A Tool for Assessing Student Learning in Computing Sciences Distance Education Classes

Yi-fang Brook Wu and Xin Chen

New Jersey Institute of Technology

Abstract

To promote STEM higher education, avenues such as distance education must be visited. Students in most of the distance-learning classes generate great amounts of textual messages for class interaction, discussion and assignments which take up most of instructor's time to respond and grade. Researches in distance learning and computer-aided grading have been well developed, but little work has been done to apply automated text process techniques to solve the problem of evaluating students' performance in virtual classrooms. This paper introduces a software application that features the assessment of student learning in distance education classes by analyzing online class messages. The assessment model is described and its implementation is discussed. Functionalities of the software for learning assessment are illustrated with examples. The result shows that the software application is a useful supplementary teaching tool.

Keywords

Distance Learning, Learning Assessment, Text Processing, Keyword Density

1. Introduction

STEM related job opportunities will grow significantly between 2000-2010, reported by Bureau of Labor Statistics. However, the supply of STEM workers cannot meet the demand, so foreign workers on work VISA are relied upon to fill STEM positions. One way to promote STEM higher education to ensure more STEM workers are produced is to provide more STEM degrees delivered via distance education or eLearning which fosters learning activities regardless of geographical limitations. Distance education has become popular with the advances of both theory and technology of "Virtual Classroom" (Hiltz, 1994). In distance education classes, students participate in class by posting messages and replying to others with the support of electronic conference systems. To assess students' performance effectively, an instructor has to read all the messages, among other evaluation techniques they use. Participation in and grading of the discussion postings take more than 50% of the instructor's time of teaching an online class (Lazarus, 2003). It is indeed a heavy workload for the instructor, especially when he/she has more than one online class to teach at the same time, or there are many students in one class. A second grader such as a teaching assistant is useful and necessary to reduce grading bias, but it is expensive and nearly impossible to have in some cases. Therefore, an automated learning assessment system would be a great aid to instructors. It not only reduces their workload by analyzing students' submissions automatically, but also increases the fairness of grading by alerting instructors with cases of disagreements to the grades the instructor and the system assign.

This study is intended to develop such a software application that can be used as a supplementary teaching tool and assist instructors in their grading. We tested the tool with eLearning courses offered in the Information Systems Department at New Jersey Institute of Technology. The software package presented in this paper has the following functionalities.

- Downloads all class messages from the electronic conference system for analysis. The downloaded copy can also serve as a backup of the class materials.
- Calculates a performance indicator score for each student from messages posted by him/her. The calculation is based on the assessment model described in section 2.
- Visualizes the results in various ways to help instructors interpret the results in their grading.
- Exports the results to Microsoft Excel worksheet for more advanced analysis and result presentation.

The rest of this paper is organized as follows. In section 2, we present the assessment model as well as some related work. Section 3 entails a detailed discussion of different modules of the software package, including the message downloading module, the learning assessment module, and the visualization module. The functionalities are illustrated with examples. Section 4 discusses the evidence of the usefulness of the tool. Section 5 concludes the paper with some final remarks.

2. The Learning Assessment Model

2.1 Related Work

Good assessment in virtual classrooms helps faculty members find appropriate ways to deliver the course content and to assess what students know and what they can do with that knowledge. It can be argued that the greater the diversity in the methods of assessment, the fairer assessment is to students (Race, 1995). Therefore, multiple measures related to an individual academic program and course objectives should be used in studying student performance (Picciano, 2002; Shea et al, 2001). In web-based distance learning classes, student participation is a key to effective collaborative learning (Hardless et al, 2001). It can be evaluated with the information of students' usage of the system, such as login times and number of posts. It can also be analyzed with the content of students' messages. Messages can be categorized manually and mapped to the learning objectives, but this approach is not suitable to less-structured online discussion and is difficult for assessors to make consistent judgments.

Automated essay grading aims at assessing the quality of essays by analyzing both the content and the surface features of essays. It can be borrowed to assess student learning. Surface-feature-based approaches, such as Project Essay Grade (PEG) (Page, 1966), are developed upon the idea that an essay's quality could be revealed by certain surface features, which would correlate to the grades assigned by human judges. Content-based approaches focus on the semantic relationships between words and the context. A semantic space, the contextual usage of words, is constructed from training essays. A test essay is compared with the documents in the space, and assigned with a score according to the grades of the nearest essay(s) (Burstein et al, 1996; Landauer and Psotka, 2000). A hybrid approach combines these two to achieve better performance (Burstein et al, 2003; Larkey, 1998). Our assessment model is drawn upon the studies in both fields.

2.2 Measures

As the core of the software application, the learning assessment model is briefly described in this section. The details of the model can be found in (Chen and Wu, 2004a and 2004b). The model assesses student learning from three aspects: the quality of their course work, the quantity of their efforts, and the activeness of their participation; the proposed three measures - keyword density, message length, and message count, are derived from the class messages to measure each assessment aspect respectively.

We assume that quality of learning is revealed by the quality of messages generated by a student. The number of key concepts appearing in the messages reflects the knowledge range of the author, so the usage of key concepts could be an indicator for the learning quality. We define keyword as a simple, non-recursive noun phrase, i.e. base noun phrase. A base noun phrase consists of a head and none or more modifiers, which can be adjectives or nouns.

The unique noun phrases extracted from all class messages form the class concept base, which represents the concepts related to the major topics in the class. Noun phrases are weighted according to their frequency, length and number of authors, and it is denoted as

$$w = (1 + \log(len)) \cdot \left(f \cdot \log \frac{N}{n} \right),$$

where w is the weight of a noun phrase, len is the length (number of words) of the phrase, f is the frequency of the phrase in the concept base, N is the total number of students in the class, and n is the number of students who use the phrase in their messages. The second part of this function is similar to the tf.idf term weighting scheme in Information Retrieval (Salton, 1989), which gives higher weight to terms that appear more often in a document (term frequency: tf) but lower weight to terms that appear in more documents (inverse document frequency: idf). We estimate each student's contribution to the class concept base by adding the weights of noun phrases in his/her messages. It is called Keyword Density (KD).

Previous research (Levenburg and Major, 2000) has found a direct and positive relationship between the amount of time students spend reading postings and engaging in virtual dialogue with their classmates and their achievement of course objectives. Message length (ML), which is the number of words in a student's messages, is defined to measure a student's effort in the class. We also define Message Count (MC), which is the number of messages posted by a student, to measure the degree of activeness of his/her class participation.

2.3 Assessment Model

Taken together, the three measures are combined to compute a Performance Indicator (PI) score, which is defined as $PI_i = \alpha KD_i + \beta ML_i + \gamma MC_i$, where PI_i is the performance indicator score assigned to student i, and the coefficients α, β, and γ are the weights of each of the three measures respectively.

The experiment results in previous studies show that there is a high correlation between the PI scores and the actual grades assigned by instructors, as well as a high correlation between the rank orders of students by the PI scores and that by actual grades. Evidence of the supplementary role of the PI scores is also found.

3. The Software Package

A tool that implements the model would be useful to instructors. We identify three requirements for the tool to be fully functional and useful. First, it should make it easier and faster for instructors to browse and navigate the class messages, and provide some statistics of the class. Second, it should be able to calculate the performance indicator scores from the class messages. Third, it should be able to visualize the results for easy interpretation.

3.1 Class Messages

Distance education classes are often supported by electronic conferencing systems that enables synchronous and asynchronous between class participants. At New Jersey Institute of Technology, two systems, WebBoard and WebCT, are provided to support all distance learning classes. They are both server side programs. Services are running at a central computer (server) and client computers access the services through standard web browser. Data are stored in the database on the server side. So far, there is no easy way to create a copy of all messages for both systems (WebBoard does not offer backup function which can create a copy of the content of a discussion board on the client's computer. WebCT offers backup function but the data has to be restored to the system in order to read the messages). However, having a copy of class messages is desired to instructors, because it enables them to (1) backup class materials for future references, (2) browse class messages offline, and (3) conduct analysis over the content.

Therefore, the prerequisite of all other functions is a message acquiring function, which can download class messages from the online conference systems. Because different messaging systems use different data formats, we designed the download function as plug-ins. The overall architecture of the software application does not need to be changed when a new system is introduced. A plug-in for the new system would be everything that is needed to integrate the new system into the software application. As of now, we have implemented the plug-in for the WebBoard system. The one for WebCT is in progress and will be finished soon. We will use messages downloaded from the WebBoard system as examples in the following illustration.

3.1.1 Download Class Messages

Given a valid user account, WebBoard plug-in downloads all accessible content of a board from the server, and saves them on the local computer in the original structure. To develop a program working independently (without changing the WebBoard server code), we implemented all communications between the WebBoard plug-in and the WebBoard server through HTTP protocol. The program simulates the web browser by sending appropriate HTTP requests to the server and parsing the returned responses for information of interests.

The downloading process consists of two steps: structuring and retrieving. During the first step the plug-in retrieves the structure of class messages. Given the URL of a class message board and a user account, the plug-in retrieves the entry page of the board and parses the HTML code to extract conference information in the board. It then expands each conference to find the root messages (discussion topics) and all replies. The structure information is saved, so that class messages will be organized in the original structure. The second step is more straightforward. The URL of each message is appropriately constructed and is used to download the message page. Each message page is saved in an HTML file to retain the original formatting.

Though the data formats in different systems are different, each download plug-in can recognize the format of the corresponding system and convert it to a unified format used by the application. Download plug-ins can be invoked from the software application. After a plug-in successfully downloads all messages, the software displays the messages in the main user interface. Figure 1 shows the discussion conferences and the structure of part of the messages under the first conference in two different interfaces - web browser and our program. The original structure of messages is retained. Users can expand a topic to read all the replies.

3.1.2 Browse Class Messages

Because the class messages are saved in their original structure, the instructor can browse the messages just like what he/she does with the WebBoard system in web browser. When a message is selected, the content and all replies are displayed in the message window. This thread view mode enables the instructor to

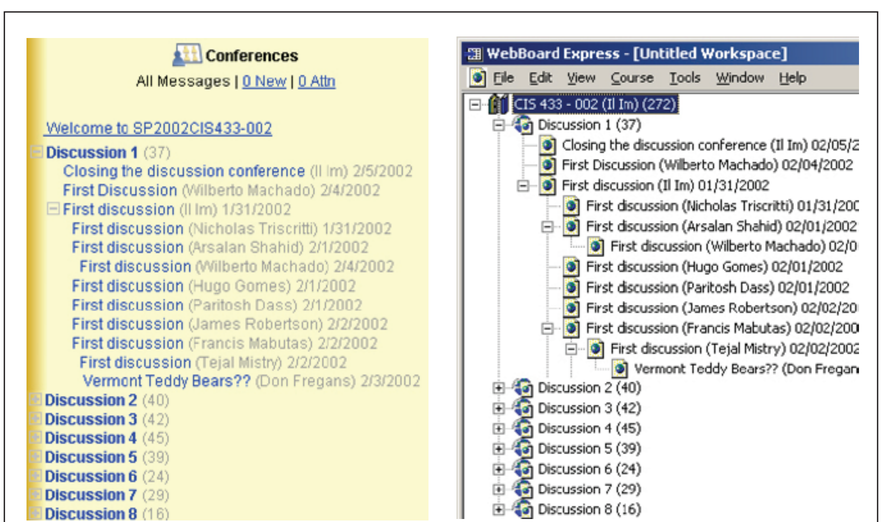


Figure 1: Class Messages in Two Different Interfaces

locate a specific discussion topic and read the entire thread. Figure 2 shows the interface for displaying messages in thread view mode.

One goal of developing the tool is to make message browsing easier and faster. Instructors might also be interested in knowing what a student has posted and replied. In addition to the standard thread view mode, the software application provides another view mode – user view mode, in which the messages are grouped by their authors. By simply expanding an author node in the tree, the instructor can see all the messages posted by a particular student. This view mode enables the instructor to read all messages from a student quickly. Figure 3 shows the user view mode to display all 34 messages posted by the instructor, Dr. Il Im.

3.2 The Learning Assessment Function

The learning assessment model described in previous section is implemented. This module analyzes the class messages and gives instructors more information about students' online participation and performance. Before calculating the performance indicator scores for students, the program displays a dialog to ask the instructor to specify the necessary parameters. It is shown in Figure 4.

The instructor first needs to select the conferences and gives each conference a weight. Some of the conferences may be created for class administration (e.g. Instructor's Announcements) or other purposes (e.g. Student Introductions), and they should not be included in learning assessment. Such conferences can be removed from analysis by deselecting the checkboxes in front of them. A weight is necessary for selected conferences because not all conferences are equally important in student learning assessment. For example, the required discussion and voluntary discussion should have different weights. Another set of important parameters are the coefficients in the assessment model. The instructor can specify a value for each measure to reflect his/her grading preferences. For example, by defining α=5, β= 3, and γ=1, the instructor would like to give higher grades to students who are more able to synthesize knowledge learned from the class, rather than to post many short, content poor messages. Other parameters in the dialog provide the instructor with more control over the process, but one can proceed with just the default values.

After the parameter setup is done, the program first converts HTML messages to plain text files. The message headers (conference

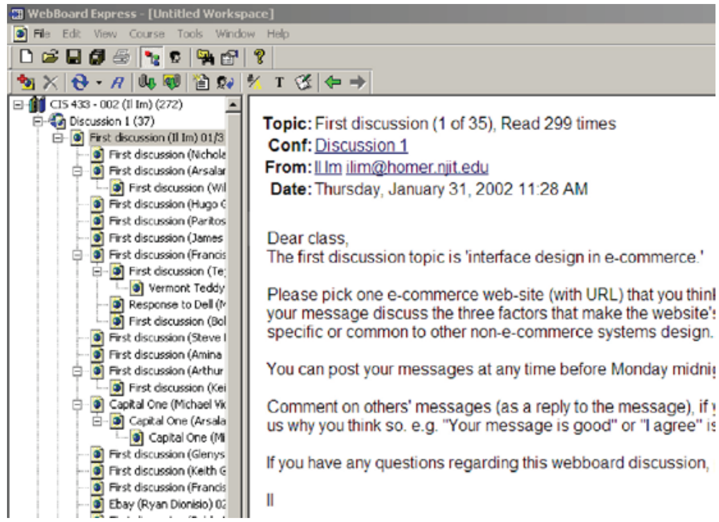


Figure 2: Browsing Class Messages in Thread View Mode

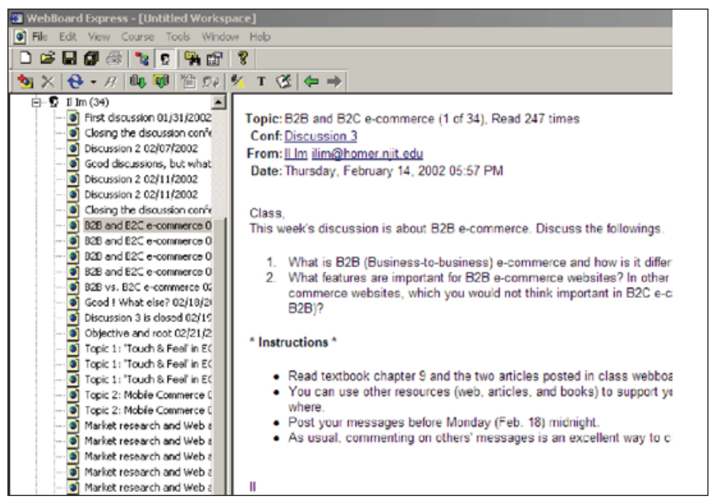


Figure 3: Browsing Class Messages in User View Mode

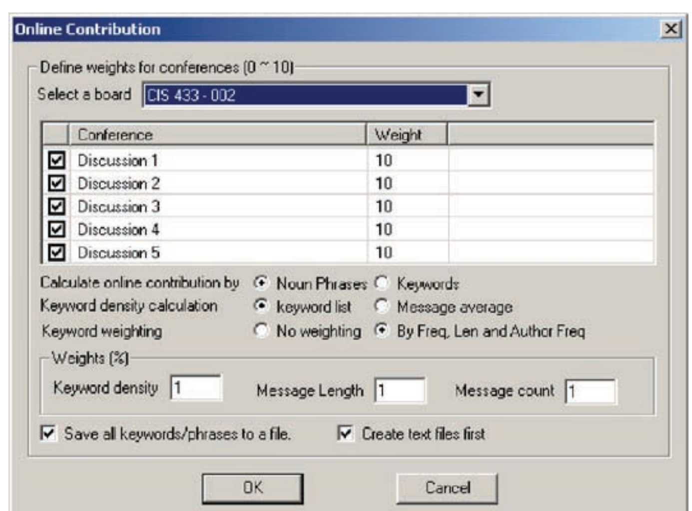


Figure 4: The Parameter Setup Dialog

name, title, post date, etc.) are removed, so are the quoted lines in reply messages. Only the main body of each message is saved in the text file for further processing. The text files are tokenized and the extracted tokens are passed to a noun phrase extractor, which tags the tokens with appropriate POS tags and identifies noun phrases by matching the POS tag sequence to predefined patterns.

The program then counts words and noun phrases for each message, each student, each conference, and the whole class. It also calculates the weight for each noun phrase according to weighting scheme proposed in the assessment model. For each student, the three measures, KD, ML and MC, are calculated based on the results obtained in previous steps. The last step is to calculate the PI score for each student, using the coefficients specified by the instructor. In addition to the PI score, the results from the intermediate steps are saved too.

3.3 Result Presentation

The program can present the results in various ways, such as list and chart. In designing the user interface, a lot of focus has been on keeping the program easy to use and as useful as possible. As a result, the program has the ability to present the results in different formats, from lists that provide comprehensive information to charts that show the statistics and distribution of class and student information. In the following illustrating examples, students' names are replaced with identification numbers for confidentiality purpose.

Figure 5 shows the list display of the detailed information of conferences and students in a class. When the user selects a class in the left-side window, the detailed information of the conferences or students will be displayed in the list in the right-side window. For each conference, the list includes the conference name, number of messages, number of attachments, number of users in the conference, total words (TW), unique words (UW), total keywords (TK) and unique keywords (UK). In addition to the similar columns of the conference list, the student list also includes Keyword Density (KD), Message Length (ML), Message Count (MC) and the Performance Indicator (PI) score. The two lists provide an instructor with comprehensive information about the messages and students in a class. The instructor can click the header of a column to sort the list in either ascending or descending order.

Though the list view can present the details of an object, it may not give instructors an overview of a class at their first glance. To present

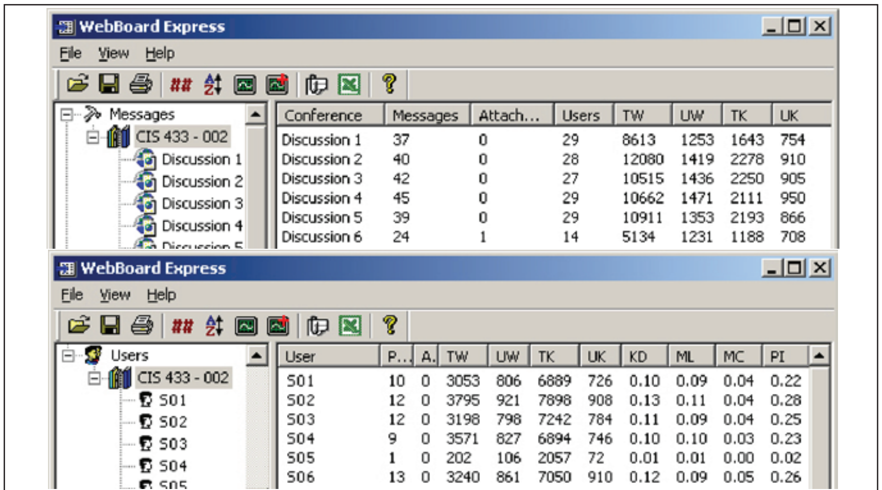


Figure 5: Detailed Information about Conferences and Students

the overall picture, we visualize the results in charts and plots. The current version of the software provides graphical presentation of the rank order of students by the PI score, the PI score distribution and some other information. The graphs are created using the Microsoft Chart ActiveX control. We will use the following screen shots to show how the information is visualized in the program.

The user opens the graph dialog by clicking the *Graph* button in the toolbar. The graph dialog is separated into two areas: graph selection and the graph. When the user selects a graph from the list, the corresponding graph will be displayed in the graph area. Figure 6 shows the rank order of students by the PI score in descending order. The mean and the median of the PI scores are marked in the plot. From the plot it is easy to see how many students are above the average level of the class.

Figure 7 presents the distribution of the PI

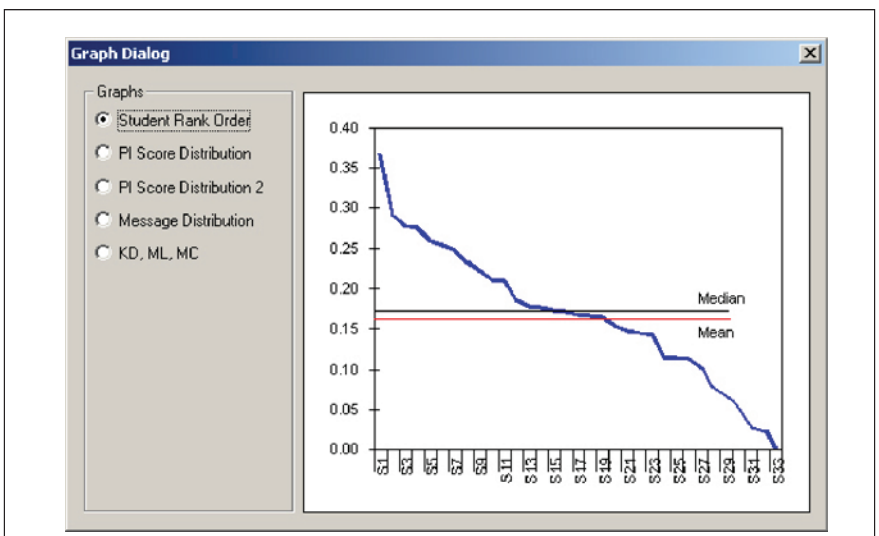


Figure 6: Student Rank Order by PI Score

scores in bar chart. The maximum PI score is divided into ten intervals. For each interval, the number of students whose PI scores fall into the interval is counted. The shape of the distribution indicates whether there is anything abnormal in the class. As we can see from figure 7, PI scores in the sample class generally follow the normal distribution, which is expected.

The distribution of the PI scores can also be displayed in pie chart, as shown in figure 8. The number of students in each interval of PI scores is shown in different colors, and the size of the pie represents the percentage of the number of students in an interval.

By selecting the desired item from the left list, the instructor can view the corresponding graph in the right side of the window. The program provides a set of predefined graphs that can be displayed with single click. Instructors do not have to do any extra work to create plots and chars. However, such pre-configuration may prevent some instructors from creating more graphic presentation of the results they need. To meet the needs of these instructors, we implement an export function that exports the results from the list to a Microsoft Excel worksheet. We utilize the Microsoft Office Automation techniques to create an Excel application and export the results to a new Excel worksheet. By simply clicking on the *Excel* icon on the toolbar, the instructor can export anything in the list to Excel worksheet (an example is shown in figure 9). After exporting the results, instructors can take advantage of the statistical and graphing function of Excel to conduct more analysis and to create more charts or plots. The discussion section entails a discussion of the usefulness of performing such analysis in Excel.

4. Discussion

When the data are available in an Excel worksheet, the instructor can perform a correlation analysis between the PI scores and the students' actual grades to ensure the correctness of grading. PI scores deviated from the actual grades may suggest either inappropriate grades, or something special in the students' messages, or both. Evidence of the supplementary role of the tool as a grading tool is found. In one class, the PI score of one student was much higher than the actual grade. By reexamining the student's messages, the instructor found that the student copied and pasted a long message along with the source URL from the web without adding his/her personal opinions. Even though the instructor had encouraged students

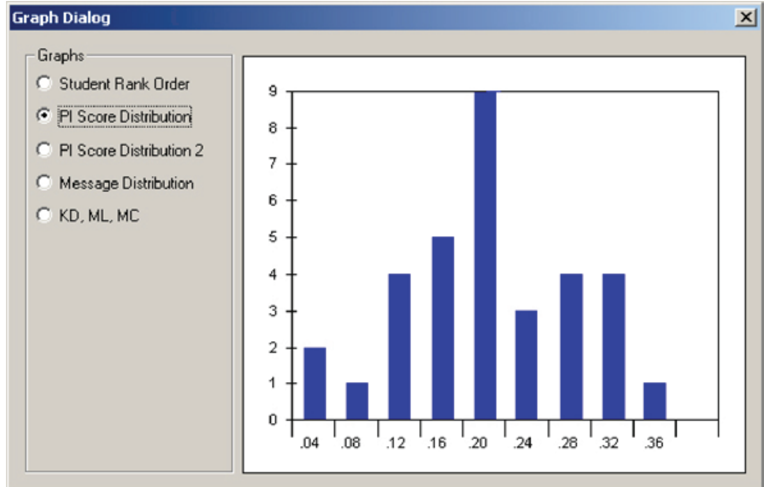


Figure 7: Distribution of PI Scores in Bar Chart

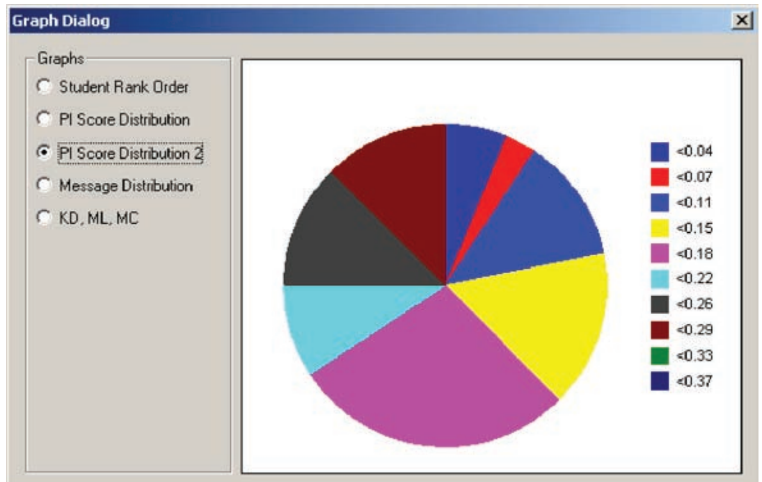
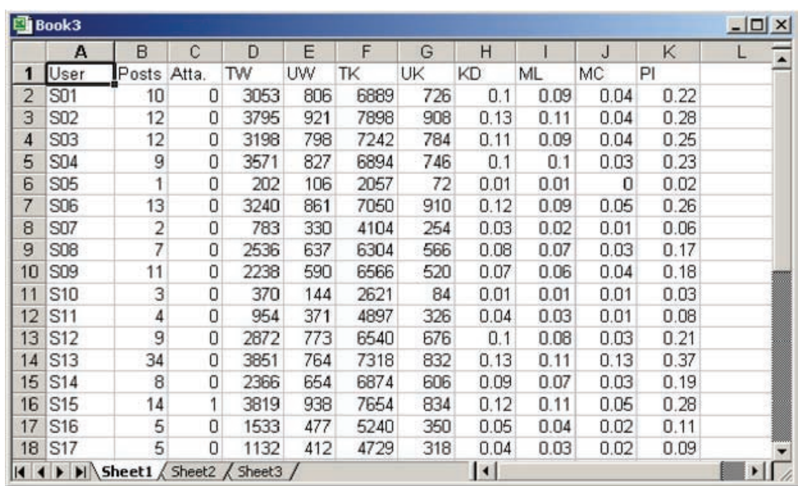


Figure 8: Distribution of PI Scores in Pie Chart



Book3

	A	B	C	D	E	F	G	H	I	J	K	L
1	User	Posts	Atta.	TW	UW	TK	UK	KD	ML	MC	PI	
2	S01	10	0	3053	806	6889	726	0.1	0.09	0.04	0.22	
3	S02	12	0	3795	921	7898	908	0.13	0.11	0.04	0.28	
4	S03	12	0	3198	798	7242	784	0.11	0.09	0.04	0.25	
5	S04	9	0	3571	827	6894	746	0.1	0.1	0.03	0.23	
6	S05	1	0	202	106	2057	72	0.01	0.01	0	0.02	
7	S06	13	0	3240	861	7050	910	0.12	0.09	0.05	0.26	
8	S07	2	0	783	330	4104	254	0.03	0.02	0.01	0.06	
9	S08	7	0	2536	637	6304	566	0.08	0.07	0.03	0.17	
10	S09	11	0	2238	590	6566	520	0.07	0.06	0.04	0.18	
11	S10	3	0	370	144	2621	84	0.01	0.01	0.01	0.03	
12	S11	4	0	954	371	4897	326	0.04	0.03	0.01	0.08	
13	S12	9	0	2872	773	6540	676	0.1	0.08	0.03	0.21	
14	S13	34	0	3851	764	7318	832	0.13	0.11	0.13	0.37	
15	S14	8	0	2366	654	6874	606	0.09	0.07	0.03	0.19	
16	S15	14	1	3819	938	7654	834	0.12	0.11	0.05	0.28	
17	S16	5	0	1533	477	5240	350	0.05	0.04	0.02	0.11	
18	S17	5	0	1132	412	4729	318	0.04	0.03	0.02	0.09	

Sheet1 / Sheet2 / Sheet3

Figure 9: An Excel Worksheet Exported from the Program

to share anything they found relevant and interesting to the class, without personal opinions and thoughts, the instructor considered this to be less effort. Therefore, the original grade was confirmed.

A similar case was found in another class, in which a student (hereafter known as S) got the highest PI score, but a low grade. After the semester ended, the instructor used our tool to verify the grading. By reviewing the student's messages and assignments, the instructor found that S received low grades on all discussions but the tool gave him the highest PI score. The instructor further investigated the case and found that S submitted almost all assignments late, even though they were good. S explained that it was because of family and personal medical problems. The instructor accepted S' late assignments and increased S' final grade. Figure 10 illustrates the close relationship between the rank orders of students by PI scores and by actual grades (outlier S is marked in red circle). Having the program serving as a second grader, instructors are able to capture outliers, and to reduce misjudgment, bias, or errors in grading.

Feedback from an instructor who used the tool also suggests that it is useful. The two view modes of class messages make message browsing easier and faster. "Using it can help me determine the quality and quantity of a student's participation, because after clicking on a student's name, I can see all of that student's postings," said the instructor, "even though I usually follow students' discussion, but keeping track of their degree of participation was very difficult without the tool." The instructor also found the PI scores helped her verify her impression of students' performance. She described the procedure as "grade students first, and compare my grading with that of the tool. If disagreement occurs, I then use the tool to read messages to verify the grading. It did help me find out some problems."

5. Conclusions

In this paper, we present a tool for learning assessment in distance education classes. The tool not only enables instructors to backup class messages and browses them offline, but also implements a learning assessment model to assist instructors in their grading. The results are visualized in various types of graphs for easy interpretation. An export function provides an interface between the tool and Microsoft Excel, so that more analysis and data presentation

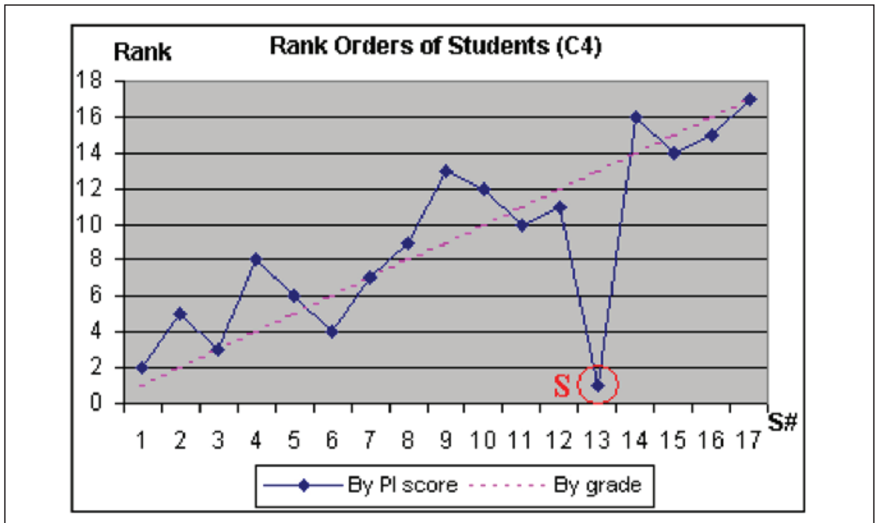


Figure 10: Correlation between Students' Rank Orders by PI Score and by Grades

can be done.

The experiment results and instructor's feedback indicate that the tool, as a supplementary teaching tool, is useful. However, it is designed as a supporting tool; instructors are not expected to grade students' work based only on the results from the tool.

This version of the software application only visualizes the most commonly used charts and plots. A desired feature of the future version would be more visualization of the results. Because the software has an *Export to Excel* function, we do not want to repeat the functions that can be easily done in Excel. We plan to add to the package more visualization functions that are useful but difficult to obtain in Excel (due to its purpose for general usage).

One possible enhancement is to show the interaction among students by analyzing their messages and replies. This would enable instructors to better understand what's going on in the class. Another useful feature would be to allow instructors to grade students in the program directly so that the grades can be compared with the PI scores within the tool, instead of being exported to Excel.

References

Burstein, J., Kaplan, R., Wolff, S. and Lu, C. (1996). Using Lexical Semantic Techniques to Classify Free-Responses. In Proceedings of SIGLEX 1996 workshop, Annual Meeting of the Association of Computational Linguistics, University of California, Santa Cruz, 1996

Burstein, J., Leacock, C. and Chodorow, M. (2003). CriterionSM: Online essay evaluation: An application for automated evaluation of student essays. . Proceedings of the Fifteenth Annual Conference on Innovative Applications of Artificial Intelligence, Acapulco, Mexico, August 2003

Chen, X. and Wu, B. (2004a). Assessing Student Learning through Keyword Density Analysis of Online Class Messages. Proceedings of the American Conference on Information Systems, New York, New York, August 2004 (Best Paper Award)

Chen, X. and Wu, B. (2004b). Automated Evaluation of Students' Performance by Analyzing Online Messages. Proceedings of IRMA 2004, New Orleans

Hardless, C., Lundin, J. and Nulden, U., (2001). Mandatory Participation in Asynchronous Learning Networks. In proceedings of HICSS-34, January, Maui, USA, 2001

Hiltz, S. R. (1994). The Virtual Classroom: Learning Without Limits Via Computer Networks. Norwood NJ, Ablex, 1994

Landauer, T. K. and Psotka, J. (2000). Simulating Text Understanding for Educational Applications with Latent Semantic Analysis: Introduction to LSA. Interactive Learning Environments, 8(2) (pp. 73-86).

Larkey, L. S. (1998). Automatic essay grading using text categorization techniques . Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval

Lazarus, B. D. (2003). Teaching Courses Online: How Much Time Does it Take?. JALN, Volume 7, Issue 3, September 2003

Levenburg, N. M. and Major, H. T. (2000). Motivating the Online Learner: The Effect of Frequency of Online Postings and Time Spent Online on Achievement of Learning Goals and Objectives. International Online Conference on Teaching Online in Higher Education

Page, E. B. (1966). Grading essays by computer: Progress report.. Notes from the 1966 Invitational Conference on Testing Problems, 87-100.

Picciano, A. (2002). Beyond Student Perceptions: Issues Of Interaction Presence, And Performance In An Online Course. JALN, Volume 6, Issue 1, July 2002

Race, P. (1995). The Art of Assessment. SEDA publication the New Academic, 1995 vol, 5 issue 3

Ramani, L. and Rocha, H. (2000). Interaction Map: Information Visualization Techniques in Web-Based Distance Education Environments. Web-based distance education environments, 2000

Reffay, C. and Chanier, T. (2003). How Social Network Analysis Can Help To Measure Cohesion in Collaborative Distance-Learning. Proceeding of Computer Supported Collaborative Learning conference, June 2003

Salton, G. (1989). Automatic Text Processing: The Transformation Analysis, and Retrieval of Information by Computer. Addison Wesley, 1989

Shea, P., Fredericksen, E., Pickett, A., Pelz, W. and Swan, K. (2001). Measures of learning effectiveness in the SUNY Learning Network. In J. Bourne & J. C. Moore (Eds) Online Education, Volume 2, Olin and Babson Colleges: Sloan Center for Online Education

Wasserman, S. and Faust, K. (1994). Social Network Analysis. Cambridge University Press, 1994

Yi-fang Brook Wu, Ph.D., is an Associate Professor in the Information Systems Department at New Jersey Institute of Technology. Her current research interests include: text mining, information extraction, knowledge organization, information retrieval, natural language processing and digital libraries. Her digital libraries projects have been supported by National Science Digital Libraries program at National Science Foundation, and Institute of Museum of Library Services. She is particularly interested in making educational resources more accessible to support digital library users.

Xin Chen, Ph.D., is a Software Design Engineer II at Microsoft. His research interests are information retrieval, text mining and digital libraries. His research has been published in journals such as Journal of the American Society for Information Science and Technology, and Journal of Asynchronous Learning Networks, and also in proceedings of Knowledge Discovery and Data Mining (ACM KDD), and Conference on Information and Knowledge Management (ACM CIKM).