Automated Essay Grading

Alex Adamson, Andrew Lamb, Ralph Ma

December 13, 2014

Abstract

Using machine learning to assess human writing is both an interesting challenge and can potentially make quality education more accessable. Using a dataset of essays written for standardized tests, we trained different models using word features, per-essay statistics, and metrics of similarity and coherence between essays and documents. Within a single prompt, the models are able to make predictions that closely match those made by human graders. We also explored methods of giving more detailed feedback for essays, such as levels of coherence and technical correctness.

1 Introduction

1.1 Data

We used essays provided for an automated essay scoring competition sponsored by the Hewlett Foundation. The data were divided into eight essay sets. The authors of the essays were American students in grades seven through ten. The essay sets had an average essay length between 150 and 650 words. Each dataset used a different prompt; some of the prompts asked for responses to source material while the rest asked students to respond to a short statement. Each essay was graded by at least two humans. Each essay set had a procedure for producing a final score if the two human scores disagreed, e.g. take the average, or use a third human score as a mediator. [1]

1.2 Measuring Agreement Between Graders

We used the Quadratic Weighted Kappa as our primary measure of how close predictions generated by the model were to human scores. Quadratic Weighted Kappa takes two equal length lists of grades ait's input, and outputs a score between -1 and 1, where -1 signals perfect disagreement, 1 perfect agreement, and 0 random agreement. [1]

2 Models

Our general pipeline involved extracting features from the raw essays, and iteratively training and using k-folds cross-validation on our model on selected essay sets in order to optimize hyperparameters.

2.1 Support Vector Regression

For each essay set, we featurized the essays and then optimized an ε-SVR via parameter sweep with C, the choice of kernel, and ε as free variables.

2.1.1 Features

- Word n-grams - N-grams tokenize the next and treat it as a "bag of words", where each feature is a count of how many times a word or combination of words appeared. We usually used unigrams. We applied the tf-idf transformation to the word counts.
- Part of speech n-grams - We used the Natural Language Toolkit part of speech tagger, and then used these tags and n-gram features. [2]
- Character counts
- Word counts
- Sentence counts
- Number of mispellings

- Reduced dimention term-vector - We used Latent Semantic Analysis (discussed below) as both an independent model and a method to reduce the dimention of the term-document matrix, which was then used as features in the SVM.

2.2 Latent Semantic Analysis

Latent Semantic Analysis is a method that attempts to find a set of concepts that run through a set of documents. LSA starts with a term-document matrix (a matrix where documents are represented along one axis, and counts of terms are represented along the other axis). We then use Singular Value Decomposition to represent the matrix as a product of two orthogonal matrices and a diagonal matrix. We can then remove the values of the diagonal matrix with the lowest magnitude, and use the resulting matrices to represent a term-document matrix with a reduced dimention term axis. By doing this, LSA is able to capture relationships between terms that are not equal, but have similar meanings or concepts, for example "dog" and "hound".

To make predictions with our reduced matrix, we can take the cosine similarity between documents, and then assign the document a score that is some combination of the k closest neighbors. Specifically, during cross-validation, we optimized the number of neighbors used, and whether the prediction was the weighted or uniform average of the neighbors' scores.

3 Results

To set a upper limit on our models, we first measured the disagreement between human graders, as a way to capture the inherently subjective nature of grading an essay. We did this by taking the Quadratic Weighted Kappa between the two human graders for each essay set. The values were between 0.629 and 0.819, signaling that human graders to agree with each other to a reasonably high level.

We found that the Support Vector Regression using all of the features described above except the reduced dimension term-vectors produced by LSA was able to make predictions that matched closely with human graders. During k-folds cross-validation (using 10 folds), we took the Quadratic Weighted Kappa between the predictions on the validation set produced by the SVR, and the resolved human scores on the validation set (the final human scores given to the essays). The Quadratic Weighted Kappa between the SVR predictions and human scores are close to or higher than the Quadratic Weighted Kappa on all of the datasets, meaning the model was able to agree with human graders quite closely. In some instances, the Quadratic Weighted Kappa of the SVR was higher than the humans - intuitively, the machine agreed more closely with the final human score than the humans agreed with each other.

Latent Semantic Analysis was less successful in producing predictions that agreed with human graders. On every dataset, the Kappa score between LSA predictions and resolved human scores is lower than the Kappa between humans graders and between SVR predictions. On some essay sets, it is significantly lower, for example, on essay set 8, LSA produces a Kappa score 0.243, signifying only a small level of non-random agreement with human scores.

We also tested a Support Vector Regression that used reduced dimension term-vectors produced by Latent Semantic Analysis. This produced agreement scores that were either close to those produced by SVR without the vectors, or between LSA and SVR without the vectors. However, it did not appear to offer significant improvement over SVR without vectors.

4 Discussion

4.1 Features

We wanted to learn which features were more useful in predicting the score of the essay. In order to do so, we removed features one at a time and ran the pipeline to calculate the kappa score. Figure 2 shows the results received from 4 different essay sets and the average. As seen, removing all word unigram/bigram features decreased the kappa score the most. Word count did not play as big of a factor as originally believed. Similarly, removing unigrams and bigrams of the part of speech caused only minimal decrease of the kappa score. Other features had minimal changes to the kappa score. From this result, we see that for most of the essay sets, the word unigram/bigram features were most significant in the regression. We wanted to investigate why unigrams/bigrams played such a big part in the regression process. In Figure 3, we calculated the average unique words per essay for essays belonging to each normalized score range. Then in Figure 4, we graphed the score ranges verse average unique words per essay for essays in the score range divided by the average unique words per essay for every essay in that whole essay set. As we can see, there is a linear relationship for each essay set in which the more unique words used, the better the score will be. This result explains why unigrams and bigrams were so important for our regression. Furthermore, it shows that number of unique words is a good feature to use.

Number of unique words of an essay might be a good indicator, because it measures the vocabulary level of the writer. We don't believe the correlation between score and average unique words is caused by a correlation between

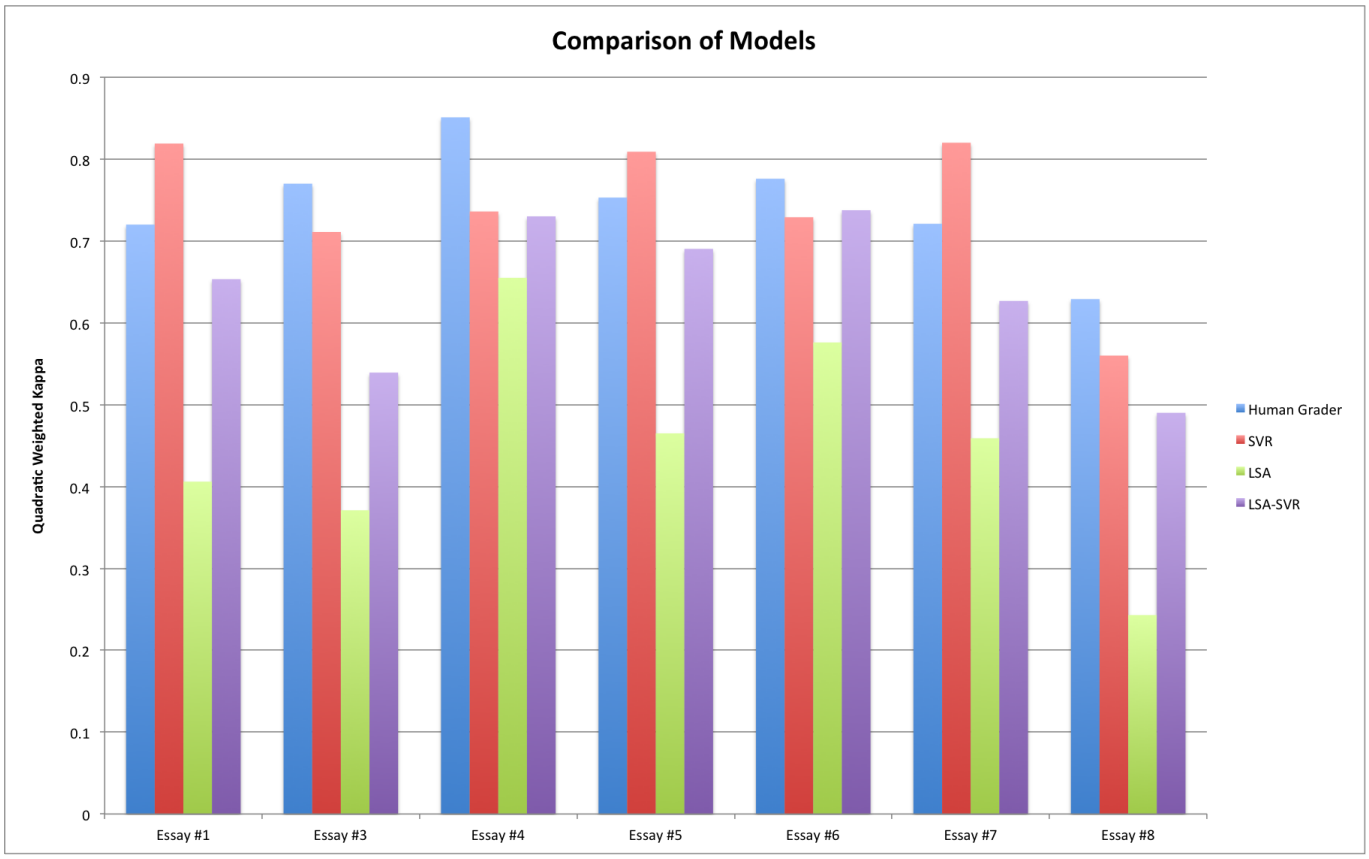

Figure 1: Model Results

Features	**Essay #1**	**Essay #3**	**Essay #6**	**Essay #8**	**Average**
Unigrams	-.24	-.037	-.022	-.49	**-.20**
wordCount	-.0031	-.0016	-.0066	-.0051	**-.0041**
tags	.0075	-.013	.038	-.036	**-.0009**

Figure 2: Change in Kappa due to Removal of Features

Score Range	**E1**	**E3**	**E4**	**E5**	**E6**	**E7**	**E8**
0-25	46	34.7	37.4	28.7	46.3	39.9	4
25-50	122.4	50.8	46.8	43.2	62.8	71.1	139.1
50-75	197.4	75.7	74.4	66.1	80.3	108.2	287.2
75-100 (inclusive)	279	108.8	101	103.9	112.4	167.3	364.3

Figure 3: Average Unique Words per Essay

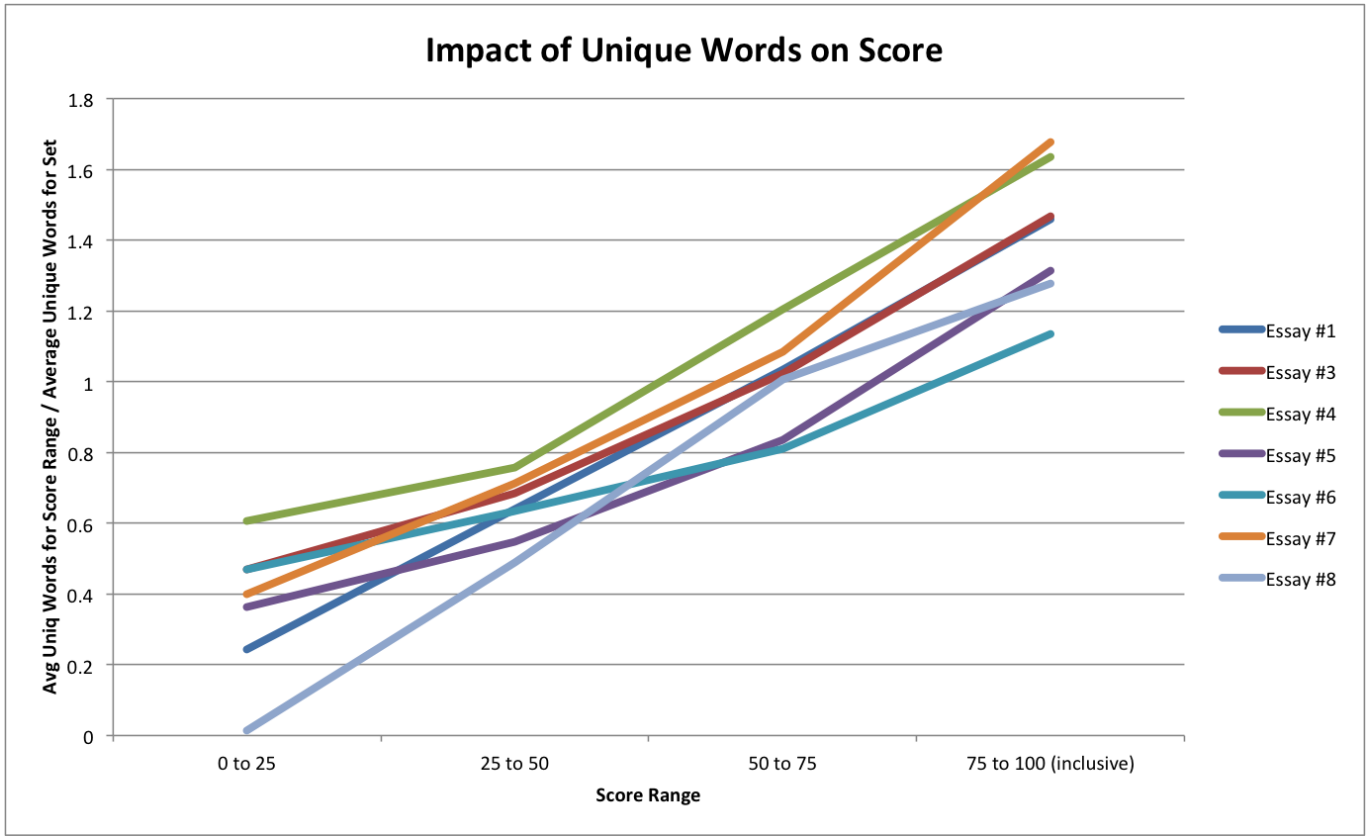

Figure 4: Unique Words and Score

score and length of essay, because, as seen in Figure 2, eliminating word count of an essay from feature vector does not affect the kappa score dramatically. We can speculate that for timed standardized tests, size of vocabulary applied in the essay is a good indicator of the sophistication of the writer. However, this correlation is limited by the characteristics of the data: timed, standardized exams, given to 7th to 10th graders with specific prompts. In the next section, we will discuss how to create a generalized predictor that is not as limited by the characteristics of the training set.

4.2 Coherence Analysis

In order to gauge the scalability of our system, we decided to experiment by training on essays of one prompt and then testing on essays of another prompt. A high kappa score would have indicated that our predictor was able to learn general traits of good essays rather than just learning traits that characterize good essays of a certain prompt/writer level. Training our predictor on data sets 3, 4, and 8 and then testing our predictor on data set 6, we received a Kappa score of .532. The low Kappa score showed that we were not capturing the characteristics of good writing as well as we wanted. A review of literature showed several features that better characterize good writing. Higgins, et al. proposed several methods for featurizing and predicting whether a sentence is coherent or not within the context of the essay and the prompt. They also found that essays that generally had a greater score would have a higher percentage of coherent sentences. Using similar methods, we decided to build a classifier that would be able to classify a sentence's coherence. The advantages of having such a classifers are two folds. Foremost, this classifier is not as prompt specific which means that we can train it on sentences from any essays based on the features described below. Second of all, we can use the percentages of coherent sentences in an essay as a feature for scoring essays.

To characterize the coherence of a sentence, we extracted the following features:

- RI score of target sentence with any sentence in prompt
- maximum RI score of target sentence with any other sentence in the essay
- sum of RI score of target sentence with the 2 previous sentences and 2 sentences after it.
- sum of RI score of sentence with all sentences in the essay
- number of sentences in the essay that has RI score higher than .2 with target sentence

- number of sentences in the essay that has RI score higher than .4 with target sentence
- number of sentences in the prompt that has RI score higher than .2 with target sentence
- number of sentences in the prompt that has RI score higher than .4 with target sentence

RI here represents Random Indexing score which is a 0 to 1 score of similarities between sentences after applying dimensionality reduction through Latent Semantic Analysis (Kanerva et al., 2000). We trained a svm based on sentence coherence of 312 sentences from data set 1 scored by 2 of the authors of this report. We then tested on 100 sentences from prompt 8 essays and received a agreement score (percentage of predictions that agreed with human graded sentences) of .49 which is both worse than random guessing (.5) and the baseline of simply marking all sentences as incoherent (.62). However, running 5-fold within prompt 1 sentences, we received an agreement score of .58 which is higher than the baseline of .52. Future projects can build on top of the pipeline that we have built by extracting features that better captures the coherence of sentences.

5 Conclusions

Using Support Vector Regression and cross-validation to optimize the hyperparameter C, we achieved Kappa agreement scores that matched agreement scores between two human graders. Our research shows that for essays of intermediate writing level (7-10th grades) and given enough human graded training examples for a writing prompt, we can automate the grading process for that prompt with fairly good accuracy. In our discussion, we talked about efforts on breaking these barriers by attempting to featurize characteristics for good writing in general, such as sentence coherence. We attempted to build a classifier for this task, with limited success.

References

[1] *Develop an Automated Scoring Algorithm for Student-written Essays.* Kaggle. 10 Feb. 2012. https://www.kaggle.com/c/asap-aes.

[2] Bird, Steven, Edward Loper and Ewan Klein (2009), *Natural Language Processing with Python.* O'Reilly Media Inc.

[3] Higgins, Derrick, Jill Burstein, Daniel Marcu, and Claudia Gentile. "Evaluating Multiple Aspects of Coherence in Student Essays." In HLT-NAACL, pp. 185-192. 2004.

[4] Kanerva, Pentti, Jan Kristofersson, and Anders Holst. "Random indexing of text samples for latent semantic analysis." In Proceedings of the 22nd annual conference of the cognitive science society, vol. 1036. Erlbaum, 2000.