Automated Essay Grading

Abhijeet Kumar and Anurag Ghosh and Vatika Harlalka

***Abstract*—We built an Automated Essay Grading System to grade 5 essay sets of around 1500 essays in each set provided by the Hewlett Foundation, each with diverse topics and written by different age groups. We extracted numerous features from our data set, including but not limited to, n-grams, word features, age of acquisition, parts of speech, sentence features etc.Graph Diffusion using Heat Matrices and Linear Regression were implemented to evaluate our model along with Support Vector Machine, Support Vector Regression and Kernel Ridge Regression to compare between various advantages and disadvantages of various classification techniques. Our model was able to make predictions that closely match those made by human graders, with accuracy computed according to the quadratic kappa metric which is the industry standard.**

***Keywords*—Essay, Grading, Scoring, NLP, Heat Diffusion**

I. Introduction

Essay are a very important tool to assess the ability of an individual with respect to academic knowledge, proficiency in a language and to cultivate a style of writing among students. It is thus an integral part of a students' education, however, manual evaluation of essay is inconvenient, expensive, time consuming and at times, subjective in nature.

The aim of this project, thus, is to learn about various classification techniques and develop a model to grade essays. These grades from the automatic grading system should, at least, agree to the extent by which human graders agree among themselves.

Many organisations have already opted for automated essay grading systems to decrease subjectivity and increase efficiency.

II. Dataset

The training and test dataset was acquired from a past competition from Kaggle.com sponsored by Hewlett Foundation. The 5 sets have approximately 8000 essays ranging from 150-550 words each provided. We split the essays for performing randomized K-Fold Cross Validation using $k = 5$ on all the Essay Sets.

III. Feature Set

We used python libraries, NLTK, py-enchant, scipy-kit, to extract features from our dataset. These include:

1) **Numerical features**: These include n-grams, average word counts, sentence counts, the number of words of different character lengths, number of sentences of different word lengths. For feature extraction, the essays were tokenized and split using python utilities. The individual tokens were then used to compute Numerical Features.
2) **Maturity features**: We used scores indicating the maturity of essays like the number of spelling errors (, average age of acquisition of words, Beautiful word score.
   a) **Average Age of Acquisition** - We try to establish the average age of acquisition by computing the age of acquisition of the most commonly used words. A higher age of acquisition means a richer vocabulary.
   b) **Beautiful Word Score** - We use a model which calculates the average beautiful word score, which is computed for words of length greater than equal to 5, where we compute the product of frequencies of letters (the frequencies come for a very general set of sentences) to come up with the score. (High perplexity-letter model)
3) **Semantic features**: These include Parts of Speech statistics like number of Nouns, Verbs, Adjectives, Bag of Words score along with Sem=ntiment Scores. These features can also be taken as a rudimentary proxy for diction. Essays were tokenized into sentences before the tagging process.
   a) **Sentiment Score** - We found the average objectivity and sentiment using a Naive Bayes classifier trained on a very generalized Movie dataset.

IV. Classification Techniques

We used the following classification techniques to implement our automated essay grading system.Accuracies are plotted later in a comparison study.

A. Linear Regression

We implemented our own linear regression model which fits a classical linear equation $Y = W^T X + W_o$ to our training set which minimizes the least squared error so as to estimate the closest matching grade. The cost is given by the following equation,

$$C = \frac{1}{2} \sum_i (y_i - W^T x_i)^2 \tag{1}$$

B. Kernel Ridge Regression

We implemented Kernel Ridge Regression which is given by the following formulation $Y = W^T X$ while we minimize the following cost function

$$C = \frac{1}{2} \sum_i (y_i - W^T x_i)^2 + \frac{1}{2} \lambda W^T W \tag{2}$$

where λ is a parameter and the second term is the regularization 'weight-decay' which prevents over fitting. Differentiating it with respect to W and equating to zero gives us,

$$W = (\lambda I + \sum_i x_i x_i^T)^{-1}(\sum_j y_j x_j) \tag{3}$$

We map the linear space to a kernel space, $x_i \rightarrow \phi_i = \phi(x_i)$ giving us the following equation,

$$W = (\lambda I_d + \phi\phi^T)^{-1}(\phi y) \tag{4}$$

Here it is evident that $K = \phi^T\phi$, also as we classify by using, $Y = W^T\phi(X)$ which becomes,

$$Y = y(\phi^T\phi + \lambda I_n)^{-1}\phi^T\phi(x) = y(K + \lambda I_n)^{-1}K(x) \tag{5}$$

eleminating the explict requirement of the mapping function and thus enabling us to employ the kernel trick.

One big disadvantage of the ridge regression is that we dont have sparseness in the W^T = $y(K + \lambda I)^{-1}$ vector, i.e. there is no concept of support vectors like SVM or SVR.

C. Support Vector Classification

We have used the C-SVM implementation from scikit-learn python library which solves the classification problem $Y = W^TX$ by minimizing the following error function.

$$E = \frac{1}{2}W^TW + C\sum_i \zeta_i \tag{6}$$

subject to the constraints

$$y_i(\langle W, x\rangle - b) \geq 1 - \zeta_i \quad \forall i \in \{1..n\} \tag{7}$$

The parameter C was tuned to improve the accuracy of the classifier using the hill climbing method. The minimization is done by introducing Lagrangian multipliers and eliminating all other variables.

D. Support Vector Regression

This is very similar to the Support Vector Classification as explained above. We used the ϵ-SVR implementation in scikit-learn python library and trained it through parameter estimation of C. The regression is solved by minimizing

$$E = \frac{1}{2}W^TW \tag{8}$$

subject to the constraints

$$(y_i - \langle W, x\rangle - b) \leq \epsilon \quad \forall i \in \{1..n\} \tag{9}$$

and

$$(\langle W, x\rangle + b - y_i) \leq \epsilon \quad \forall i \in \{1..n\} \tag{10}$$

The parameter C was tuned to improve the accuracy of the classifier using the hill climbing method. The minimization is done by introducing Lagrangian multipliers and eliminating all other variables.

E. Heat Diffusion

We implemented our own spectral analysis method involving heat matrices. The method can be compactly described through the following equations. We find the laplacian of the graph by

$$L = I - D^{\frac{-1}{2}}WD^{\frac{-1}{2}} \tag{11}$$

(where D and W are degree matrix and adjacency matrix respectively) and then decompose the laplacian matrix into its corresponding eigenvectors (U) and eigenvalues (λ).

$$L = U\lambda U^T \tag{12}$$

We then take the q smallest non zero eigenvalues and corresponding eigen vectors, and form the heat matrix over different scales (t).

$$H(t) = U_q exp(-\lambda_q t)U_q^T \tag{13}$$

Now, f is the initial distribution of the grades and the grades are obtained for time scales by the follow equation.

$$Y(t) = H(t)f \tag{14}$$

We thus used a transductive setup in this section to perform the graph diffusion.

Different similarity measures were used to formulate graphs edges assuming each essay from essay set to be a vertex. Of all the measures we tried, RBF kernel was found to be the best one. The Graph itself is formulated in two ways considering the k nearest neighbours of a vertex designated to have edges, first is a dense formulation with a high value of k and second is the sparse formulation considering a low value of k.

Class problem is a major issue in our data set as few grades are very sparsely represented. Multi-Scale Heat Diffusion is known to solve this problem by employing the use of Heat matrices at different scales and the initial label distribution.

In our setup we have diffused the values at different scales(varying t at reasonable steps) and to merge the values found at different scales where low scale represents the local neighbourhood information and the high scale represents the global neighborhood information, we use different voting procedures such as average,stochastic and exponential.

V. Performance Measure

The standard agreement function that is widely used to calculate agreement between graders is the Quadratic Weighted Kappa, also known as the Cohen's Kappa. Given a list of grades (integral and constrained) by each grader it generates an output between -1 and 1, -1 indicating that there is perfect disagreement, 1 indicating that there is perfect agreement and 0 indicating agreements in specific instances, if any, were random in nature. [1]

We start by computing the confusion matrix, say C from the two grade lists where each element $c_{i,j}$ represents the number of times the first grader gave grade i while the second gave grade j for the same essay.

The Observed Accuracy is then calculated, as the number of times both graders agreed divided by the total number of instances. Or we can succinctly put it as,

$$p_o = \frac{Tr(C)}{\sum_i \sum_j C[i,j]} \tag{15}$$

The Expected Accuracy is then calculated, which is the product of the marginal frequencies of an awarded grade for both the graders, summed for each grade which is then divided by total number of instances, ie,

$$p_e = \frac{\sum_k \sum_i C[k,i] * \sum_i C[i,k]}{\sum_x \sum_y C[x,y]} \tag{16}$$

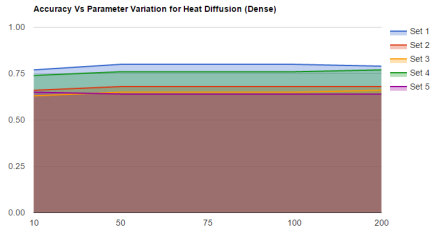
Lastly, Kappa metric is calculated as,

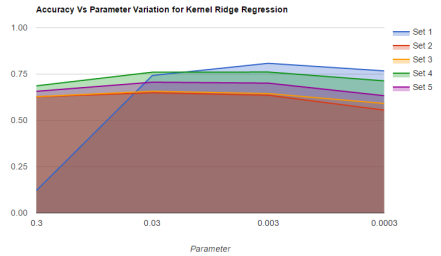
$$\kappa = (p_o - p_e)/(1 - p_e) \tag{17}$$

VI. Results

Here we present the results we obtained from the techniques used above measured by the Quadratic Weighted Kappa metric. The Quadratic Weighted Kappa between the 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 was higher than the humans - intuitively, the machine agreed more closely with the nal human score than the humans agreed with each other.

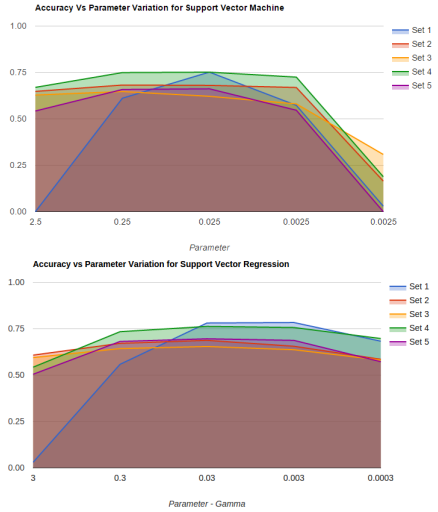
Firstly we estimated the parameters for all the classifiers we trained using a hill climbing method. It is notable that the graph diffusion method showed the least variation when parameters were changed, while the Kernel Ridge Regression method showed the largest variation in the acceptable range.





The SVM and SVR showed similar variation levels in the accuracy which is pretty easy to understand because they are very similar to each other. Heat Diffusion method seems pretty impervious to changes in parameters and Kernel Ridge Regression seemed to be constant too except for Set 1.

We have then used K-Fold Cross Validation (with k = 5) to evaluate the dataset with all the classifiers (with same optimal parameters over all sets). We used the Quadratic Weighted Kappa, the accuracy measure which has been described earlier.

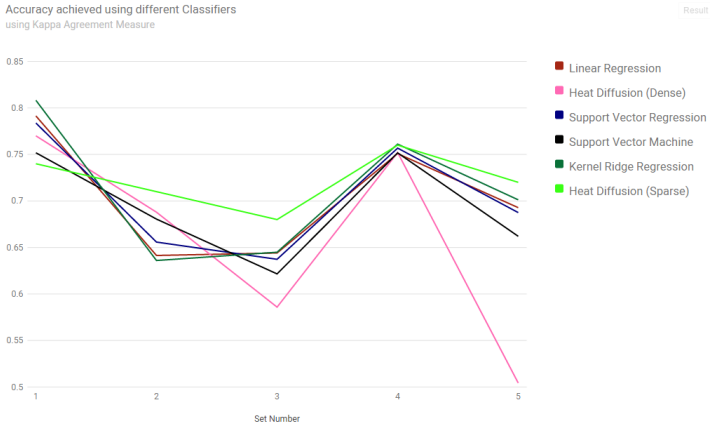


It is important to note that Linear Regression has performed admirably with respect to SVM and SVR. This means that simpler models sometimes are more effective in nature than models which are complicated.

It is notable that Heat Diffusion using Sparse Graph Formulation shows the least variation over all sets and except for set 1, outperforms all other classifiers. Heat Diffusion using the Dense Graph formulation performs nearly at par with the other classifiers, however it is worth noting that it also shows the same amount variation (with respect to to other classifiers).

The performance of the Heat Diffusion using Dense Formulation may be due to the fact that the underrepresented classes are very sparsely represented or points towards a use of better similarity criteria which is data specific which we didn't pursue. It is thus very easily understood that Graph Construction is very important while using spectral methods such as Heat Diffusion.

However, some classes are severely underrepresented in the dataset and thus it is not possible to represent them which contributes to the lesser accuracy. A bigger dataset having more number of underrepresented class elements would help.



VII. Future Work

The results obtained using Graph Diffusion are very promising, however feature engineering would help improve the results considerably. Removing features which do not correlate well with the score and adding more features by forming more hypotheses would be the next step forward. Also, it would be important to come up with a model which uses the feature set to generate good suggestions.

VIII. Acknowledgement

We would like to thank Prof. Avinash Sharma for his encouragement in pursuing this project. We have thoroughly enjoyed learning about state-of-the-art techniques and hope to use them further in our research. We would also like to thank our mentors, Ajitesh Gupta and Gaurav Mishra for their help and support.

References

[1] Anthony Viera et al, *Understanding Interobserver Agreement: The Kappa Statistic*. 2005, url:http://virtualhost.cs.columbia.edu/ julia/courses/CS6998/Interrater_agreement.Kappa_statistic.pdf

[2] bx, *Answer on Kappa Statistic in Plain English?, StackExchange Statistics website*. 2014, url:http://stats.stackexchange.com/questions/82162/kappa-statistic-in-plain-english

[3] Max Welling, *Kernel ridge Regression*. , url:http://www.ics.uci.edu/w̃elling/classnotes/papers_class/Kernel-Ridge.pdf

[4] Derrick Higgins et al, *Evaluating Multiple Aspects of Coherence in Student Essays*. 2004, url:http://www.aclweb.org/anthology/N041024

[5] Alex Adamson et al, *Automated Essay Grading*. 2014

[6] Manvi Mahana et al, *Automated Essay Grading Using Machine Learning*. 2012

[7] Shihui Song et al, *Automated Essay Scoring Using Machine Learning*. 2013

[8] Kuperman V, *Age-of-acquisition ratings for 30,000 english words*. 2012. url:http://link.springer.com/article/10.3758%2Fs13428-012-0210-4/fulltext.html

[9] Louis A et al, *What makes writing great? first experiments on article quality prediction in the science journalism domain.* 2013. url:http://www.transacl.org/ wp-content/uploads/2013/07/paper341.pdf

[10] Kaggle.com, *The hewlett foundation: Automated essay scoring - evaluation*. 2012, url:http://www.kaggle.com/c/asap-aes/ details/evaluation.

[11] Adam Kilgra, *BNC database and word frequency lists*. 1996, url:http://www.kilgarriff.co.uk/ bnc-readme.html.