A semantic model for evaluating essays using reflective random indexing

Reinald Kim Amplayo
Ateneo de Davao University
E. Jacinto St., Davao City 8000
Philippines
rktamplayo@addu.edu.ph

Jason Occidental
Ateneo de Davao University
E. Jacinto St., Davao City 8000
Philippines
jtoccidental@addu.edu.ph

ABSTRACT

Essays serve as an excellent gauge in evaluating the test-taker's knowledge of a given concept. However, checking them require evaluators to spend time reading and comprehending the answer before providing a grade. There are existing applications that automate these processes but are memory intensive since they employ the Latent Semantic Analysis (LSA). Thus, the use of this existing checking system is discouraged. In order to provide a more scalable semantic model for evaluation, this study made use of the Reflective Random Indexing (RRI). Reflective random indexing is used to retrieve a content similarity value of an essay response from a test-taker to a provided template response. The word count, keyword count, and content similarity value were used as input features from the gradient descent algorithm. The model shows a 74.41% accuracy based on the quadratic weighted kappa error (QWKE) metric. This shows that the proposed model is competitive to the existing model and can be implemented as a usable alternative for evaluating essay responses.

Categories and Subject Descriptors

I.2.7 Natural Language Processing: Text Analysis

General Terms

Design and Languages

Keywords

reflective random indexing, gradient descent, semantic model, essay evaluation

1. INTRODUCTION

Tests are administered in order to measure the level of knowledge of an individual. Among the types of exams, essays have been preferred to gauge a test-taker's level of understanding. This is because there is freedom in providing an answer which is based on a personal understanding of a specific concept or based on a personal stand on a specific problem or issue. Despite the effective ability of gauging a test-taker's understanding, evaluation of these tests require more time as compared to identification and multiple-choice types of exams. This is due to the checker's responsibility to read and comprehend the answer in order to have a basis for grading. Oftentimes, checkers even go over the responses several times in order to properly judge the answer. Thus, test results with essays are not posted immediately because of the time spent in checking essays.

Computer applications that automate the checking of exams are widely available but do not have features that allow evaluation of essays. This is a widely known problem for these kinds of programs. Researches have been made to implement algorithms that aids in grading essays with accuracy. Latent Semantic Analysis (LSA) is a computational method to aid in deriving the strength of semantic connection between groups of words used in an essay response.

However, these existing applications are memory intensive. This is because performing LSA requires the use of a full term-document matrix that is stored in memory. Thus, these programs are only usable for test checkers that have the required computational resources.

In this study, an alternative method for generating a semantic model is developed. The model will be based on Reflective Random Indexing (RRI) which is a scalable alternative to LSA. Input features, such as the word count, keyword count, and content similarity are implemented for the gradient descent algorithm.

2. AUTOMATED ESSAY SCORING (AES)

Automated Essay Scoring is the computer technology that evaluates and scores the written prose [1]. These systems are developed to assist institutions that administer essay type tests. A clear example is a school assessing their students through essay-type exams. Primary benefits of using these systems are its ability to save time, ability to save costs, and high reliability. Current AES systems have undergone intensive research in which allowed them to be highly accurate. This is determined through the high agreement rates between the results from human graders and the system.

Modern AES systems are improving and providing good results in terms of efficiency but there has been opposition which points out the lack of human interaction [2]. Writing prose is a medium of communication between people and allowing machines to be at the other end sets aside the response's emotion and style. This implies a form of devaluation of writing as simply a means to assess and not a means to communicate with people.

2.1 Current Algorithms for AES

Essay scoring applications often use a linear regression model. Input features such as word count and keyword count are used to train the linear regression model [3]. Prediction for grading of essay answers is then done using the trained model. A research conducted by Murray and Orii [4] made use of linear regression models to predict essay scores. Dense and parse features of generalized linear models were used in their study. Dense features try to determine the total number of various components in an essay. Word count, misspelled word count, character count, stop word count, and symbol count are some of the prominent dense features. Sparse features, on the other hand, cover n-grams. The unigram, bigram, and trigram are the sparse features. The study pointed out that the features differ in impact to score prediction. The high impact features that have been identified were the word count, keyword count, misspelled word count, and some n-grams.

Another algorithm, the k-Nearest Neighbor (kNN) algorithm for text categorization, has been applied for automating test evaluation of essays in the College English Test CET-4, a national English level test in the People's Republic of China [5]. Each essay is represented by vector space model. Stop words are removed in order to utilize the words, phrases, and arguments as input features of the essay. The value of each vector is represented by the term frequency and inverse document frequency (TF-IDF) weight. Checking the similarity of two word space models is done using the cosine similarity algorithm. The result of the study showed a 76% accuracy.

Finally, a Backpropagation Neural Network and Latent Semantic Analysis (LSA) were used to evaluate the quality of high school students in Thailand [6]. The goal of the study was to assess the improvement of introducing LSA in the scoring procedure. LSA was done before the vectors were fed to the neural network. Samples were made using 40 scored essays which was evaluated by high school teachers. The study implemented the experiment twice, one using the LSA and the other without LSA. The study found out that the introduction of an LSA technique reduced the mean error of their sample set by 36%.

3. LATENT SEMANTIC ANALYSIS

Latent Semantic Analysis is used as a technique for indirect inference on semantic relatedness among terms that don't co-occur directly in one document. The terms are projected into a high-dimensional semantic space through a term-document matrix. Reduction is then done using Singular Value Decomposition (SVD). SVD is a dimension reduction technique that is used to construct a semantic space of a topic contained in the set. Semantic similarity can now be determined from the distance between elements of the text passage in the semantic space. This is done using the cosine similarity or normalized scalar product.

However, there are problems in using LSA. SVD requires input of a high dimensional semantic space that needs to be stored in memory. LSA relies on SVD and makes the technique unusable for computers with limited RAM size.

4. REFLECTIVE RANDOM INDEXING

Other indirect inference techniques have been used to implement an alternative for LSA. One alternative, called random indexing (RI), is a technique incorporates an indirect inference that eliminates the reliance to SVD [7]. Instead of placing all the terms into the semantic space, the values are randomly allocated on zero document vectors. The will then be used for term document corpus training by calculating the weights of term occurrence. Weights are solved using the log-entropy function written in eqn (1).

$$\log entropy = \sum \frac{P_{i,j} \log_2 P_{i,j}}{\log_2 n} \quad (1)$$

$$P_{i,j} = \frac{tf_{i,j}}{gf_{i,j}} \quad (2)$$

where $tf_{i,j}$ is the local term frequency of term i; $gf_{i,j}$ is the global term frequency of term i; and n is the number of documents in the corpus. Among applications that use the RI approach is for searching large RDF (Resource Description Framework) graphs [8]. RI is used to generate a semantic index to find similarities among nodes within the graph. In their experiments with these graphs in the sphere of life sciences, they were able to generate a set of similar terms for each keyword of interest.

Though document vectors are given randomly allocated values, they are identified to be orthogonal to each other. The computational complexity of LSA and RI are cubic and quadratic respectively [9].

However, RI is not optimal for derivations of meaningful indirect connections. Thus, modifications were done to increase its inference ability. This gave birth to Reflective Random Indexing (RRI) which cyclically retrains the term vectors in RI. This iterative variant generates new inferences by considering existing output from a dataset produced by the previous iteration.

RRI's basic concept is to get the same dimensionality reduction achieved by latent semantic indexing. It does it without using mathematically complex and intensive singular value decomposition. This also rids the use of its matrix methods. Figure 1 shows a typical procedure of term-based RRI.

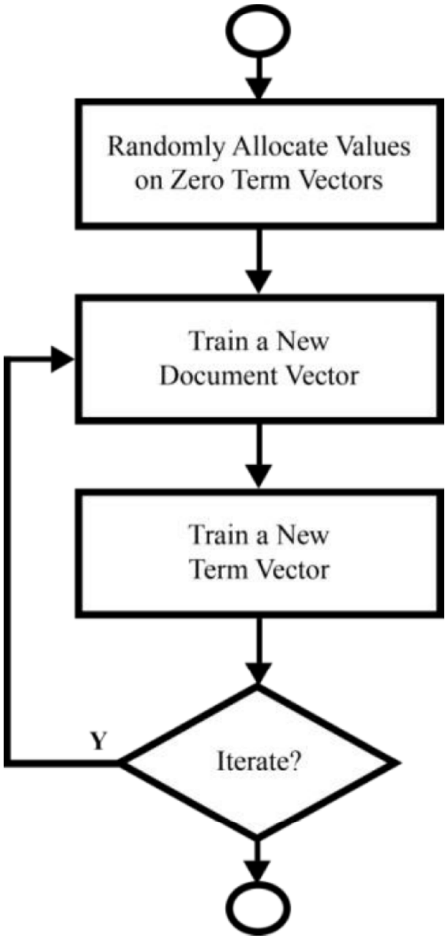
Randomly Allocate Values on Zero Term Vectors
Train a New Document Vector
Train a New Term Vector
Y
Iterate?

Figure 1. Steps in a Term-based RRI Procedure.

Training of a new vector is done by adding the document vector of each document it occurs in. When necessary, repeating the training steps is done to improve the output.

The application of RRI has been significant. Among existing studies [10], RRI has been used as methods for indexing medical literature. This study made use of the MEDLINE research database and the Medical Subject Heading to evaluate a novel approach in indexer assistance by combining the nearest neighbor classification and RRI. The study was able to outperform the existing system. This suggests that RRI can be a potential scalable alternative in semantic modeling.

5. GRADIENT DESCENT ALGORITHM

Gradient descent is a first-order optimization algorithm that finds a local minimum of a function by taking steps proportional to the negative of the gradient of the function at the current point. It is an algorithm that is used to discuss basic discussions like finding the minimum of a parabola when the derivative of a function cannot be solved directly for x. It would start at some value, use a derivate at that value to tell which way to move, and then repeat the process until an approximation of the true value of x is arrived.

The algorithm is used for neural networks for a general framework of dynamical systems. This general approach organizes and simplifies all the known algorithms and results which have been originally derived for different problems such as fixed point or trajectory learning, for different models such as discrete and continuous, for different architectures such as forward or recurrent, and using different techniques like backpropagation, variational, etc. It can also be applied to derive new algorithms.

Gradient descent is used to solve a multi-variable linear regression problem. The problem states that given the list of features x, output the estimated value of y by equating a linear function of x to y. For example, when x consists of only one feature, the hypothesis function or y is given using eqn (3).

$$h_\theta(x) = \theta_0 + \theta_1 x \quad (3)$$

In this formula, θ_i are parameters where the 0th θ is the zero condition, and the rest are gradient. In summary, a hypothesis takes in some variable (features x), uses parameters determined by a learning system (θ_i parameters), and outputs a prediction based on that input ($h_\theta(x)$). For the general linear regression, the equation is given in eqn (4).

$$h_\theta(x) = \sum_{i=0}^{n} \theta_i x_i \quad (4)$$

A cost function is needed to implement a linear regression. It helps in figuring out how to fit the best straight line to the data presented. This also allows selecting which values to put on the parameters to make a straight line based on the training set. The cost function is defined to minimize the summation of the difference between the hypothesis function and the supposed output $(h_\theta(x) - y)^2$. The cost function that should be minimized is showin in eqn (5).

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 \quad (5)$$

Gradient descent is applied in linear regression as follows:

- Start with initial guesses of the parameters
- Change the parameters to try and reduce the cost

Table 1. Sample Dataset

Dataset	#Test	#Training	Sample Essay	Score
1	431	1295	The story is basically talking about more people with cars should ride bicycles to work or to school's so it want be alot of polution in the air.	0/3
2	442	1329	She said that she was going to take the test agen because to show her mom that she can do it and Past it this time. instad of worring over her old home.	1/3
3	451	1354	"The mood set by the author is thankful. The author says that he is happy to have his parents. He thanks them for giving him a life in Cuba, but still carrying out their Cuban ways. He was happy to have others come into his apartment and sit down and act like family. He likes that his parents are selfless and …"	3/4
4	450	1350	"The builders faced many obstacles while trying to add a mooring mast on top of the Empire State Building. As stated in paragraph fourteen, nature was the biggest obstacle the builders faced. The ""violent air currents"" would cause dirigibles to move around the mooring mast, even if it was tethered. …"	4/4

function

- Repeat the first two steps until convergence to a local minimum happens

A significant application of gradient descent is presented in a feature selection process called grafting [11]. Predictor models are created using the algorithm wherein iteration optimizes the model.

6. METHODOLOGY

6.1 Data Gathering

The essay datasets used in the study were essay data uploaded by the Hewlett Foundation. The essays were written by Grade 7 and Grade 10 students and have an average length of 550 words. The essay datasets include essay descriptions and training materials. The training materials provide the criteria used by the teachers on different essay datasets. The dataset was downloaded from kaggle. Minimal preprocessing was done such as converting all the letters to lowercase and non-letters to spaces.

There are four datasets that are dependent to a source information. These source dependent essays are used as datasets in the study. The given essay sets have unique characteristics that are able to test the developed semantic model. Table 1 shows a sample from the dataset.

Furthermore, a corpus should also be collected as this will be the basis of the semantic model. The corpus is a single file that contains a structured set of texts. This is essential in evaluating the linguistic rules of a response. RRI uses the semantic model to find similarities between two documents. The corpus applied for the model is the Wikipedia Extended Abstracts (WEA) which is in the DBpedia website[1].

Preprocessing of the corpus is required before building the semantic model. Non informative strings such as HTML tags, XML tags, and Unicode symbols are removed. Then, the corpus is divided into several text files, one extended abstract per text file. The file contains 3.6 million extended abstracts and would contain 3.6 million text files after preprocessing. Finally, file indexing is used to allow optimized searching for words in the corpus. This is possible by having a directory of indices of the corpus. The indexed folder will now be used for building the semantic model.

6.2 Semantic Model Building

In order to establish an efficient semantic model, elements that have no impact to the accuracy of the output evaluation are removed.

Semantic model building involves the use of RRI. This starts off by removing terms with frequencies less than 10. Also, stop words are removed and numbers are filtered. The model is built with 800 dimensions and two training cycles. Finally, a semantic model is produced from the RRI.

6.3 Linear Regression and Model Training

The gradient descent algorithm with multiple variables is used in order to train the linear regression model. Gradient descent algorithm is a first-order optimization algorithm that finds a local minimum of a function by taking steps proportional to the negative of the gradient of the function at the current point.

The features of the dataset need to be extracted in order to perform the gradient descent algorithm. The features extracted are spelling error value, grammar error value, word count, keyword count, and the content similarity value.

The values of the first two features, the spelling error and the grammar error, are calculated using spelling and grammar collection tools. The spelling error value is the frequency of spelling corrections of the essay input while the grammar error value is the frequency of grammar corrections. The word count and keyword count is simply the number of words and keywords in the essay input. The training materials included in the dataset contain the keywords. The content similarity value is the cosine similarity between two document vectors: the document vector of the essay made by the test taker and the document vector the test checker's expected essay response. The cosine similarity is shown in eqn (6).

$$\text{cossim} = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i x B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2 x}\sqrt{\sum_{i=1}^{n} (B_i)^2}} \qquad (6)$$

Where A and B are the document vectors of the two essays. Also,

Table 2. Training Guidelines

Dataset	Guideline	Feature Used
1	• Understanding of the complexities of the text and question • Insightful observations are interwoven with textual quotes and meticulous details • Main answer is supported with expressed information • Ideas are extended through evaluation of the situation and appropriate comparison	• Word count • Keyword count • Content similarity value
2		
3	• Idea/Topic Development • Organization • Details • Language/Style • Structure • Grammar and Usage • Mechanics	• Spelling error value • Grammar error value • Word count • Keyword count • Content similarity value
4		

[1] http://downloads.dbpedia.org/3.9/en/long_abstracts_en.nq.bz2

n is the number of terms in the document vector. The cosine similarity will have a result in the range [-1, 1] where -1 indicates a dissimilar response and 1 indicates a similar response.

The training materials also include the rubrics and criteria of evaluating essay responses. Table 2 shows the provided guidelines.

The mean and standard deviation of all extracted features are then calculated. In order to retrieve a parameter vector, the gradient descent algorithm is called. Evaluating of essays can now be done after the linear regression model is trained.

7. RESULTS

A quadratic weight kappa error (QWKE) metric is used to assess the performance of the model. QWKE measures the agreement between two variables: the machine rater and human rater. The metric typically varies from 0 to 1 and may go below 0 for instances with lesser agreement between variables. The QWKE metric is performed as follows:

1. An N-by-N histogram matrix is created over the essay ratings. Each cell in the matrix has a corresponding column c and row r which represents the number of essays that received a rating c by the machine rater, and a rating r by the human rater.

2. A matrix of weights is calculated based on the difference between raters' scores. This calculation is shown in en (7).

$$\text{weight} = 1 - \frac{distance^2}{maxdistance^2} \tag{7}$$

3. The sum of the product of the weight matrix and the score matrix (Pobserved) is calculated for each cell.

4. The QWKE metric is solved as shown in en (8).

$$\text{QWKE} = \frac{P_{observed} - P_{expected}}{1 - P_{expected}} \tag{8}$$

The predicted scores are plotted along the horizontal axis and the actual scores are plotted along the vertical axis in an N by N Histogram matrix shown on Table 3.

Table 3. N by N Histogram Matrix (Predicted vs. Actual)

	0	1	2	3	4
0	26	61	6	1	0
1	15	350	132	12	0
2	0	97	433	71	1
3	0	7	182	256	10
4	0	0	5	79	29

Table 4. Frequencies of Agreement

Rate	Maximum Possible	Chance Expected	Observed
0	41	2.17	26
1	509	147.85	350
2	602	257.37	433
3	419	107.53	256
4	40	2.55	29
Total	**1611**	**517.47**	**1094**

The maximum possible agreement between two scores is observed from the frequencies of agreement. The QWKE is positive when the observed agreement is greater than the expected agreement in every rating. Table 4 shows the frequency of agreement and table 5 shows the result of QWKE evaluation.

Table 5. Kappa with Quadratic Weighting

Observed Kappa	Standard Error	0.95 Confidence Interval	
		Lower Limit	Upper Limit
0.7441	0.0302	0.685	0.8032
0.8288	**Maximum possible quadratic-weighted kappa**		
0.8978	**Observed as proportion of maximum possible**		

The metric gave the semantic model an observed kappa of 74.41% with 3.02% standard error. Thus, this can go up to 80.32%. The metric gives a maximum possible QWKE of 82.88%. Therefore the ratio of the QWKE metric over the maximum possible QWKE yields an accuracy of 89.78% for the given model.

8. Conclusion

An approach in the automated scoring problem was presented in the study using a gradient descent algorithm. With the use of reflective random indexing technique in semantic analysis, a content similarity value was extracted to the essay responses. The content similarity value was used as input vectors for the gradient descent algorithm for multiple variables. This included the spelling error value, grammar error value, word count, and keyword count, the content similarity.

The semantic model was rated using the quadratic weighted kappa error and has an observed kappa of 74.41%. This translates to a competitive semantic model as opposed to the existing models utilized in existing AES systems. Furthermore, using reflective random indexing in developing a semantic model is a more usable technique in evaluating essays due to reductions in memory usage. Thus, the reflective random indexing is viable alternative in implementing an essay evaluation method against the use of latent semantic analysis.

9. ACKNOWLEDGEMENT

This study would not have been possible without the support of many people. The authors wish to express their gratitude to their adviser, Mr. Dominique Gerald Cimafranca and to their panelists, Ms. Pelobello, Ms. Banawan, and Mr. Freires who has contributed their time and expertise on the study. Also, the authors would like to thank the creators of the Semantic Vectors package, especially Dr. Dominic Widdows, Ph.D, Finally, the authors would like to thank their friends and family for their undying support for this academic endeavor.

10. REFERENCES

[1] S. Dikli, "Automated essay scoring," *Turkish Online Journal of Distance Education*, vol. 7, pp. 49-62, Jan. 2006

[2] B. Code, C. Vojak, S. Kline, S. McCarthey, and M. Kalantzis, "New spaces and old places: An analysis of writing assessment software," *Computers and Composotion*, vol. 28, pp. 97-111, Jun. 2011.

[3] S. J. Haberman and S. Sinharay, "The application of the cumulative logistic regression model to automated essay scoring," *Journal of Educational and Behavioral Statistics*, vol. 35, pp. 586-602, Oct. 2010.

[4] K. W. Murray and N. Orii (2012) Automatic Essay Scoring. [Online]. Available:

http://www.cs.cmu.edu/afs/cs.cmu.edu/Web/People/norii/pub/aes.pdf

[5] B. Li, J. Lu, J. M. Yao, and Q. M. Zhu, "Automated essay scoring using the KNN algorithm," in *Proc. International Conference on Computer Science and Software Engineering*, 2008, pp. 735-738.

[6] C. Loraksa and R. Peachavanish, "Automatic Thai-language essay scoring using neural network and latent semantic analsysis," in *Proc. First Asia International Conference on Modelling & Simulation,* 2007, pp. 400-402.

[7] T. Cohen, R. Schvaneveldt, and D. Widdows, "Reflective random indexing and indirect inference: A scalable method for discovery of implicit connections," *Journal of Biomedical Informatics,* vol. 43, pp. 240-256, Sep. 2009.

[8] D. Damljanovic, J. Petrak, M. lupu, H. Cunningjam, M. Carlsson, G. Engstrom, and B. Andersson, Grafting, "Random Indexing for Finding Similar Nodes within Large RDF Graphs," *Lecture Notes in Computer Science*, vol. 7117, pp. 156-171, 2003.

[9] D. Widdows and T. Cohen, "The semantic vectors package: New algorithms and public tools for distributional semantics," in *Proc. Fourth IEEE International Conference on Semantic Computing,* 2010, pp. 9-15

[10] V. Vasuki and T. Cohen, "Reflective random indexing for semi-automatic indexing of the biomedical literature," *Journal of Biomedical Informatics*, vol. 43, pp. 694-700, Oct. 2010.

[11] S. Perkins, K. Lacker, and J. Theiler, Grafting, "Fast, Incremental Feature Selection by Gradient Descent in Function Space," *The Journal of Machine Learning Research*, pp. 1333-1356, 2003.