A framework for the computerized assessment of university student essays

Rehab M. Duwairi *

Department of Computer Information Systems, Jordan University of Science and Technology,
Medical Building PH 4 LI, P.O. Box 3030, Irbid 22110, Jordan

Available online 11 November 2004

Abstract

This paper presents an approach for automatic grading of essays. Student essays are compared against a model or key essay provided by the teacher. The similarity between a student essay and the model essay is measured by the cosine of their contained angle in an n-dimensional semantic space. The model essay is preprocessed by removing stopwords, extracting keywords, assigning weights to keywords to reflect their importance and finally by linking every keyword to a subject-oriented synonym list. The student essay, by comparison, is preprocessed by removing stopwords and then by extracting keywords. The keywords extracted from the model essay and the keywords extracted from students essays together with weights provided by teacher are used to build feature vectors for teacher and students essays. The obtained grade depends on the similarity between these vectors (calculated by using the cosine formula). A simulator was implemented to test the viability of the proposed approach. It was fed with student essays (at the university level) gathered from database management course over three semesters. The results were very encouraging and the agreement between the auto-grader and human grader was as good as the agreement between human graders.

Keywords: Computerized assessment; Automatic-assessment; Automatic grading; Computer based grading; On-line assessment

* Tel.: +962 7201000x23402; fax: +962 2 7095014.
E-mail address: rehab@just.edu.jo.

0747-5632/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
1. Introduction

Due to the low costs of personal computers, increased networking capabilities and availability of software, large number of universities and schools are incorporating computerized testing and assessment of students. Researchers investigated the involvement of computers in student assessment as early as 1960s. Early research focused, however, on automatic grading of programming exercises (Hext & Winngs, 1969; Higgins, Symeonidis, & Tsintsifas, 2002; Jackson & Usher, 1997; Joy & Luck, 1998; Mansouri, Gibbon, & Higgins, 1998; Matt, 1994). Recent research, on the other hand, addressed the automatic grading of essays (Burstein et al., 1998; Foltz, Laham, & Landauer, 1999; Lanejiya, Kumar, & Prasad, 2003; Landauer, Laham, & Foltz, 2003; Larkey, 1998; Lewis David, 1992; Lonsdale & Strong-Krause, 2003; Rose et al., 2003; Shermis et al., 2001). Manual essay grading is a time-consuming effort, and this contributed to the phenomena that most teachers heavily rely on multiple-choice, fill in the blanks or short answer questions for student assessment. The need for involving computers in the process of essay grading is apparent in the presence of the Web, distance learning and intelligent tutoring systems. In distance learning model of teaching students need not be engaged in traditional student–teacher classrooms. On the contrary, students and teachers could be physically apart and therefore, the presence of an automatic grader system that can accept students responses (or essays) online and produce a grade in a short time is necessary. Intelligent tutoring systems are systems that mimics the human teacher and as assessment is a major role for human teachers, intelligent tutors should also contain a computerized assessment component.

This article presents a framework for the automatic grading of student essays. Even though there is resistance for letting computers be the sole judge of the quality of essays, they can play an auxiliary role such as providing feedback to students (especially when distance learning is used), replace one of the human graders when more than one human is involved in grading essays, and be an assistant to the teacher when he/she is the only judge of essays’ quality. This argument is applicable to situations where students’ essays are supposed to mirror instructor’s knowledge and preferences.

The work reported here automatically determines a student’s grade by comparing his/her essay to a model essay provided by the teacher. The final grade is obtained after applying the following steps, in order, to the student and model essays: essay preprocessing, weight vector generation and cosine angle calculation between vectors. In the preprocessing step, model essay is handled by removing stopwords (Stopwords includes words that do not affect the knowledge contained in essays such as is, a, an, the, that, etc.), extracting keywords (keywords are words that are important in reflecting level of knowledge), assigning weights to these keywords (to reflect correlation of reliability between the teacher’s model keywords and related acceptable synonyms given in the student’s essay) and by liking extracted keywords to synonym lists. Student essay is treated, on the other hand, by removing stopwords, and by extracting keywords. Weight vectors generation involves generating a teacher-vector that corresponds to the model essay and contains real numbers from 0 to 1 that
reflect the weights assigned to keywords present in the model essay. By comparison, we generate for every student essay a student-vector that corresponds to the student essay and contains numbers in the range 0–1 derived from the teacher vector based on the closeness of the student essay to the model essay. Finally, the cosine between student vector and teacher vector is calculated to obtain the final grade as explained in details in Section 3.

This paper is organized as follows: Section 2 presents related work. Section 3, by comparison, describes in details the approach we used for the automatic grading of students essays. Section 4 provides a description of the experimentations and result analysis that were carried out to assess the validity of the proposed methodology. Finally, Section 5 summarizes the work reported in this article and highlights future work.

2. Literature review

This section describes some of the related work that was reported in the literature. Larkey (1998) trained binary classifier to distinguish bad from good essays and then used that classifier to predict scores for new essays. He defined a large number of objectively measurable features in essays such as essay length, average word length, and average sentence length as input parameters to the classifier. A secondary goal of his work was to compare binary classifier with k-nearest neighbors and with linear regression approaches.

PILOT (Bridegeman et al., 2000) is a platform independent tool used for testing computer science concepts. In particular, PILOT supports graph problems such as finding minimum spanning tree, tree search algorithms and shortest path algorithms. PILOT has an automated grading mechanism that supports partial credits for the above-mentioned problem types.

Foltz et al. (1999) and Landauer et al. (2003) used latent semantic analysis (LSA) to assess the quality of essays. LSA is a machine-learning model that computes the semantic similarity of words and passages by first analyzing large bodies of domain relevant text (learning phase) and then marking student essays (grading phase).

Lanejiya et al. (2003) used syntactically enhanced latent semantic analysis (SELSA) technique that takes into consideration word-order or syntactic information that can improve the knowledge representation and therefore lead to a better performance than LSA. However, they found out that LSA has better correlation with human graders than SELSA; but SELSA was as good as LSA in terms of the absolute mean difference measure. SELSA was able to correctly evaluate a few more essays than LSA.

Burstein et al. (1998) exploited statistical redundancy inherent in natural languages to automatically grade essays. They employ a hybrid feature identification method, which includes syntactic structure analysis, rhetorical structure analysis and topical analysis, to grade essays.

Shermis et al. (2001) developed Project Essay Grader (PEG) software to evaluate web-based student essays. PEG is based on statistical models configured
specifically for the type of writing to be assessed. The system was trained on student essays to learn or extract observed variables that would contribute to the final grade. About 30 variables were found such as the essay length and the average word length.

The auto-grader presented in this paper differs from the work reported in Bridegeman et al. (2000) in that it handles a wider range of problems; not only problems related to graphs and trees. Auto-grader also has a relatively short preparation time. The only preparation required by the teacher is to provide a model answer, assign weights to keywords and assign synonyms to these keywords. On the other hand, the work reported in this section either takes a relatively long time to train the auto-grader (as in (Foltz et al., 1999; Landauer et al., 2003; Larkey, 1998)) or it relies on extensive processing of words and phrases to obtain a grade (as in Shermis et al. (2001)).

The following section presents the approach we used for the automatic grading of student essays.

3. Methodology

The student grade is obtained by applying the following three measures to student and model essays: essay preprocessing, weight vector generation and finally similarity calculation between the weight vectors that is based on the cosine formula. The types of essays that are handled by the proposed approach are content-based answers which are typically written for classroom tests, assignments and end-of-chapter review questions for university students. Since the suggested methodology requires the existence of a model answer (provided by the teacher), questions should have a single answer or a known number of answers. i.e. the work reported here do not deal with open-ended questions.

Student essays are preprocessed by removing stopwords and then by extracting keywords or terms (sorted in ascending alphabetical order). The removal of stopwords and extraction of keywords is done simultaneously during the parsing of essays. Words that appear in an essay and also appear in the Stopwords list are ignored; remaining words are considered keywords. The model essay, however, is preprocessed by removing stopwords, extracting keywords or terms (sorted in ascending alphabetical order), assigning weights to keywords to reflect their importance (weights are real numbers in the range from 0 to 1 specified by the teacher) and by linking every keyword to a subject-oriented synonym list (synonym lists are created by the teacher). These synonyms are domain-specific and one keyword could be linked to several synonyms. Note that the terms may contain phrases to reflect the fact that correct essays contain sequences of words.

Weight vectors generation involves generating a teacher-vector that corresponds to the model essay and contains real numbers from 0 to 1 that reflect the weights assigned to keywords present in the model essay. For every student essay we generate a student-vector that corresponds to the student essay and contains numbers in the
range 0 to 1. The student-vector is filled in the following manner: for every keyword present in both student essay and model essay (taking into consideration the synonym lists) a value equal to the weight assigned to that keyword in the model essay is placed in the cell corresponding to that keyword in the student-vector. For every keyword that is present in the model essay but absent from the student essay a 0 is assigned to the cell corresponding to that keyword. Keywords that are present in the student essay but absent from the model essay are ignored.

When student and teacher vectors are generated the similarity between the student essay and model essay is estimated by calculating the cosine between the teacher and student vectors.

The following pseudocode describes the algorithm that we employ to calculate the student mark based on the teacher model answer. Let:

\( U_s \) be the string that contains the student essay.
\( U_t \) be the string that contains the teacher or model essay.
\( T_s \) be the keyword or term list of the student answer.
\( T_t \) be the keyword or term list of the model essay.
\( V_s \) be vector that contains the weights of keywords found in the student essay.
\( V_t \) be vector that contains the weights of keywords found in the teacher essay.

**Input:** \( U_s, \ U_t, \ T_s, \ T_t \) and set of weights assigned by the teacher.
**Output:** an essay score.

**Steps:**
- Remove stopwords from \( U_t \) and \( U_s \)
- Extract keywords from \( U_t \) and store them in \( T_t \)
- Extract keywords from \( U_s \) and store them in \( T_s \)
- Sort \( T_t \) in ascending order.
- Sort \( T_s \) in ascending order.
- While \( T_t \) is not empty do the following:
  - For every term \( t_i \in T_t \) store its corresponding weight \( w_i \) in cell \( i \) of \( V_t \).
  - For every term \( t_i \) in \( T_s \)
    - Check if \( t_i \) exists in \( T_t \). This includes checking the synonym list associated with \( t_i \) in \( T_t \).
    - If \( t_i \) exists in \( T_t \) in cell \( i \) then assign \( w_i \) to that cell in \( V_s \) otherwise assign 0 to that cell.
  - Remove \( t_i \) from \( T_t \) and \( T_s \).
- Use the cosine measure to calculate the similarity between \( V_t \) and \( V_s \):

\[
\text{cosine}(V_t, V_s) = \frac{V_t \cdot V_s}{||V_t|| \ |V_s|}
\]

**Student grade = cosine(\( V_t, V_s \)) * 100** (assuming that the grade is out of 100)
where cosine is a function that returns a real valued number in the range 0–1 that represents the similarity between teacher and student essays.

\[ V_t \cdot V_s \] is the dot product of the vectors \( V_t \) and \( V_s \). 

\[ | V_t | = \sqrt{V_t \cdot V_t} \] and 

\[ | V_s | = \sqrt{V_s \cdot V_s}. \]

4. Experimentation and result analysis

We collected four datasets from university student essays written as part of exercises and exams to a database management course (a 3rd year level course) over a period of three semesters. All the four datasets expected certain facts to be present as part of the correct answer. Every dataset contains 50 essays.

The first dataset contains students’ essays written as a response to a question asking them to enumerate and briefly explain the benefits of a clustering file organization. The second dataset, on the other hand, consists of students’ responses to a question asking them to enumerate and briefly explain the procedures a standard recovery algorithm would take. The third dataset contains students’ essays written as answer to a question asking them to explain why the cost of a write operation is always greater than the cost of a read operation. Finally, the fourth dataset contains students’ essays collected as answers to a question requiring students to explain why the size of an index file is always smaller than the size of a data file.

Two human graders manually graded all essays in the four datasets. The first grader was the course teacher and the second was the teaching assistant for the course. Both the teacher and the teaching assistant used the same key or model essay provided by the teacher. We fed student essays and model answers to the simulator (or auto-grader) and it automatically graded the essays, then the correlation coefficient between human and auto-grader were calculated. Table 1 shows the correlation coefficient between the automatic and human graders (when grades were represented as real numbers in the range 0–100).

We noticed from Table 1 that the agreement between auto-grader and human grader is close enough to the agreement between human graders. We also notice that the correlation coefficient between human graders was higher than the coefficient between human and automatic graders in the four experiments. This is due to two reasons; firstly, some students used algebraic expressions, or mathematical notations to convey their answers, and such answer forms are not currently handled by our automatic gra-

<table>
<thead>
<tr>
<th>Dataset</th>
<th>1st Human grader against auto-grader</th>
<th>2nd Human grader against auto-grader</th>
<th>1st Human grader against 2nd human grader</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.8</td>
<td>0.8897</td>
<td>0.91</td>
</tr>
<tr>
<td>2</td>
<td>0.7609</td>
<td>0.7695</td>
<td>0.8784</td>
</tr>
<tr>
<td>3</td>
<td>0.8138</td>
<td>0.7248</td>
<td>0.8571</td>
</tr>
<tr>
<td>4</td>
<td>0.7173</td>
<td>0.6919</td>
<td>0.7665</td>
</tr>
</tbody>
</table>
der; and, secondly, English is a second language to the students who wrote the essays therefore, a human grader was more understanding to this fact. A final comment on the experiments is that the accuracy of the automatic grader is highly dependent on the model or key answer, list of synonyms and weights provided by the teacher.

5. Conclusions and future work

This paper has presented a framework for the automatic grading of student essays. Student essays are compared against a model answer provided by the teacher. The similarity between the student essay and model essay is measured by the cosine of their contained angle in an \textit{n}-dimensional semantic space. The final grade is obtained by applying the following three steps: essay preprocessing, weight vector generation and finally calculating the cosine between the weight vectors.

A simulator was implemented to test the viability of the proposed methodology. Four datasets of student essays were collected from essays written by students in a database management course over a period of three semesters. Every dataset contains 50 essays. These essays were graded manually by two instructors and automatically by the simulator. The correlation coefficient between automatic grader and human graders was calculated and the agreement between the automatic grader and human grader was as good as the agreement between human graders.

A future enhancement to the above mentioned framework would be to assign contexts to keywords extracted from model and student essays. This would allow the automatic grader to accurately grade essays that involve comparisons between several objects and a student mistakenly swaps the characteristics of the compared objects. For example, assume that students were asked to compare between hard disks and tapes (two examples of secondary storage devices); a student who incorrectly answers the question and writes “tape is faster than hard disk” will not score zero using the auto-grader. This is due to the fact that auto-grader uses a bag-of-words model and therefore word order is unimportant. We agree here with Larkey (1998). Other models, on the other hand, such as the word reported in Leacock & Chodorow (2003) take word order into consideration. Taking word order into consideration involves a large processing overhead when processing students essays. In the above example, if contexts were employed then the word \textit{faster} would be associated with the context \textit{disk} and therefore the auto-grader would be able to flag the error mentioned above as \textit{faster}, in the example, is associated with \textit{tape} not \textit{disk}.

References


