in order to improve the semantic knowledge of the system. These texts (three French novels) were not related to the course but they were useful for the system to better "understand" the non domain-dependent terms of the student essay.

To date, only the content-based assessment has been included into the web-based version of *Apex*. The two others run only in the text-based version, and are not yet implemented on the web-based version.

4.2.1. Content-based assessment

At the content level, the system identifies how well the notions are covered by requesting LSA to measure a semantic similarity (in the range [–1, 1]) between

the student text and each notion of the selected topic. For instance, if the student selects the topic `The solar system`, he or she has to write an essay about that topic. There are three notions in that topic; therefore the student text will be successively compared with each of them. If the similarity is high, it means that the student has covered the corresponding notion well. Actually, the system provides a message to the student according to the value of the similarity. The current version of *Apex* relies on four categories:

1. if the similarity $\in$ [–1; 0.1], the message is "the notion was covered very poorly";
2. if the similarity $\in$]0.1; 0.5] the message is "the notion was covered poorly";
3. if the similarity $\in$]0.5; 0.7] the message is "the notion was covered well";
4. if the similarity $\in$]0.7; 1] the message is "the notion was covered very well".

However, the number of categories as well as the corresponding values and the text of the message can be fully parametrized depending on the content of the course. To do so, the teacher has to edit and modify a text file.

In addition to this notion by notion assessment, *Apex* provides a general grade of the student text. This grade is the average grade for each notion, multiplied by 20 in order to get a grade between 0 and 20 as is usual in the French school system. Figure 2 shows an example of a content-based evaluation of a student essay in the domain of educational sociology.

Insert Figure 2 about here

Figure 2. Example of a content-based assessment

One problem we had to tackle concerns very short student texts. When the student has written out just a few words, the content-based assessment could yield a high score, although one might want the assessment to be considered low. Therefore, if the number of words of the student text is below 300 words (that value can be easily changed in the parameter file), all the content thresholds described earlier are arbitrarily raised so that the assessment is more severe: the shorter is the text, the closer to 1 becomes the thresholds. Basically, with N being the number of words of the essay, all content thresholds T are changed to:

$$T + (1 - T)\left(\frac{300 - N}{300}\right) \qquad [1]$$

For instance, a 100-word text would lead to the thresholds 0.7; 0.83; 0.9 instead of 0.1; 0.5; 0.7.

4.2.2. Outline-based assessment

At the outline level, the system displays the most similar notion of the course for each paragraph of the essay. The goal is to provide the student with an outline view of the essay. If the similarity computed by LSA is too low, the system prints that there is no predicted notion. This threshold is currently set to 0.4 but this can be changed easily by means of the parameter file we mentioned earlier. Figure 3 shows an example of an outline-based assessment in the domain of cognitive ergonomics. For instance, the first paragraph of the student text has been recognized by *Apex* as being concerned with the definition of activity. If this is not what the student intended to do, he or she should probably rework this paragraph.

```
Paragraph 1: Before describing Rasmussen's model, it is neces...
Predicted notion (0.72): The definition of activity
Paragraph 2: At level n, Rasmussen considers the behavior as ...
Predicted notion (0.77): Applications of Rasmussen's model
Paragraph 3: The hierarchy is justified by the necessity to t...
No predicted notion.
Paragraph 4: Suppose we select a specific application of the ...
Predicted notion (0.91): Applications of Rasmussen's model
Paragraph 5: Reason has established a list of possible errors...
Predicted notion (0.60): Rasmussen's model
...
```

Figure 3. Example of an outline-based assessment. Text-only version, not yet included into the web-based version

4.2.3. Coherence-based assessment

LSA has been used already to measure text coherence and has proven to be successful [FOL et al. 98]. At the coherence level, the system relies on LSA to measure semantic proximities between adjacent sentences. Therefore *Apex* can detect coherence breaks. Then it gives an average measure of coherence and if necessary an example of an important conceptual break between two sentences. Figure 4 shows an example of a coherence-based assessment: the student is required to work on the text again, and in particular to correct the linking of ideas between sentence 2 and sentence 3.

```
coherence sentence 1 - sentence 2: 0.67
coherence sentence 2 - sentence 3: 0.16
coherence sentence 3 - sentence 4: 0.58
coherence sentence 4 - sentence 5: 0.79
coherence sentence 5 - sentence 6: 0.52
...
Poor inter-sentence coherence (0.49)
For instance, there is a big conceptual break between the
following sentences:
2: The activity is a set of unobservable behaviors that...
3: There are generally two kinds of procedures in the ...
```

Figure 4. Example of a coherence-based assessment. Text-only version, not yet included into the web-based version

4.3. Testing *Apex* correlations with human grades

We performed an experiment based on a graduate course on sociology of education. We took 31 essays that were written a year ago by students at the university of Grenoble. We typed them out and ran *Apex* in order to get the general grade between 0 and 20 for each student essay. We first compared these grades with the grades the teacher had given a year ago. We got a good significant correlation ($r = 0.59$, $p < 0.001$). Then all essays were ranked by two judges who were teachers in similar domains. They had studied the relevant part of the course several times before the task. They graded each of the 31 texts in the same way that *Apex* does: each text was given 14 grades corresponding to the estimated adequacy between each of the 14 notions. Then an average grade was computed. We got good significant correlations with *Apex* grades ($r_1 = 0.59$, $p < 0.001$; $r_2 = 0.68$, $p < 0.0001$).

All of these results are in the range of correlations found in the literature. Wiemer-Hastings [WIE 99] has also compared LSA grades and human grades and none of the correlations were above 0.5. Foltz's [FOL 96] similar correlations were in the range [0.3; 0.55]. Wolfe et al. [WOL et al 98] indicated correlations around [0.6; 0.7]. E. Kintsch et al. [KIN et al to appear] mentioned a value of 0.64. The highest correlations ever found [0.8; 0.86] were obtained by Foltz and colleagues [FOL et al 99]. These repeated experiments show that LSA is a adequate tool for assessing knowledge, one that rivals the ability of humans.

The main criticism of all these approaches is that the student could fool the machine by writing an essay with only keywords. However, we claim that if a student can provide the right keywords, then this student should have a good knowledge of the domain and that is exactly what we want to measure. Another criticism often encountered is that spelling and syntax are not taken into account. A solution would be to supplement these systems with adequate third-party softwares.

4.4. Integrating *Apex* into a virtual campus

Most of the softwares reviewed in this paper are production-centered rather than learning-centered. The authors of these softwares are more interested in providing adequate prompts to the users than in providing rich environments for active learning [GRA 96]. In an experiment we tested a non web-based version of *Apex* in a real-life context [LD to appear]. Three groups of students were asked to write out an essay in order to review the main features of a sociology of education course. A group worked with an automated version of *Apex*—the assessment prompts were delivered online—; another group used an *Apex*-demand version—the assessments were delivered on student requests—; the third group was a control-group using a classical text editor. We plan to replicate this experiment using the web-based version of *Apex*. The context will be a virtual

campus environment, in which students would not be only receivers of assessments, but active planers and producers. The holistic format of the assessments provided by *Apex* allows a more authentic assessment than MCQ marking. Although the student involved in a distance learning program has to write text (e.g. in forums, e-mails, discussion lists), these productions are seldom assessed by the teacher. Furthermore, the emergence of self-assessment, for instance by writing portfolio, is becoming an interesting means to promote student learning.

Acknowledgements

This research is partially supported by a grant from the University Pierre-Mendès-France of Grenoble. We would like to thank Susan Dumais and the Bellcore Labs to have allowed us to use and hack the code of the basic LSA programs. We are also grateful to Pascal Bressoux who provided us with the text from one of his courses and who helped us in formatting this course for our system, and to Dora Gaspar and Arnaud Serret who assisted the third author of this paper in developing the web-based version of *Apex*.

References

[BLA et al 98] Blank, D., Holmes, G., Wells, R. & Wolinski, P. (1998). Interactive Gradebook: the Missing (Hyper)Link, *ACM SIGCSE*.

[BUR et al 98] Burstein, J., Kukich, K., Wolff, S., Lu, C., & Chodorow, M. (1998). Computer Analysis of Essays. *NCME Symposium on Automated Scoring*.

[CG 90] Crossley, K., Green, L. (1990). *Le design des didacticiels [Designing Educational Software]*. A.C.L./O.T.E., Paris.

[CO 97] Chung, G. & O'Neil, G. (1997). *Methodological Approaches to online scoring of essays*, Los Angeles, Center for the Study of Evaluation, Technical Report # 461, CRESST.

[DEE et al 90] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshmann, R. (1990). Indexing by Latent Semantic Analysis, *Journal of the American Society for Information Science, 41*, pp. 391-407.

[DUM 91] Dumais, S. T. (1991). Improving the retrieval of information from external sources, *Behavior Research Methods, Instruments, & Computers*, *23*:2, pp. 229-236.

[FOL 96] Foltz, P. W. (1996). Latent Semantic Analysis for text-based research, *Behavior Research Methods, Instruments, & Computers, 28*:2, pp. 197-202.

[FOL et al 98] Foltz, P. W., Kintsch, W., & Landauer, T. K. (1998). The Measurement of Textual Coherence with Latent Semantic Analysis, *Discourse Processes, 25*:2-3, pp. 285-307.

[FOL et al 99] Foltz, P. W., Laham, D. & Landauer, T. K. (1999). Automated Essay Scoring: applications to Educational Technology, *Proceedings of the ED-MEDIA'99 Conference*, AACE, Charlottesville.

[GRA 96] Grabinger, R. S. (1996). Rich environments for active learning. *In* D. H. Jonassen (Ed.), *Handbook of Research for Educational Telecommunications and Technology*, MacMillan, New York, p. 665-692.

[KIN et al to appear] Kintsch, E., Steinhart, D., Stahl, G., & the LSA Research Group (to appear). Developing Summarization Skills through the Use of LSA-based Feedback, *Interactive Learning Environments*.

[LAN et al 98] Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An Introduction to Latent Semantic Analysis, *Discourse Processes, 25*:2-3, pp. 259-284.

[LAR 98] Larkey, L. S. (1998). Automatic Essay Grading Using Text Categorization Techniques. *Proc. SIGIR'98*. Melbourne.

[LD 97] Landauer, T. K., & Dumais, S. T. (1997). A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction and Representation of Knowledge, *Psychological Review, 104*:2, pp. 211-240.

[LD to appear] Lemaire, B., & Dessus, P. (to appear). A System to Assess the Semantic Content of Student Essays, *Journal of Educational Computing Research*.

[MW 98] McKnight, K. S., & Walberg, H. J. (1998). Neural Network analysis of Student Essays. *Journal of Research and Development in Education, 32*:1, 26-31.

[PAG 66] Page, E. (1966). The imminence of grading essays by computer, *Phi Delta Kappan, 47*, pp. 238-243.

[PAG 94] Page, E. B. (1994). New computer grading of student prose. *Journal of Experimental Education, 62*:2, 127-142.

[STA et al to appear] Stahl, G., dePaula, R., & the LSA Research Group (to appear). Evolution of an LSA-Based Interactive Environment for Learning to Write Summaries, *Interactive Learning Environments*.

[STO 99] Stockburger, D. W. (1999). Automated Grading of Homework Assignments and Tests in Introductory and Intermediate Statistics Courses Using Active Server Pages, *Behavior Research Methods, Instruments, & Computers*, *31*:2, pp. 252-262.

[VAN 98] Vantaggiato, A. (1998). Automatic Exams and Course Administration on the Web. *Proc. WebNet'98*. AACE, Orlando.

[VIG 99] Vigilante, R. (1999). Online Computer Scoring of Constructed-Response Questions. *Journal of Information Technology Impact, 1*:2, 57-62.

[WG to appear] Wiemer-Hastings, P. & Graesser, A. (to appear). Select-a-Kibitzer: A computer tool that gives meaningful feedback on student compositions, *Interactive Learning Environments*.

[WH 99] Whittington, D., & Hunt, H. (1999). Approaches to the Computerized assessment of free text responses. *Proc. of the Third Annual Computer Assisted Assessment Conference*. Loughborough.

[WIE 99] Wiemer-Hastings, P. (1999). How Latent is Latent Semantic Analysis?, in *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI'99)*, Stockholm.

[WOL et al 98] Wolfe, M. B., Schreiner, M. E., Rehder, B., & Laham, D. (1998). Learning from text: Matching readers and texts by Latent Semantic Analysis, *Discourse Processes*, *25*:2-3, pp. 309-336.

[WRE 93] Wresch, W. (1993). The imminence of grading essays by computer—25 years later, *Computers and Composition*, *10*:2, pp. 45-58.