ISSN: 2229-6956(ONLINE)
DOI: 10.21917/ijsc.2014.0111

ASSESSMENT OF PERFORMANCES OF VARIOUS MACHINE LEARNING ALGORITHMS DURING AUTOMATED EVALUATION OF DESCRIPTIVE ANSWERS

C. Sunil Kumar[1] and R.J. Rama Sree[2]
[1]*Research and Development Center, Bharathiar University, India*
E-mail: sunil_sixsigma@yahoo.com
[2]*Rashtriya Sanskrit Vidyapeetha, India*
E-mail: rjramasree@yahoo.com

Abstract

Automation of descriptive answers evaluation is the need of the hour because of the huge increase in the number of students enrolling each year in educational institutions and the limited staff available to spare their time for evaluations. In this paper, we use a machine learning workbench called LightSIDE to accomplish auto evaluation and scoring of descriptive answers. We attempted to identify the best supervised machine learning algorithm given a limited training set sample size scenario. We evaluated performances of Bayes, SVM, Logistic Regression, Random forests, Decision stump and Decision trees algorithms. We confirmed SVM as best performing algorithm based on quantitative measurements across accuracy, kappa, training speed and prediction accuracy with supplied test set.

Keywords:

Descriptive Answers, Automated Evaluation, LightSIDE, Machine Learning Algorithms

1. INTRODUCTION

On-time delivery of results post completion of evaluation of answer scripts is a challenge for most educational institutions. This challenge has been discussed in recent times very intensely by media [1]. Analysis of this challenge reveals that the scarce availability of qualified examiners to evaluate the answer scripts is one of the reasons contributing to the delay. An obvious side effect of this problem is overloading of the examiners with more number of answer scripts for evaluation in limited time which lead to quality issues in evaluation. There were numerous cases reported in media recently about huge increase or decrease of marks when students apply for revaluation of their answer scripts [2] [3].

Most current software systems available in the market offer capabilities for auto-evaluating non-descriptive or objective type answers such as multiple choice answers, fill in the blank type answers, true or false answers. However, there are very few systems available for non-objective type answers evaluation and the technology behind these systems are kept as black boxes to the world by the software manufacturers.

There have been numerous researches by educationalists that proved that a holistic perspective on individuals learning can be obtained through evaluating the individual by means of descriptive essays, short answers. Objective type evaluation is just not enough to obtain the holistic perspective [4]. Therefore the need for systems that offers capabilities of auto-evaluation of descriptive answers.

In real world, when a human evaluator evaluates a descriptive answer, the evaluator interprets the answer based on a pre-defined answer key. Depending on how close the answer is to that of the answer key, the human evaluator allocates a predefined rubric out of the possible rubrics for the answer. The answer key in this context can be reference material provided to the human evaluator or it could be the experience of the human evaluator. We trust the same methodology can be applied in automated descriptive answers evaluation system as well however by replacing a human evaluator with supervised machine learning classifiers. The supervised machine learning system learns from the set of training samples provided. Post the training, the system predicts the rubric for a new answer based on the training it got.

For the scope of this paper, we considered only supervised learning. The research goal of this paper is to evaluate various supervised machine learning algorithms with the same set of training data so as to identify better performing algorithm for evaluation of descriptive answers. The parameters considered to evaluate algorithms are both prediction accuracy obtained from trained models as well as training time required to train each of the models. We have explicitly excluded the memory used by training algorithm from the criteria for evaluation as we have allocated 1 GB heap memory for the training algorithm and the memory gets automatically recycled by LightSIDE when the maximum threshold is hit during training.

In research problems such as automated evaluation of descriptive answers, it may be a very challenging task to obtain a huge training data set. In most cases, the number of samples that can be used as training set may be limited and one should be able to leverage the data available in order to obtain optimum results, therefore the question of "which supervised machine learning algorithm to use with a limited number of training data set available", this statement forms the motivation for our research presented in this paper.

Classification of an answer into an appropriate rubric is a nominal classification task i.e., predicting labels and not a continuous value score for each answer. From the training sets selected, it is clear that the rubrics predicted can be a whole number discrete value labels such as 0, 1, 2, 3 etc., Due to this fact, we decided to consider this classification task as nominal classification task therefore applying algorithms suitable for this classification typology.

The rest of this paper is organized as follows. Section 2 discusses related work; section 3 discusses experimental setup and the preliminaries of the tools and techniques used. Section 4 describes the measurements obtained from the experiments, conclusion remarks and future directions.

2. RELATED WORK

While there was huge amount of research done on document classification area, it appears from our research that minimal research was done on document classification application for

"automated evaluation of descriptive answers". We found that there are some commercial systems available and being used for automated evaluation but the field has not been researched in depth by academics. Due to this fact, we were unable to identify any direct research papers published in the area of automated assessment of descriptive answers through machine learning specifically using LightSIDE workbench. However, some interesting researches presented around the area of automated essay scoring using machine learning are presented in this section.

Mark D. Shermis analysed and contrasted the capabilities of eight already existing commercial machine scoring systems and LightSIDE open source software. The systems included in the study were AutoScore, from American Institutes for Research (AIR), Bookette from CTB McGraw-Hill, e-rater from Educational Testing Service, Lexile Writing Analyzer from MetaMetrics, Project Essay Grade (PEG) from Measurement, Inc., Intelligent Essay Assessor (IEA) from Pearson Knowledge Technologies, CRASETM from Pacific Metrics, IntelliMetric from Vantage Learning and LightSIDE, Carnegie Mellon University, TELEDIA Lab. [5]

Syed M. Fahad Latifi et al. in their research tested the prediction accuracy of three recommended machine learning algorithms in LightSIDE, namely Naïve Bayes, Sequential minimal optimization (SMO), and J48. They have not been able to test the predictions with multi-layer perceptron. The conclusion from their research is that although differences between human and machine classification for transcription variables were generally not large, they are fair enough that they should not be ignored. [6]

Sunil Kumar et al. in their experimented with various training sample sizes in order to determine the best training sample size required for automated evaluation of descriptive answers through sequential minimal optimization. It was determined that when the training sample size is 900, the best prediction accuracies were obtained. [7]

3. EXPERIMENTAL SETUP

The setup in which the experiments are conducted for this paper are specified and the related work of each topic is introduced.

3.1 DATA COLLECTION AND DATA CHARACTERISTICS OF TRAINING DATA

In February 2012, The William and Flora Hewlett Foundation (Hewlett) sponsored the Automated Student Assessment Prize (ASAP) [8] to machine learning specialists and data scientists to develop an automated scoring algorithm for student-written essays. As part of this competition, the competitors are provided with hand scored essays under 10 different prompts. 5 of the 10 essays prompts are used for the purpose of this research.

All the graded essays from ASAP are according to specific data characteristics. All responses were written by students of Grade 10. On average, each essay is approximately 50 words in length. Students are given source text prior to taking up the task of answering the questions and all the questions asked are based on source material provided. The answers used in this research are all in ASCII formatted text and each answer was double evaluated and scored by two independent human scorers. Wherever, the scores provided did not match, another final evaluator's score is provided as the finally resolved score. For the purpose of evaluation of the performance of the model, we considered the score predicted by the model to comply with one of the human scores given the situation of multiple scores.

The data used for training, validation and testing the models are answers written by students for 5 different questions. Data for a question is considered as one unique dataset. So, we have a total of 5 datasets. The questions that students are asked to provide responses to are from Chemistry, English Language Arts and Biology.

3.2 LIGHTSIDE PLATFORM

All experiments performed were executed on a Dell Latitude E5430 laptop. The laptop is configured with Intel Core i5 - 3350M CPU @ 2.70 GHz and with 4 GB RAM however LightSIDE workbench is configured to use a maximum of 1 GB. The laptop runs on Windows 7 64 bit operating system.

3.3 THE LIGHTSIDE WORKBENCH

For the purpose of designing and evaluating our experiments, we have used a machine learning workbench called LightSIDE. LightSIDE (Light Summarization Integrated Development Environment) is a free and open source offering from Carnegie Mellon University (TELEDIA lab). This program has a user-friendly interface and it incorporates numerous options to develop and evaluate machine learning models. These models can be utilized for a variety of purposes, including automated essay scoring. LightSIDE focuses on the syntactical elements of the text rather than semantics. [9]

LightSIDE cannot evaluate any random content or creative content. The automated evaluation we are referring to is for a specific context. LightSIDE can be trained with answers on specific questions and later automated assessment is possible and relevant only for those answers written for those specific questions that the earlier training data set belongs to.

Using LightSIDE to achieve AES involves 4 different steps [10] -

a) Data collection and data input file formatting - LightSIDE Labs recommends at least 500 data samples for each question that the system to get trained on. Once the training data set is available, Data should be contained in a .csv file, with every row representing a training example, except the first, which lists the names of the fields of the data. At least one column in the data should be the label and the other columns can be text and meta-data related to the training example. LightSIDE's GUI interface provides the user with an option to load the input file.

b) Feature extraction - From the input training data set file, user can specify on the LightSIDE GUI the features to be extracted for the purpose of creating a feature table which can later be used to create machine learning models.

c) Model building - With the feature table in hand, one can now train a model that can replicate human labels by selecting the desired machine learning algorithm from LightSIDE's GUI interface and also the GUI can be used to set the various parameters applicable. Model's performance

can also be tested with default 10 fold cross validation or other validation options available on LightSIDE GUI.

d) Predictions on new data - Using the model that is built, new data can be loaded and the classification auto essay scoring task can be carried so as to get the resultant predications on the new data. New data presented for evaluation by LightSIDE also need to abide the input formatting rules as mentioned in steps a and b above.

3.4 STATISTICAL FEATURE EXTRACTION

Though LightSIDE offers capabilities to extract advanced features from training data set, we have limited ourselves to basic bag of words features for the purpose of this research. Below features are focused on from input training data set to build feature table -

a) Unigrams - An n-gram of size 1 is referred to as a "unigram".
b) Bigrams - An n-gram of size 2 is a "bigram" (or, less commonly, a "digram").
c) Trigrams - An n-gram of size 3 is a "trigram".
d) Stop words - The most common, short function words, such as the, is, at, which, and on.
e) Stemming - It is a process of reducing inflected (or sometimes derived) words to their stem, base or root form-generally a written word form.
f) Punctuations - unigrams representing things like periods, commas, or quotation marks

For each of the 5 training data sets, we built a baseline models by -

- Included features - Unigrams, Bigrams, Trigrams, and Stemming.
- Excluded features - Stop words, Punctuations.

3.5 TRAINING DATA, TEST DATA SIZE

In each of the 5 training data sets used for our research, the training set is 900 samples in size. Our previous research for determining appropriate sample size for automated essay scoring using SMO revealed that using 900 samples for training proved to yield slightly better results than using other sample sizes therefore the decision to use 900 samples as the training sample size. [7]

For each data set, we have a separate set of 100 samples to use as test data set. We ensured that the test data sets are non-intersecting with training data sets i.e., none of the test samples are used as part of training data sets.

3.6 SUPERVISED MACHINE LEARNING ALGORITHMS

For the purpose of our research in the paper, we considered the below common nominal class prediction supervised machine learning algorithms -

- Naive Bayes
- Logistic Regression
- Random Forests
- Support Vector Machine
- Decision Stump
- Decision Tree

3.7 MEASUREMENT OF PREDICTIONS AND TRAINING TIME

Due to the implementation of our experiments with 10 fold cross validation, each model we built resulted in Kappa and Accuracy. For each training set, kappas and accuracies obtained with models built using that specific training set were compared as separate categories for ranking. We used the MS-Excel rank function to separately rank the kappas obtained and the accuracies obtained across models build using various training algorithms. For each training set, the Rank function ranks the kappa that has the highest value with rank 1 and the lowest kappa with last rank. Wherever two kappa values are the same, same rank is assigned to both kappa values however the next rank is skipped while assigning a rank to a lower kappa value. Same ranking principles were applied for ranking the accuracy too. We then added up kappa ranks given for each model so as to arrive at a Kappa rank sum for each model. Similarly Accuracy rank sum is obtained for each model. For conclusion purposes, we compared the kappa rank sums across obtained the models. The one with lowest sum is judged the best performed algorithm and the second lowest sum is the second best performing algorithm etc., similar deductions were made using accuracy rank sums too.

Test datasets were used to predict scores using the models built. We compared the obtained predicted scores with that of the manual scores provided by human evaluators. We considered the predicted score to be correctly predicted if it complies with at least one of the two scores provided by human evaluators. For each prompt, we calculated the percentage of test samples correctly predicted, we named it test data prediction accuracy percentage. For each training set, test data prediction accuracy percentages recorded using the models built using the training set were compared for ranking. We used the MS-Excel rank function to rank the prediction test data prediction accuracy percentage. The ranking principles explained in the above paragraph holds good here as well. We then added up ranks given for each model so as to arrive at a prediction accuracy rank sum for each model. For conclusion purposes, we compared the prediction accuracy rank sums across obtained the models. The one with highest sum is judged the least best performed algorithm and the second highest sum is the second least best performing algorithm etc.

Average time taken for training in each fold while building a model was recorded. For each training set, the training time recorded for models built using the training set were compared for ranking. We used the MS-Excel rank function to rank the recorded training time. Rank function ranked the highest training time with rank 1 and the lowest training time with the lowest rank. We are conscious of the fact that, in reality the lowest ranked model is the best performed algorithm and the algorithm that was ranked as number 1 had the worst training time. We then added up ranks given for each model so as to arrive at an average training time rank sum for each model. For conclusion purposes, we compared the average training time rank sums across the obtained models. The one with highest sum is judged

the best performed algorithm and the second highest sum is the second best performing algorithm etc.

There were too many values based on which the final judgement on best performing algorithm need to be chosen therefore we needed a single value that specifies the performance of models. To fulfil this need, we summed up the consolidated ranks for each model type. Again, we ranked the summed values obtained using the MS-Excel rank function. Rank function by default ranked the one with highest sum with rank 1 and the one with lowest sum with the last rank. We are conscious of the fact that, in reality the lowest ranked model is the best performed algorithm and the algorithm that was ranked as number 1 is the worst performed one. Therefore the conclusion that the lowest ranked algorithm is the best performed algorithm to use for automated evaluation and scoring of descriptive answers.

4. MEASUREMENTS OBTAINED, CONCLUSION REMARKS AND FUTURE DIRECTIONS

Various models built during the experiments, the measurements obtained and various conclusions made through analysis of the measurements done during the experiments are described in this section.

4.1 MEASUREMENTS

Table.1. Ranking performance of the algorithms based on kappas achieved with 10 fold cross-validation

Models built using	Bayes	Logistic Regression	Random Forests	SVM Liblinear	Decision Stump	Decision Tree -J48
Training set1	0.141	0.351	0.244	0.324	0.231	0.225
Rank	6	1	3	2	4	5
Training set2	0.02	0.102	0.081	0.131	0	0.022
Rank	5	2	3	1	6	4
Training set3	0.317	0.446	0.316	0.456	0	0.104
Rank	3	2	4	1	6	5
Training set4	0.189	0.415	0.169	0.448	0.463	0.334
Rank	5	3	6	2	1	4
Training set5	0.082	0.489	0.16	0.524	0.446	0.501
Rank	6	3	5	1	4	2
Sum of Ranks	25	11	21	7	21	20
Kappa Consolidated Rank	**6**	**2**	**4**	**1**	**4**	**3**

Table.2. Ranking performance of the algorithms based on accuracies achieved with 10 fold cross-validation

Models built using	Bayes	Logistic Regression	Random Forests	SVM Liblinear	Decision Stump	Decision Tree - J48
Training set1	0.404	0.521	0.446	0.499	0.45	0.434
Rank	6	1	4	2	3	5
Training set2	0.534	0.506	0.526	0.499	0.534	0.466
Rank	1	4	3	5	1	6
Training set3	0.651	0.707	0.619	0.708	0.543	0.542
Rank	3	2	4	1	5	6
Training set4	0.771	0.811	0.778	0.817	0.811	0.767
Rank	5	2	4	1	2	6
Training set5	0.837	0.88	0.844	0.883	0.848	0.871
Rank	6	2	5	1	4	3
Sum of Ranks	21	11	20	10	15	26
Accuracy Consolidated Rank	**5**	**2**	**4**	**1**	**3**	**6**

Table.3. Ranking performance of the algorithms based on accuracies achieved with supplied test data set

Models built using	Bayes	Logistic Regression	Random Forests	SVM Liblinear	Decision Stump	Decision Tree - J48
Training set1	37	57	44	51	50	53
Rank	6	1	5	3	4	2
Training set2	72	61	66	60	72	60
Rank	1	4	3	5	1	5
Training set3	76	80	64	76	61	58

Rank	2	1	4	2	5	6
Training set4	84	94	85	93	88	88
Rank	6	1	5	2	3	3
Training set5	83	73	81	72	67	67
Rank	1	3	2	4	5	5
Sum of Ranks	16	10	19	16	18	21
Prediction accuracy Consolidated Rank	**2**	**1**	**5**	**2**	**4**	**6**

Table.4. Ranking performance of the algorithms based on training time

Models built using	Bayes	Logistic Regression	Random Forests	SVM Liblinear	Decision Stump	Decision Tree -J48
Training set1	4.525	0.408	17.697	0.391	4.008	345.588
Rank	3	5	2	6	4	1
Training set2	4.02	0.39	18.337	0.341	3.907	466.688
Rank	3	5	2	6	4	1
Training set3	4.782	0.491	19.34	0.398	4.662	497.917
Rank	3	5	2	6	4	1
Training set4	2.934	0.315	11.019	0.223	2.54	127.756
Rank	3	5	2	6	4	1
Training set5	2.595	0.271	8.491	0.205	2.129	134.995
Rank	3	5	2	6	4	1
Sum of Ranks	15	25	10	30	20	5
Average Training Time Consolidated Rank	**4**	**2**	**5**	**1**	**3**	**6**

Table.5. Final rank computation based on overall performance of algorithms across accuracy, kappa, tests data accuracy & training time

Models built using	Bayes	Logistic Regression	Random Forests	SVM Liblinear	Decision Stump	Decision Tree - J48
Prediction accuracy Consolidated Rank	2	1	5	2	4	6
Accuracy Consolidated Rank	5	2	4	1	3	6
Kappa Consolidated Rank	6	2	4	1	4	3
Average Training Time Consolidated Rank	4	2	5	1	3	6
Sum of Consolidated Ranks	17	7	18	5	14	21
Final Rank	**3**	**5**	**2**	**6**	**4**	**1**

4.2 CONCLUSIONS DERIVED OF MEASUREMENTS

Based on the experiments and measurements, it is very clear that Support vector machines (SVM) out performs all other algorithms when used for automated evaluation of descriptive answers. Logistic regression and Naive Bayes algorithms are positioned as runner up and second runner up based on their overall performance.

4.3 FUTURE DIRECTIONS

The variant of SVM we used and proved as best performing algorithm is SVM – Liblinear. There exists another variant of SVM called Sequential Minimal Optimization (SMO). A Comparative study between SVM – Liblinear and SVM – SMO for automated evaluation of descriptive answers is an area to explore through research. The future research scope can be widened by verifying the effects of altering the exponents with SVM – SMO on kappa, accuracy and training time etc. Ensemble classifiers for automated evaluation of descriptive answers is another area to progress our research.

REFERENCES

[1] Staff Reporter, “Protest over delay in evaluation work”, The Hindu, Available at: http: // www.thehindu.com

/news/cities/ bangalore/protest-over-delay-in-evaluation-work/article4214480.ece, 2012.

[2] The Times of India, Nagpur, "80 out of 83 score more after revaluation", Available at: http: // articles.timesofindia.indiatimes.com /2011-07-15/nagpur/ 29777272_1_revaluation-results-rechecking-redressal-system, 2011.

[3] Sridhar Vivan, "Revaluation fails 100 'passed' PU students", Bangalore Mirror, Available at: http://www.bangaloremirror.com/index.aspx?page=article&sectid=10&contentid=20110628201106282358189681f9dbf8, 2011.

[4] Siddhartha Ghosh, "e-Examiner: A System for Online Evaluation and Grading of Essay Questions", Available at: http://elearn.cdac.in/eSikshak/eleltechIndia05/PDF/05-e-Examiner%20A%20system%20for%20online%20evaluation%20&%20grading%20of%20essay%20questions-Sidharth-05.pdf, 2013.

[5] Mark D. Shermis, Ben Hammer and Kaggle, "Contrasting State-of-the-Art Automated Scoring of Essays: Analysis", *Contrasting Essay Scoring*, pp. 1-54, 2012

[6] Syed M. Fahad Latifi, Qi Guo, Mark J. Gierl, Amin Mousavi and Karen Fung, "Towards Automated Scoring using Open-Source Technologies", *Annual Meeting of the Canadian Society for the Study of Education*, pp.13-14, 2013.

[7] Sunil Kumar and R. J. Rama Sree, "Experiments towards determining best training sample size for automated evaluation of descriptive answers through sequential minimal optimization", *ICTACT Journal on Soft Computing*, Vol. 4, No. 2, pp. 710 -714, 2014.

[8] Kaggle, "Develop a scoring algorithm for student-written short-answer responses", The Hewlett Foundation: Short Answer Scoring, Available at: http:// www.kaggle.com/c/asap-sas, 2012.

[9] http://lightsidelabs.com/our-technology/

[10] Elijah Mayfield, David Adamson and Carolyn Penstein Rose, "*LightSIDE Researcher's user manual*", pp. 5-9, 2013.