Regression based Automated Essay Scoring

Sai Srivatsa R
IIT Kharagpur

Anirud Thyagarajan
IIT Kharagpur

Plaban Kumar Bhomick
IIT Kharagpur

Abstract

This paper proposes a regression based approach for automatically scoring essays written in English. We use standard NLP techniques for obtaining the features from the text and integrate it with an existing framework that uses vector space model to improve the results. We extensively evaluate our approach on a benchmark database (Automated Student Assessment Prize), and demonstrate that the results obtained are comparable to professional human raters while at a much faster rate. We also analyse how the essays are scored in order to get a better understanding of the proposed model.

1 Introduction

Essays are a measure of a students metacognitive abilities and descriptive abilities, which makes it tedious to establish a systematic grading system for essays, considering the expense of time and effort. Automated Essay Scoring (AES) is the process of evaluating and grading a written essay using computer programs. Computers were initially used for word processing requirements till AES revealed that they have the potential to act as more effective cognitive tools.

E-rater, by Educational Testing Services (ETS) is a commercial AES system which is currently being used in Graduate Record Examination (GRE) and the Test of English as a Foreign Language (TOEFL). The E-rater system also utilises similar NLP techniques to extract various kinds of linguistic features of essays, such as lexical, syntactic and grammatical features. Then it predicts the final score by the stepwise regression method. (Attali and Burstein, 2006)

Essay scoring can either be viewed as a regression problem (Attali and Burstein, 2006) or a classification problem (L.S. Larkey, 1998). The regression models utilize NLP techniques to automatically rate essays written for given prompts (S Dikli, 2006) . The classification-based approach uses the assigned scores as class labels and uses classification algorithms like the K-nearest neighbor (KNN) or a naive Bayesian model to predict the class an essay belongs to. (Foltz et. al, 2011) Recent approaches build up on the above methods. These methods have also been put into use for grading essays in other langugages by including language-specific features. (stling et. al, 2013) (Peng et. al, 2010)

One more common approach is to view the essay as a semantic vector obtained by Latent Semantic Analysis (Dumais, 2005) and the essays are scored based on the extent to which it is similar to the already scored essays. (Yannakoudakis et. al, 2011) (T.Y. Liu, 2009) (Chen et. al, 2013) propose a preference ranking based approach for learning a rating model, where a ranking function or model is learned to construct a global ordering of essays based on writing quality.

The model built for AES can either be prompt specific or generic. In a generic model, the features extracted for all the essays which are to be assessed . The models that extract features based on NLP techniques alone are usually generic models. In a prompt-specific model, the features are dependent on prompt. Vector-space models are usually prompt-specific models as the words/units that represent the features as different for different prompts.

We consider the Automatic Essay Scoring as a regression based problem and not as a classification problem, as a marginal misgrading will not result in total misclassification. We utilize several NLP techniques to extract features from the essay and use regression based supervised machine learning methods to allot a grade to the particular essay. Apart from extracting general syntactic and low level lexical features, we analyse a few higher level features as well. We integrate our model with an existing improvised vector space model (Peng et. al, 2010) that takes into account relationship between words/small units rather than considering them as independent. We also overcome the loss of sequential information by segmenting the essay and constructing the feature vectors independently and then combining them. As we use a vector-space model also for assessment, the proposed approach is prompt specific.

The paper is organized as follows. Section 2 describes various modules of the proposed approach including features and vector space models. Experiments and results are discussed in section 3. Conclusion and future work is discussed in section 4.

2 Methodology

The workflow of the proposed approach is as follows: (i) We propose features that can used to assess the essays. (ii) We integrate the features with an existing vector space model proposed by []. (iii) We train a Support Vector Regressor on these features to obtain the score.

2.1 Data

We use the publicly available dataset release for the Automated Student Assessment Prize (ASAP), which is sponsored by the William and Flora Hewlett Foundation. The Dataset consists of eight essay sets, each generated from a single prompt. Each set has 900-1800 essays. Essays range from an average length of 150 to 550 words per response. The responses were written by students in grade levels from Grade 7 to Grade 10. Each of the eight sets has its own unique characteristics and variability which makes it one of the best publically available database to test on. We split each set into two parts, the training part has 75% of the essays and the remaining 25% is used for testing.

2.2 Features

We describe the features used in our approach in this section. The features can be classified in the following categories

Spelling Errors:

- The ratio of number of misspelt words to the total number of words in a given essay was computed using the *pyenchant* package. As expected, the number of misspelt words has a negative correlation with the essay scores assigned.

Statistical Features:

- We use a number of simple statistical features obtained from the text such as character count, word count, sentence count, paragraph count. Text length is asscoiated with the students fluency of writing.

- We also include the count of unique words and lexical diversity features (the ratio of number of unique words to the total number of words in the text). These features are indicative of a students vocabulary level. We also use the features such as the count of stop words, NER counts. The use of punctuation is associated with how organised the text is. Hence we also include the puncutation count.

POS counts:

- This set of features includes the frequency distribution of parts of speech of the text (Noun, Verb, Adjective, Adverbs, Prepositions and Pronouns). We use the *nltk* package to obtain the POS tags.

Grammatical and Fluency Features:

- We use link grammar for parsing the sentences and use the count of links in the parsed sentence as a feature. This is a fast and good indicator of grammaticality of a sentence as the ones that are grammatically incorrect have no links when parsed.

- We evaluate the fluency of a given text by computing the bigram, trigram, 4-gram and 5-gram probabilities of the text. We use COCA for obtaining the n-gram data.

- We also adapt a similar approach for computing POS n-gram probabilties of the text.

Readability score:

- The readability score is a measure of organisation of text as well as the syntactic and semantic complexity of the text. We use Kincaid's readability score as one of the features which is given by the formula.

$$r = 206.835 - 1.015\frac{|w|}{|s|} - 84.6\frac{|syl|}{|w|} \quad (1)$$

where $|w|, |s|, |syl|$ denote the number of words, sentences and syllables in the given text.

Ontological Features :

- Following works of (Ong et. al, 2010) who showed that automatically identifying diagram ontology elements in essays has a postive correlation with the expert graders, we use a simplified implementation of their approach for automatically tagging sentence as CurrentStudy, Hypothesis, Claim, Citation supports and opposes.

2.3 Vector-Space Models

In the traditional VSM model, a document containing words/units $w_1, w_2..., w_n$ is represented as a single dimensional vector $D_{[n\times1]}$ where D_i is the weight associated with the word/unit w_i. In this model' the base vectors are represented as $B_i = (0, 0, ..0, 1, 0, ..0)_{n\times1}$. All vectors are orthogonal and hence the words / units do not have any relationship with each other. The Improved vector space model proposed by (Peng et. al, 2010) overcomes this unrealistic assumption by introducing a relation beween the word vectors using a linear transform.

$$D_{new(n\times1)} = W_{n\times n}D_{old(n\times1)} \tag{2}$$

We represent the documents in the vector space by applying a Tf-Idf transfrom. We use word2vec model to generate the word vectors and obtain $W(i, j)$ by computing the cosine similarity between the word vectors corresponding to w_i and w_j. In order to account for order of words, we divide the essay into k segments and compute the new vectors separately and then combine then.

2.4 Regression

With the generated features, we train a support vector regressor (Vapnik et. al, 2010) and perform a grid-search to get the best parameter setting for each prompt. We used both linear and rbf kernels. We perform 3-fold cross validation and we optimized the parameters C for linear kernel and γ and C for rbf kernels through grid search. We tested for $C = [2^{-5}, 2^{-3}..., 2^{9}]$ and $\gamma = [2^{-15}, 2^{-13}..., 2^{5}]$.

Set	Kernel	Kappa	correlation
1	linear	0.8276	0.8268
1	rbf	0.8520	0.8279
2	linear	0.6590	0.7330
2	rbf	0.6589	0.7351
3	linear	0.6527	0.7036
3	rbf	0.6460	0.6979
4	linear	0.6903	0.7683
4	rbf	0.6811	0.7589
5	linear	0.7707	0.8323
5	rbf	0.7679	0.8107
6	linear	0.6722	0.7343
6	rbf	0.7257	0.7545
7	linear	0.7710	0.7704
7	rbf	0.7761	0.7843
8	linear	0.6946	0.7417
8	rbf	0.7322	0.7589

Table 1: Performance of the proposed appraoch evaluated using Cohen's Kappa and Spearmans Correlation coefficient for different sets with different kernels

3 Result

3.1 Evaluation

Following recent approaches we use Spearman's rank correlation coefficient ρ , which is a measure of monotonicity of relationship. We alo use Cohen's Kappa (Brenner et. al, 1996) which is a robust measure to quantify inter-rater agreement compared to percentage agreement as it also accounts for agreement occuring by chance. Quadratic weighted Cohen's Kappa is given by the formula.

$$\kappa = 1 - \frac{\sum_{i,j} w_{ij}O_{ij}}{\sum_{i,j} w_{ij}E_{ij}} \tag{3}$$

where O_{ij} is the number of times the annotators assign the grade i and grade j respectively, E_{ij} is the is the expected number of times for the same event, given that both annotators randomly assign grades according to a multinomial distribution and $w_{ij} = \frac{(i-j)^2}{(N-1)^2}$ and N is the number of possible grade levels. Cohen's Kappa is 1 when there is a perfect agreement and 0 when the agreement is random.

3.2 Performance

Table 1 shows the results obtained by the proposed approach measured by Cohen's Kappa and correlation. We present the results obtained using both

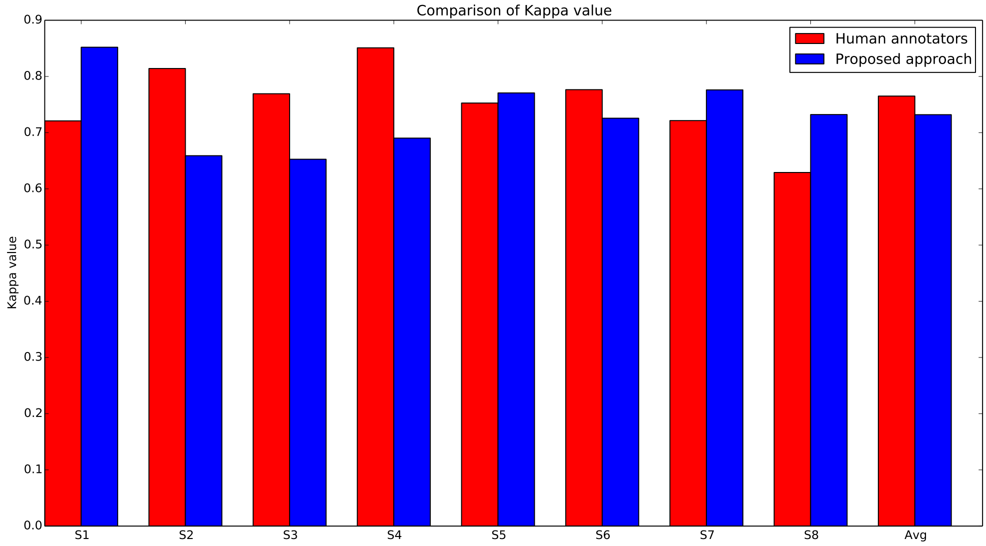

Figure 1: Plot of Kappa values for various sets and the average Kappa value. We use kappa value to compute the inter-annotator agreement using the two raters score in the dataset. We compute the agreement between the score obtained using the proposed and the resolved score assigned to the essays. This plot shows that the proposed approach performs on par when compared expert human raters. the inter-annotator agreement

the kernels, $linear$ as well as rbf. The agreement between professional human raters ranges from 0.70 to 0.80, measured by quadratic weighted Kappa (Powers et. al, 2000) (Williamson, 2009). In the experiments, our approach achieves a quadratic weighted Kappa within the specified range for prompt-specific rating.

3.3 Feature Analysis

We start by analysing the weights corresponding to each feature learnt by the SVM. We list the relevant features based on the weights assigned.

Large Negative weights: Spelling Error Count and the count of words per sentence Length. This is consistent with the fact the essays that have more spelling errors are likely to get lower scores. Also an increase in the average sentence length would lead to decrease in comprehensibilty and organisation.

Negligible weights: ngram probabilities of words and ngram probabilities of POS tagged text get negligible weights. Most of the remaining features get a slightly positive weights.

Highest positive weights: Adjective counts and punctuation counts. Higher weights assigned to punctuation counts suggests that essays that are more organised are likely to receive better scores. Adjectives make the essays more descriptive and vivid and hence use of more adjectives leads to a better score.

3.4 Speed

We implement our method in Python which we plan to make public soon. The average running time for feature extraction and grading is 4.33 seconds per essay on a laptop with Intel i5-4200U CPU @ 1.80 GHz with 6GB RAM.

4 Conclusion and Future Works

In this paper, we propose relevant features for AES and integrate it with an existing improvised vector space model to achieve results comparable to to expert raters. Although the proposed AES systems provides lucrative advantages such as saving time and better reliability in scoring, on some outliers, the absence of a human rater could result in missing out on inferential skills, critical thinking and abstract ideas. These form a scope for improvement for future essay evaluation systems to come.

References

S Dikli 2006. An overview of automated scoring of essays *The Journal of Technology, Learning and Assessment.*

L.S. Larkey 1998 Automatic essay grading using text categorization techniques *In Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pages 9095. ACM.*

Y. Attali, Brent Bridgeman, and Catherine Trapani. 2006 Automated essay scoring with e-rater v. 2 *The Journal of Technology, Learning and Assessment/*

H. Yannakoudakis, T. Briscoe, and B. Medlock. 2011. A new dataset and method for automatically grading esol texts *In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, volume 1, pages 180 189..*

D.M. Williamson. 2009. A framework for implementing automated scoring. *In Annual Meeting of the American Educational Research Association and the National Council on Measurement in Education, San Diego, CA.*

Peter W Foltz, Darrell Laham, and Thomas K Landauer.k. 1999. Automated essay scoring: Applications to educational technology *In World Conference on Educational Multimedia, Hypermedia and Telecommunications, volume 1999, pages 939944.*

S.T. Dumais. 2005. Latent semantic analysis. *Annual Review of Information Science and Technology*

T.Y. Liu 2009. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*

X. Peng, D. Ke, Z. Chen, and B. Xu, 2010 Automated Chinese Essay Scoring Using Vector Space Models *Universal Communication Symposium (IUCS)*

Vladimir Vapnik, Steven E. Golowich, and Alex Smola. 1996. Support vector method for function approximation, regression estimation, and signal processing. *In Advances in Neural Information Processing Systems 9, pages 281287. MIT Press*

stling, Robert, Andr Smolentzov, Bjrn Tyrefors Hinnerich and Erik Hglin 2013. Automated Essay Scoring for Swedish. *Proceedings of the 8th Workshop on Innovative Use of NLP for Building Educational Applications, NAACL-HLT 2013, Atlanta, USA.*

Donald E Powers, Jill C Burstein, Martin Chodorow, Mary E Fowles, Karen Kukich, and Graduate Record Examinations Board. 2000. Comparing the validity of automated and human essay scoring. *Research-report ETS*

Hongbo Chen and Ben He. 2013. Automated essay scoring by maximizing human-machine agreement. *In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 17411752, Seattle, Washington, USA, October. Association for Computational Linguistics.*

Hermann Brenner and Ulrike Kliebsch. 1996. Dependence of weighted kappa coefficients on the number of categories. *Epidemiology, pages 199202.*

Mark D Shermis and Jill C Burstein. 2002. 2002 Automated essay scoring: A cross-disciplinary perspective.

Nathan Ong, Diane Litman, and Alexandra Brusilovsky. 2014. Ontology-based argument mining and automatic essay scoring. *In Proceedings of the First Workshop on Argumentation Mining, pages 2428, Baltimore, Maryland, June. Association for Computational Linguistics.*