Automated Scoring of Secondary Student Summaries and Short Answer Tests

David Jones, Innovation Assessments

December 2019

Introduction

Research and development of software to harness artificial intelligence for scoring student essays has many significant obstacles. Using machine learning techniques requires massive amounts of data and computing power far beyond what is available to the typical secondary public school. The cost and effort to devise such technology does not seem to be juice worth the squeeze, since it is still more time efficient and cost effective to just have a human do the job. However, the potential exists to devise AI-assisted grading software whose purpose is to increase the speed and accuracy of human raters. AI grading that is "assisted" applies natural language processing strategies to student writing samples in a narrowly defined context and operates in a mostly "supervised" fashion. That is, a human rater activates the software and may make scoring judgments with the advice provided by the AI. A promising area for this, more narrowly contextualized application of artificially intelligent natural language processing, is in scoring summaries and short answer tests. This also poses interesting possibilities for automated coaching for students while they write. This study examines a set of algorithms that derives a suggested score for a secondary level student summary and short answer test response by comparing a corpus of model answers selected by a human rater with the student work. The human rater stays on duty for the scoring process, adding full credit student work to the corpus such that the AIs is trained and selecting student scores.

Features of Text for Comparison

The AI examines the following text characteristics to evaluate a student work by comparison to one or more models:

1. "readability" as determined by the Flesch-Kincaid readability formula
2. the percent difference in number of unique words after pre-processing[1]

[1] "Preprocessing" refers to text that has been scrubbed of irrelevant characters like HTML tags and extra spaces, has been lemmatized, synonymized, and finally stemmed.

3. intersecting noun phrases
4. Jaccard similarity
5. cosine similarity of unigrams
6. cosine similarity of bigrams
7. cosine similarity of trigrams
8. intersecting proper nouns
9. cosine similarity of T-score
10. intersecting bigrams as percent of corpus size
11. intersecting trigrams as percent of corpus size
12. analysis score[2]

The program first compares the student text to each model using cosine similarity of n-grams. The most similar model in the corpus is then compared to the student work. Four hundred and twenty-six short answer questions that had been scored by a human rater were compared using the algorithm. From these results was developed scoring ranges within each text feature typifying scores of 100, 85, 65, 55, and 0. Outliers were removed from the dataset. Next, sets of student summaries were scored using the ranges for each text feature and the program's scoring accuracy was monitored. With each successive scoring trial, the profiles were adjusted, sometimes more intuitively that methodically, until over the course of months the accuracy rate was satisfactory.

When analyzing a student writing sample for scoring, the score on each text feature is compared to the profiles and the program keeps a tally of matches for each scoring category (100, 85, 65, 55, and 0). The "best fit" is the first stage of suggested score. Noun phrases, intersecting proper nouns, and bigram cosine were found to correlate most highly with score matching the human rater, so an additional calculation is applied to the profile scores to weight these factors. Next, a set of functions calculates partial credit possibilities for scores in the category of 94, 76 and 44

[2] "Analysis score" is a metric devised to evaluate the "analytical richness" of a student writing sample. See Jones, D. (2019). *Developing Semi-Automated Evaluation of Analysis in Secondary Student Writing.* Retrieved from https://www.innovationassessments.com/demo/Developing%20Semi-Automated%20Evaluation%20of%20Analysis%20in%20Student%20Writing%20Samples.pdf.

using statistics from the data analysis of the original dataset of 426 samples. Finally, samples where analysis are important in the response have their score adjusted one final time.[3]

The development of the scoring ranges for text features proceeded somewhat methodically and at times more intuitively or organically. Over the course of months, when error patterns in AI scoring became apparent, adjustments were made to improve performance. Natural language processing, even at this basic level, is very demanding on computer memory and processing resources. At this writing, the server running this software has 6GB of RAM and work is often being done on the code to reduce processing time. One strategy is to store both "raw" and processed versions of the student work products as they are written so that processing time can be shortened at the end. the corpus of model responses is also saved in this way.

Training the AI

Upon creation of an assignment, the teacher can save model responses to the corpus. Once students have completed the assignment, the teacher can begin by reviewing and scoring the work product of students who usually score full credit. Upon confirming that these are indeed full credit models, the teacher can click a button to add the student sample to the corpus of model answers. The software limits the teacher to five models in short answer testing and seven models in composition assessment.

Once trained, the teacher can run the scoring algorithm on each student submission. At this writing, processing takes about nine seconds on average per sample, depending on the text size. This program works best for assignments where there is a narrow range of full credit responses. Its primary purpose is to score writing samples by comparing to a limited number of full credit responses. Its strength is in recognizing similar meaning across texts in varying ways to say the same thing. This program does not assess spelling or technical / mechanical writing conventions, although it does rely on student accuracy for scoring to the extent that adherence to certain conventions are necessary for the program to operate. Examples: proper noun count requires that

[3] Precise details of this process and the scoring ranges are confidential for reasons of commercial viability.

students capitalize them; sentence count requires that students apply standard rules of punctuation.