Project Report: Automated Essay Scoring

1. Introduction

The rapid growth of technology and the emphasized importance for education equality has inevitably led to the marriage of the two concepts. Today, there is a growing interest in innovative instruction delivery and support systems that would make education more accessible. Many outstanding Massive Open Online Courses (MOOCs) such as Coursera and edX are already doing a great job in replacing the role of a physical lecture, but there is still a great demand for an online grading system, especially for essays. Manually grading essays is time consuming and not applicable to MOOCs where the students to instructors ratio is much higher than that in physical classes. If we can automatically grade essays or at least provide a ballpark measure of each essays grade (classify them into certain grade ranges), then we will be able to save a huge amount of time and effort. In this report, we will talk about our proposed automated essay grading system. Among various essay grading criteria, we focused on surface features, which are statistical values including total number of distinct words, percentage of spelling errors, and part of speech distribution, and on-topic-ness of the essay, which we defined as how well the essay answers the given prompt. To measure on-topic-ness of the essay, we computed the relative weight of each word in an essay to the corpus, all essays in the same essay set, using tf-idf, and calculated the essays topical distribution of pre-generated list of topics through Latent Dirichlet Allocation(LDA). Representing each essay with these features in a vector form, we used KNN and SVM regression test to measure the accuracy of each essay set. Contrary to what we initially believed, surface features were more effective than tf-idf or LDA.

2. Problem Definition and Methods

2.1. Task Definition

First, we needed to come up with a feature function such that given an essay, a feature function will output a feature vector that would represent the essay (If E_p is a set of essays for prompt p and X is feature space, we wanted to come up with $f : E_p \to X$). Then, given such a representation of an essay in the form of feature vector, we wanted to be able to predict the score of the essay (we wanted to compute $h : X \to Y_p$, where Y is the set of possible scores for prompt p): because score is a continuous measure, we modeled the problem as a regression problem. The above task is very interesting, as delving into different feature functions will provide insight as to what properties really constitute an essay and how these properties actually affect the score of an essay. Even with such insight, the students can focus on these properties to improve their essays, and the graders can mainly focus on these properties in order to quicken the process of grading an essay.

2.2. Algorithms and Methods

There are 3 components to our feature function, essay statistics, tf-idf vector, and topical distribution through LDA. Essay statistics involved total number of words, total number of unique words, average lengths of words, percentage of spelling errors, and then finally a part of speech distribution within the essay. As for tf-idf, prior to calculating tf-idf vectors for the essays, we used Porter Stemmer to tokenize the words and removed stop words to get more refined tf-idf vectors. Finally, we first ran LDA on each essay set corpus to k topics (we used $k = 20$), where a topic is simply a distribution of words. Then, with the obtained topics, we were able to come up with topic distribution for each essay. Also, to improve the topical coherency of LDA, we tried various methods. We manually expanded the stopword list, filtered out words whose appearance is below or above certain threshold, used univariate statistical tests to filter out statistically unimportant words, and finally fixed spelling errors to get the correct word. Then, we concatenated these components in different combinations to obtain the feature vectors for all the training data and the test data. Also, in order to have equal weight for each component, we normalized each component to 1. Then, we used KNN algorithm and SVM to predict the score of the test essays.

3. Experimental Evaluation

3.1. Methodology

3.1.1. Error Metric

Because we modeled our problem as a regression problem, we cant simply use accuracy (number of correct predictions / size of test set). Hence, we had to use

mean absolute error and mean squared error, which will both tell on average how much was our prediction off from the actual value. In other words, by coming up with a model that has very low mean absolute and mean squared error (our prediction is not that much off from the actual score), this model could be used to give a grader a general idea of what the grade of the essay should be even before reading through the whole essay and also give automated-feedback to the students. We also used a dummy regressor (predicting the mean of the training data) as a baseline to make sure that our models are actually doing something.

3.1.2. Experimental Methodology

First, we divided up the dataset into training set and test set with the proportion of 0.8 and 0.2 respectively. After coming up with a different components for the feature vector, we tried a bunch of different combinations of the components to see which component was the most effective and see which pair of components complement each other. Because we had 3 different components to the feature vector, there were a total of 7 possibilities in terms of combinations of components (excluding the case where none of the components is used). Also, after trying different methods in order to improve the LDA result, we looked through the word distribution for each topic in order to see if the topical coherence increased.

3.1.3. Performance Data

1. Mean absolute error mean square error for different combinations of components in the feature vector on KNN and SVM

2. After obtaining the topics of the corpus through LDA, we manually looked through the word distribution in the topic to check the coherence in the topics.

3. In order to validate our assumptions about the relationship between the scores and certain essay statistics, we made a graph with X-axis being the scores and the Y-axis being the specific statistic.

3.2. Results

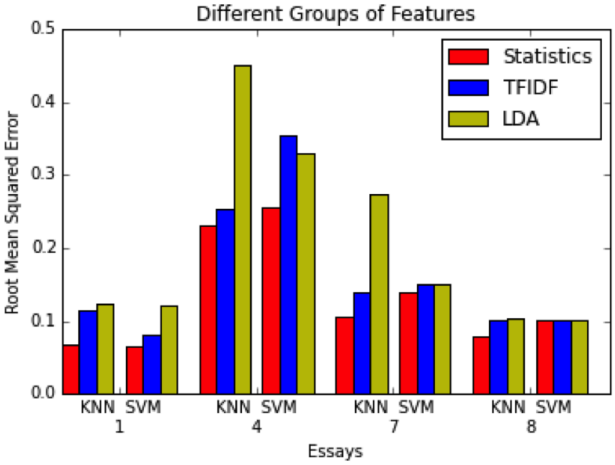

Figure 1. graph

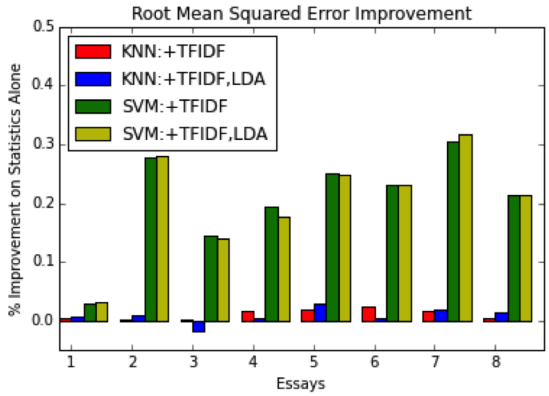

Figure 2. graph

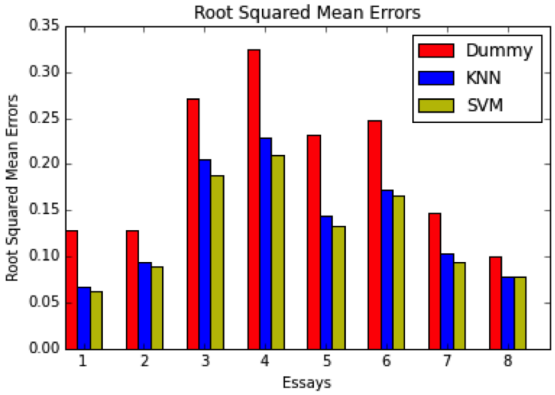

Figure 3. graph

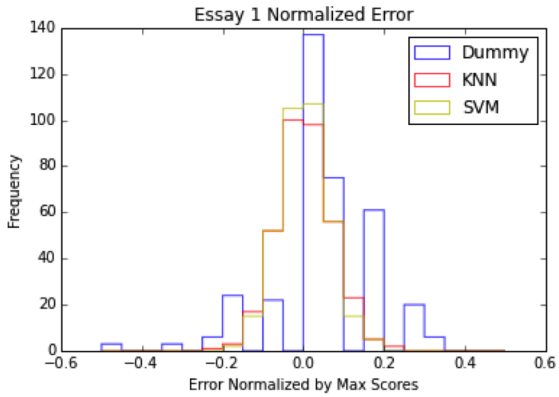

Figure 4. graph

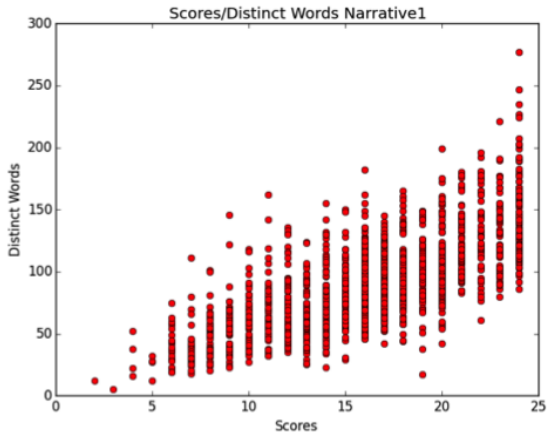

Figure 5. graph

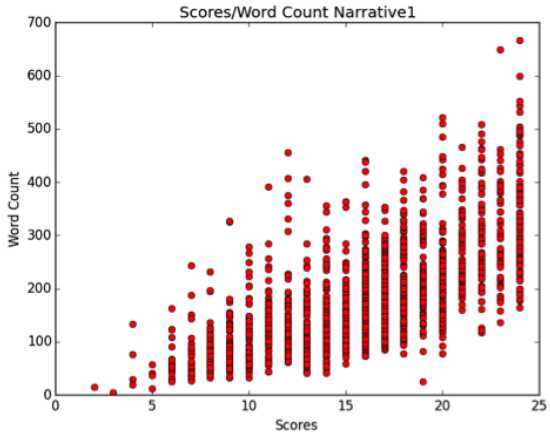

Figure 6. graph

Accuracy was measured by looking at the error between the predicted score and the actual score. Errors were normalized with respect to the maximum possible scores for each essay. A Dummy model which predicted the median of the score distribution was used as a baseline for comparison. Figure 3.2.1 shows the accuracy of KNN and SVM for different groups of features. Figure 3.2.2 shows the percentage improvement in root mean squared error when compared to using statistical features only. Figure 3.2.4 represents the distribution of errors for essay 1, the distributions were similar for the other essays.

3.3. Discussion

Overall, the combination of all three components, essay statistics, tf-idf vectors, and topic distribution through LDA, performed the best when used with SVM regression prediction. Interestingly, essay statistics alone provided much of the predictive power of our model. Figure 3.2.2 shows that for KNN regression, the addition of tf-idf and topical analysis contributed very little to the overall accuracy. Although the addition of tf-idf and topics improved accuracy significantly for SVM, statistics still provided very impressive predictions as seen in Figure 3.2.1. The reason statistics were so powerful can be seen in their strong correlations with score. As shown in Figure 3.3.1 and 3.3.2, there was a strong correlation between the score and number of words used in an essay.

On the other hand, LDA was not as effective as we initially conjectured. In most of the essay sets, LDA vector had the highest mean square error and mean absolute error among other single component vectors. This may in part due to the way we pre-generated the topics. We used each essay set as a corpus to generate a list of topics. However, most of these topics did not make an intuitive sense. Thus, the topic distribution was not doing its job of representing each essay as a collection of logically sound topics, but instead, it was serving a similar purpose as tf-idf, coming up with the relative weight of each word in an essay to the corpus.

Generally, SVM showed better accuracy results than KNN did. This is due to high dimensionality and sparseness of the vector. Vector representation using tf-idf and LDA returned a large number of features (10000) and high proportion of nonzero entries. As a result, the distance between each vectors all resulted in similar values, and thus KNN was not as effective.

We believe that the major portion of inaccuracy came from the lack of structural information of the essay in our vector representation. The vector represented the various uses of words and their relative importance in the essay, but did not take the order in which the words appear into account.

4. Related Work

ETS uses an automated essay scoring machine called E-rater to score standardized TOEFL essays. It builds new models for each essay prompt by evaluating 270 training essays, and extracts relevant features based on the predictive feature set obtained by linear regression. Its scores for essays exactly match with human graders or off by one 92% of the time [1].

EdX, a massive open online course developed by Harvard and the Massachusetts Institute of Technology, also incorporates an automated essay scorer called Discern. It requires human graders to grade 100 essays by hand to use the essays as training examples [2].

5. Future Work

One major problem that we have in our current methods is that the order of words in essays is not taken into account. Since each essays score is calculated using tf-idf, essay statistics like number of words, and topic distribution, it is not possible for the current model to recognize difference between two essays that are vastly different but have the same bag of words. In order to mitigate this problem, it is necessary to take the order of words in essays into account. We can use the order of part of speech tags in essays as another feature when classifying. Then, the similarity function will be able to recognize the difference between two essays that have the same words in different orders.

Another issue with the current setting is that the number of essays in each essay set is too small(1500 train examples) to train a good LDA model. Because of this issue, there wasnt that much coherency in the topics that we acquired. The topic distribution of an essay extracted by using the model is less effective than other features when predicting the score, especially for the less-complicated essays that are in the first and second essay sets, which can be seen in figure 3.2.1.

To fix the problem of not having enough essays in each essay set, we obtained pre-trained model for word2vec that were trained on a corpus of Google News dataset, with about 100 billion words: this model used deep learning with the idea of continuous bag of words and skipgram architecture, allowing us to calculate similarity measure between two sets of words. The model consists of 300-dimensional vectors for 3 million words and phrases. Viewing each essay as a set of words and using the similarity function given by the model, we implemented KNN. With this model, the mean square error and mean absolute error came out to be (1.175, 2.40) on essay set 1 and (4.12, 26.22) on essay set 8, which was still worse than our current model. In future, we can also train another LDA model that uses a big corpus like Google News or the entire English wikipedia and see if it predicts better than the other model. It would also be ideal to have more essays collected for each essay set so that the topic distributions drawn from the corpus can represent an essay more clearly.

6. Conclusion

Overall, using a combination of statistics, tf-idf and topical features, we were able to predict essay scores with significantly higher accuracy than the baseline of predicting the median. Observing how well the features did separately, however, provided interesting insight into the strong predictive abilities of statistics on essays and the difficulties with using LDA for topical distributions. The corpus of essays themselves turned out to be a poor source of topics mainly because it was too small so a larger separate corpus was used. Still, LDA topics did not improve accuracy when combined with tf-idf features suggesting they ultimately represented similar information. Looking forward, more structural features could add greater diversity to the features used and potentially improve predictions. The dominance of statistics in predicting essay scores perhaps also reveals that because graders have so many essays to grade, they will try to use easily discernible features, like word count, use of vocabulary, and grammar, things that can be represented by statistics, to assign scores. This idea both furthers the case for automated essay scoring because graders are already analyzing essays to pick out certain features and also motivates their use. By using automated essay scoring to provide one of two scores for an essay, graders will have a smaller workload and can focus on looking at literary aspects of essays that cannot be gleaned through statistics.

7. References