4 SEPTEMBER 2016

Term Paper On

Data Warehousing and Mining

ICT - 6522

AN OVERVIEW OF CURRENT RESEARCH ON AUTOMATED ESSAY SCORING

Submitted by

Mahedi Hasan

1014312019

An Overview of Current Research on Automated Essay Scoring

Mahedi Hasan
Institute of Information and Communication Technology (IICT), BUET
email: mahedi0803@gmail.com

***Abstract*- Essay scoring has traditionally relied on human raters, who understand both the content and the quality of writing. In language tests writing essays is considered as an essential part to assess students' language skill. But In real life scoring students' writing is one of the most time consuming a labor intensive activity for human raters. Moreover human scoring methods sometime lead to an inaccuracy of grading. Automated scoring has the potential to provide solutions to some of these obvious shortcomings in human essay scoring. Today's state-of-the-art systems for automated scoring involve construct-relevant aggregation of quantifiable text features in order to evaluate the quality of an essay. Several Automated Essay Scoring (AES) have been developed under academic and commercial initiative many statistical and natural language processing technique. Amongst many others. Latent Semantic Analysis (LSA) is a powerful Information Retrieval technique that uses statistics and linear algebra to discover underlying "latent" meaning of text. LSA forms a word by document matrix and then the matrix is decomposed using Singular Value Decomposition (SVD) technique. Existing AES systems based on LSA can't achieve higher level of performance to be a replica of human grader. The Author of Paper 1 and Paper 3 have developed an AES system using Generalized Latent Semantic Analysis (GLSA) which makes n-gram by document matrix instead of word by document matrix and experimentally proved that their system outperforms the existing system. The author of Paper 2 introduce a new incremental method of LSA to score essays effectively when the dataset is massive. By comparison of the traditional method their incremental method, concerning the running time and memory usage has a huge advantage over the traditional method.**

I. INTRODUCTION

Scoring students' writing is one of the most expensive and time consuming activity for educational assessment. Especially in language tests writing is considered as an essential part to assess students' language skill. In such tests students are usually require to write different essays and raters score their essays based on some given criteria. As such language tests become more and more popular and number of the testees become larger scoring essays become a huge labor intensive work [7] [8].

Moreover human scoring methods will lead to an inaccuracy of grading [6]. First of all scoring task is much more time consuming and expensive. It requires teacher recruiting, training, calibration and monitoring total system. Even human raters often subject to perception difference error, drift error and inconsistency error. Therefore a more accurate and fastest automated essay scoring method has been desired for a long time.

Automated essay grading uses computer algorithms designed to emulate human scoring. This is achieved by extracting linguistic features from essays and then using machine learning and modelling to establish a correspondence between these features. Then test essays are scored based on a sample of essays that have been scored by human raters. These processes are iterative and require replication to achieve the most optimal solutions for scoring of essays. Consequently, automated scoring solutions require training before they can be used to mark student essays in a live test administration.

Usually, there are three levels upon which to evaluate an essay [15]: *word*, *sentence* and *paragraph*. *Word* is the basic part and testees are required to use correct characters, spelling and meaningful words; *sentence* means students are required to think about the confluence of sentences and inter-sentences, and the relations between the topic and these sentences; *paragraph* is a consideration about the logic relations among paragraphs and even the whole passage. As to the current technique on automated essay scoring, *word* remains a basic and crucial part of assessing an essay [16].

LSA is a technique that has been successfully applied into a wide range of fields and industries [17] [18] [19] [20], such as bio-information [21] [22], web document comprehending [23], language processing [24] [25] [26] and signal processing [27]. It is used for comparing the essays in a reduced dimensionality semantic space based on the words they contain [28] [29]. Its main tasks are to avoid the surface of the language complexity, and to understand the true meaning the words are expressed from a semantic perspective. According to the words appeared in the dataset, a weighted matrix is used to reduce the dimensionality, and to represent the true significance of the words.

Singular value decomposition (SVD) is the main algorithm for LSA to produce low-rank approximations [30]. By SVD, the original dictionary-based space will be divided into three subspaces whose elements are often regarded as semantics. Obsoleting useless semantics and multiplying the new matrices will reconstruct the semantic-based space. This technique which shows viability theoretically, however, cannot be used effectively even practically when faced with massive streams of data. Earlier experiments showed that neither memory usage nor time consuming can support such huge a dataset for SVD.

In order to resolve the problems regarding memory usage and time consuming, incremental SVD has been successfully introduced by Matthew Brand [31], and has been implemented into the fields of image processing [32], information retrieval such as recommender systems [33] [34] and natural language processing [23]. Automated essay scoring using incremental LSA, however, is a new method. The Author of Paper 2 experimentally shows that this incremental method is effective to reduce the usage of memory and time consuming without lowering the performance of automated scoring in comparison with human scoring.

II. EXISTING ESSAY GRADING SYSTEMS

First generation automated essay scoring system have been developed in the mid to late 20th century. Limited language processing methods and algorithms (e.g. word counts, grammar/spell checks) are used to extract and evaluate lexical and syntactic properties of essays. Regression analyses were then typically applied to generate an essay score.

As early as 1966, Page developed Project Essay Grader (PEG) were first developed which uses multiple regression technique to grades essays. On this system grading was done on the basis of writing quality, taking no account of content [9]. Vector–space model was developed by Sparck Jones at 1972 which starts with co-occurrence term-document matrix formed from the essay. TF-IDF is used for weighting the elements of matrix and Cosine correlation is used for scoring. It is less successful at judging overall quality essay [1].

E-rater is an essay-scoring system developed in 1990 by Educational Testing Service. E-rater uses multiple regressions with NLP to extract writing features of essays. By comparing human and E-Rater grades essays with 87% accuracy [9], [10]. Bayesian Essay Test Scoring System (BETSY) [14] is based on Multivariate Bernoulli Model (MBM) and the Bernoulli Model (BM). BETSY is a program that classifies text based on trained material and is being developed by Lawrence M. Rudner. Using BESTSY an accuracy of over 80% was achieved. IntelliMetric uses a blend of Artificial Intelligence (AI), Natural Language Processing (NLP), and statistical technique. IntelliMetric process assists to examine the essay according to the main characteristics of standard written English.

Bin L. et al. designed an essay grading technique that used text categorization model which incorporates K-Nearest Neighbor (KNN) algorithm. Transforming the essays into the vector space model (VSM), TF-IDF and IG are applied for feature selection from the feature pool of words, phrases and arguments. After training for the KNN algorithm, a precision over 76% is achieved on the small corpus of text [11].

Intelligent Essay Assessor (IEA) developed by Hearst et al. based on the Latent Semantic Analysis (LSA) technique was originally designed for indexing documents and text retrieval. Whittington et al. defined that LSA represents documents and their word content in a large two-dimensional matrix semantic space. Using a matrix decomposition technique known as Singular Value Decomposition (SVD), new relationships between words and documents are built and existing relationship are improved to more accurately represent their true significance. The correlation from 0.59 to 0.89 has been achieved between the IEA and human raters [12], [13].

III. PRELIMINARIES

A. Singular Value Decomposition

The underlying algorithm of LSA is SVD, which can construct a semantic space of a given dataset. Given *r*-rank matrix M, upon which we apply SVD:

$$M = U \sum V^T \tag{1}$$

Where U and V are orthogonal matrices, and the elements in $\sum$ are singular values those are in descending order. Specially, in natural language processing, maintaining only $k \ll r$ will produce a lower dimensionality and better approximation about the original matrix M. By removing the (r - k) diagonal elements, (r - k) columns in U and (r - k) rows in V and (r - k) elements in $\sum$ where the elements with far too small values are considered to be noise and unnecessary, we can multiply the matrices and get the approximation to the original matrix:

$$M'_{m\,x\,n} = U_{m\,x\,k} \sum_{k\,x\,k} V^T_{k\,x\,n} \tag{2}$$

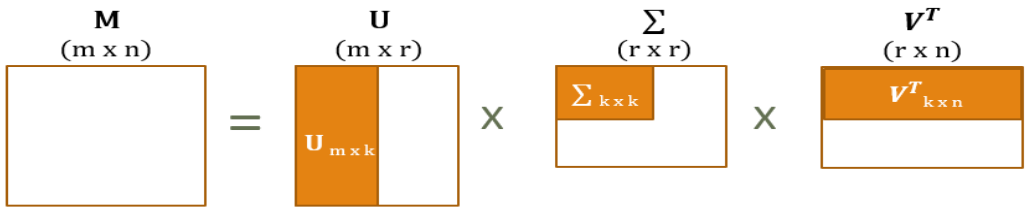

Figure 1. The Formation of singular value decomposition

The dimensionality reduction operation has been done by removing one or more smallest singular values from singular matrix S and also deleted the same number of columns and rows from U and V, respectively. The purpose of the dimensionality reduction is to reduce the noise and unimportant details in the data so that the underlying semantic structure can be used to compare the content of essays. The truncated SVD matrices have been used for making the training essay vectors.

B. Generalized Latent Semantic Analysis

LSA forms a word by document matrix and then the matrix is decomposed using Singular Value Decomposition (SVD) technique. The author of Paper 1 and 3 have developed an AES system using Generalized Latent Semantic Analysis (GLSA) which makes n-gram by document matrix instead of word by document matrix. In GLSA n-gram by document matrix is created instead of a word by document matrix of LSA. According to GLSA, a bi-gram vector for “carbon dioxide” is atomic, rather than the combination of “carbon” and “dioxide”. The GLSA preserve the proximity of word in a sentence. We have used GLSA because it generates clearer concept than LSA.

C. Update SVD by adding columns

Suppose the matrix C, which consists of additional columns, will be added to the original matrix M, then we firstly apply traditional SVD on M and get the result of formula (1). With being mathematically proved, by adding additional C, we can get:

$$[M\ C] = [U\ J] \begin{bmatrix} \Sigma & L \\ 0 & K \end{bmatrix} \begin{bmatrix} V^T & 0 \\ 0 & I \end{bmatrix}^T \tag{3}$$

Where $L = U^TC$. Let $H = C - UL$, then by QR decomposition or other methods upon H, we will get $H \rightarrow JK$. The matrix in the middle $\begin{bmatrix} \Sigma & L \\ 0 & K \end{bmatrix}$ will be continually applied with traditional SVD. Finally, we multiply all medium results and get the updated result: [26]

$$[M\ C] = [U\ J]\ U' \cdot \Sigma' \cdot \begin{bmatrix} V^T & 0 \\ 0 & I \end{bmatrix}^T V'^T \ = U'' \cdot \Sigma' \cdot V''^T \tag{4}$$

IV. AES: SYSTEM ARCHITECTURE

The author of Paper 3 have developed a system for essay grading using Generalized Latent Semantic Analysis (GLSA). Their whole system architecture has been spited into two main parts: generation of training essay set and evaluation of submitted essay. Fig. 2 shows training set generation where essays are graded first by more than one human experts of that subject. The number of human graders may increase for the non-biased system. The average value of the human grades has been treated as training score of a particular training essay. The preprocessing has been done on training essay set. In preprocessing steps the stop-words have been removed from the essay and words have been stemmed to their roots. For stemming we have used M. F. Porter's stemming algorithm.

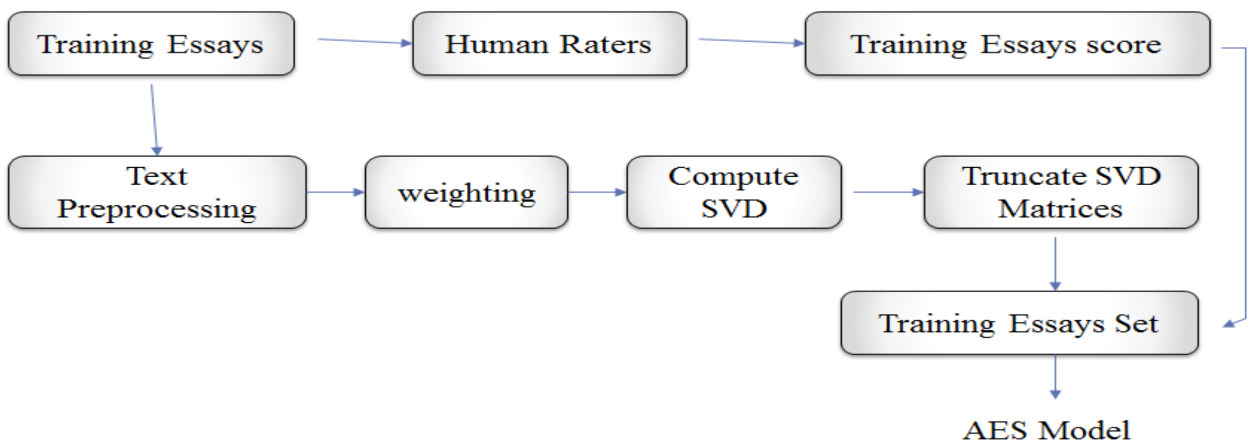

Figure 2. AES Model Generation

V. THE PROPOSED METHOD

Traditionally, LSA can be performed through five sub-steps and these five sub-steps are text preprocessing, weighting, calculating SVD, correlation measurement and correlation method [35].

A. Text pre-processing

Text pre-processing is the first sub-step and it includes two parts: segmenting and producing *t-d* matrix. The *t-d* (term-document) matrix is based on the amount of words (terms) appeared in the essays (documents). For example, $a_{i,j}$ is the number of times of the i^{th} word appeared in the j^{th} essay in the dataset. Additionally, eliminating stop-words is a necessary step, since they are of high frequencies without real meaning, such as preposition, verbal auxiliaries and so forth.

In Paper 2 each column as an *essay vector* which is denoted as *dj* , whose feature is the number of words in the training set or that of latent semantics. Because the essays in the dataset are represented as texts without space as delimiter of words, it is necessary to segment the texts reasonably.

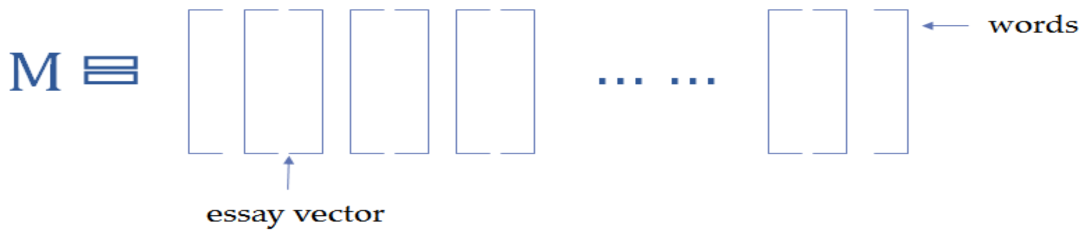

Figure 3. Original dataset matrix

There are many methods that have been developed, for example, The author of Paper 1 and 3 use N-gram-based method for disperse string detection. In this approach some important words as unigram and then their neighbors' are being selected. For example for bigram the first and second neighbors' of the selected words have been selected, for trigram first, second and third neighbors' have been selected. In this way n-grams have been generated. The n-gram by document matrix has been created by using the frequency the n-gram present in an essay. Each cell of the matrix has been filled by the frequency of n-grams in the document.

After the *t-d* matrix of the dataset has been generated, the word corpus will be confirmed as well. The subsequent steps, including matrix weighting, will base on this word corpus. Specifically, the training set and the testing set are sharing the same word corpus confirmed by the training set.

B. Weighting

The TF-IDF matrix, where TF stands for the term frequency and IDF for inverse document frequency, is a common method for weighting [37]. Every element in the *m-n* matrix can be weighted as

$$W_{i,j} = TF_{i,j} \ x \ IDF_{i,j} \tag{5}$$

$$\text{Term Frequency, } \mathrm{TF}_{i,j} = \frac{\mathrm{num}_{i,j}}{\sum_{k=1}^{m} \mathrm{num}_{k,j}} \tag{6}$$

where $\mathrm{num}_{i,j}$ is the number of the i^{th} word appeared in the j^{th} essay, and *m* is the number of the words in the training set. As to IDF, it can be calculated as

$$IDF_{i,j} = log \frac{n}{1 + DF_i} \tag{7}$$

where *n* is the size of the training set (or the amount of essay vectors), and DF_i is the number of essays which contain the *i*-th word. For generating the TF-IDF matrix of the testing set, a special situation exists that no essay contains the i^{th} word, so, in case of dividing zero, 1 is added to the denominator.

C. Calculating SVD

However, calculating SVD becomes a hard even impossible task when faced with a massive dataset. To solve this effectively the author of Paper 2 use incremental LSA. Its flowchart is given in Fig. 4. At first, all the essays will be segmented into words and constructed into the essay vectors which form the original dataset matrix as Fig. 3 shows. Next, applying incremental algorithm on that matrix will establish a semantic space, where all the essay vectors can be re-projected. Finally, all the essay vectors with their human scoring results as labels will be sent into the support vector machine for training. Thus, all the essays in the dataset will have their own predicted automated scoring.

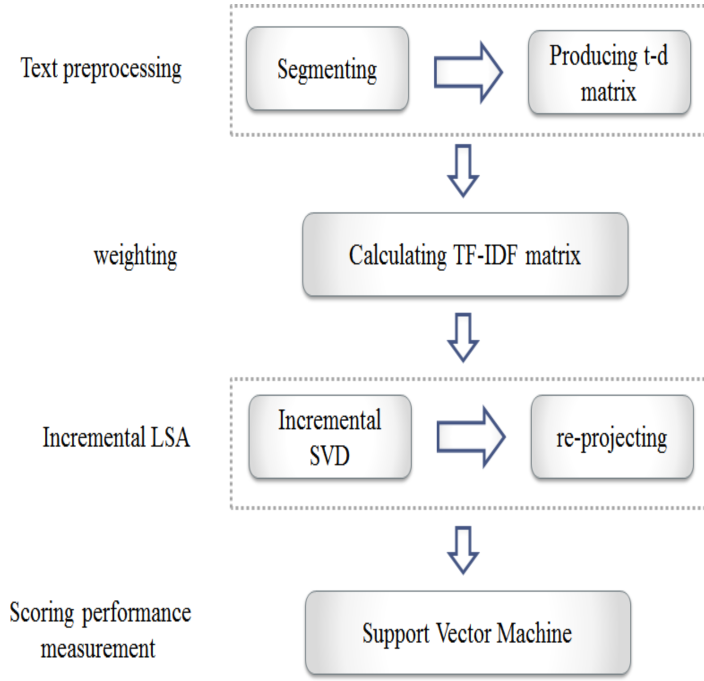

Figure 4. The flowchart of the proposed method

Incremental LSA is composed of two parts: incremental decomposition of a dictionary-based space and re-projection of any essay vector under the reconstructed semantic space. The first part can be completed by incremental SVD, which avoids the synonyms and ambiguities about words and enables the essay vectors to be projected onto the low-dimension semantic space. The second part is called re-projection, and any essay vector based on term-frequency can be represented under that space.

These components will be described below.

1. Incremental SVD: The core of LSA is performing SVD on TF-IDF matrix for training. Algorithm 1 is our method of incremental SVD. We get the first *M* essay vectors, called *initialized value*, to apply the conventional SVD, which is faster because of the less essay vectors and hence far too smaller matrix. Then, we obtain a mediate result of decomposition.

The next step is to update the mediate result by adding *N* columns, called *batch size*, from the rest part of TFIDF matrix. According to the mathematical derivation we introduced previously, it is easier and faster to update the mediate result, and it has been proved that this updating result is approximate to the conventional SVD of the original matrix

[31]. This process of updating will repeat until all the essay vectors in the TF-IDF has been added to the mediate result. Finally, we can get the result of incremental SVD.

During the process, an important problem is to maintain necessary semantics used to construct the semantic space. As the decomposition goes, the rank of $\sum$ will possibly increase, and thus, the redundant semantics will be produced. Producing too much noise will not only lower the preciseness of the semantic space, but also has impact on computational performance, i.e. larger memory usage for saving data and longer running time. Therefore, we maintain *k* latent semantics, called *threshold value*, to guarantee the effectiveness during the updating procedure and removing useless semantics immediately.

2) Re-projecting: Applying formula (2), where *k* is the threshold value or the number of semantics remained, will construct a semantic space of the dataset. Any essay vector based on term-frequency and the same word corpus d_j , can be re-projected to the semantic space. Suppose that $U \cdot \sum \cdot V^T$ is the final result of incremental SVD performed on TF-IDF matrix of the training set, and d_j is an essay vector based on the same word corpus. Then by Formula (8), we will re-project d_j to the semantic space as :

$$\widehat{d_J} = \sum^{-1} \text{ x } U^T \text{ x } d_j \tag{8}$$

D. Correlation Method:

The author of Paper 3 determines the score of the submitted essays by finding the maximum value cosine similarity with the training essay. The author Paper 2 use Support Vector Machine (SVM) to automatically score the essays. The incremental method is effective for processing massive datasets, especially for automated essay scoring. The word corpus is established on the training set, which is used to produce *t-d* matrix for the testing set. Additionally, all the essay vectors in the testing set will be re-project to the semantic space constructed by the training set.

According to [38], the optimized solution of SVM can be expressed as follows:

$$\min_{w,b,\aleph} \ \frac{\mathbf{1}}{\mathbf{2}} \boldsymbol{\omega}^{\boldsymbol{T}} \boldsymbol{\omega} + \boldsymbol{C} \sum_{\boldsymbol{i=1}}^{\boldsymbol{l}} \xi_i \tag{9}$$

Constrains:

$$y_i (\boldsymbol{\omega}^{\boldsymbol{T}} \ \emptyset(x_i) + b) \geq 1 - \xi_i$$

$$\xi_i \ \geq 0, i = 1, \ldots\ldots\ldots\ldots, l$$

where y_i can be used as human scoring in our experiments. The number *n* is the threshold value, $xi \in R^n$ is the essay vector which will be mapped into a high-dimensional space by $\phi(xi)$, and *C* is the regularization parameter.

VI. EXPERIMENTS

A. Environment and datasets

In Paper 2 for text pre-processing a tool WFST written in C++ language [36] is used to segment the essays. In the incremental procedure, MATLAB and C language are used to compute matrices. Finally LIBSVM [39] is used to complete automated scoring. For the training set, the corresponding human scores (from 0 to 6 at intervals of 0.5) will be added to the essay vectors in order to train the support vector machine model. The author of Paper 3 use 960 essays written by undergraduate students have been used to train the model. The model is tested by 120 essays. The total mark is 100. The score of each essay ranged from 2 point to 4 points, where a higher point represented a higher quality.

B. Computing time

From the experimental result GLSA based approach as proposed by the author of Paper 3 seems to perform slower compare to traditional LSA based approach due to performing additional steps mentioned in the algorithm.

Table I and Figure 5 shows the experimental results for Incremental LSA based approach as proposed by the author of Paper 2. It shows that when the size grows larger and larger, incremental SVD is far more efficient than conventional SVD. When the size grows to 110,432, the running time of conventional SVD is more than two hours, as is shown in Table III and Fig. 5, so it is obvious that incremental SVD performs much better.

TABLE I. Comparison of conventional SVD and incremental SVD

Training set size	Time(s)	
	Incremental	Conventional
31552	141.661	1111.324
47328	231.759	1692.691
63104	339.327	2617.137
78880	460.742	3858.184
94656	592.244	5466.019
110432	680.372	7929.217
126208	824.693	N/A
141984	978.254	N/A
157760	1145.381	N/A

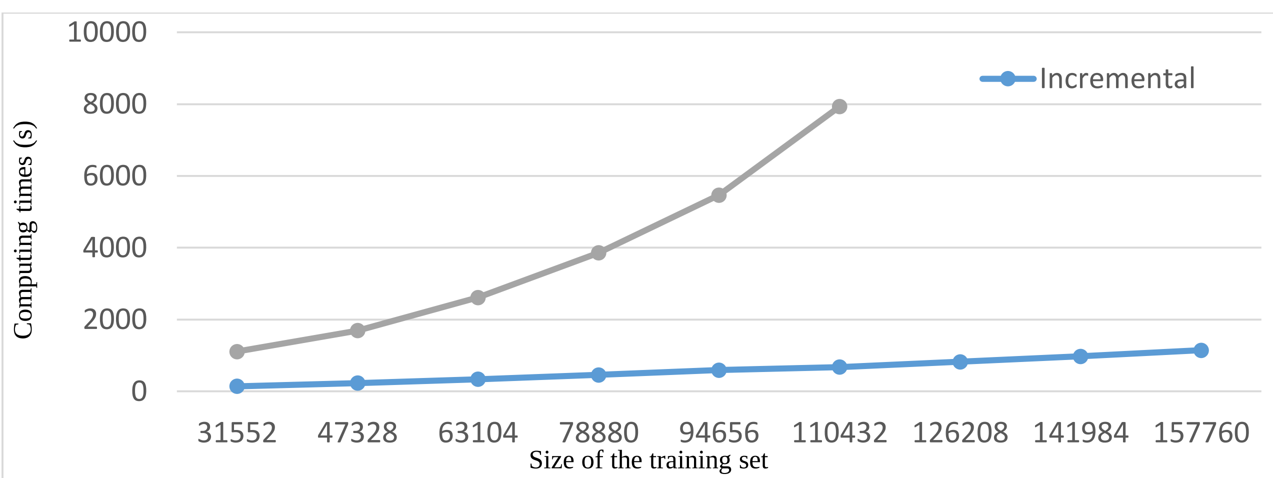

Figure 5. Comparison of conventional SVD and incremental SVD

C. Memory usage

In addition to computing time, economic memory usage is another huge advantage of incremental SVD. Fig. 6 shows the comparison of these two methods in memory usage as the size of the training set grows. From Fig. 7, we can see that the maximum memory usage of incremental method as described by the author of Paper 2 is only 492M and incremental method performs relatively stable. In contrast, conventional method uses much more memory and increases distinctly. In practical application, language tests usually produce a massive dataset, so even given huge memory, the decomposition task of conventional SVD seems impossible, but it is viable for incremental SVD to perform it.

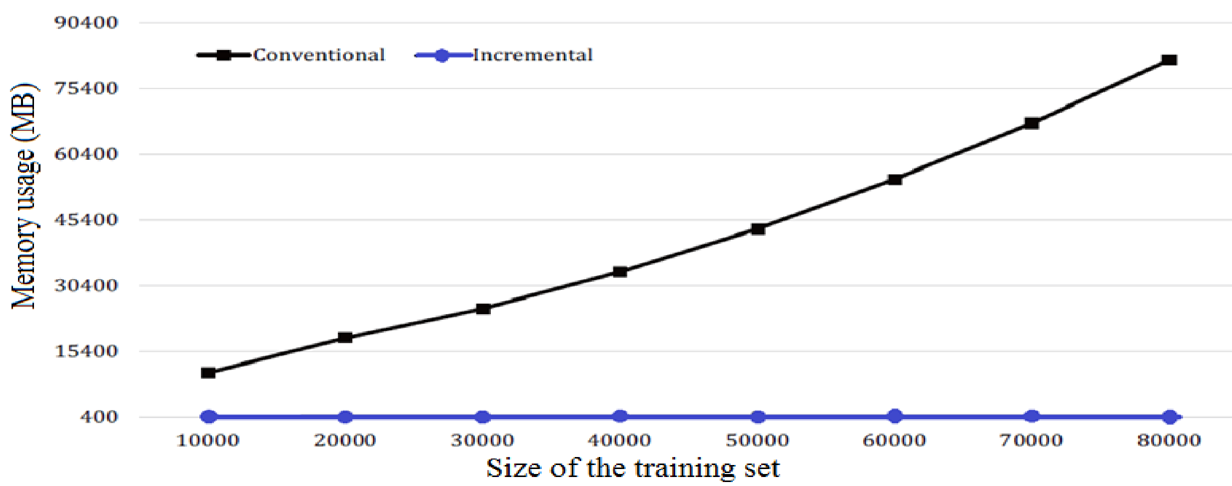

Figure 6. Comparison of conventional and incremental SVD in memory usage

D. Scoring performance

The performance of a method for scoring essays can be evaluated by measuring how much the automated score closer to the human score. The more closely the automated score to the human score is more accurate. The performance for GLSA based approach as proposed the author of Paper 3 are shown in TABLE II. From table it seen that their system perform well.

TABLE II: Performance of AES using GLSA

Human Score	No. of Test Essay	GLSA Score					
		4.00	3.50	3.00	2.50	2.00	0.00
4.00	20	10	5	5	0	0	0
3.50	22	5	10	5	2	0	0
3.00	20	0	8	7	5	0	0
2.50	20	0	0	4	9	5	2
2.00	20	0	0	1	5	10	4
0.00	18	0	0	0	1	2	14

TABLE III shows the comparison table of GLSA based approach with traditional LSA.

TABLE III: comparison of LSA with GLSA

Experiments	LSA score	GLSA score
Mean of Errors	0.80	0.33
Standard deviation of Errors	0.82	0.22

The Author of Paper 2 calculates scoring accuracy in following way

$$\text{Scoring accuracy} = \frac{\sum_{i=1}^{n} \text{t(hsi , psi))}}{n} \tag{10}$$

Where n is the size of the testing set, and hs_i and ps_i are the human scoring and the automated scoring of the i^{th} essay respectively. The function $t(hs_i, ps_i)$ is binary which can be defined as:

$$t(hsi, psi) = \begin{cases} 1 & |hs - ps| \leq 1 \\ 0 & Otherwise \end{cases} \tag{11}$$

From the experimental results of the author of Paper 2 verifies that incremental SVD is as effective as conventional method, and both their values of scoring accuracy are 88.8%, which is acceptable.

VII. CONCLUSIONS

Latent semantic analysis is of prevalence in the fields of natural language processing. But Document representation as Generalized Latent Semantic Analysis (GLSA) vectors are able to give performance improvement on several task such compared to traditional Latent Semantic Analysis (LSA) based systems.

The Author of Paper 3 and Paper 4 proves experimentally that GLSA based essay scoring system have better accuracy than LSA based essay scoring system. It minimalize difference between grading done by human grader and computer based system, higher threshold of Pearson Product Moment Correlation. But the actual problem in GLSA based approach which I think is it slightly slower performance on processing time. Because additional calculation steps require in GLSA based systems make it slower than traditional LSA based systems.

Another main disadvantage in LSA and GLSA based approach is in its primary algorithm, singular value decomposition, however, cannot deal with huge datasets. To solve this problem The Author of Paper 2 introduce a new incremental method of LSA to score essays effectively when the dataset is massive. By comparison of the traditional method and their incremental method, concerning the running time and memory usage, experimental results make it obvious that the incremental method has a huge advantage over the traditional method.

But again Incremental LSA has a problem in accuracy. Unlike time and memory usage it can't increase accuracy further than traditional LSA based approach. So I think in order to improve running time and memory usage we can use ILSA based approach. But to improve accuracy further we should can refine it with combining other method which improve accuracy like approaches with analyze the true meaning of the essays.

ACKNOWLEDGMENT

This overview is based on the work of first 5 papers entitled in references. I am very much grateful to them and also to the anonymous referees fort their valuable suggestions.

REFERENCES

[1] Andrew M. Olney, "Generalizing latent semantic analysis," *IEEE international conference on semantic computing,* 2009

[2] Mingqing Zhang, Shudong Hao and Yanyan Xu, "Automated essay scoring using incremental latent semantic analysis," *Journal of software,* vol. 9, no. 2, 2014

[3] M. M. Islam and A. Hoque, "Automated essay scoring using generalized latent semantic analysis," *Journal of Computers*, vol. 7, no. 3, pp. 616–626, 2012.

[4] Purti Ratna, Henry Artajaya and Anantasatya Adhi, "GLSA based online essay grading system," *IEEE international conference on teaching, assessment and learning for engineering,* 2013

[5] Y. Cao and C. Yang, "Automated chinese essay scoring with latent semantic analysis," *Examinations Research*, vol. 3, no. 1, pp. 63–71, 2007.

[6] H. Peng and Y. Yu, "Research on controlling central rating in net-based scoring of subjective test items," *China Examinations*, no. 6, pp. 3–9, 2013.

[7] A. Fazal, T. Dillon, and E. Chang, "Noise reduction in essay datasets for automated essay grading," in *On the Move to Meaningful Internet Systems: OTM 2011 Workshops*. Springer, 2011, pp. 484–493.

[8] R. A. Hardy, "Examining the costs of performance assessment, *Applied Measurement in Education*, vol. 8, no. 2,pp. 121–134, 1995.

[9] S. Valenti, F. Neri, and A. Cucchiarelli, "An overview of current research on automated essay grading," Journal of Information Technology Education, vol. 2, 2003, pp. 319-330.

[10] Y. Attali, and J. Burstein, "Automated essay scoring with e-rater® V.2," The Journal of Technology, Learning and Assessment, vol. 4, no. 3, 2006.

[11] L. Bin, L. Jun, Y. Jian-Min, and Z. Qiao-Ming, "Automated essay scoring using the KNN algorithm," In Proceedings of the International Conference on Computer Science and Software Engineering (CSSE 2008), IEEE Conferences, 2008, pp. 735–738.

[12] S. Ghosh, and S. S. Fatima, "Design of an Automated Essay Grading (AEG) system in Indian context," In Proceedings of TENCON- 2008, IEEE Region 10 Conference, 2008, pp. 16-21.

[13] T. Kakkonen, N. Myller, J. Timonen, and E. Sutinen, "Comparison of dimension reduction methods for automated essay grading," International Forum of Educational Technology & Society (IFETS), 2008, pp. 275–288.

[14] L. M. Rudner and T. Liang, "Automated essay scoring using bayes' theorem," *The Journal of Technology, Learning and Assessment*, vol. 1, no. 2, 2002.

[15] H. Peng, "The minorities-oriented chinese level test," *China Examinations*, no. 10, pp. 57–59, 2005.

[16] D. Ke, X. Peng, Z. Zhao, Z. Chen, and J. Wang, "Wordlevel- based automated chinese essay scoring method," in *Proceedings of National Conference on Man-Machine Speech Communication*, Xi'an China, 2011, pp. 57–59.

[17] Y. Tonta and H. R. Darvish, "Diffusion of latent semantic analysis as a research tool: A social network analysis approach," *Journal of Informetrics,* vol. 4, no. 2, pp. 166– 174, 2010.

[18] J. McInerney, A. Rogers, and N. R. Jennings, "Improving location prediction services for new users with probabilistic latent semantic analysis," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 2012, pp. 906–910.

[19] W. Wang and B. Yu, "Text categorization based on combination of modified back propagation neural network and latent semantic analysis," *Neural Computing and Applications*, vol. 18, no. 8, pp. 875–881, 2009.

[20] Y. Jin, Y. Gao, Y. Shi, L. Shang, R. Wang, and Y. Yang, "P2lsa and p2lsa+: Two paralleled probabilistic latent semantic analysis algorithms based on the mapreduce model," in *Intelligent Data Engineering and Automated Learning-IDEAL 2011*. Springer, 2011, pp. 385–393.

[21] S. Ismail, R. M. Othman, and S. Kasim, “Pairwise protein substring alignment with latent semantic analysis and support vector machines to detect remote protein homology,” in *Ubiquitous Computing and Multimedia Applications*. Springer, 2011, pp. 526–546.

[22] L. Lin, B. Liu, X. Wang, X. Wang, and B. Tang, “Protein remote homology detection and fold recognition based on features extracted from frequency profiles,” *Journal of Computers*, vol. 6, no. 2, pp. 321–328, 2011.

[23] S. Zhong, M. Shang, and Z. Deng, “A design of the inverted index based on web document comprehending,” *Journal of Computers*, vol. 6, no. 4, pp. 664–670, 2011.

[24] L. Wang and Y. Wan, “Sentiment classification of documents based on latent semantic analysis,” in *Advanced Research on Computer Education, Simulation and Modeling*. Springer, 2011, pp. 356–361.

[25] C. Liu, Y. Wang, F. Zheng, and D. Liu, “Using lsa and text segmentation to improve automatic chinese dialogue text summarization,” *Journal of Zhejiang University Science A*, vol. 8, no. 1, pp. 79–87, 2007.

[26] J. Yeh, H. Ke, and W. Yang, “Chinese text summarization using a trainable summarizer and latent semantic analysis,” in *Proceedings of the 5th international conference on Asian digital libraries (ICADL’02)*. Springer, 2002, pp. 76–87.

[27] A. Mesaros, T. Heittola, and A. Klapuri, “Latent semantic analysis in sound event detection,” in *19th European Signal Processing Conference*, 2011, pp. 1307–1311.

[28] G. Gorrell, “Generalized hebbian algorithm for incremental singular value decomposition in natural language processing.” in *Proceedings of the 11th Conference of the European Chapter of the Association for ComputationalLinguistics (EACL-06)*, 2006, pp. 97–104.

[29] A. Atreya and C. Elkan, “Latent semantic indexing (lsi) fails for trec collections,” *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 2, pp. 5–10, 2011.

[30] Y. Li, “On incremental and robust subspace learning,” *Pattern recognition*, vol. 37, no. 7, pp. 1509–1518, 2004.

[31] M. Brand, “Incremental singular value decomposition of uncertain data with missing values,” in *Computer Vision ECCV 2002*. Springer, 2002, pp. 707–720

[32] T. J. Chin, K. Schindler, and D. Suter, “Incremental kernel svd for face recognition with image sets,” in *7th International Conference on Automatic Face and Gesture Recognition. FGR.* IEEE, 2006, pp. 461–466.

[33] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Incremental singular value decomposition algorithms for highly scalable recommender systems,” in *5th International Conference on Computer and Information Science*, 2002, pp. 27–28.

[34] M. Brand, “Fast online svd revisions for lightweight recommender systems,” in *SIAM International Conference on Data Mining*, 2003, pp. 37–46.

[35] F. Wild, C. Stahl, G. Stermsek, and G. Neumann, “Parameters driving effectiveness of automated essay scoring with lsa,” in *9th Internaional Computer-Assisted Assessment Conference*, 2005, pp. 485–494.

[36] S. Hao, Z. Gao, M. Zhang, Y. Xu, H. Peng, K. Su, and D. Ke, “Automated error detection and correction of chinese characters in written essays based on weighted finite-state transducer,” in *12th International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2013, pp. 763–767.

[37] M. Sahami and T. D. Heilman, “A web-based kernel function for measuring the similarity of short text snippets,” in *Proceedings of the 15th international conference on World Wide Web*. ACM, 2006, pp. 377–386.

[38] X. Peng and Y. Wang, “Cch-based geometric algorithms for svm and applications,” *Applied Mathematics and Mechanics*, vol. 30, no. 1, pp. 89–100, 2009

[39] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1– 27:27, 2011, software available at http://www.csie.ntu.edu/tw/~cjlin/libsvm.