Automated Essay Scoring

Alen Lukic and Victor Acuna

Rice University

1 Abstract

The purpose of this project was to build an automated essay scoring system. We used a set of 12,978 essays, subdivided into 8 sets, provided on Kaggle.com by the William and Flora Hewlett Foundation as part of a competition. The essay set was split into a training set of 8,652 essays and a validation set of 4,326 essays. We extracted several features of the essays, ranging from simple numerical data such as word count and average word length, content-specific numerical data such as misspelled word count and adjective count, and structural data such as average degree and diameter of the essays' noun-verb relatedness graphs. We used a simple naive Bayes predictor and the closed-form solution to linear regression. In addition to the fraction of exactly correct predictions t, we used average percent error n_ϵ over the eight essay sets as a correctness measure. The closed-form solution approach outperformed the naive Bayes prediction slightly but not conclusively, yielding $n_\epsilon = 9.52\%$ and $t = 50.18\%$. The Kaggle competition used the quadratic weighted kappa error metric to measure the agreement between the predicted scores and actual scores on a separate validation set of 4,258 essays, with $\kappa = 0$ indicating no agreement and $\kappa = 1$ indicating complete agreement. The closed-form solution approach yielded $\kappa = 0.63631$. Comparatively, the best submission to the competition yielded $\kappa = 0.80135$.

2 Introduction

An automated grading system can serve as a useful assistive tool for instructors. If the system has a high degree of accuracy, instructors can use it to compare grades assigned to student essays by the system to those assigned by human graders. This would provide a means of ensuring consistency and fairness. A particularly sophisticated grading system could save vast amounts of time and effort by eliminating human graders altogether; in a situation such as this, the system would assign preliminary grades to all student essays, and the instructors would only become involved in the process to address student disputes and regrade requests.

The William and Flora Hewlett Foundation presented provided the data necessary to build an automated essay system on the machine learning and data mining competition website Kaggle [1]. The data consists of training and validation essay sets. The training set is presented in spreadsheet form. Each row in the spreadsheet consists of the following columns:

an essay ID, a set number (there are 8 different essay sets), the essay itself, the score given to the essay from 2 graders, the resolved essay score, and for essay set 2, the score given for following language conventions (spelling, grammar, etc.). The validation set contains only the essay ID, set number and essay.

We use and compare two methods for predicting scores for the essays in the validation set. The first is a simple naive Bayes prediction, which assumes that the scores are independent of each other. The model is trained on the features of the training essays and predicts scores for the validation essays given their features.

The second method we use is the closed-form solution to linear regression. The design matrix X is defined such that each row represents an essay in the training set and each column represents the score given to some feature of the essay. The target vector y_{train} contains the corresponding resolved score given to the essay. We calculate the parametrization w from the closed form solution $w = (X^T X)^{-1} X^T y$. We use this value of w to predict the scores of the essays in the validation set by simply using the equation $Xw = y_{pred}$.

3 Hypothesis

We will use three scoring metrics to measure accuracy. The first is simply the percentage of exactly correct predictions t. The second is an average percent error n_ϵ, which averages the difference between the predicted score and actual score for the validation essays in each set, divides this average by the number of total possible scores in the set, and then finds the average value of these percent errors. The third metric is the quadratic weighted kappa error κ and is used by the Kaggle competition [2]. This metric measures the agreement between the predicted scores and actual scores, with 0 indicating no agreement and 1 indicating complete agreement. We believe that we can attain values of $t = 30\%$, $n_\epsilon = 20\%$, and $\kappa = 0.6$, and that the closed-form solution to linear regression model will outperform the naive Bayes model. We believe that the following numerical essay features will be good predictors of essay score:

- Word count, character count, and average word length
- Misspelled word count, adjective count, and transition/analysis word count
- Total occurrences of words in the prompt in the essay
- Sum of KL divergence between word pair cooccurrence probability and corpus probability

Additionally, we think that the following structural properties, derived from constructed noun-verb relatedness graphs for each essay:

- Average node degree of graph's three highest node degrees
- Graph diameter

4 Methodology

4.1 Feature extraction

We used a Python natural language processing package, NLTK, to tokenize essays, strip the essays of punctuation and stop words (which was performed prior to any feature extraction), and for part-of-speech tagging. The following numerical features were straightforward to extract from the essays: word count, character count, average word length, number of transitional and analytical words, number of misspelled words, number of adjectives used, and the number of words found in both the prompt and the essay.

For each essay, we also formulated graphs representing noun-verb pairs within sentences. Each time a noun and a verb appeared together in a sentence and appeared to relate to one another, which was determined by their adjacency in the sentence, the noun was connected to the verb in the graph. A noun could connect to many verbs in the graph (and vice-versa), and the edges were weighted by the number of times a noun-verb pair appeared in the essay. From these graphs, we drew the average node degree (of the three highest degrees in the graph), where node degree is defined as the sum of the edge weights connecting to that node, and the diameter of the graph, which is defined as the longest shortest path between any pair of nodes in the graph. These measurements taken from each essay's graphs were meant to indicate noun usage within essays. We believe that a high average node degree of the three highest-degree nodes in the graph indicates topicality; that is, an essay concentrated on a few particular topics rather than a discursive, off-topic essay. Similarly, we believe that a higher diameter indicates topical variation; that is, discussing different aspects the same subject or discussing similar aspects of different subjects. Together, these two measures provide an indication of cohesiveness.

Additionally, we used the sum of KL divergences between cooccurring word pair probability and corpus probability in the essay as a measurement of topicality. KL divergence is a measure of difference between two probability distributions. For each essay set, we selected the $N = 200$ most frequently occurring words within the set. Call this mapping of word to frequency $f(w_i)$, and define the corpus probability p_c of w_i as $\frac{f(w_i)}{\sum_N f}$. We also selected a "window size" for the set, which was defined for each essay set as roughly one-sixth of the average word count. Since the essays were not delineated by paragraph, this window size aimed to approximate them. The notion of window size and word pair associativity are drawn from the paper by Hassan and Mihalcea [3]. Call the number of unique word pairs M. For each pair of unique words, we counted how frequently they occurred within adjacent, non-overlapping windows within each essay. Call this association of word pair to frequency $c(w_1, w_2)$ and define the cooccurrence probability p_t of (w_1, w_2) as $\frac{c(w_1,w_2)}{\sum_M c}$. Upon calculation of these frequencies, each essay was partitioned into adjacent, non-overlapping window sizes. Each time a word pair in c appeared in one of these partitions, the KL divergence was calculated as follows:

$$d = p_t \ln(\frac{p_t}{p_c}) + (1 - p_t) \ln(\frac{1-p_t}{1-p_c})$$

Word pairs whose cooccurrence was high relative to cooccurrence of other word pairs and the individual occurrence of the words in the corpus had a higher KL divergence. Essays with higher KL divergence sums tended to contain word pairs which tend to cooccur often, indicating a topical association between the words.

4.2 Training and Validation

We selected to compare two models for training and validation: a naive Bayes predictor and the closed-form solution to linear regression. We selected naive Bayes as a baseline model to compare with the closed-form model. We selected the closed-form model for the following reasons: (1) the underlying feature data is static; (2) incorporating additional features into the model is straightforward; (3) scaling is unnecessary; and (4) computational complexity is small given the size of the data and number of features.

The naive Bayes predictor is trained by calculating the prior probability of each score and the conditional probability of each score given the respective essay features are calculated from the training essays. These probabilities are then used to predict the most probable score of the validation set essays given their respective features.

The closed-form solution to linear regression model is trained by using the equation $w = (X_{train}^T X_{train})^{-1} X_{train}^T y$ to calculate the model prediction parametrization, where X_{train} is the feature matrix of the training set and y is the target vector of corresponding essay scores. Validation essay scores are predicted by the equation $X_{valid} w = y_{pred}$.

We trained and validated with both models first using only word count as a benchmark, and then using the rest of the essay features.

4.3 Evaluation

We selected two metrics to evaluate the goodness of our predictions: the fraction of exactly correct predictions t and the average percent error n_ϵ. Over the eight essay sets we calculated $n_\epsilon = \frac{1}{8} \sum_{i=1}^{8} \frac{1}{C_i N_i} \sum_{j=1}^{N_i} |a_j - p_j|$, where C_i is the number of possible classifications (scores) in the essay set, N_i is the total number of essays in the validation set, a_j is the actual score of a validation essay, and p_j is the predicted score of a validation essay. The third metric, used by Kaggle, is the quadratic weighted kappa error κ, where $\kappa \in [0, 1]$, with a higher κ value indicating greater agreement between actual and predicted scores. We only obtained a κ value for the predictions determined with the closed-form solution to linear regression model.

5 Results

Using the benchmark feature (word count) only, the naive Bayes predictor yielded $n_\epsilon = 11.03\%$ and $t = 57.28\%$, and the closed-form predictor yielded $n_\epsilon = 11.20\%$ and $t = 54.49\%$.

Using all of the features, the naive Bayes predictor yielded $n_\epsilon = 11.76\%$ and $t = 56.03\%$, and the closed-form predictor yielded $n_\epsilon = 10.55\%$ and $t = 56.97\%$.

Note that the above data were only determined across essay sets 1 through 6, as both sets 7 and 8 contained negative within-class variances between the word count feature and some subset of the classes, precluding the use of the naive Bayes predictor. Across all of the essay sets, the closed form predictor yielded $n_\epsilon = 9.53\%$ and $t = 50.18\%$.

Benchmark (Essay Sets 1-6)		
Model	n_ϵ	t
Naive Bayes	11.03%	57.28%
LR: CF	11.20%	54.49%

Full Results (Essay Sets 1-6)		
Model	n_ϵ	t
Naive Bayes	11.76%	56.03%
LR: CF	10.55%	56.97%

Full Results (All Sets)			
Model	n_ϵ	t	κ
LR: CF	9.53%	50.18%	0.6363

The naive Bayes predictor slightly outperformed the closed-form model in the benchmark test. The quality of the naive Bayes predictions decreased with the addition of features other than word count, whereas that of the closed-form model increased. The closed-form model slightly outperformed the naive Bayes model with the addition of the remaining features. Across all of the essay sets, we see that the average percent error of the closed-form model drops to 9.53% from 10.55%. Though t does decrease, this can be accounted for by the fact that essay sets 7 and 8 are graded on a 30 and 60 point scale, respectively, which is much more variable than the scales of the essays from the other sets.

These results are in line with the hypothesis. For the closed-form model, the n_ϵ is approximately 10.5% better than hypothesized, the t value is approximately 20% higher than hypothesized, and the κ value is approximately 0.0363 higher than hypothesized. However, the closed-form model does not outperform the naive Bayes model as distinctly as predicted. The goodness of this model compared to simple naive Bayes prediction for the problem of automatic essay grading is inconclusive.

There were numerous sources of error in our feature extraction process. The NLTK tokenizer, parts-of-speech tagger, and spell-checker are all imperfect, leading to noise in any features which rely on these tools. The association of noun-verb pairs was similarly an imperfect process; some pairings were likely spurious. The essays also censored nouns which were considered to be personally identifying information. We stripped these censored nouns from the essay corpus, and this also likely affected the noun-verb pairing process.

6 Conclusion and Future Work

Although our closed-form prediction model exceeded our hypothesized expectations, its improvement over the predictions obtained from both naive Bayes and the closed-form model using word count as a benchmark feature was modest at best. It is inconclusive from our data whether the linear regression closed-form solution is a better predictor of essay scores than a simple naive Bayes predictor or whether it is an appropriate model to use to address this problem at all. A prime area to explore in the future is the use of other prediction models for essay score classification (for example, support vector machines for classification). Another possibility for improvement is the extraction of more complex essay features. For example, the noun-verb relatedness graphs which we constructed may have been too local, as they only connected nouns and verbs if they were the respective adjacent parts of speech to each other in a sentence. Another feature which could be added by extending the notion of window sizes is a measure of relatedness between adjacent window sizes within essays; such a feature could be used to measure the relatedness of adjacent sentences or paragraphs.

7 References

1 ”The Hewlett Foundation: Automated Essay Scoring” [online] 2012, http://www.kaggle.com/c/asap-aes (Accessed: 25 April 2012)

2 ”Evaluation” [online] 2012, http://www.kaggle.com/c/asap-aes/details/Evaluation (Accessed: 25 April 2012)

3 Hassan, Samer and Mihalcea, Rada. ”Semantic Relatedness Using Salient Semantic Analysis” [online] 2012, http://www.cse.unt.edu/ rada/papers/hassan.aaai11.pdf (Accessed: 25 April 2012)