Automated Essay Scoring by Maximizing Human-machine Agreement

Hongbo Chen and Ben He
School of Computer and Control Engineering
University of Chinese Academy of Sciences
100190 Beijing, China
chenhongbo11@mails.ucas.ac.cn, benhe@ucas.ac.cn

Abstract

Previous approaches for automated essay scoring (AES) learn a rating model by minimizing either the classification, regression, or pairwise classification loss, depending on the learning algorithm used. In this paper, we argue that the current AES systems can be further improved by taking into account the agreement between human and machine raters. To this end, we propose a rank-based approach that utilizes listwise learning to rank algorithms for learning a rating model, where the agreement between the human and machine raters is directly incorporated into the loss function. Various linguistic and statistical features are utilized to facilitate the learning algorithms. Experiments on the publicly available English essay dataset, Automated Student Assessment Prize (ASAP), show that our proposed approach outperforms the state-of-the-art algorithms, and achieves performance comparable to professional human raters, which suggests the effectiveness of our proposed method for automated essay scoring.

1 Introduction

Automated essay scoring utilizes the NLP techniques to automatically rate essays written for given prompts, namely, essay topics, in an educational setting (Dikli, 2006). Nowadays, AES systems have been put into practical use in large-scale English tests and play the role of one human rater. For example, before AES systems enter the picture, essays in the writing assessment of Graduate Record Examination (GRE) are rated by two human raters. A third human rater is needed when the difference of the scores given by the two human raters is larger than one in the 6-point scale. Currently, GRE essays are rated by one human rater and one AES system. A second human rater is required only when there exists a non-negligible disagreement between the first human rater and the machine rater. With the help of an AES system that highly agrees with human raters, the human workload can be reduced by half at most. Therefore, the agreement between the AES system and the human rater is an important indicator of an AES system's effectiveness.

There have been efforts in developing AES methods since the 1960s. Various kinds of algorithms and models based on NLP and machine learning techniques have been proposed to implement AES systems. Existing approaches consider essay rating as a classification (Larkey, 1998), regression (Attali and Burstein, 2006) or preference ranking problem (Yannakoudakis et al., 2011), where the loss function is the regression loss, classification loss and pairwise classification loss, respectively. In this paper, we argue that the purpose of AES is to predict the essay's rating that human raters would give. If an AES system frequently disagrees with the first human rater, a second human rater will be needed in most cases. Thus, the introduction of the AES system does not bring much benefit in reducing the human workload. It is therefore desirable to minimize the disagreement between the machine and human raters. However, this disagreement is not explicitly, if any, addressed in the current AES methods.

To this end, we propose a rank-based approach in this paper that utilizes a listwise learning to rank

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1741–1752,
Seattle, Washington, USA, 18-21 October 2013. ©2013 Association for Computational Linguistics

algorithm to address automated essay scoring in the view of directly optimizing the agreement between human raters and the AES system. Different from the preference ranking-based approach (Yannakoudakis et al., 2011) that maximizes the pairwise classification precision (Liu, 2009), our rank-based approach follows the listwise learning paradigm and the agreement between the machine and human raters is directly integrated into the loss function that is optimized by gradient boost regression trees.

To the best of our knowledge, this work is the first to apply listwise learning to rank approach for AES, which aims at the optimization of the agreement between the human and machine raters. Experimental results on the publicly available dataset ASAP indicate that our proposed method achieves high agreement with human raters, that is about 0.80, measured by quadratic weighted Kappa (Brenner and Kliebsch, 1996). Our proposed method also outperforms the previous classification, regression and preference ranking based approaches. As it is widely accepted that the agreement between human raters, measured by either quadratic weighted Kappa or Pearson's correlation coefficient, ranges from 0.70 to 0.80 (Powers et al., 2000) (Williamson, 2009), our proposed approach therefore performs as well as human raters.

The rest of this paper is organized as follows. In section 2, we introduce the research background of automated essay scoring and give a brief introduction to learning to rank. In section 3, a detailed description of our listwise learning to rank approach for automated essay scoring is presented. Section 4 explains the experimental setup and section 5 presents the experimental results. Finally, in section 6 we conclude this research.

2 Related Work and Background

Firstly, we give a brief description of existing approaches for AES in section 2.1. Then, an introduction to learning to rank is presented in section 2.2.

2.1 Existing AES Methods

In general, existing solutions consider AES as a learning problem. Based on a large number of predefined objectively measurable features, various learning techniques, including classification, regression and preference ranking, are applied (Larkey, 1998) (Yannakoudakis et al., 2011).

Regression-based approach treats feature values and essay score as independent variables and dependent variable, respectively, and then learns a regression equation by classical regression algorithms, such as support vector regression (Vapnik et al., 1996). In 1966, the first AES system, Project Essay Grader, was developed by Ellis Page upon the request of the American College Board. The PEG system defines a large set of surface text features from essays, e.g. fourth root of essay length, and uses regression-based approach to predict the score that human raters will give. E-rater, developed by Educational Testing Services (ETS) in America, in late 1990s, is a commercial AES system which has been put into practical use in the Graduate Record Examination (GRE) and the Test of English as a Foreign Language (TOEFL). The E-rater system uses natural language processing techniques to extract various kinds of linguistic features of essays, such as lexical, syntactic, grammar, etc.. Then it predicts the final score by the stepwise regression method (Attali and Burstein, 2006).

The classification-based approach sees essay scores as in-discriminative class labels and uses classical classification algorithms, e.g. the K-nearest neighbor (KNN) and the naive Bayesian model, to predict to which class an essay belongs, where a class is associated to a numeric rating. Intelligent Essay Assessor (IEA) (Foltz et al., 1999), developed also in late 1990s, evaluates essay by measuring semantic features. Each ungraded essay, represented by a semantic vector generated by Latent Semantic Analysis (LSA) (Dumais, 2005), is rated according to the similarity degree with semantic vectors of graded essays. Bayesian Essay Test Scoring sYstem, developed by Larkey in 2003, is based on naive Bayesian model. It is the only open-source AES system, but has not been put into practical use yet.

Besides classification and regression-based approaches, (Yannakoudakis et al., 2011) proposed 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. It is also the first study of rank-based approach in automated essay scoring. Although "learning to rank" is not mentioned in their paper, the algorithm they used, Ranking SVM (svm-light package with "-z p" option), is actually a pairwise approach. We will give a brief introduction to learning to rank in section 2.2.

The AES systems can be deployed in two different manners, namely prompt-specific and generic. A prompt-specific rating model is built for a specific prompt and designed to be the best rating model for the particular prompt (Williamson, 2009). For different prompts, the features used, their weights, and scoring criteria, may be different. It usually requires several hundreds of graded essays for training, which is time-consuming and usually impractical in a classroom environment. Generic rating model is trained from essays across a group of prompts and designed to be the best fit for predicting human scores for all prompts. It usually does not consider prompt-specific features and just takes writing quality into account. Generic rating model evaluates essays across all prompts with the same scoring criteria, which is more consistent with the human rubric that is usually the same for all prompts, and therefore has validity-related advantages (Attali et al., 2010).

2.2 Learning to Rank

Learning to rank, also called machine-learned ranking, was originally proposed to settle the ranking problem in information retrieval (IR) (Liu, 2009). It is a type of supervised or semi-supervised machine learning algorithm that automatically construct a ranking model or function from training data.

Current learning to rank algorithms fall into three categories, that is, the pointwise, pairwise, listwise approaches. Pointwise approach takes individual documents as training examples for learning a scoring function. In fact, both multiple linear regression and support vector regression (Vapnik et al., 1996), which have been widely used in automated essay scoring (Shermis and Burstein, 2002), can be seen as pointwise approaches. Pairwise approaches process a pair of documents each time and usually model ranking as a pairwise classification problem. Thus, the loss function is always a classification loss. Representative algorithms are ranking SVM (Joachims, 2006), RankNet (Li et al., 2007), etc.. (Yannakoudakis et al., 2011) apply pairwise approach, ranking SVM, to automated essay scoring and achieve better performance than support vector regression. In listwise approaches, ranking algorithms process a list of documents each time and the loss function aims at measuring the accordance between predicted ranking list and the ground truth label. Representative algorithms are LambdaMart (Wu et al., 2008), RankCosine (Qin et al., 2008), etc.. Listwise approach has not yet been used in automated essay scoring.

3 Automated Essay Scoring by Maximizing Human-machine Agreement

The main work-flow of our proposed approach is as follows. Firstly, a set of essays rated by professional human raters are gathered for the training. A listwise learning to rank algorithm learns a ranking model or function using this set of human rated essays represented by vectors of the pre-defined features. Then the learned ranking model or function outputs a model score for each essay, including both rated and unrated essays, from which a global ordering of essays is constructed. Finally, the model score is mapped to a predefined scale of valid ratings, such as an integer from 1 to 6 in a 6-point scale

In this section, we give a detailed description of our listwise learning to rank approach for AES in section 3.1. And features used in our approach are presented in section 3.2.

3.1 Listwise Learning to Rank for AES

Our choice of the listwise learning to rank algorithm is due to the fact that it takes the entire set of labeled essays associated to a given prompt, instead of the individual essays or essay pairs as in (Yannakoudakis et al., 2011), as training examples. This brings us the convenience of easily embedding the inter-rater agreement into the loss function for the learning.

In this paper, we deploy LambdaMART (Wu et al., 2008), a listwise learning to rank algorithm and then use Random Forests (RF) (Breiman, 2001) for the bagging of LambdaMART learners. Having been widely used in information retrieval applications, LambdaMART is one of the most effective learning to rank algorithms. For instance, it achieves the top results in the 2010 Yahoo! Learning to Rank challenge (Burges, 2010). Random Forests is an ensemble learning method for classification and regression.

Previously, the loss function of LambdaMART is defined as the gradient loss of the retrieval effectiveness, measured by IR evaluation criteria such as Normalized Discounted Cumulative Gain (nDCG) (Wu et al., 2008). More specifically, it is a heuristic method that directly defines λ, the gradient of nDCG with respect to the model score of each document, and has been shown to work empirically for particular loss functions NDCG (Yue and Burges, 2007). Then, Multiple Additive Regression Trees (MART) (Friedman, 2000), also called Gradient Boosting Decision Tree (GBDT)[1], are used to "learn" these gradients iteratively. MART is a class of boosting algorithms that performs gradient descent in function space, using regression trees. Its output $F(x)$ can be written as $F(x) = \sum_i \alpha_i f_i(x), i = 1, 2, ...N$. Each $f_i(x)$ is a function modeled by a single regression tree and the α_i is the corresponding weight. Given that *n* trees have been trained, the *(n+1)*th regression tree, $f_{i+1}(x)$, models the derivative of the cost with respect to the current model score at each training point. Thus, what remains is to compute the derivative.

As for the automated essay scoring, LambdaMART is not readily available since its loss function is defined as a function of the gradient of IR evaluation measures. While such measures focus on the top-ranked documents that are of great importance to the IR applications, they are not suitable to our study. This is because for AES, the rating prediction of all essays equally matters, no matter what ratings they receive.

It is therefore necessary to re-define the λ. Specifically, we need to define the gradient of the evaluation criteria in AES, e.g. quadratic weighted Kappa (Brenner and Kliebsch, 1996) and Pearson's correlation coefficient, with respect to the model score of each essay. In this paper, we use quadratic weighted Kappa as the evaluation metric. Kappa (Cohen and others, 1960) is a statistical metric which is used to measure inter-rater agreement. Quadratic weighted Kappa takes the degree of disagreement between raters into account. This measuring method is widely accepted as a primary evaluation metric for the AES tasks. For instance, it is the official evaluation metric in the Automated Student Assessment Prize sponsored by Hewlett Foundation[2]. We denote our modified LambdaMART as K-LambdaMART in which K stands for the Kappa-based gradient function. Specific steps include the following:

To begin with, we re-define the $\lambda_{i,j}$ for each pair of essays. For a pair of essays, essay i and essay j, $\lambda_{i,j}$ is defined as the derivative of RankNet (Li et al., 2007) loss function multiplied by the Quadratic weighted Kappa gain after exchanging the two essays' ratings.

$$\lambda_{i,j} = \frac{-\delta}{1 + e^{\delta(s_i - s_j)}} |\Delta_{Kappa}| \tag{1}$$

s_i and s_j are the model scores for essay i and essay j, respectively. δ is a parameter which determines the shape of the sigmoid. Quadratic weighted Kappa are calculated as follows:

$$\kappa = 1 - \frac{\sum_{i,j} \omega_{i,j} O_{i,j}}{\sum_{i,j} \omega_{i,j} E_{i,j}} \tag{2}$$

In matrix O, $O_{i,j}$ corresponds to the number of essays that received a score i by human rater and a score j by the AES system. In matrix ω, $\omega_{i,j}$ is the difference between raters scores $\frac{(i-j)^2}{(N-1)^2}$, where N is the number of possible ratings. Matrix E is calculated as the outer product between the two raters vectors of scores, normalized such that E and O have the same sum.

It is necessary to define the quadratic weighted Kappa gain, namely Δ_{Kappa}, in an explicit manner. In each iteration, every essay is ranked by its model score and then rated according to its ranking position. For example, for five essays e_1, e_2, e_3, e_4, e_5 with actual ratings $5, 4, 3, 2, 1$, if the ranking (by model score) is e_3, e_4, e_1, e_5, e_2, we assume that e_3, e_4, e_1, e_5, e_2 will get ratings of $5, 4, 3, 2, 1$, over which quadratic weighted kappa gain can be calculated.

After the definition of $\lambda_{i,j}$ for each pair of essays, it is time to re-define the λ, the gradient for each essay. Let I denote the set of pairs of indices $\langle i, j \rangle$, in which essay i receive a higher rating than essay j.

[1] For space reason, we refer the readers to (Friedman, 2000), (Breiman, 2001) for details of MART, GBDT and Random Forests.

[2] http://www.kaggle.com/c/asap-sas

Set I must include each pair just once. Then, the λ gradient for each essay, e.g. essay i, is defined as,

$$\lambda_i = \sum_{j:\langle i,j\rangle \in I} \lambda_{i,j} - \sum_{j:\langle j,i\rangle \in I} \lambda_{i,j}; \qquad (3)$$

The rational behind the above formulae is as follows. For each of the essays in the whole essay collection associated with the same prompt, e.g. essay i, the gradient λ_i is incremented by a positive value $\lambda_{i,j}$ when coming across another essay j that has a lower rating. The value of $\lambda_{i,j}$ is weighted by the quadratic weighted Kappa gain after exchanging the two essays' ratings. On the contrary, the gradient λ_i will be incremented by a negative value $-\lambda_{i,j}$ when the another essay has a higher rating. As a result, after each iteration of MART, essays with higher rating tend to receive a higher model score while essays with lower rating tend to get a lower model score.

After the training process, the ranking model outputs an unscaled model score for each ungraded essay. To determine the final rating of each given unrated essay, we have to map this unscaled model score to the predefined scale, such as an integer from 1 to 6 in a 6 point scale. The mapping process is as follows. To begin with, the learned ranking model also computes an unscaled model score for each essay in the training set. As the model is trained by learning to rank algorithms, essays with higher model scores tend to get higher actual ratings. In other words, essays with close model scores tend to get the same rating. Therefore, we select the *k* essays whose model scores are closest to the given essay. We then remove the essays with the very highest and lowest model scores within the *k*. The final rating is the mean of the remaining $k - 2$ essays' ratings. In this paper, *k* is empirically set to 5, obtained in our preliminary experiments on the ASAP validation set.

Finally, the Random Forests algorithm is used to bag K-LambdaMART learners. During the training process, both features and samples are randomly selected for each K-LambdaMART learner. In the testing phase, it outputs a score for each testing sample that is the mode of the scores output by each K-LambdaMART learner.

3.2 Pre-defined Features

We pre-define four types of features that indicate the essay quality, including lexical, syntactical, grammar and fluency, content and prompt-specific features. A brief description of these four classes of features is given below.

Lexical features: We define 4 subsets of lexical features. Each subset of features consists of one or several sub features.

– *Statistics of word length*: The number of words with length in characters larger than 4, 6, 8, 10, 12, respectively. The mean and variance of word length in characters.

– *word level*: All words in Webster dictionary [3] are divided into 8 levels according to the College Board Vocabulary Study (Breland et al., 1994). The higher level a word belongs to, the more sophisticated vocabulary usage it indicates. For example, words like thoroughfare, percolate are in level 8, while words with the same meanings, street, filter, belong to level 1. We count the number of words that belong to each level and calculate the mean word level of a given essay.

– *Unique words*: The number of unique words appeared in each essay, normalized by the essay length in words.

– *Spelling errors*: The number of spelling errors detected by the spelling check API provided by Google [4].

Syntactical features: There are 4 subsets of syntactical features.

– *Statistics of sentence length*: The number of sentences with length in words larger than 10, 18, 25, respectively. The mean and variance of sentence length in words.

– *Subclauses*: The mean number of subclauses in each sentence, normalized by sentence length in words. The mean subclause length in words. Subclauses are labeled as "SBAR" in the parser tree generated by a commonly used NLP tool, Stanford Core NLP (Klein and Manning, 2003), which is an integrated suite of natural language processing tools for English in Java[5], including part-of-speech tagging, parsing, co-reference, etc..

– *Sentence level*: The sum of the depth of all nodes in a parser tree generated by Stanford Core NLP. The height of the parser tree is also incorporated into the feature set.

– *Mode, preposition, comma*: The number of modes, prepositions and commas in each sentence respectively, normalized by sentence length in words. Part of speech (POS) is detected by Stanford Core NLP (Toutanova et al., 2003). The POS tags of modal verb and preposition are “MD” and “IN”, respectively.

Grammar and fluency features: There are two subsets of grammar and fluency features.

– *Word bigram and trigram*: We evaluate the grammar and fluency of an essay by calculating mean tf/TF of word bigrams and trigrams (Briscoe et al., 2010) (tf is the term frequency in a single essay and TF is the term frequency in the whole essay collection). We assume a bigram or trigram with high tf/TF as a grammar error because high tf/TF means that this kind of bigram or trigram is not commonly used in the whole essay collection but appears in the specific essay.

– *POS bigram and trigram*: Mean tf/TF of POS bigrams and trigrams. The reason is the same with word bigrams and trigrams.

Content and prompt-specific features: We define four subsets of content and prompt-specific features.

– *Essay length*: Essay length in characters and words, respectively. The fourth root of essay length in words is proved to be highly correlated with the essay score (Shermis and Burstein, 2002).

– *Word vector similarity*: Mean cosine similarity of word vectors, in which the element is the term frequency multiplied by inverse document frequency (tf-idf) (Salton, 1971) of each word. It is calculated as the weighted mean of all cosine similarities and the weight is set as the corresponding essay score.

– *Semantic vector similarity*: Semantic vectors are generated by Latent Semantic Analysis (Dumais, 2005). The calculation of mean cosine similarity of semantic vectors is the same with word vector similarity.

– *Text coherence*: Coherence in writing means that all the ideas in a paragraph flow smoothly from one sentence to the next. We only consider nouns and pronouns in each sentence as they convey more information. The relevance degree between one sentence and its next in the same paragraph is calculated as the sum of the similarity degrees between nouns and pronouns appeared in the two sentences, normalized by the sum of the two sentences’ length in words. The similarity degree between words is set to 1 if coreference exists, indicated by Stanford Core NLP (Lee et al., 2013). Otherwise, it is measured by WordNet similarity package (Pedersen et al., 2004). Finally, text coherence is computed as the average relevance degree of all pairs of neighbored sentences.

The rating model is learned off-line using a set of training essays. For a given target essay, it is the feature extraction that mainly accounts for the overhead. In our experiments, it usually costs in average no more than 10 seconds on a desktop PC with an Intel i5-2410M CPU running at 2.3GHZ to extract the pre-defined features and predict a rating for a given essay, which is affordable, compared to the cost of a human rater.

4 Experimental Setup

This section presents our the experimental design, including the test dataset used, configuration of testing algorithms, feature selection and the evaluation methodology.

4.1 Test Dataset

The dataset used in our experiments comes from the Automated Student Assessment Prize (ASAP)[1], which is sponsored by the William and Flora Hewlett Foundation. Dataset in this competition[6] consists of eight essay sets. Each essay set was generated from a single prompt. The number of essays associated with each prompt ranges from 900 to 1800 and the average length of essays in word in each essay set ranges from 150 to 650. All essays were written by students in different grades and received a resolved score, namely the actual rating, from professional human raters. Moreover, ASAP comes with a validation set that can be used for parameter training. There is no overlap between this validation set and the test set used in our evaluation.

In AES, the agreement between human-machine rater is the most important measurement of success. We use quadratic weighted Kappa to evaluate the agreement between the ratings given by the AES algorithm and the actual ratings. It is widely accepted

[3] http://www.merriam-webster.com/

[4] http://code.google.com/p/google-api-spelling-java/

[5] http://nlp.stanford.edu/software/corenlp.shtml

as a reasonable evaluation measure for AES systems (Williamson, 2009), and is also the official evaluation measure in the ASAP AES competition. It is calculated on all essay topics. If there are essays that come from n essay topics, we calculate the agreement degree on each essay topic first and then compute the overall agreement degree in the z-space. In addition, analysis of variance (ANOVA) (Scheffe, 1999) is conducted to test whether significant difference exists between the two groups of scores given by human and machine raters.

4.2 Configuration of Testing Algorithms

Random Forests bagging K-LambdaMart We denote our proposed method K-LambdaMART where K stands for the Kappa-based gradient. Our implementation of RF bagging K-LambdaMART is based on the open-source RankLib toolkit[7], a library of learning to rank algorithms, in which many popular learning to rank algorithms have been implemented, e.g. LambdaMART and RankNet (Li et al., 2007). Empirical settings of parameters obtained by preliminary experiments on the ASAP validation set are as follows. For bagging: the number of bags is set to 300, subsampling rate is 0.80 and feature sampling rate is 0.50. For LambdaMART in each bag: the number of trees is set to 1, the number of tree leaves is 100 and other parameters are set to default.

Baseline algorithms We use classical machine learning algorithms, support vector machine (SVM) for classification, regression (Vapnik et al., 1996) and preference ranking (Joachims, 2006), respectively, as the baselines. These three algorithms have been used for AES in the literature (Briscoe et al., 2010) (Yannakoudakis et al., 2011). Especially, the state-of-the-art AES approach proposed by (Yannakoudakis et al., 2011) utilizes the SVM for preference ranking, a pairwise learning to rank algorithm, for training a rating model. The linear kernel is used in the experiments. The parameter C, which controls the trade-off between empirical loss and regularizer, is set by grid search on the ASAP validation set.

The original LambdaMART is not included in the baseline algorithms as it has been shown that the performance of LambdaMART is inferior to ranking SVM on the same dataset (Chen et al., 2012).

4.3 Feature Selection

Although machine learning approaches usually use the all features available for training, we try to obtain an carefully selected feature set that can withstand the scrutiny of construct validity in assessment development (Chen and Zechner, 2011). Specific steps of feature selection conducted on individual features are as follows:

To begin with, the importance of the features is determined by computing each features Pearson correlation coefficient with human raters scores based on the training set (Chen and Zechner, 2011). Features whose absolute Pearson correlation coefficient with human scores are lower than 0.20 are removed from the feature set.

Next, we calculate the inter-correlation degrees between these features. For each pair of features whose Pearson correlation coefficient larger than 0.90, one of them should be removed. The criteria for feature removing is as follows. Firstly, at least one feature in each subset of features should be remained. Satisfying the first prerequisite condition, the removed one should be linguistically less meaningful than the remaining one.

For prompt-specific rating model, feature selection is conducted on the essays associated with the same prompt. For generic rating model, the final feature set used for training is the intersection of the 8 feature sets for prompt-specific rating model.

For space reason, here we briefly summarize the feature selection results. Among the lexical features, word length in characters larger than 8 and 10, number of words in each of the levels from 3 to 6, number of unique words, and number of spelling errors are mostly selected. As for the syntactical features, sentence length in words larger than 18 and 25, number of commas, mean clause length and the mean depth of parser tree are usually selected. Among the grammar and fluency features, mean tf/TF of word bigrams and mean tf/TF of POS trigrams are always selected. For content and prompt-specific features, essay length in words, word vector and semantic vector similarity with high rated essays, text coherence are usually selected for training a prompt-

[6] http://www.kaggle.com/c/asap-sas/data

[7] http://people.cs.umass.edu/ vdang/ranklib.html

Table 1: Cross-validation on ASAP dataset measured by quadratic weighted Kappa.

Algorithm	Prompt-specific	ANOVA		Generic	ANOVA
SVMc (baseline)	0.7302(9.75%)	Significant		0.6319(23.93%)	Significant
SVMr (baseline)	0.7861(1.95%)	Significant		0.7022(11.52%)	Significant
SVMp (baseline)	0.7876(1.75%)	Significant		0.7669(2.11%)	Not significant
RF bagging K-LambdaMART	**0.8014**	**Not significant**		**0.7831**	**Not significant**

specific rating model. When it comes to the generic rating model, the prompt-specific features like word vector similarity and semantic vector similarity, are removed.

4.4 Evaluation Methodology

We conduct three sets of experiments to evaluate the effectiveness of our listwise learning to rank approach for automated essay scoring.

The first set of experiments evaluates our proposed approach under a prompt-specific setting. We conduct 5-fold cross-validation, where the essays of each prompt are randomly partitioned into 5 subsets. In each fold, 4 subsets are used for training, and one is used for testing. To avoid bias introduced by the random partition, we repeat the 5-fold cross-validation for 5 times on 5 different random partitions. The overall quadratic weighted Kappa is averaged on all 25 test subsets.

It should be noticed that in random partition of the whole dataset, the overlap between any two partitions should be kept below $1.5*1/(\#folds)*100\%$. For example, in five-fold cross validation, the overlap should be kept below 30%. This is because: according to the Dirichlet principle (Courant, 2005), each subset in one partition overlaps more than 20% with at least one subset in another partition in five-fold cross-validation. The tolerance boundary parameter is then set to 1.5.

The objective of the second set of experiments is to test the performance of our listwise learning to rank approach for generic rating models. We also conduct 5 times 5-fold cross-validation like the first experiment. In 5-fold cross-validation, essays associated with the same prompt are randomly partitioned into 5 subsets. In this way, each fold consists of essays across all prompts. The overall performance is averaged on all 25 test subsets.

In the third set of experiments, we evaluate the quality of the features used in our rating model by feature ablation test and feature unique test. In ablation test, we evaluate our essay rating model's performance before and after the removal of a subset of features from the whole feature set. The performance difference indicates the removed features' contribution to the rating model's overall performance. In unique test, only a subset of features are used in the rating model construction and all other features are removed. The learned rating model's performance indicates to which extent the features are correlated with the actual essay ratings.

5 Experimental Results

5.1 Evaluation Results

Table 1 presents the first set of experimental results obtained on the ASAP dataset, measured by quadratic weighted Kappa. In Table 1, RF stands for random forests. SVMc, SVMr, SVMp are SVM for classification, regression and preference ranking, respectively. ANOVA stands for variance analysis, which aims to test whether significant difference exists between the scores given by human and machine raters. The improvement of our RF bagging K-LambdaMART over each baseline in percentage is also given.

For prompt-specific rating model, all of these algorithms achieve good performance comparable to human raters as literatures have revealed that the agreement between two professional human raters (measured by statistics for correlation analysis, e.g. quadratic weighted Kappa) is around 0.70 to 0.80 (Williamson, 2009) (Williamson, 2009). It is clear that our listwise learning to rank approach, Random Forests bagging K-LambdaMART, gives the best performance on the ASAP dataset. The variance analysis result on the six groups of scores (scores given by five times of five-fold cross-validation and the scores provided by human rater), no significant difference, suggests the robustness of our proposed approach. On the contrary, although preference ranking based approach, SVM for ranking, and regression based approach, SVM for regression, give very good result in human-machine agreement, their variance analysis results indicate that there exists significant difference between the scores given by human and machine raters. The result of the first set of experiments suggests the effectiveness and robustness of our listwise learning to rank approach in the building of prompt-specific rating model.

For generic rating model, one can conclude from Table 1 that RF bagging LambdaMART performs better than SVM for classification, regression and preference ranking on the ASAP dataset. The dataset used in our experiment consists of essays generated by 8 prompts and each prompt has its own features. With such a training set, both classification and regression based approaches produce not good results, as it is commonly accepted that rating model whose performance measured by interrater agreement lower than 0.70 is not applicable (Williamson, 2009). And the variance analysis results also reveal that there exists statistically significant difference between the scores given by human and machine raters, indicating a low robustness of these two baselines. The performance comparison of the generic rating models suggest that the rank based approaches, SVMp and RF bagging K-LambdaMART, are more effective than the classification based SVMc and the regression based SVMr, while our proposed RF bagging K-LambdaMART outperforms the state-of-the-art SVMp. Moreover, we find that there is no obvious performance difference when our proposed method is applied to prompt-specific and generic rating models. Considering the advantages generic rating models have, the result of the second set of experiments suggests the feasibility of building a rating model which is generalizable across different prompts while performs slightly inferior to the prompt-specific rating model.

5.2 Feature Analysis

Table 2 gives the results of feature ablation and unique test. In the table, “All features” stands for the use of all the features available, apart from the prompt-specific features that are not applicable to learning a generic model. In other rows, the feature subset name stands for the feature subset to be ablated in ablation test and the feature subset to be used in unique test. Note that we ablate (as in the ablation test) or use (as in the unique test) a subset of features such as the different statistics of word length as a whole since features belonging to the same subset are usually highly correlated.

Among the lexical features, the two feature subsets, word level and statistics of word length, are highly correlated with essay score in both prompt-specific and generic rating models. This observation was expected since word usage is an important notion of writing quality, regardless of essay topics.

In the syntactical features, the feature subset, sentence level, measured by the height and depth of the parser tree, correlates the most with essay score. One can infer that long sentences with nested subclauses tend to improve the final ratings.

All grammar and fluency features achieve performance around 0.60 in feature unique test for prompt-specific rating model. What is more, during feature selection, we find that the Pearson’s correlation coefficient between the feature values and the final ratings in each essay prompt ranges from -0.20 to -0.60, which suggests that our method to estimate the number of grammar errors is applicable because it is widely accepted that in the evaluation of student essays, essays with more grammar errors tend to receive lower ratings.

Among the content and prompt-specific features, essay length and word vector similarity features give good results in feature unique test. The fourth root of essay length in words has been proved to be a highly correlated feature by many works on AES (Shermis and Burstein, 2002). Word vector similarity feature measures prompt-specific vocabulary usage, which is also important to essay evaluation.

In ablation test, there is no significant performance decrease no matter what feature subset is removed. It seems that each feature subset contributes little to the overall performance and therefore can be removed. However, the result of feature unique test suggests that most features used in our rating model are in fact highly correlated with the writing quality.

6 Conclusions and Future Work

We have proposed a listwise learning to rank approach to automated essay scoring (AES) by directly incorporating the human-machine agreement

Table 2: Results of feature ablation and unique test

Feature subset	Prompt-specific			Generic	
All Features	0.8014			0.7831	
	Ablation	Unique		Ablation	Unique
Lexical features					
Statistics of word length	0.7763	0.7512		0.7801	0.7350
Word level	0.7834	0.7582		0.7779	0.7306
Unique words	0.7766	0.6737		0.7692	0.6786
Spelling errors	0.7724	0.6863		0.7730	0.6742
Syntactical features					
Statistics of sentence length	0.7856	0.6410		0.7684	0.7025
Subclauses	0.7862	0.5473		0.7813	0.5050
Sentence level	0.7749	0.7046		0.7796	0.6955
Mode, preposition, comma	0.7847	0.5860		0.7807	0.5606
Grammar and fluency features					
Word bigrams and trigrams	0.7813	0.6017		0.7824	0.4395
POS bigrams and trigrams	0.7844	0.6410		0.7786	0.6022
Content and prompt-specific features					
Essay length	0.7930	0.7502		0.7736	0.7390
Word vector similarity	0.7658	0.7001		–	–
Semantic vector similarity	0.7924	0.5683		–	–
Text coherence	0.7863	0.6947		0.7798	0.6367

into the loss function. Experiments on the public English dataset ASAP show that our approach outperforms the state-of-the-art algorithms in both prompt-specific and generic rating settings. Moreover, it is widely accepted that the agreement between professional human raters ranges from 0.70 to 0.80, measured by quadratic weighted Kappa or Pearson's correlation (Powers et al., 2000) (Williamson, 2009). In the experiments, our approach achieves a quadratic weighted Kappa around 0.80 for prompt-specific rating and around 0.78 for generic rating, suggesting its potential in automated essay scoring.

Most existing research on AES focus on training a prompt-specific rating model. While such approaches have the advantage of providing a satisfactory rating accuracy for essays written for a specific topic, they also suffer from validity and feasibility problem as a significant amount of training data, namely essays with human ratings, are required for every given essay topic (Attali et al., 2010). It is therefore appealing to develop an approach that learns a generic model with acceptable rating accuracy, since it has both validity-related and logistical advantages. In our future work, we plan to continue the research on generic rating model. Because of the diversification of writing features of essays associated with different prompts, a viable approach is to explore more generic writing features that can well reflect the writing quality.

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China (61103131/F020511), the President Fund of UCAS (Y15101FY00/Y25102HN00), and the National Key Technology R&D Program of China (2012BAH23B03).

References

Y. Attali and J. Burstein. 2006. Automated essay scoring with e-rater® v. 2. *The Journal of Technology, Learning and Assessment*, 4(3).

Yigal Attali, Brent Bridgeman, and Catherine Trapani. 2010. Performance of a generic approach in automated essay scoring. *The Journal of Technology, Learning and Assessment*, 10(3).

L. Breiman. 2001. Random forests. *Machine learning*, 45(1):5–32.

H.M. Breland, R.J. Jones, and L. Jenkins. 1994. *The college board vocabulary study*. College Entrance Examination Board.

Hermann Brenner and Ulrike Kliebsch. 1996. Dependence of weighted kappa coefficients on the number of categories. *Epidemiology*, pages 199–202.

T. Briscoe, B. Medlock, and Ø. Andersen. 2010. Automated assessment of esol free text examinations. Technical report, University of Cambridge Computer Laboratory Technical Reports, UCAM-CL-TR-790.

C. Burges. 2010. From ranknet to lambdarank to lambdamart: An overview. *Learning*, 11:23–581.

Miao Chen and Klaus Zechner. 2011. Computing and evaluating syntactic complexity features for automated scoring of spontaneous non-native speech. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 722–731.

Hongbo Chen, Ben He, Tiejian Luo, and Baobin Li. 2012. A ranked-based learning approach to automated essay scoring. In *Cloud and Green Computing (CGC), 2012 Second International Conference on*, pages 448–455. IEEE.

Jacob Cohen et al. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Richard Courant. 2005. *Dirichlet's principle, conformal mapping, and minimal surfaces*. Courier Dover Publications.

S. Dikli. 2006. An overview of automated scoring of essays. *The Journal of Technology, Learning and Assessment*, 5(1).

S.T. Dumais. 2005. Latent semantic analysis. *Annual Review of Information Science and Technology*, 38(1):188–230.

Peter W Foltz, Darrell Laham, and Thomas K Landauer. 1999. Automated essay scoring: Applications to educational technology. In *World Conference on Educational Multimedia, Hypermedia and Telecommunications*, volume 1999, pages 939–944.

Jerome H. Friedman. 2000. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232.

T. Joachims. 2006. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 217–226. ACM.

Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics.

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 90–95. ACM.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4).

P. Li, C. Burges, and Q. Wu. 2007. Learning to rank using classification and gradient boosting. In *Proceedings of the International Conference on Advances in Neural Information Processing Systems (NIPS)*.

T.Y. Liu. 2009. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. Wordnet::similarity: measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, pages 38–41, Boston, Massachusetts, 2-7 May. Association for Computational Linguistics, Stroudsburg, PA, 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-EDUCATIONAL TESTING SERVICE PRINCETON RR*, (10).

Tao Qin, Xu-Dong Zhang, Ming-Feng Tsai, De-Sheng Wang, Tie-Yan Liu, and Hang Li. 2008. Query-level loss functions for information retrieval. *Inf. Process. Manage.*, 44(2):838–855, mar.

G. Salton. 1971. *The SMART Retrieval System-Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Henry Scheffe. 1999. *The analysis of variance*, volume 72. Wiley. com.

M.D. Shermis and J.C. Burstein. 2002. *Automated essay scoring: A cross-disciplinary perspective*. Lawrence Erlbaum.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics.

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 281–287. MIT Press.

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*.

Q. Wu, C.J.C. Burges, K.M. Svore, and J. Gao. 2008. Ranking, boosting, and model adaptation. Technical report.

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.

Yisong Yue and C Burges. 2007. On using simultaneous perturbation stochastic approximation for learning to rank, and the empirical optimality of lambdarank. Technical report, Technical Report MSR-TR-2007-115, Microsoft Research.