Contextualized Latent Semantic Indexing: A New Approach Applied to Automated Chinese Essay Scoring

SHUDONG HAO, BIN HUANG, YANYAN XU, Beijing Forestry University
DENGFENG KE, Chinese Academy of Sciences
KAILE SU, Griffith University

As Chinese becomes more popular, the scales of related tests are growing larger than ever before, and the deficiency of conventional human scoring becomes more obvious. Writing assessment in Chinese language tests is in badly need of a mature automated essay scoring system. Although automated English essay scoring systems have been widely used, there has been relatively little research on automated Chinese essay scoring. In this paper, we propose a new approach applied to automated Chinese essay scoring, called Contextualized Latent Semantic Indexing (CLSI). Genuine CLSI and Modified CLSI are two versions of this new approach. The *N*-gram Language Model and Weighted Finite-State Transducer (WFST), two critical components, are used to extract the context information. Not only does CLSI improve conventional Latent Semantic Indexing (LSI), but bridges the gap between latent semantics and their context information which is absent in LSI. Moreover, CLSI can score essays from the perspectives of language fluency and content, and addresses the local overrating and underrating problems caused by LSI to some extent. Experimental results show that CLSI outperforms LSI and other methods from many aspects, and thus, proves to be an effective approach in automated Chinese essay scoring.

Categories and Subject Descriptors: I.2.7 [**Artificial Intelligence**]: Natural Language Processing

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Automated Chinese Essay Scoring, Context Information, Educational Assessment, Educational Measurement

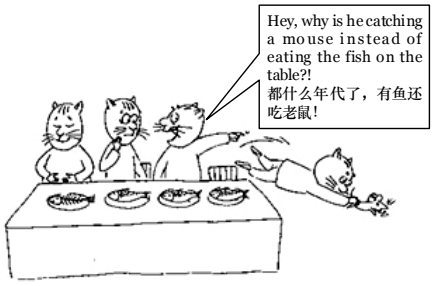
ACM Reference Format:
Shudong Hao, Bin Huang, Yanyan Xu, Dengfeng Ke, and Kaile Su, 2014. Contextualized Latent Semantic Indexing: A New Approach Applied to Automated Chinese Essay Scoring. *ACM Trans. Asian Lang. Inform. Process.* 9, 4, Article 1 (February 2014), 33 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

In language tests, writing is the most crucial part to evaluate a test taker's language proficiency. Generally, there are two kinds of writing tests. One requires test takers to write using their native languages, such as the writing part in The National

This work is supported by the Fundamental Research Funds for the Central Universities, under grant YX2014-18, the Beijing Higher Education Young Elite Teacher Project, under grant YETP-0768 and the National Natural Science Foundation of China, under grant 61103152 and 61472369.
Authors' addresses: S. Hao, B. Huang and Y. Xu (the corresponding author), School of Information Science and Technology, Beijing Forestry University, Beijing, China, email: xuyyxu@gmail.com; D. Ke, Institute of Automation, Chinese Academy of Sciences, Beijing, China; K. Su, Institute for Integrated and Intelligent Systems, Griffith University, Brisbane, Australia.

DOI: http://dx.doi.org/10.1145/0000000.0000000



根据上面的图片，写一篇不少于800字的作文。文体不限（诗歌除外）。
Write an essay inspired by the picture above. There are no limitations to genres except poems. An effective response must contain a minimum of 800 words.

The National Higher Education Entrance Examination

一位老和尚为了给自己选一个接班人，对他的两个徒弟说：“去找一片你们最满意的树叶回来。”结果，第一个徒弟很快就回来了，递给师傅一片树叶，说：“虽不完美，但却是我看到的最好的树叶。”第二个徒弟很晚才归，两手空空，对师傅说：“我见到的树叶很多，但没有一片是完美的，所以没有一片是我最满意的。”老和尚看着第一个徒弟满意地笑了。

请根据上面的内容，结合自己的认识，写一篇读后感。自定题目，全文不少于350字。
Please write an essay, rephrasing the fable and expressing your thoughts according to your experiences. An effective response must contain a minimum of 350 words.

The Minorities-Oriented Chinese Level Test (MHK)

Fig. 1. Examples of two kinds of writing tests.

Higher Education Entrance Examination in mainland China. This kind of test usually gives test takers an enlightening picture or a statement which can inspire deeper thoughts, so that test takers can write down their own thoughts combining reasoning or narration with personal experiences. The other kind of writing test requires test takers to write using foreign languages, such as The Minorities-Oriented Chinese Level Test (MHK) [Peng 2005], whose purpose is to test the test takers' basic abilities to use a foreign language, so they are required to write relatively easier essays, expressing their thoughts according to a specific topic or scene. Figure 1 shows these two kinds of writing tests respectively.

How to evaluate an essay in an objective and effective way is important for both educators and researchers. For the first type of writing, because of different genres (such as Argumentation and Narration), various contents and sophisticated writing, there always exist different opinions about one essay. For the second type, however, the essays are relatively simple, leading to less disagreements. Similar to computers, to accurately score essays of the first type is much more difficult than those of the second one. Therefore, the strategies developed and discussed in this paper are used to score the second type of essays, especially those in the MHK tests.

Traditionally, one essay needs to be assessed by at least two human raters, who are trained by studying the scoring criteria and samples of essays with different scores. If the difference between them surpasses an acceptable bias, the essay will be sent to an expert to get a final score. Otherwise, the essay's score will be the average of those two scores. When the scale of this test grows larger, this method is obviously time-consuming, labor-consuming and far less accurate caused by subjective opinions and rater fatigue. Although these disadvantages can be controlled to some degree, it is still difficult for traditional human scoring to deal with larger-scale tests. However, automated essay scoring, a powerful intelligent system, has advantages over traditional human scoring in:

— less time consuming;
— less human labor; and
— less subjective judgments.

Therefore, automated Chinese essay scoring necessarily becomes an indispensable part of computer-aided education, from classrooms to large-scale tests.

1.1. Background

Automated essay scoring originated from related research on English [Shermis and Burstein 2003]. Earlier to 1970s, Ellis Page has developed Project Essay Grader (PEG) for large-scale essay scoring upon the request of the College Board, which broke the dawn of the field of scoring by intelligent systems and computers [Page 1994]. Since then, many studies have been conducted to improve the effectiveness and accuracy, along with some commercial softwares emerged in the education market and applied to the industry. Automated essay scoring not only assists testers to assess the proficiency of test takers, but also helps faculties and teachers to get knowledge about their students.

Several representative applications in automated English essay scoring have been applied to educational markets. PEG utilizes surface information to score essays, such as the length of passages, the number of words and misspellings in an essay, *etc.*, but ignores deeper semantic structures. It is this methodology that makes PEG vulnerable: students can use more complex words, and write longer passages without any logical contents within essays to trick the system and get high scores. Intelligent Essay Assessor (IEA) utilizes Latent Semantic Indexing (LSI) to avoid the deficiencies of PEG and focuses on semantics instead of surface features [Landauer et al. 2001]. It uses such information as pre-scored essays, expert model essays and knowledge source materials to analyze essays [Landauer et al. 2003]. Educational Testing Service (ETS) applies its E-rater to the writing section in the Test of English as a Foreign Language (TOEFL) and analytical writing in Graduate Record Examination (GRE) [Burstein et al. 2003; Attali and Burstein 2006; Ramineni et al. 2012]. E-rater first trains a model using a corpus containing articles from newspapers. Then it uses three modules – syntax, discourses and topical-analysis – to analyze essays. IntelliMetric, the first automated scoring system based on artificial intelligence combining Natural Language Processing (NLP) with statistics, was developed by Vantage Learning. It is supported by two methods proposed by Vantage Learning, one of which is CogniSearch, which enables IntelliMetric to understand natural languages, and the other is Quantum Reasoning, which can discover the characteristics of each score point [Elliot 2003; Learning 2001; 2003]. Based on statistics and the Bayesian theorem, Lawrence M. Rudner developed the Bayesian Essay Test Scoring sYstem (BETSY) [Rudner and Liang 2002]. Similarly, BETSY uses large samples to train models, and evaluates database statistics and determines the classification accuracy. The essay to be assessed will get a relative level such as Pass or Fail.

Due to that automated English scoring emerged earlier and it is relatively easier to process English texts than Chinese ones, the automated scoring systems for English have been applied widely. Automated Chinese essay scoring which started later, however, is left behind.

1.2. Related Work

Automated Chinese essay scoring attaches much more attention currently. Especially, when the scale of a Chinese language test grows larger, it is more necessary to replace conventional human scoring by automated scoring. Unfortunately, there still lacks such a mature architecture caused by difficulties in processing Chinese texts.

Hurdle 1. For Chinese text processing, the first hurdle is segmentation. Unlike English or other western languages, Chinese does not provide inter-word delimiters within a sentence, leading to that one sentence can have different meanings according

南京市长江大桥。

Segmentation 1: 南京／市长／江大桥。
(Daqiao Jiang is the major of Nanjing City.)
Segmentation 2: 南京市／长江大桥。
(The Yangtze River Bridge in Nanjing.)

Fig. 2. One sentence can have two ambiguous meanings according to different segmentations.

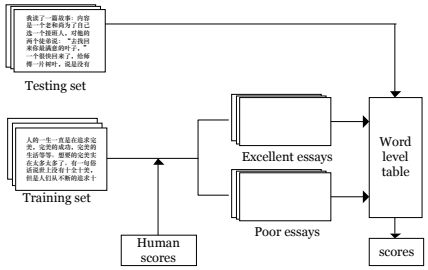


Fig. 3. Feature extraction using a word level method.

to different segmentations, as Figure 2 shows. If a word-based analysis technique is built on a low-precise segmentation, all improvements on the subsequent methods are useless. It is this limitation that makes research on Chinese text processing need an extra step as pre-processing, such as domain ontology [Chen and Huang 2013], Chinese text proofread [Zhang et al. 2001], machine translation [Chu et al. 2013] and so forth. Currently, there are several methods focusing on Chinese word segmentation, among which, the integrate generative and discriminative character-based model proposed in [Wang et al. 2012] is the most comprehensive one. The methodology, however, should be adapted to automated Chinese essay scoring. Therefore, along with a stricter requirement on segmentation, automated Chinese essay scoring is faced with this problem.

Hurdle 2. The second hurdle for automated Chinese essay scoring is feature extraction, which is also faced by automated essay scoring for any other language. Inspired by the studies in English, similar methods have been researched from various angles [Li 2006]. For instance, the word level method [Ke et al. 2011; Peng et al. 2012] uses statistics and predicts scores according to the overall level of word usage, assuming that excellent essays use more suitable and higher-level words whereas poor essays use ordinary even improper words, as shown in Figure 3. This method, though, seems appropriate and advanced since it combines statistical strategies and reasonable assumptions, it still works on surface information without deeper analysis on contents. Depending on word usage will make a system vulnerable as PEG. Topic feature extraction methods using topic modeling strategies like Regularized Latent Semantic Indexing (RLSI) [Wang et al. 2013; Hao et al. 2014b] and Latent Dirichlet Allocation (LDA) [Blei et al. 2003; Hao et al. 2014b] are novel ones. Other methods also have been tested but their results are relatively rare [Cai et al. 2010]. Semantic feature extraction using LSI [Cao and Yang 2007; Zhao 2011] explores the contents reflected by words, which is the most classical and widely applied one among these novel methods.

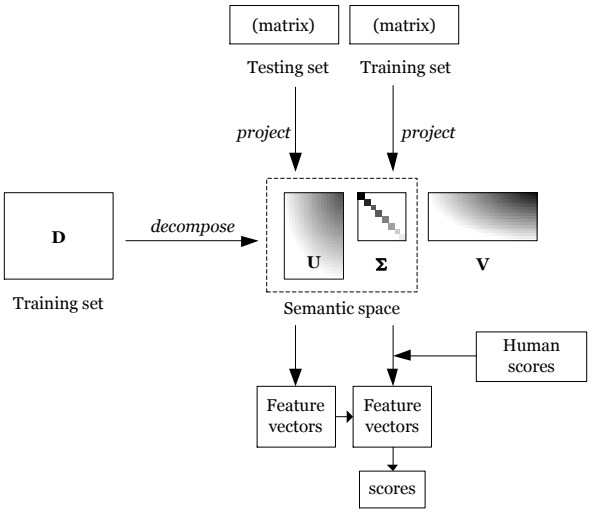


Fig. 4. Feature extraction using LSI. The feature vectors of the training set and testing set can use various methods to predict scores, such as similarity and classification strategies.

Currently, several LSI-based methods have been applied to automated essay scoring [Cao and Yang 2007; Islam and Hoque 2012; 2012], which are:

(1) *Conventional LSI*
LSI was proposed by Scott Deerwester, et al in [Deerwester et al. 1990], which starts with a term-document matrix $\mathbf{D}$ where every row $\mathbf{d}_{i,:}$ is a word and every column $\mathbf{d}_{:,j}$ is an essay. Typically, $\mathbf{D}$ is weighted by the Term Frequency – Inverse Document Frequency (TF-IDF) method [Nakov et al. 2001]:

$$\begin{aligned} d_{i,j} &= TF_{i,j} \cdot IDF_{i,j} \\ &= \frac{count_{ij}}{\mid \mathbf{d}_{:,j} \mid} \cdot \log \frac{\mid \mathcal{D} \mid}{\mid \{\mathbf{d} \in \mathcal{D} : i \in \mathbf{d}\} \mid} \end{aligned}$$

where $count_{ij}$ is the number of the i-th word appeared in the j-th essay, and $\mid \mathbf{d}_{:,j} \mid$ is the number of words in the j-th essay, and $\mid \mathcal{D} \mid$ is the number of essays in the dataset, and $\mid \{\mathbf{d} \in \mathcal{D} : i \in \mathbf{d}\} \mid$ is the number of essays where the i-th word appears. Next, LSI uses a mathematical strategy called Singular Value Decomposition (SVD) to construct a semantic space by

$$\mathbf{D} \approx \mathbf{D}^* = \mathbf{U}^* \mathbf{\Sigma}^* \mathbf{V}^{\top *}$$

where $\mathbf{\Sigma}^*$ is a diagonal matrix with $\sigma_{1,1} \geqslant \sigma_{2,2} \geqslant \ldots \geqslant \sigma_{k,k}$ as the diagonal elements, and $\mathbf{D}^*$ is the least square best-fit approximation of $\mathbf{D}$. Finally after transformation, we can train a scoring model. See how LSI works in Figure 4.
LSI focuses on the word usage and the contents it reflects, solving the problem that "*What are these words?*". The deficiency of LSI is that it ignores the contexts where the words appear. According to the essays in MHK tests, sometimes there are few distinct differences between a higher-score essay and a lower-score essay from the perspective of word usage and occurrence merely. The reason may be that the essay with a higher score uses the words in an appropriate order and context, while that with a lower score does not. If two essays have identical word occurrence but they are used in different contexts, they will get the same features

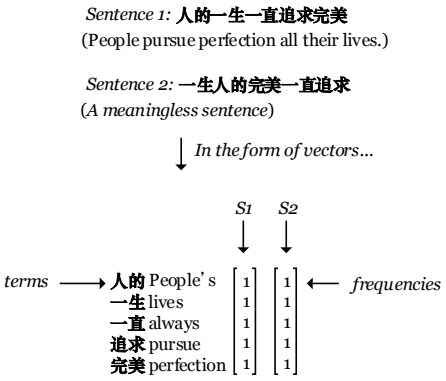


Fig. 5. An example showing that a meaningless sentence ($Sentence_2$) shares the same occurrence information (vectors S_1 and S_2) with a normal sentence ($Sentence_1$).

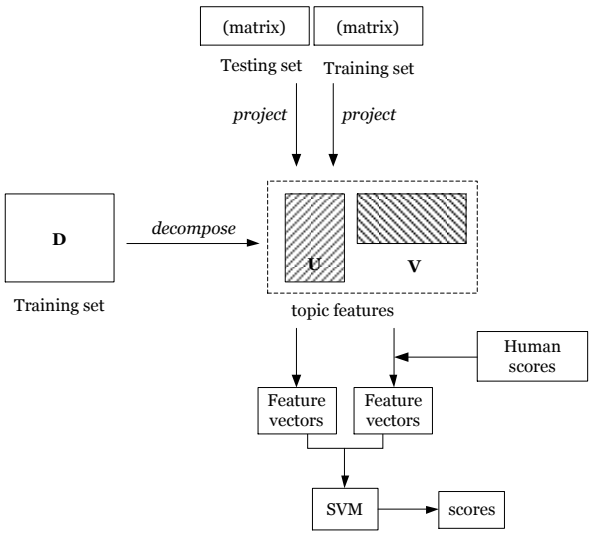


Fig. 6. Topic feature extraction using RLSI.

when applying LSI on them, as Figure 5 shows. Consequently, the higher-score and the lower-score essays will get the same automated scores, and it causes the so-called *local overrating* problem when low-score essays cannot be recognized, or the *local underrating* problem when high-score essays cannot be recognized.

(2) *Generalized LSA*
Considering the lacked context information in conventional LSI, a method called Generalized Latent Semantic Analysis (GLSA) has been proposed [Islam and Hoque 2012]. By using n-gram as the atomic element of each row, it produces a very huge and sparse matrix D as the learning sample. For flexible languages such as Chinese, it is computationally prohibitive.

(3) *Regularized LSI*

Regularized LSI (RLSI) is a recently proposed method in [Wang et al. 2013] in topic modeling and information retrieval, including the batch version and the online version. The first try of scoring Chinese essays from the topic perspective has been proposed, using the batch version of RLSI [Hao et al. 2014b]. The target of RLSI is to decompose a term-document matrix $\mathbf{D}$ into a term-topic matrix $\mathbf{U}$ and a topic-document matrix $\mathbf{V}$, minimizing the Frobenius norm $\|\mathbf{D} - \mathbf{UV}\|_F^2$ with ℓ_1-norm regularization on $\mathbf{U}$ and ℓ_2-norm regularization on $\mathbf{V}$. It can be described as:

$$\min_{\mathbf{U},\mathbf{V}}(\|\mathbf{D} - \mathbf{UV}\|_F^2 + \lambda_1\|\mathbf{U}\|_1 + \lambda_2\|\mathbf{V}\|_F^2) \tag{1}$$

where $\mathbf{U}$ reflects the relations between documents and latent topics. In [Hao et al. 2014b], RLSI is combined with a Support Vector Machine (SVM) to score Chinese essays. See how RLSI is applied to automated Chinese essay scoring in Figure 6. RLSI is a good method to discover the topics in a collection of test essays and is effective to score essays from topic perspective, but a comprehensive scoring system requires more than one perspective.

1.3. Main Contributions

Facing with the hardships of automated Chinese essay scoring and the limitation of LSI, we propose a new method called Contextualized Latent Semantic Indexing (CLSI), which has three main contributions.

(1) Integrate Chinese text segmentation and context information extraction. Traditionally, in an automated Chinese essay scoring system, one needs to segment essays into words first, and then perform certain algorithms to extract features. In CLSI, we perform feature extraction and text segmentation simultaneously, meaning that we do not need to perform segmentation explicitly and separately.
(2) Score essays from the perspective of language fluency. In MHK tests, there are several scoring perspectives, from the *character and word usage*, *topic*, *language fluency* to the *logical structure* [Peng 2005]. There is no algorithm to score essays from the perspective of language fluency and CLSI will use context information to finish this task.
(3) Address the overrating and underrating problems. As mentioned earlier, LSI may cause the overrating and underrating problems. In language tests, low-score essays weigh much more than ordinary essays, because they can be used to discover the latent learning problems of students and test takers. If a system cannot recognize low-score essays, it will lose important test information, and hurt the fairness of tests to some extent.

Instead of the traditional term-document matrix recording the occurrence information, CLSI uses a probabilistic matrix reflecting the context information from an n-gram language model, solving the problems of both "*What are these words?*" and "*How these words are used?*". More specifically, we propose two versions of CLSI: Genuine CLSI which uses originally written essays and Modified CLSI which modifies possibly erroneous characters in essays. Experimental results show that both genuine and modified versions are effective in feature extraction and automated essay scoring.

The rest of the paper is organized as follows. In Section 2, we take a snapshot on the new methods and explain the terms used in this paper. Then we thoroughly introduce each part of Genuine CLSI and Modified CLSI from Section 3 to Section 5, including the language model construction, CLSI and the SVM-based scoring model. Section 6 shows the experimental results and some discussions. Finally, in section 7 we conclude this paper and point out some future work.

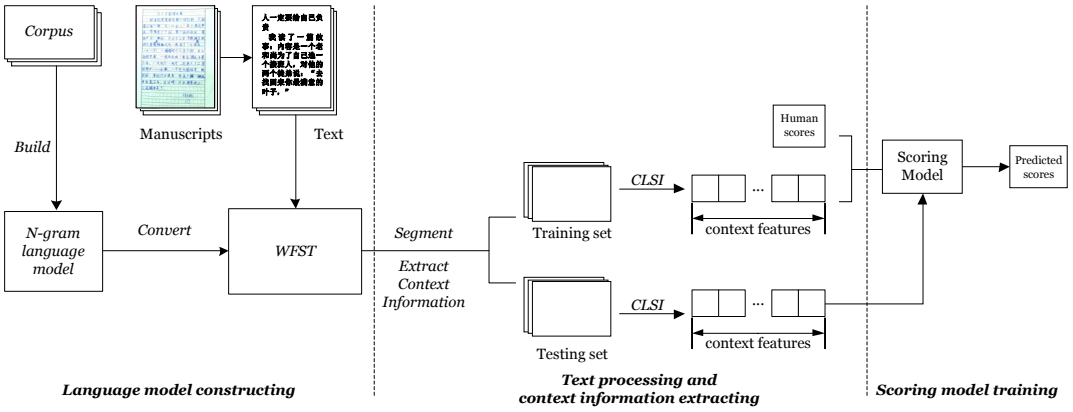


Fig. 7. The Framework of automated Chinese essay scoring using CLSI

2. AN OVERVIEW OF CONTEXTUALIZED LATENT SEMANTIC INDEXING

A common method for automated essay scoring includes three steps:

— Text pre-processing;
— Feature extracting; and
— Scoring model training.

According to the characteristics of Chinese tests, we propose the following framework:

— Language model constructing;
— Context information extracting; and
— Scoring model training.

Figure 7 is the framework we use in this paper. In next sections, we introduce each step respectively, omitting the manually typing stage.

In Language model constructing, we use a large corpus to build an n-gram language model, and then convert it to a Weighted Finite-State Transducer (WFST). In context information extracting, we use WFST to extract context information. If Modified CLSI is used, WFST will use a Confusing-character Table including frequently confusing characters to detect and correct erroneous characters. Then, essays will be organized as a matrix, and transformed under the contextualized-semantic space. The essays, finally, will be sent to SVM to train the scoring model.

In order to be clearer, we introduce some terms and notations as preliminaries for the rest of this paper.

(1) *Essay collection*: When the written essays in the electronic text form are obtained, they are organized as a data set, and we call it a essay collection $\mathcal{D}$. We denote $|\mathcal{D}|$ as the size of $\mathcal{D}$.
(2) *Essay word list*: We define an ordered word list $\mathcal{V}$ containing the words appeared in an essay, and $|\mathcal{V}|$ is the size of $\mathcal{V}$. The elements w_i for $i = 1, \ldots, |\mathcal{V}|$ in $\mathcal{V}$ are ordered by the sequence they occur in the essay, and the list may have identical words.
(3) *Collection word set*: We use $\mathcal{V}^\dagger$ to define a set containing all the words in an essay collection, namely, $\mathcal{V}^\dagger = \bigcup_{i=1}^{|\mathcal{D}|} \mathcal{V}_i$. Similarly, $|\mathcal{V}^\dagger|$ is the size of the $\mathcal{V}^\dagger$. There are no identical elements in $\mathcal{V}^\dagger$.
(4) *Essay vector*: If we have an essay with a word list $\mathcal{V}$, then it can be presented as a vector $\mathbf{d} = [d_1, d_2, \ldots, d_M]^\top \in \mathbb{R}^M$ where $M = |\mathcal{V}|$. If d_i is the term frequency of the i-th term in the essay (like in LSI) or context information (like in CLSI), this

vector is called a term-based essay vector. After mathematical transformation, this vector will be a transformed essay vector. Therefore, in the following discussion, an essay vector can be referred to either a term-based or transformed one, and it can be easily distinguished without ambiguity.

(5) *Essay matrix*: When essays in an essay collection have been processed into essay word lists and presented as essay vectors, they will be arrayed together to a matrix $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, ...\mathbf{d}_N] \in \mathbb{R}^{M \times N}$ where $M = |\mathcal{V}^\dagger|$ and $N = |\mathcal{D}|$. Since M is usually larger than a single essay vector, the corresponding elements in an essay vector where terms do not appear will be filled with certain numbers (usually zero if the term frequency is used). Similarly, a matrix that contains transformed essay vectors is also called an essay matrix, where M will be a certain number such as a specific dimension rather than the size of the word set. In practice, if we separate the essay collection into a training set, a validation set and a testing set, we use $\mathbf{D}$ to denote the essay matrix of the training set, and $\mathbf{Q}$ to denote the essay matrix of either the validation set or the testing set.

(6) *Human score*: A score of an essay given by human raters is called a human score.

(7) *Predicted score*: A score given by an automated essay scoring system is called a predicted score.

3. LANGUAGE MODEL CONSTRUCTION

The first part of CLSI is to build an n-gram language model from a large-scale corpus, which will be used to construct a WFST to recognize specific word sequences of sentences in essays and extract context information.

3.1. *N*-gram Language Model

In NLP, an n-gram language model is a representative statistical inference strategy. Assume that a sentence S is a sequence of words:

$$S = w_1 w_2 \dots w_n. \tag{2}$$

Then predicting the word w_n can be formularized as a probability:

$$P(w_n) = P(w_n | w_1, \dots, w_{n-1}). \tag{3}$$

Therefore, predicting a sentence S can be formularized as:

$$\begin{aligned} P(S) &= \textstyle\prod_{m=1}^{n} P(w_m) \\ &= P(w_1) P(w_2|w_1) \dots P(w_n|w_1, \dots, w_{n-1}). \end{aligned} \tag{4}$$

A more common method is to use the logarithm of $P(S)$, denoted as the Log Probability $LP(S)$. Thus, we rewrite Equation (4) to:

$$\begin{aligned} LP(S) &= \log P(S) \\ &= \log(\textstyle\prod_{m=1}^{n} P(w_i)) \\ &= \textstyle\sum_{i=1}^{n} LP(w_i). \end{aligned} \tag{5}$$

The intuition behind Equation (5) is obvious. The higher the probability is, the more probable we observe the i-th word w_i after we observe the previous word sequence.

Formally, it is called a uni-gram when there is no dependency between the current word and other words. When the first-order Markov assumption is applied, namely, that the current word depends on one previous word, it is called a bi-gram. Similarly, we call it a tri-gram when the second-order Markov assumption is applied where

the current word depends on two previous words. In this paper, we use the tri-gram language model, including tri-grams, bi-grams and uni-grams. This model can be built on any large-scale corpus. In our study, we use a word list including $47,500$ words, which are used as uni-grams to train an n-gram language model by the SRI Language Modeling Toolkit (SRILM)[1] [Stolcke 2002], a very successful tool in Speech Recognition. During the training, we use common smoothing methods, including the linear interpolation and Kneser-Ney discount (KN discount) [Chen and Goodman 1999]. According to different corpora, some words will be pruned, so the sizes of uni-grams are slightly different. Since this is not our focus, we omit these details.

Generally speaking, the choice of a corpus has impact on the following processing and conclusions, because we assume that all the sentences in the corpus are reasonable and grammatically correct. The subsequent analysis is meaningful only if the corpus can provide a relatively accurate context background, reflected as probabilities of words, so choosing an appropriate corpus is the first step we concern.

3.2. Weighted Finite-State Transducer

A finite-state transducer is a finite automaton whose transitions are presented as arcs with input and output labels on it [Mohri 2004]. When designated with probabilities on arcs, it becomes a weighted finite-state transducer.

In this paper, WFST is adapted for language processing, which can be regarded as a four-tuple $\{S, \epsilon, A_{forward}, A_{backoff}\}$, where S stands for the set of states, $A_{forward}$ for that of forward arcs, $A_{backoff}$ for that of back-off arcs and ϵ for the starting state.

Each state represents an order in an n-gram language model. For example, a one-tuple term {今天(today)} will be a first-order state in WFST. Similarly, a two-tuple term {今天(today), 天气(weather)} and a three-tuple term {今天(today), 天气(weather), 不错(good)} will be a second-order state and a third-order one respectively. ϵ is the starting state of a complete WFST, from which the decoding process starts.

We preserve the set of forward arcs $A_{forward}$ in WFST, starting from lower-order states ($(i-1)$-gram states) to higher-order states (i-gram states), along with corresponding input and output labels. Since we only need one label on one arc, the input and output labels on a certain forward arc are the same, denoted as *word*. Additionally, the arcs are weighted according to the n-gram language model. For the previous example, we see that a forward arc can start from the state {今天} to the state {今天, 天气} with the corresponding probability on it and the word "天气". In addition to forward arcs, we add arcs in reverse directions from higher-order states to lower-order ones, called *back-off arcs*. Note that each state (except the starting state ϵ) has exactly one back-off arc and the probability on the arc is called a *back-off coefficient*. There are no words on back-off arcs. Figure 8 shows an example of a conversion from an n-gram language model to WFST. For more specific discussions, please refer to our previous study [Hao et al. 2013; Hao et al. 2014a].

4. CONTEXTUALIZED LATENT SEMANTIC INDEXING

CLSI methods proposed in this paper, both the Genuine version and the Modified one, consist of three main parts:

— Context information extracting;
— Constructing contextualized-semantic space; and
— Presenting the essays under the space.

[1] available at http://www.speech.sri.com/projects/srilm/.

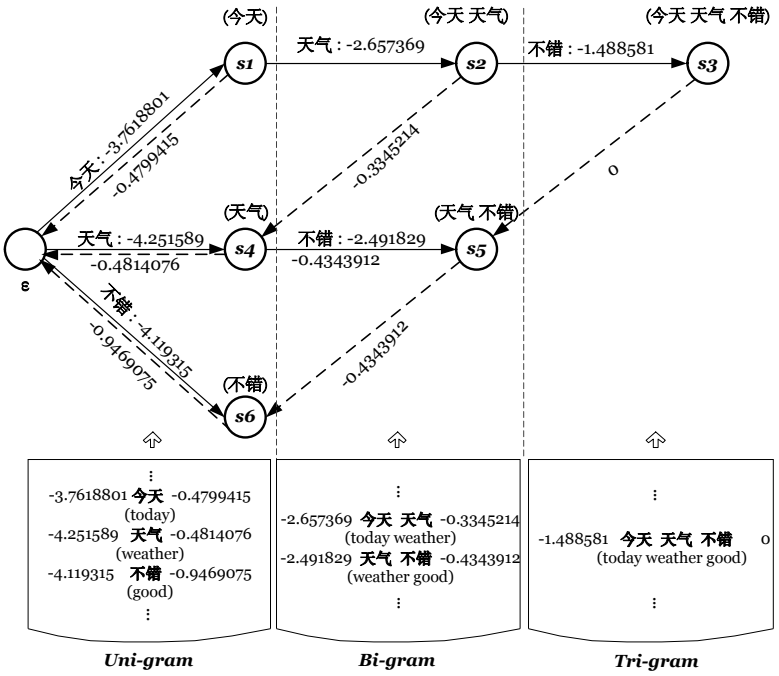


Fig. 8. Conversion from an n-gram model to WFST. We only show a small part of the whole WFST. In this figure, solid lines are forward arcs and dash lines are back-off ones. ϵ is the starting state.
Source: All the data in this paper are from our corpus.

For the first part, Genuine and Modified CLSI perform differently, so we introduce them in section 4.1 and 4.2 respectively. For the remaining two parts, they perform in the same way, so we use sections 4.3 and 4.4 to describe them.

4.1. Genuine Contextualized Latent Semantic Indexing

As previously stated, Genuine CLSI does not modify characters in essays. When we get an essay collection $\mathcal{D}$, WFST is used to recognize all essay word lists, and present each essay using a term-based essay vector $\mathbf{d}$. After that, CLSI is used to extract each word with its context information, and thus finish the feature extraction.

CLSI has an advantage that it extracts context information along with segmenting. As we know, a sentence can have different segmentations, and hence different context information. Our objective is to recognize the most reasonable sequence in an essay and extract its context information as accurately as possible, that is, to find out a word list $\mathcal{V} = \{w_i\}$ in the n-gram language model with the maximum probability, which can be formulated as:

$$\widehat{S} = \arg\max_{\{w_i\}} LP(S). \tag{6}$$

In WFST, it is a search problem whose goal is to find an optimal path from the starting state ϵ to a certain state. The intuition behind this method is obvious; that is, the more reasonable a sentence S is, the higher its probability $LP(S)$ is.

Given a specific context, context information is used to decide whether a word appears in a reasonable way. The n-gram language model provides a general standard, constructed from a large corpus with probabilities, and judging whether a sentence is

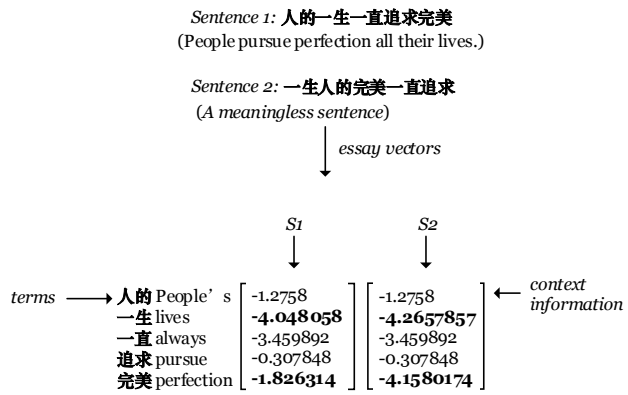


Fig. 9. Context information applied to the example in Figure 5. When we consider the context information extraction using WFST and the n-gram language model, we see that the two sentences are not identical anymore.

understandable or not, which is helpful for detecting fluency and coherence among words. In Figure 5, there is no explicit information about how these words are organized in two sentences. However, if we consider the processing in WFST and the n-gram language model, and replace term frequency or TF-IDF with probabilities, we will get a different picture as Figure 9 shows. A simple calculation, like summation, can make a difference: $PL(S1) > PL(S2)$, which makes sense because Sentence 2 is poorly written without a reasonable order of words.

This context information provides an important indicator for scoring essays. According to our assumption about test essays, Sentence 1 may appear in good or excellent essays whereas Sentence 2 may in poor essays. Thus, different from LSI, CLSI can make more obvious difference between poor essays and good ones. Figure 10 illustrates how to recognize a word list in a sentence using WFST, and Algorithms 1 to 4 show the details of Genuine CLSI.

In WFST, each possible solution is represented as a candidate, consisting of a segmented string (*segmented*), an unsegmented string (*unsegmented*), a log probability of the segmented string (*probability*), a current state (*state*), a word list ($\mathcal{V}$) and an essay vector (d). In Algorithm 1, to recognize a word in a sentence will pass a forward arc according to the word on the arc ($arc.word$), which is a step of context information extraction. A *candidate set* is used to define the set of promising solutions. Since there is more than one candidate caused by different paths, we use a *beam width* to maintain the size of the candidate set to avoid unnecessary expansions. We sort the members in the candidate set in descending order, and prune the set during each step of segmenting, according to the members' probabilities.

In each iteration, we first pass each candidate S through all of its back-off arcs, arriving at its corresponding lower-order states (Algorithm 2). For example, if one candidate is stopping on a tri-gram state, we pass it to its corresponding bi-gram state, and then to the uni-gram state, and finally arrive at ϵ. Thus, three new candidates $S_i^*(i = 0, 1, 2)$ will be generated whose current states are the bi-gram state, the uni-gram state and ϵ respectively, and corresponding log probabilities on the back-off arcs will be added. Although no segmentation happens, we still extract some important context information by using the statement "`Apply a certain Punishment Rule;`" in Algorithm 2, which will be explained later.

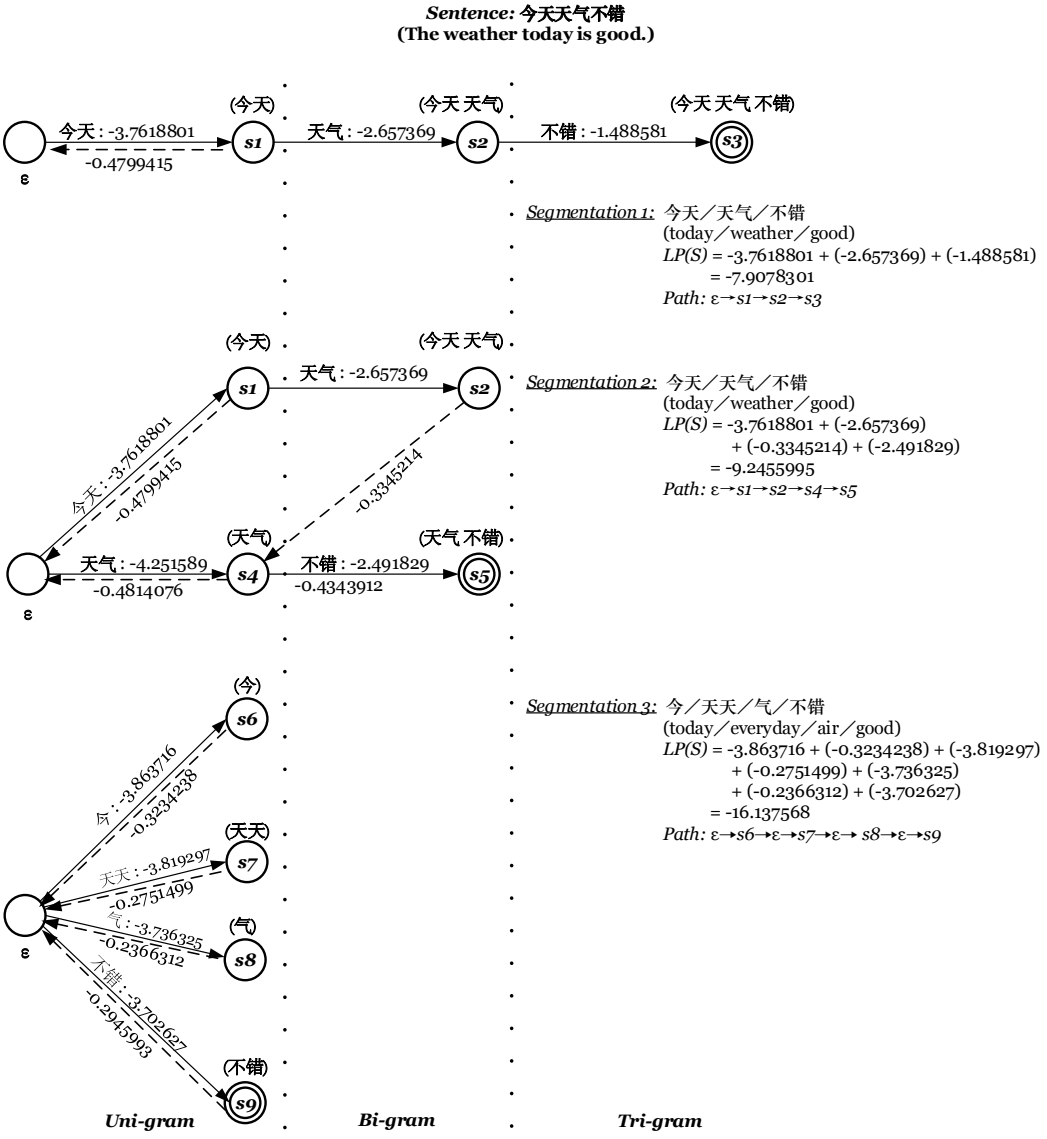


Fig. 10. An example showing how to recognize a word list in a sentence using WFST. For clarity, we separate the WFST into three parts, though actually they are in the same WFST. It is worth noting that one segmentation can have different paths, as Segmentation 1 and Segmentation 2 show. We use Segmentation 1 as the final result instead of Segmentation 2 because the log probability of the former is higher than that of the latter. The probability of Segmentation 3 is reasonably low because segmenting this sentence in this way is unreasonable from perspectives of both grammar and daily use. If we use $\mathcal{V}_i$ to denote the word list of Segmentation i (for $i = 1, 2, 3$), we see that $\mathcal{V}_1 = \mathcal{V}_2$ ={今天，天气，不错} and $\mathcal{V}_3$ ={今，天天，气，不错}.

ALGORITHM 1: Context Information Extraction Using WFST in CLSI

Input: Unsegmented Chinese sentence S;
Output: Essay vector d;

```
1  S.state ← ϵ;
2  S.V ← ∅;
3  the candidate set C ← ∅;
4  C ← C ∪ {S};
5  repeat
6      for each S_i ∈ C with the unsegmented part do
7          C* ← ∅;
8          for all the back-off arcs to ϵ do
9              S* ← Pass Back-off Arc (S_i, back-off arc arc_i);
10             C* ← C* ∪ {S*};
11             if |C*| > the beam width then
12                 prune C*;
13             end
14         end
15         for all the forward arcs arc_i starting from S_i.state do
16             if Judge Passable Forward Arcs in Genuine/Modified CLSI (S_i.unsegmented,
               arc_i.word) == TRUE then
17                 S* ← Pass Forward Arc (S_i, arc_i);
18                 C* ← C* ∪ {S*};
19                 if |C*| > the beam width then
20                     prune C*;
21                 end
22             end
23         end
24     end
25     for each S_i ∈ C that has been segmented completely do
26         C* ← C* ∪ {S_i};
27         if |C*| > the beam width then
28             prune C*;
29         end
30     end
31 until ∀S_i(S_i ∈ C ∧ S_i has been completely processed);
32 C ← C*;
33 Sort C according to S_i.probability in descending order;
34 return essay vector S_0.d
```

ALGORITHM 2: Pass Back-off Arc

Input: S_i and back-off arc arc_i
Output: S^*

```
1  S* ← S_i;
2  S*.state ← back-off arriving state;
3  S*.probability += back-off coefficient;
4  Apply a Punishment Rule;
5  return S*
```

Next, in each iteration, we check every arc starting from the current state of the candidate S (Algorithm 3). Note that in line 16 in Algorithms 1, we use the statement "`Judge Passable Forward Arcs in Genuine/Modified CLSI`", which means in the Genuine version, it will call the function of Algorithm 3. If the unsegmented part of S starts with the word of the forward arc $arc.word$, then we pass this arc, and

ALGORITHM 3: Judge Passable Forward Arcs in Genuine CLSI

Input: $S_i.unsegmented$ and $arc_i.word$
Output: TRUE or FALSE
if $S_i.unsegmented$ *starts with* $arc_i.word$ **then**
  return *TRUE*
end
return *FALSE*

ALGORITHM 4: Pass Forward Arc

Input: S_i and arc_i
Output: S^*
$S^* \leftarrow S_i$;
$S^*.unsegmented$ -= $arc_i.word$;
$S^*.\mathcal{V} \leftarrow S^*.\mathcal{V} \cup \{arc_i.word\}$;
$S^*.\mathbf{d} \leftarrow S^*.\mathbf{d} \cup arc_i.probability$;
$S^*.probability$ += $arc_i.probability$;
$S^*.state \leftarrow$ forward arriving state;
return S^*

transduce the state to its arriving state (Algorithm 4). The word on the arc will be added to the word set $\mathcal{V}$, and the corresponding word in the unsegmented part will be eliminated. Simultaneously, we extract the context information $arc_i.probability$ from the arc and append it to the tail of the essay vector $\mathbf{d}$ (Note that we use $\cup$ to denote the appending).

In WFST, back-off arcs are crucial, because usually when a sequence w_i, w_{i+1} is unreasonable, it needs to backtrack from the higher-order state $\{w_i\}$ to the lower-order state. Therefore, adding a back-off coefficient lowers the probability, meaning that this sequence perhaps is meaningless. However, a problem is encountered, that is, of which word, w_i or w_{i+1}, it is supposed to add the back-off coefficient to the context information. There exists three assumptions. First, we assume that w_i is correct but it is unreasonable to observe w_{i+1} after w_i. Second, we assume that w_{i+1} is correct but it is unreasonable to observe w_i before w_{i+1}. Third, we do not introduce back-off coefficients to the essay vector $\mathbf{d}$. Based on these assumptions, we give three Punishment Rules, and Figure 11 can help to understand this procedure.

Punishment Rule 1. Add the back-off coefficient to the first word, meaning that the first word is not appropriate while the second is.
Punishment Rule 2. Add the back-off coefficient to the second word, meaning that the second word is not appropriate while the first is.
Punishment Rule 3. Do not add any back-off coefficient.

In practice, one can choose a Punishment Rule that performs best. In Algorithm 2, if we use Punishment Rule 1, we will add the back-off coefficient to the last dimension of the essay vector $\mathbf{d}$; if we use Punishment Rule 2, we will appened the back-off coefficient to $\mathbf{d}$ in advance. Thus when the next word is recognized, its $arc.probability$ will be added to this dimension; applying Punishment Rule 3 just skips the statement "`Apply a certain Punishment Rule;`".

4.2. Modified Contextualized Latent Semantic Indexing

Essays with erroneous characters may interfere with both a human rater's and a machine's scoring. For example, "今天(today)" is a correct word whereas "今田" is not. Genuine CLSI extracts the context information of "今田" into "今(now)" and "田(field)"

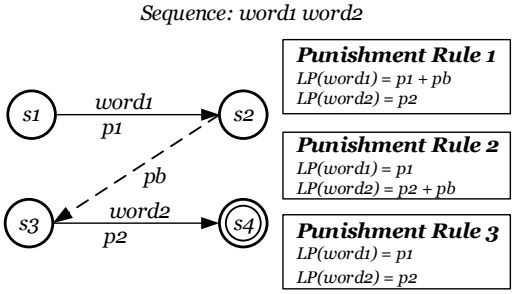
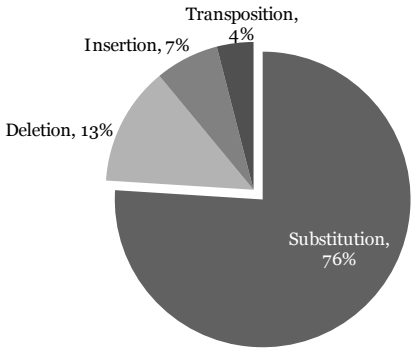


Fig. 11. Three Punishment Rules. In the figure, *p1* and *p2* stands for the probabilities, and *pb* for the back-off coefficient. $LP(word_i)$ is the context information of the *i*-th word. Note that the plus sign in this figure can have different meanings in the experiments, which we will explain later.



	Substitution(76%)	Deletion(13%)	Insertion(7%)	Transposition(4%)
Correct	休息 (rest)	五彩斑斓 (colorful)	爱好 (hobby)	原谅 (forgive)
Wrong	体息	五斑斓	爱好好	谅原

Fig. 12. Examples of four kinds of errors with their proportions respectively.

mistakenly. If we detect and correct the erroneous character "田" to "天", we may extract a more accurate feature. For this reason, we propose Modified CLSI.

The difference between Modified CLSI and Genuine CLSI is that Modified CLSI detects and corrects erroneous characters before extracting context features. For instance, if we apply Modified CLSI, the erroneous character "田" in the above example will be corrected to "天", and the correct word "今天" with its context information will be extracted.

Generally speaking, there are four kinds of errors in essays, which are substitution, deletion, insertion and transposition. By manual statistics from 200 essays from the MHK test, we find that the most common mistake is substitution, as Figure 12 shows. Therefore, we focus on this kind of mistake.

A key component of Modified CLSI is the Confusing-character Table, including approximate and homophonic characters, as shown in Figure 13. We do not separate

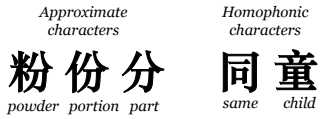


Fig. 13. Examples of approximate and homophonic characters. The two homophonic characters are pronounced “tóng”. The three approximate characters are pronounced “fen” with different tones.

……
天 夭 犬 大 太
王 汪 旺
放 方 房 防
……

Fig. 14. A part of the Confusing-character Table.

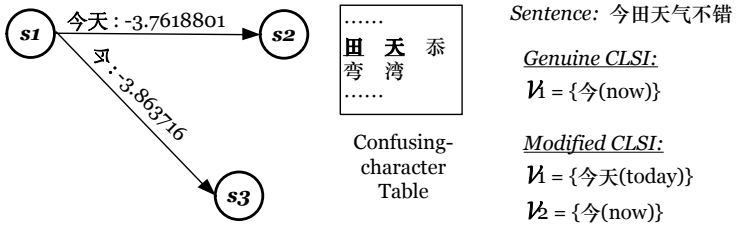


Fig. 15. In this example, the second character “田(field)” of the input sentence is erroneous. If we use Genuine CLSI, we keep it unchanged, and thus, pass only one forward arc. When we use Modified CLSI, if we replace the erroneous character with its corresponding characters in the Confusing-character Table, then there will be another forward arc to pass, and thus, the erroneous character is corrected. “ 田” has several confusing characters in the table, like “天(sky)”. Since the word “今天(today)” includes one identical character “今” to the input sentence, and a confusing character “天”, this arc is passable. Therefore, using Modified CLSI will have two possible word sets $\mathcal{V}_1$ and $\mathcal{V}_2$, whereas using Genuine CLSI has only $\mathcal{V}_1$.

this two kinds of characters, because those approximate characters are usually homophonic with different tones in Chinese.

Figure 14 is a part of the Confusing-character Table. Characters in the same line are easily confused ones. Thus, when tentatively extracting a word and its context information in WFST, we can detect probable erroneous characters, and replace them with characters in the same line. For a certain character, its confused characters have the same chance to replace it. Figure 15 is an example to show how this table helps to detect and correct erroneous characters.

The only difference between the algorithm of Genuine CLSI and that of Modified CLSI is the judgement of whether an arc is passable or not. Therefore, we still use

ALGORITHM 5: Judge Passable Forward Arcs in Modified CLSI

Input: $S_i.unsegmented$, $arc_i.word$ and the Confusing-character Table
Output: TRUE or FALSE

```
if S_i.unsegmented starts with arc_i.word then
    return TRUE;
end
if S_i.unsegmented starts with arc_i.word after replacing certain characters in S_i.unsegmented
according to the Confusing-character Table then
    return TRUE;
else
    return FALSE;
end
```

Algorithm 1 as the framework of Modified CLSI. However, in line 16, Algorithm 5 is used to replace Algorithm 3.

Sometimes, this method mistakenly change correct characters to erroneous ones, but according to our experiments (the Recall Rate of 85.68%, the Detection Precision of 91.22% and the Correction Precision of 87.30%), it still performs effective. Since the relative experimental results have been shown in [Hao et al. 2013], we omit them in this paper.

4.3. Constructing Contextualized-Semantic Space

The first step to construct a contextualized-semantic space is to convert essay vectors to an essay matrix. Then, SVD is used to reduce the dimension and construct a less noisy contextualized-semantic space. Thus, any term-based essay vector can be transformed to that space, which provides a uniform metric under the same space for automated scoring.

4.3.1. From Essay Vectors to an Essay Matrix. After context information extraction, each essay is associated with its word list $\mathcal{V}$ and corresponding essay vector $\mathbf{d}$. As in LSI, we need to array the essay vectors into an essay matrix. Note that $\mathcal{V}$ may contain identical words whereas the collection word set $\mathcal{V}^\dagger$ of an essay matrix does not contain identical words, so we have to merge the elements that represent identical words in $\mathbf{d}$. We propose two Merging Rules to merge them, which are also applied in the statement "`Applying a Punishment Rule`;" where the back-off coefficient is merged to the context information of a certain word.

Merging Rule 1 (Plus). Using this rule, we simply add the same word's probabilities:

$$d^* = d_i + d_j \tag{7}$$

where d_i and d_j are context information of the same word in original essay vector $\mathbf{d}$.

Merging Rule 2 (Log). Since we use log probabilities in the n-gram language model, another way to calculate is:

$$d^* = \log(10^{d_i} + 10^{d_j}). \tag{8}$$

After merging, the new word list $|\mathcal{V}^*| \leqslant |\mathcal{V}|$. Moreover, the collection word set $\mathcal{V}^\dagger = \bigcup_{i=1}^{|\mathcal{D}|} \mathcal{V}_i^*$, and thus, $|\mathcal{V}^\dagger| \leqslant \sum_{i=1}^{|\mathcal{D}|} |\mathcal{V}_i^*|$.

After all essay vectors are merged, they will be converted to an essay matrix. For certain words not included in an essay vector, we fill the corresponding elements with the log probability -99 (meaning the probability of 0, since $\log 0 \to -\infty$ and all the

context information are log probabilities.), denoting that these words are "impossible" words. Moreover, low-frequency words (those only appears in less than two essays) and stop words (like auxiliary verbs) are eliminated, both from $\mathcal{V}^\dagger$ and the corresponding essay vectors. Additionally, if an essay collection is used as a testing set, it will use the same $\mathcal{V}^\dagger$, and the words absent in the training set will be eliminated as well. In other words, if the essay collection is separated into a training set and a testing set (or a validation set), $\mathcal{V}^\dagger$ is determined by the training set.

4.3.2. Singular Value Decomposition. We use the essay matrix to construct a semantic space. Given an essay matrix $\mathbf{D}$, the optimization problem of CLSI performs as follows:

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V}} \quad & \|\mathbf{D} - \mathbf{U}\mathbf{V}\|_F^2 \\ \text{subject to} \quad & \mathbf{U}^\top \mathbf{U} = \mathbf{I} \\ & \mathbf{V}^\top \mathbf{V} = \mathbf{I} \end{aligned} \tag{9}$$

where $\mathbf{I}$ is an identity matrix. This problem can be solved by truncated SVD.

The standard mathematical strategy SVD decomposes the matrix into three matrices [Loan 1976; Klema and Laub 1980].

$$\mathbf{D} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top \tag{10}$$

$$\mathbf{D}^{M\times N} = \begin{bmatrix} \begin{bmatrix} u_{11} \\ \vdots \\ u_{M1} \end{bmatrix} & \cdots & \begin{bmatrix} u_{1M} \\ \vdots \\ u_{MM} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_{rank(\mathbf{D})} \\ 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} [v_{11} & \cdots & v_{1N}] \\ & \vdots & \\ [v_{N1} & \cdots & v_{NN}] \end{bmatrix} \tag{11}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices and $\mathbf{\Sigma}$ is a diagonal matrix with $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_{rank(\mathbf{D})}$ and extra zero.

We can truncate the original decomposition results into lower-dimensional matrices whose factorization is highly approximate to the matrix $\mathbf{D}$:

$$\mathbf{D} \approx \mathbf{D}^* = \mathbf{U}^*\mathbf{\Sigma}^*\mathbf{V}^{\top *} \tag{12}$$

$$\mathbf{D}^{*M\times K} = \begin{bmatrix} \begin{bmatrix} u_{11} \\ \vdots \\ u_{M1} \end{bmatrix} & \cdots & \begin{bmatrix} u_{1K} \\ \vdots \\ u_{MK} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_K \end{bmatrix} \begin{bmatrix} [v_{11} & \cdots & v_{1N}] \\ & \vdots & \\ [v_{K1} & \cdots & v_{KN}] \end{bmatrix} \tag{13}$$

This truncated SVD solves the optimization problem proposed in Equation (9), and construct a contextualized-semantic space.

4.4. Presenting the Essays Under the Space

Any essay vector is supposed to be projected to the contextualized-semantic space, in order to gain a uniform representation under the space for the scoring model. Therefore, given an term-based essay vector $\mathbf{d} = [d_1, d_2, ..., d_M]^\top$, it can be projected as follows:

$$\mathbf{d}^* = \mathbf{\Sigma}^{*-1}\mathbf{U}^{*\top}\mathbf{d} \tag{14}$$

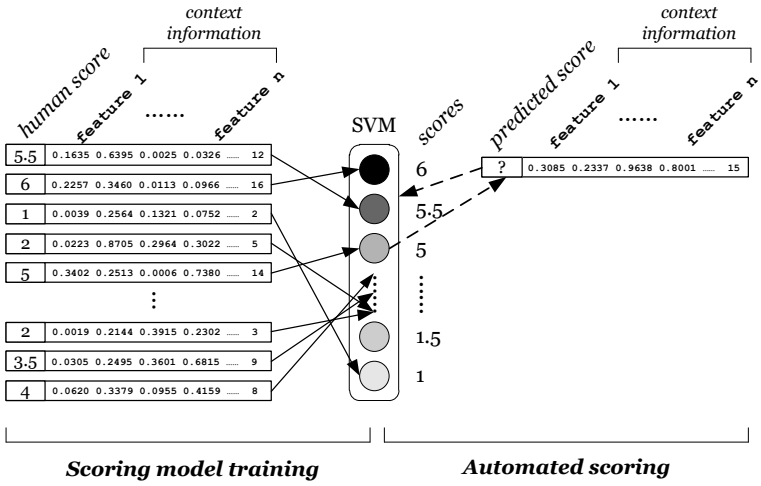


Fig. 16. The SVM-based scoring model.

where $\mathbf{d}^* \in \mathbb{R}^K$, and any test collection $\mathbf{Q}$ can be projected as well:

$$\mathbf{Q}^* = \mathbf{\Sigma}^{*-1}\mathbf{U}^{*\top}\mathbf{Q}. \tag{15}$$

So far, we complete the whole processing of CLSI methods.

5. SVM-BASED SCORING MODEL

If we consider each scoring point as a label, automated essay scoring can be processed as a kind of multi-class classification. We use the samples from the training set of$(\mathbf{d}^*, y)$ to train a multi-class classifier as the scoring model, where $\mathbf{d}^*$ is a transformed essay vector and y is its human score. After training, given any transformed essay vector, we use this model to predict its label to be the predicted score. SVM is a state-of-the-art and effective solution that has been applied widely [Cortes and Vapnik 1995]. Therefore, we choose SVM to train the scoring model and score essays. Figure 16 shows how this model works.

6. EXPERIMENTS

6.1. Experimental Settings

The programs in our experiments are written in C++. SVD and other matrix-related calculations are performed using MATLAB. In scoring model training, we use `LIBSVM`[2] [Chang and Lin 2011] to classify the essays. All the experiments run on a Linux machine with a 2.9GHz Intel Xeon E5-2690 CPU and 256G memory.

6.2. Sources and Statistics of Data

(1) *Language Models (LM):* We have three kinds of corpora for training n-gram language models: Literature, Mixture and News. The Literature Corpus includes all the articles awarded the Mao Dun Literature Prize, the highest honor for Chinese writers. The Mixture Corpus includes various sources like micro-blog posts, magazines, web pages and so forth. The News Corpus comes from The

[2]available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Table I. Statistics of three language models

	Uni-gram	Bi-gram	Tri-gram
Literature	47,489	6,557,265	7,173,881
Mixture	47,493	3,246,941	908,188
News	47,489	3,195,164	1,768,174

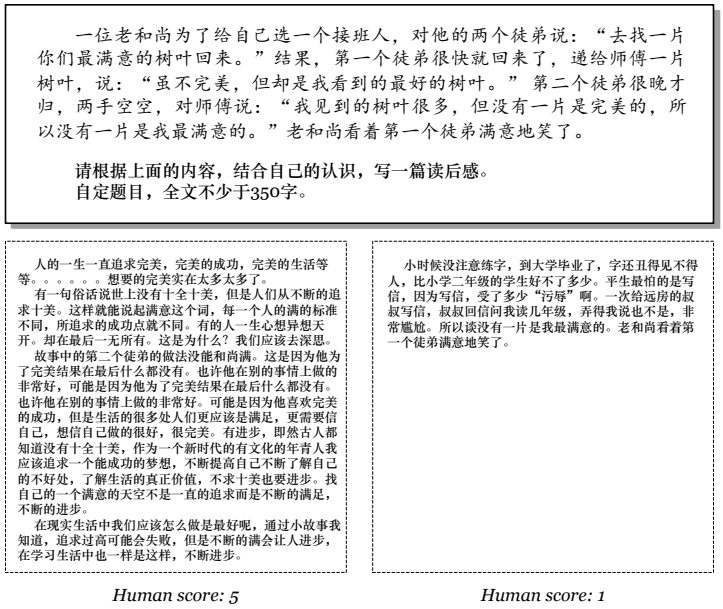


Fig. 17. The assignment of the essays in the sample sets is shown above. In this assignment, test takers are required to read a fable talking about the perfect leaf three monks are searching for, and then write an essay, rephrasing this fable and expressing their thoughts according to their experiences. An effective response must contain no less than 350 words. Two representative essays are shown below without translation.

People's Daily, the best-seller newspaper in mainland China. These corpora are very representative which stand for daily use of Chinese, mixture of language usage and standard Chinese, respectively. In Table I, we list the statistics of these three language models. Because of the shortage of huge corpora, we properly prune the language models, and thus the sizes of bi-grams and tri-grams vary.

(2) *The Confusing-character Table:* The Confusing-character Table comes from Modern Chinese Dictionary, including the most confusing homophonic and approximate characters. Additionally, according to previous research about the MHK test, we manually add some frequently confusing characters. Thus, the table includes $6,674$ characters in total.

(3) *The Sample Set:* We use the written essays with their human scores from the MHK test assigned by a same topic as the sample set. We show the topic and two essays with their scores in Figure 17. It is easy to see that the high-score essay usually contains fluent sentences, whereas the poor essay presents incoherent sentences. Omitting the empty essays and those without human scores, we obtain two training sets with the sizes of $10,000$ and $1,400$, a validation set with the size of 600 and a testing set with the size of 600. The training set is used to

Table II. Statistics of numbers of characters, words, and sentences per essay in the datasets

	Training Set 1			Training Set 2		
	min	max	average	min	max	average
Character / essay	7	547	347.55	7	544	356.98
Word / essay	5	365	244	5	358	227.49
Sentence / essay	2	46	11.89	2	46	12.13
	Validation set			Testing set		
	min	max	average	min	max	average
Character / essay	7	1,094	343.13	108	526	346.33
Word / essay	4	718	220.04	67	334	221.97
Sentence / essay	1	30	12.41	2	44	11.77

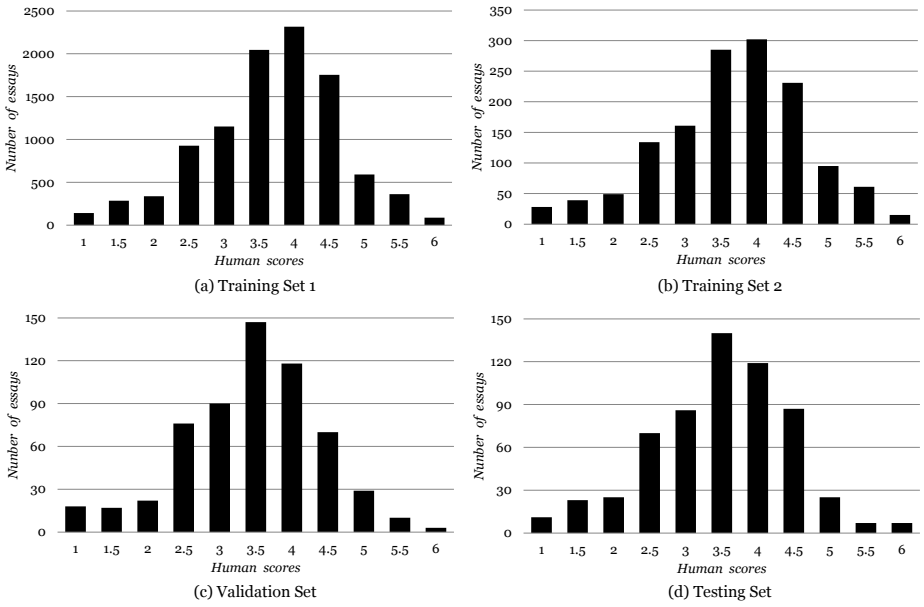


Fig. 18. The human score distributions of Training Set 1 (a), Training Set 2 (b), the validation set (c) and the testing set (d).

train a model; the validation set is to determine details of models due to various settings; the testing set is to test and make comparison among different versions of CLSI, conventional LSI and other methods. Figure 18 shows the human score distributions of these sets and in Table II, we list the statistic results of them.

6.3. Performance Criteria

There are several criteria from different scopes to evaluate the performance of an automated essay scoring system [Yannakoudakis et al. 2011]. In this paper, we use Rating Agreement, Fairness Coefficient, Quadratic Weighted Kappa, Spearman's Correlation and Pearson's Correlation.

(1) *Rating Agreement:* Rating Agreement is an overall evaluation criterion, including *exact agreement* and *adjacent agreement* [Dikli 2006]. The exact agreement calculates the exact equal scoring results between human scores and predicted scores, whereas the adjacent agreement is used to count the results with an acceptable bias. In our experiments, we use adjacent agreement (*R.A.*):

$$Rating\ Agreement(bias) = \frac{\sum_{i=1}^{N} \mathbf{1}\{|hs_i - ps_i| \leqslant bias\}}{N} \tag{16}$$

where N is the size of the testing set, and hs_i and ps_i stand for the human score and predicted score of the i-th essay respectively, and *bias* can be set to an acceptable value according to various tests. In the experiments, we set bias to 1 point according to the MHK test and the indicator function is shown as below:

$$\mathbf{1}\{Expression\} = \begin{cases} 1 & : \quad Expression\ is\ TRUE \\ 0 & : \quad Expression\ is\ FALSE \end{cases} \tag{17}$$

(2) *Fairness Coefficient:* Fairness Coefficient looks into each class. From the human score distribution in Figure 18, we see that the majority lies in the score of 3.5. If the classifier gives all essays a score 3.5, Rating Agreement $R.A.(1)$ can rise up to 83.5%, but that apparently violates the fairness of the test. Some poorly written essays may get higher scores than what they deserve, whereas some excellent ones may get lower scores. Therefore, we design a criterion named Fairness Coefficient (*F.C.*):

$$Fairness\ Coefficient(bias) = \frac{\sum_{i=1}^{C}(log(\frac{n_i}{N}) \cdot R.A(bias)_i)}{\sum_{i=1}^{C} log(\frac{n_i}{N})} \tag{18}$$

where C is the number of classes; N is the size of the validation or testing set; n_i is the size of class i; $R.A.(bias)_i$ is the Rating Agreement of class i with the *bias*. The intuition of this coefficient is since low- and high-score essays weigh more but have less number than ordinary essays in the sampling set, we increase their weights by using the logarithm of their proportions, focusing on the minorities.
Using this coefficient, we can evaluate an automated scoring system in the sense of "fairness". When $F.C.(1)$ is 1, the system is perfectly fair because each class has $R.A.(1) = 100\%$, meaning that all the essays get their deserved scores. The lower $F.C.$ is, the relatively less "unfair" the system is.

(3) *Quadratic Weighted Kappa:* Quadratic Weighted Kappa κ takes chance agreement into account, because there exists the risk that a system just uses a Baseline Classifier (which will be explained later) and guesses all the scores by pure chance. To calculate this value, we can construct a confusion matrix, showing the human scores and the predicted scores.

(4) *Spearman's Correlation:* As in evaluations of other scoring systems, we calculate Spearman's correlation $\rho(Spearman)$:

$$\rho(Spearman)_{X,Y} = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2(y_i - \bar{y})^2}} \tag{19}$$

where X and Y stand for the ranks of human scores and predicted scores respectively, and $\bar{x}$ and $\bar{y}$ for their averages respectively, and N for the size of the dataset.

(5) *Pearson's Correlation:* For the sake of completeness, we introduce the last criterion, Pearson Correlation $\rho(Pearson)$:

$$\rho(Pearson)_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y} \tag{20}$$

where $cov(X,Y)$ is the covariance between X and Y, and σ_X is the standard deviation of X, and μ_X is the mean of X, which is applied to Y as well.

6.4. Experimental Results with Different Models

6.4.1. Genuine CLSI. We use Table III to V to list the scoring details of Genuine CLSI with different models. Since the language model has impact on the results, we perform Genuine CLSI with each language model. Literature, Mixture and News produced collection word sets of sizes of $3,720$, $3,724$ and $3,722$ respectively. For each model, we use various combinations of Punishment Rules (Punishment Rule 1, 2 and 3) and Merging Rules showed in Formula 7 and 8 (Plus and Log). In the tables, S is the number of a certain predicted score, and F is the multi-class classification F-measure, calculated by $F = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$, and PR is the Punishment Rules. From the results, we see that Genuine CLSI successfully recognizes low-score essays.

As to the performance criteria *R.A.* and *F.C.*, we show the experimental results in Table VI. When choosing an appropriate model, Rating Agreement is the first criterion to be considered, while Fairness Coefficient is the second one. Therefore, we select the combination of Mixture corpus, Punishment Rule 2 and Merging Rule Log as the best combination for Genuine CLSI, because Rating Agreement of this combination reaches the highest.

6.4.2. Modified CLSI. For Modified CLSI, we follow the same settings as for Genuine CLSI to choose the best combination. In corpus choosing, Literature, Mixture and News produce collection word sets of sizes of $3,711$, $3,734$ and $3,774$ respectively. We show the scoring details of different combinations in Table VII to IX.

From Table X, we see that when we use the corpus News, Punishment Rule 2 and Merging Rule Log, both $R.A.(1)$ and $F.C.(1)$ are the best.

6.4.3. Conventional LSI. We run conventional LSI on the same training and validation sets to choose the best language model. Similarly, we list the scoring details in Table XI. From Table XI, it is obvious that LSI fails to recognize the low- and high-score essays, from point 1 to 2 and from 5.5 to 6 respectively, which is unacceptable because it has local overrating and underrating problems.

From Table XII, we see that LSI performs best when using the News corpus to train the model. Therefore, in further experiments, we choose News for it.

6.5. Further Experiments on Selected Models

6.5.1. Sensitivity to Dimension. In previous experiments, dimension K in truncated SVD is set to an empirical value 100. This value is important, because it will produce much noise if it is too high, or lose information if too low. In order to test the sensitivities of conventional LSI, Genuine CLSI and Modified CLSI to this value, we run experiments from $K = 25$ to 200 at intervals of 25, and plot them in Figure 19.

In Figure 19, we see that for $R.A.(1)$, the performance of conventional LSI decreases sharply as K increases, whereas those of CLSIs decrease more smoothly and are better than the conventional method. For $F.C.(1)$, the performance of conventional LSI fluctuates continually, whereas those of CLSIs remain relatively stable and are better, too. Hence, we conclude that CLSIs are more stable considering the dimension K. For the other three criteria, κ, $\rho(Spearman)$ and $\rho(Pearson)$, we see clearly that CLSIs stay relatively stable, but conventional LSI fluctuates and is worse than CLSIs.

Table III. Scoring details of Genuine CLSI (LM: Literature)

Human scores	**1**		**1.5**		**2**		**2.5**		**3**		**3.5**		**4**		**4.5**		**5**		**5.5**		**6**	
Validation set	18		17		22		76		90		147		118		70		29		10		3	
Models	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F
PR 1 + Log	8	N/A	8	N/A	10	N/A	64	0.93	79	1.00	145	0.82	112	0.91	68	0.99	23	0.86	2	N/A	0	N/A
PR 1 + Plus	9	0.52	7	0.56	8	0.56	60	0.78	83	0.93	146	0.90	112	0.88	66	0.88	23	0.79	1	0.46	0	N/A
PR 2 + Log	8	0.50	9	0.58	10	0.61	64	0.84	81	0.99	146	0.84	112	0.90	68	0.97	23	0.82	2	0.18	0	N/A
PR 2 + Plus	9	0.55	8	0.62	8	0.52	59	0.81	82	0.94	146	0.90	112	0.88	66	0.92	23	0.88	2	0.33	0	N/A
PR 3 + Log	8	0.50	8	0.59	10	0.62	64	0.85	81	0.93	146	0.91	112	0.90	68	0.93	23	0.88	2	0.33	0	N/A
PR 3 + Plus	9	0.55	8	0.62	8	0.52	60	0.83	82	0.94	146	0.90	112	0.88	67	0.93	23	0.88	2	0.33	0	N/A

Table IV. Scoring details of Genuine CLSI (LM: Mixture)

Human scores	**1**		**1.5**		**2**		**2.5**		**3**		**3.5**		**4**		**4.5**		**5**		**5.5**		**6**	
Validation set	18		17		22		76		90		147		118		70		29		10		3	
Models	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F
PR 1 + Log	8	0.48	7	0.56	8	0.53	57	0.78	81	0.93	146	0.90	110	0.89	68	0.89	21	0.84	5	0.67	0	N/A
PR 1 + Plus	9	0.56	9	0.67	9	0.58	61	0.84	82	0.95	146	0.90	113	0.89	68	0.93	22	0.86	1	0.18	0	N/A
PR 2 + Log	9	0.55	9	0.62	10	0.62	63	0.85	82	0.94	145	0.91	113	0.90	68	0.94	25	0.93	1	0.18	0	N/A
PR 2 + Plus	9	0.56	9	0.67	9	0.58	61	0.84	81	0.94	146	0.90	114	0.90	68	0.92	22	0.86	1	0.18	0	N/A
PR 3 + Log	8	0.48	8	0.62	8	0.53	61	0.81	81	0.94	146	0.90	110	0.90	69	0.92	21	0.84	4	0.57	0	N/A
PR 3 + Plus	9	0.56	9	0.64	9	0.58	60	0.83	80	0.94	146	0.90	114	0.89	68	0.91	22	0.86	1	0.18	0	N/A

Table V. Scoring details of Genuine CLSI (LM: News)

Human scores	**1**		**1.5**		**2**		**2.5**		**3**		**3.5**		**4**		**4.5**		**5**		**5.5**		**6**	
Validation set	18		17		22		76		90		147		118		70		29		10		3	
Models	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F
PR 1 + Log	9	0.53	7	0.54	10	0.62	63	0.84	83	0.95	146	0.91	113	0.90	69	0.95	23	0.88	1	0.18	0	N/A
PR 1 + Plus	13	0.62	7	0.50	10	0.61	58	0.81	78	0.91	144	0.90	112	0.90	69	0.93	21	0.84	1	0.18	0	N/A
PR 2 + Log	13	0.53	8	0.59	10	0.62	59	0.84	78	0.94	144	0.91	113	0.91	69	0.95	21	0.88	1	0.18	0	N/A
PR 2 + Plus	13	0.62	8	0.55	10	0.61	59	0.82	78	0.91	144	0.90	112	0.90	69	0.94	21	0.84	1	0.18	0	N/A
PR 3 + Log	13	0.63	9	0.60	11	0.67	61	0.84	80	0.92	144	0.91	113	0.91	69	0.95	21	0.84	2	0.33	0	N/A
PR 3 + Plus	13	0.62	8	0.55	10	0.61	59	0.83	78	0.91	144	0.90	113	0.90	69	0.94	21	0.84	1	0.18	0	N/A

Table VI. Results of Genuine CLSI

LM	Literature		Mixture		News	
	$R.A.(1)$ (%)	$F.C.(1)$	$R.A.(1)$ (%)	$F.C.(1)$	$R.A.(1)$ (%)	$F.C.(1)$
Punishment Rule 1 + Log	86.50	0.512884	85.17	0.523624	**87.33**	0.503083
Punishment Rule 1 + Plus	85.83	0.487807	86.67	0.504911	85.50	0.512351
Punishment Rule 2 + Log	**87.17**	0.521066	**87.50**	0.520829	84.33	0.520196
Punishment Rule 2 + Plus	85.83	0.505677	86.67	0.504682	85.83	0.519765
Punishment Rule 3 + Log	87.00	0.514504	86.00	0.521734	87.17	0.547303
Punishment Rule 3 + Plus	86.17	0.507489	86.33	0.503171	86.00	0.520196

Table VII. Scoring details of Modified CLSI (LM: Literature)

Human scores	**1**		**1.5**		**2**		**2.5**		**3**		**3.5**		**4**		**4.5**		**5**		**5.5**		**6**	
Validation set	18		17		22		76		90		147		118		70		29		10		3	
Models	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F
PR 1 + Log	10	0.59	10	0.71	9	0.58	63	0.82	85	0.94	145	0.91	110	0.90	68	0.94	20	0.82	2	0.33	0	N/A
PR 1 + Plus	10	0.63	11	0.79	11	0.67	60	0.80	85	0.96	146	0.91	110	0.90	68	0.93	21	0.84	4	0.57	0	N/A
PR 2 + Log	10	0.59	10	0.71	9	0.58	62	0.82	85	0.95	145	0.91	111	0.90	69	0.95	21	0.84	1	0.18	0	N/A
PR 2 + Plus	10	0.63	11	0.79	11	0.67	61	0.82	84	0.95	146	0.91	112	0.91	68	0.92	21	0.84	4	0.57	0	N/A
PR 3 + Log	10	0.57	11	0.79	11	0.67	63	0.83	85	0.96	145	0.91	111	0.91	68	0.95	20	0.82	4	0.57	0	N/A
PR 3 + Plus	10	0.63	11	0.79	11	0.67	61	0.82	84	0.95	146	0.91	112	0.91	68	0.92	21	0.84	4	0.57	0	N/A

Table VIII.Scoring details of Modified CLSI (LM: Mixture)

Human scores	**1**		**1.5**		**2**		**2.5**		**3**		**3.5**		**4**		**4.5**		**5**		**5.5**		**6**	
Validation set	18		17		22		76		90		147		118		70		29		10		3	
Models	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F
PR 1 + Log	9	0.58	9	0.64	14	0.78	61	0.84	85	0.94	144	0.91	115	0.91	68	0.94	21	0.82	1	0.18	0	N/A
PR 1 + Plus	9	0.56	8	0.64	12	0.71	58	0.81	87	0.97	146	0.91	114	0.90	69	0.94	23	0.88	1	0.18	0	N/A
PR 2 + Log	9	0.56	9	0.64	12	0.71	62	0.84	84	0.94	144	0.91	116	0.92	67	0.92	22	0.85	2	0.33	0	N/A
PR 2 + Plus	9	0.56	7	0.58	12	0.71	60	0.83	85	0.96	142	0.90	113	0.89	70	0.93	23	0.85	3	0.46	0	N/A
PR 3 + Log	10	0.54	9	0.67	16	0.78	61	0.82	80	0.88	140	0.92	111	0.91	64	0.90	22	0.81	2	0.31	0	N/A
PR 3 + Plus	9	0.56	7	0.58	12	0.71	60	0.83	85	0.96	142	0.90	113	0.89	70	0.94	23	0.85	3	0.46	0	N/A

Table IX. Scoring details of Modified CLSI (LM: News)

Human scores	**1**		**1.5**		**2**		**2.5**		**3**		**3.5**		**4**		**4.5**		**5**		**5.5**		**6**	
Validation set	18		17		22		76		90		147		118		70		29		10		3	
Models	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F
PR 1 + Log	11	0.56	9	0.60	14	0.76	63	0.85	79	0.89	143	0.91	113	0.92	66	0.93	22	0.86	2	0.33	0	N/A
PR 1 + Plus	9	0.60	8	0.62	11	0.67	64	0.85	85	0.96	145	0.91	113	0.91	68	0.94	22	0.86	6	0.75	0	N/A
PR 2 + Log	10	0.53	9	0.58	14	0.78	63	0.84	79	0.89	143	0.92	114	0.92	66	0.93	22	0.86	2	0.33	0	N/A
PR 2 + Plus	9	0.60	8	0.62	12	0.71	64	0.85	85	0.96	145	0.91	113	0.91	68	0.94	22	0.86	6	0.75	0	N/A
PR 3 + Log	11	0.56	9	0.60	14	0.76	64	0.85	79	0.89	143	0.91	113	0.91	66	0.92	21	0.84	1	0.18	0	N/A
PR 3 + Plus	9	0.60	8	0.62	12	0.71	64	0.85	85	0.96	145	0.91	113	0.91	68	0.93	21	0.84	6	0.75	0	N/A

Table X. Results of Modified CLSI

LM	Literature		Mixture		News	
	$R.A.(1)$ (%)	$F.C.(1)$	$R.A.(1)$ (%)	$F.C.(1)$	$R.A.(1)$ (%)	$F.C.(1)$
Punishment Rule 1 + Log	87.00	0.525937	**87.83**	0.527405	87.00	0.550349
Punishment Rule 1 + Plus	87.67	0.568560	**87.83**	0.517870	88.50	0.576078
Punishment Rule 2 + Log	87.17	0.516931	**87.83**	0.533746	**88.67**	0.580783
Punishment Rule 2 + Plus	**88.00**	0.569614	87.33	0.536656	87.00	0.544682
Punishment Rule 3 + Log	**88.00**	0.567975	85.83	0.548933	86.83	0.535112
Punishment Rule 3 + Plus	**88.00**	0.569614	87.33	0.536656	88.50	0.577512

Table XI. Scoring details of conventional LSI

Human scores	**1**		**1.5**		**2**		**2.5**		**3**		**3.5**		**4**		**4.5**		**5**		**5.5**		**6**	
Validation set	18		17		22		76		90		147		118		70		29		10		3	
Models	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F
LM: Literature	0	N/A	0	N/A	0	N/A	65	0.90	87	0.97	147	0.83	118	0.89	70	0.98	22	0.86	0	N/A	0	N/A
LM: Mixture	0	N/A	0	N/A	0	N/A	62	0.89	89	0.99	147	0.83	118	0.89	70	0.98	22	0.86	0	N/A	0	N/A
LM: News	0	N/A	0	N/A	1	0.18	65	0.90	88	0.98	147	0.83	118	0.89	70	0.98	22	0.86	0	N/A	0	N/A

Table XII. Results of conventional LSI

	$R.A.(1)$ (%)	$F.C.(1)$
LM: Literature	84.83	0.346885
LM: Mixture	84.67	0.345652
LM: News	**85.17**	0.352250

Table XIII. Scoring details among various methods

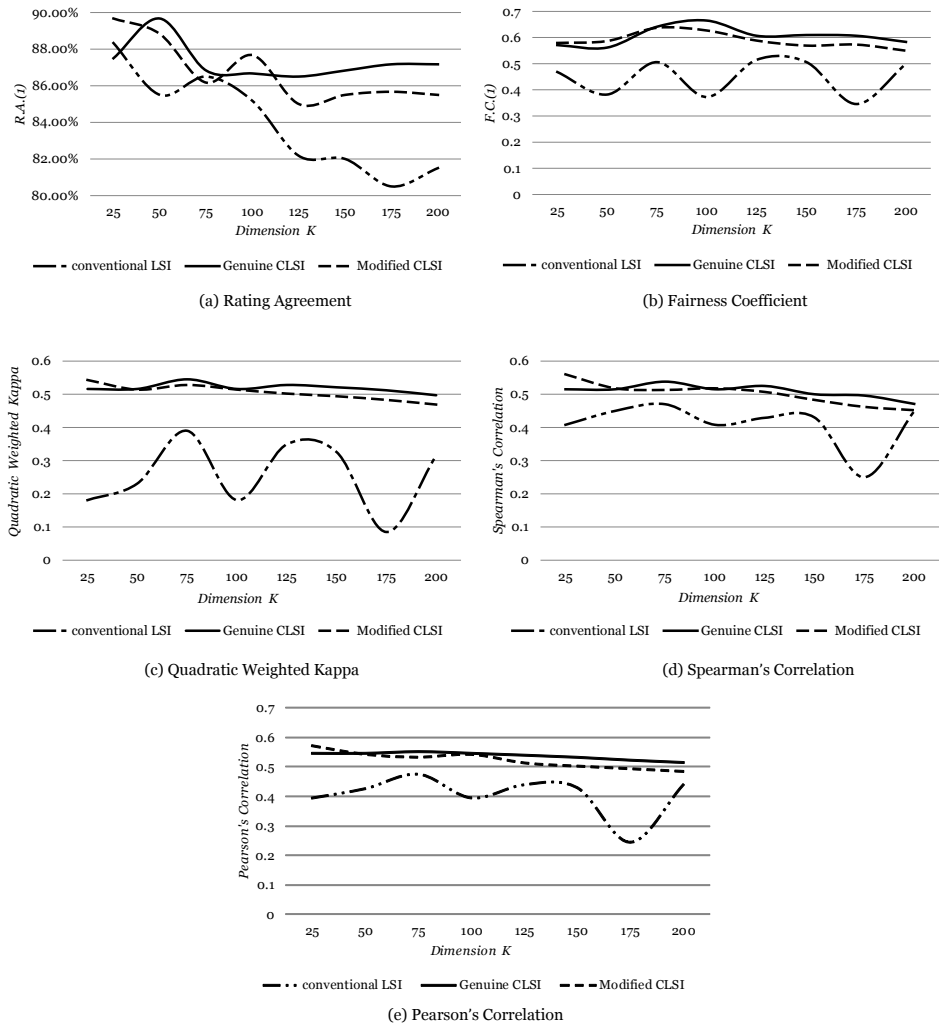
Human scores	**1**		**1.5**		**2**		**2.5**		**3**		**3.5**		**4**		**4.5**		**5**		**5.5**		**6**	
Testing set	11		23		25		70		86		140		119		87		25		7		7	
Methods	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F
Genuine CLSI	8	0.76	8	0.48	11	0.61	64	0.91	83	0.96	138	0.92	117	0.92	85	0.94	23	0.96	1	0.25	0	N/A
Modified CLSI	6	0.63	9	0.56	12	0.62	63	0.88	81	0.94	137	0.93	114	0.90	84	0.93	24	0.98	3	0.60	0	N/A
conventional LSI	0	N/A	1	0.08	1	0.08	56	0.87	85	0.99	140	0.86	118	0.86	87	0.99	25	1.00	0	N/A	0	N/A
RLSI	2	0.31	3	0.23	9	0.53	50	0.79	67	0.86	137	0.90	111	0.85	87	0.84	24	0.92	4	0.62	0	N/A
LDA	1	0.14	3	0.22	1	0.08	32	0.62	80	0.96	139	0.89	117	0.77	87	0.94	23	0.96	0	N/A	0	N/A

Table XIV. Comparisons among various methods

	$R.A.(1)$ (%)	$F.C.(1)$	κ	$\rho(Spearman)$	$\rho(Pearson)$
Genuine CLSI	**89.67**	0.561517	**0.516**	0.515	**0.5460**
Modified CLSI	88.83	**0.586551**	0.514	**0.518**	0.5423
conventional LSI	85.50	0.381993	0.231	0.450	0.4265
RLSI	82.33	0.499882	0.318	0.386	0.3941
LDA	80.50	0.428769	0.308	0.465	0.4367
Baseline Classifier	76.17	0.320264	0	0	0

Table XV. Comparisons of Scoring Deviation

	Genuine CLSI	Modified CLSI	conventional LSI	RLSI	LDA
(1, 1.5]	5.00%	6.33%	7.17%	9.50%	10.33%
(1.5, 2]	4.17%	3.83%	5.17%	5.00%	5.50%
(2, 2.5]	0.67%	0.17%	2.17%	2.67%	3.17%
(2.5, 3]	0.00%	0.00%	0.00%	0.50%	0.50%
Overall	**9.84%**	**10.33%**	**14.51%**	**17.67%**	**19.05%**



(a) Rating Agreement

(b) Fairness Coefficient

(c) Quadratic Weighted Kappa

(d) Spearman's Correlation

(e) Pearson's Correlation

Fig. 19. Comparisons of conventional LSI, Genuine CLSI and Modified CLSI (Rating Agreement, Fairness Coefficient, Quadratic Weighted Kappa, Spearman's Correlation and Pearson's Correlation), with different dimensions (K).

These experiments justify that CLSIs are less sensitive to K, meaning that it performs well no matter how much latent semantics is required, which is of great importance, because it is often hard to find the appropriate dimension K, and less K means less memory usage.

6.5.2. Sensitivity to the Size of the Training Set. When encountered a supervised learning problem in Machine Learning or NLP, it raises the problem of how small or big the training set is supposed to be. We randomly select essays from Training Set 1, from $1,000$ to $10,000$ at intervals of $1,000$. K is set to 100. The results are shown in Figure 20.

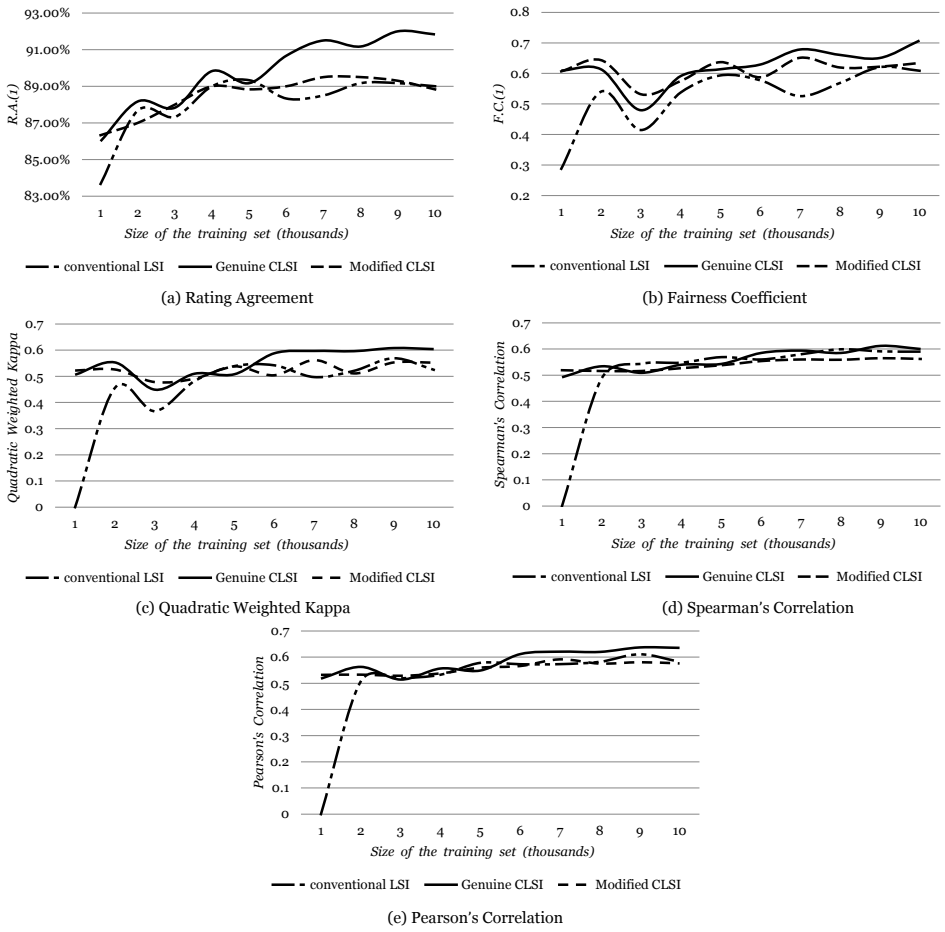


Fig. 20. Comparisons of conventional LSI, Genuine CLSI and Modified CLSI, using different size training set. The sizes vary from $1,000$ to $10,000$ at intervals of $1,000$.

In Figure 20 (a) and (b), we see that as the sizes grow, both $R.A.(1)$ and $F.C.(1)$ of CLSIs grow and are better than conventional LSI, especially for the Rating Agreement of Genuine CLSI. In Figure 20 (c), (d) and (e), differences among these three methods as the sizes grow are not obvious, but Genuine CLSI still performs better slightly than LSI. It is worth noting that when the size of the training set is $1,000$, the five results of LSI are very low. Therefore, when there are not many essays to be used to train the model, the deficiency of LSI becomes very obvious.

According to the experimental results, we have two conclusions. First, in automated Chinese essay scoring, the larger the training set is, the better the performance is, especially for CLSIs. Second, when there are only a small number of essays to be used as the training set, *i.e.*, less than $1,000$, CLSIs have great advantages over conventional LSI.

6.5.3. Comparisons Using Selected Models. Previously, we use various combinations and corpora to choose the best models for Genuine CLSI, Modified CLSI and LSI respectively. In order to explicitly demonstrate that CLSI outperforms LSI for automated Chinese essay scoring, we compare CLSIs and LSI using the selected models with other methods. Table XIII lists the scoring details and Table XIV shows comparisons of all methods.

In Table XIII and Table XIV, three additional methods are added. The first one is a Baseline Classifier, which counts the most frequent class in the training set and guesses all essays' scores in the testing set. In our experiments, since the most frequent class in the training set is point 4, all the essays in the testing set score 4. The extremely low κ, $\rho(Spearman)$ and $\rho(Pearson)$ in the table justify that this classifier is not applicable, although it has relatively close $F.C.(1)$ to the LSI method.

The second one is RLSI, which is also an LSI-based method and has been applied to automated Chinese essay scoring, though its scoring perspective is different from CLSI: RLSI is from the topic perspective, whereas CLSI from language proficiency and content perspectives. In this test, we use the News corpus to train the language model as we have done in LSI, and set the latent topic number to 100.

The third one is Latent Dirichlet Allocation (LDA)[3], which also scores essays from the topic perspective. Though this method is a probabilistic one and not based on LSI, we also include it for the sake of completeness. For LDA, we use the same settings: the News corpus to train the language model and the latent topic number 100.

From Table XIV, we see that both CLSI methods outperform other methods in Rating Agreement and Fairness Coefficient. More specifically, Genuine CLSI performs better than Modified CLSI in Rating Agreement slightly, and is lower in Fairness Coefficient slightly. Therefore, we conclude that both versions of CLSI perform well and can be applied to automated Chinese essay scoring.

6.5.4. Scoring Deviation. In order to further demonstrate the performance of CLSI, we define Scoring Deviation as:

$$Scoring\ Deviation(lower, upper) = \frac{\sum_{n=1}^{N} \mathbf{1}\{lower < |hs_i - ps_i| \leqslant upper\}}{N} \tag{21}$$

where *lower* and *upper* are the lower bound and upper bound of a given interval.

From the results in Table XIV, we calculate Scoring Deviation and show comparisons in Table XV. We see that conventional LSI has a high deviation because there are over 2% essays whose differences between human scores and predicted scores are 2.5. Both RLSI and LDA present unacceptable high deviations, because there are still 0.83% and 0.17% essays with the differences higher than 2.5. By sharp contrast, corresponding deviations of both Genuine CLSI and Modified CLSI are much lower. These results further confirm that CLSI is superior to other methods.

6.6. Discussions

6.6.1. Scoring Assumption. A human score consists of several scores from various perspectives, as mentioned in section 1.3. Usually a human rater considers an essay from all of these perspectives. If an essay contains several erroneous characters, the score of character and word usage will be low. Whenever he/she encounters an erroneous character, the human rater may correct it based on context and go on reading. Therefore, although an essay may contain erroneous characters, it will still get a higher score from other perspectives, like language proficiency, logical structure

[3]available at `http://www.cs.princeton.edu/~blei/lda-c/index.html`.

and so forth, as long as it meets these criteria. By detecting and correcting erroneous Chinese characters, Modified CLSI simulates this processing using WFST.

CLSI can score essays from the perspectives of language proficiency and content. It is impossible to get a human score in details from those perspectives, so there may be limitations in the experiments where we use the human scores as the benchmark. However, since language proficiency and content weigh most in the human score, the bias is negligible. For a more mature and comprehensive scoring system, we will combine all the factors to give more precise predicted scores in the future.

6.6.2. Exceptions in Processing. Generally speaking, the language proficiency requirement in the MHK test is basic, so a corpus should include all the characters appeared in an essay collection. However, one cannot rule out the exception where some characters in some essays are not in the corpus, which are called *unknown characters*. In CLSI, the unknown characters can surely be recognized as single characters, denoted as $\langle unk \rangle$ arcs in WFST.

Essay writing, even in tests for foreigners, can be a creative task. By analyzing essays, we find that some test takers use symbols to simplify a title, like "富有≠ 完美(Fortune ≠ Perfection)" for "Fortune does not mean perfection". Such an expression is simple but clear. Human raters can switch the symbol "≠" to the phrase "does not mean". CLSI, however, will extract the context information of two words: "fortune" and "perfection", and the underlying meaning cannot be recognized, and thus lose some content. We ignore such exceptions because such essays are not abundant according to previous MHK tests, which have little impact on overall performance.

6.6.3. Result Analysis. Genuine CLSI analyzes essays without any modification, where Modified CLSI detects and corrects erroneous characters. From the results, we see that Genuine CLSI is slightly better than Modified CLSI, which is a little counterintuitive because we expect that after correction of erroneous characters, Modified CLSI will extract context information more accurately than Genuine CLSI. As previously mentioned, there exists the risk that some correct characters are mistakenly modified to erroneous characters, which may explain why Modified CLSI performs slightly poorer than Genuine CLSI. We expect to enhance the precision of this technique further in future, and gain a better performance.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a new LSI method based on context information which can be applied to automated Chinese essay scoring – Contextualized Latent Semantic Indexing (CLSI), and developed two versions: Genuine CLSI and Modified CLSI. Using Genuine CLSI, one can extract the context information directly without segmenting the essays in advance. Detection and correction for erroneous characters is integrated to the processing in Modified CLSI, which can execute in the text preprocessing stage without extra computation. Both methods can effectively address the local overrating and underrating problem caused by conventional LSI, and score essays from the perspective of language fluency and content.

Although the local overrating and underrating problems have been alleviated to some extent, from the experimental results we see that these problems are not solved completely, leading to that Fairness Coefficient is still not high. Especially, the best essays whose points are 6 cannot be recognized. Context information can be the metric for language fluency, and able to classify poorly written essays and others. This information, however, has little power to discern topics, motifs and rhetoric in ordinary and excellent ones. The reason may be that the differences among them lie in other deeper information instead of the context or the word order.

As for the deficiencies analyzed above, we point out two ways to address these problems using CLSI. One is to reinforce the context information so that we can have a stronger degree of differentiation among poor, ordinary and excellent essays. The other is to add other deeper features to help the scoring. Moreover, deep learning is also a potentially powerful way to be applied to this field.

In addition to being an independent system, CLSI can be incorporated as a part into a more comprehensive automated Chinese essay scoring system where the overall score is obtained from various perspectives. In the future, we are going to develop a complete system for automated Chinese essay scoring, combining CLSI with other methods. Finally, this system can be used for MHK tests and any other tests of Chinese as a second language.

REFERENCES

Yigal Attali and Jill Burstein. 2006. Automated Essay Scoring With E-rater v.2.0. *Journal of Technology, Learning and Assessment* 4 (2006).

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *The Journal of Machine Learning Research* 3 (2003), 993–1022.

Jill Burstein, Martin Chodorow, and Claudia Leacock. 2003. Criterion^{SM} Online Essay Evaluation: An Application for Automated Evaluation of Student Essays. In *Proceedings of the 15th Annual Conference on Innovative Applications of Artificial Intelligence*. 3–10.

Li Cai, Xingyuan Peng, Dengfeng Ke, and Jun Zhao. 2010. Research of the Feature for Automated Essay Scoring System for Chinese Proficiency Test for Minorities. In *Proceedings of the 5th Youth Worshop of Computational Linguistics (YWCL)*.

Yiwei Cao and Chen Yang. 2007. Automated Chinese Essay Scoring with Latent Semantic Analysis. *Examinations Research* March (2007), 63–71.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* (2011), 2:27:1–27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Stanley F. Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language* 13 (1999), 359–394.

Shyi-Ming Chen and Ming-Hung Huang. 2013. A New Method for Generating the Chinese News Summary Based on Fuzzy Reasoning and Domain Ontology. *Intelligent Information and Database Systems* 7802 (2013), 70–78.

C. Chu, Nakazawa, D. T., Kawahara, and S. Kurohashi. 2013. Chinese-Japanese Machine Translation Exploiting Chinese Characters. *ACM Trans. Asian Lang. Inform. Process* 12, Article 16 (2013), 25 pages. Issue 4. http://dx.doi.org/10.1145/2523057.2523059

Corinna Cortes and Vladimir Vapnik. 1995. Support-Vector Networks. *Machine Learning* 20 (1995), 273–297.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science* 41 (1990).

Semire Dikli. 2006. An Overview of Automated Scoring of Essays. *Journal of Technology, Learning, and Assessment* 5 (2006).

Scott Elliot. 2003. IntelliMetric: From here to validity. In *Automated Essay Scoring: A Cross-disciplinary Perspective*. Lawrence Erlbaum Associates, INC., Mahwah, NJ, 71–86.

Shudong Hao, Zongtian Gao, Mingqing Zhang, Yanyan Xu, Hengli Peng, Kaile Su, and Dengfeng Ke. 2013. Automated Error Detection and Correction of Chinese Characters in Written Essays Based on Weighted Finite-State Transducer. In *Proceedings of 12th International Conference on Document Analysis and Recognition*. 763–767.

Shudong Hao, Yanyan Xu, Dengfeng Ke, Kaile Su, and Hengli Peng. 2014a. SCESS: A WFSA-based Automated Simplified Chinese Essay Scoring System with Incremental Latent Semantic Analysis. *Natural Language Engineering* (2014). To appear.

Shudong Hao, Yanyan Xu, Hengli Peng, Kaile Su, and Dengfeng Ke. 2014b. Automated Chinese Essay Scoring From Topic Perspective Using Regularized Latent Semantic Indexing. In *Proceedings of the 22nd International Conference on Pattern Recognition*. 3092–3097.

Md. Monjurul Islam and A. S. M. Latiful Hoque. 2012. Automated Essay Scoring Using Generalized Latent Semantic Analysis. *Journal of Computers* 7 (March 2012), 616–626.

Dengfeng Ke, Xingyuan Peng, Zhi Zhao, Zhenbiao Chen, and Shijin Wang. 2011. Word-level-based Automated Chinese Essay Scoring Method. In *Proceedings of National Conference on Man-Machine Speech Communication*. 57–59.

Virginia Klema and Alan Laub. 1980. The Singular Value Decomposition: Its Computation and Some Applications. *IEEE Trans. Automat. Control* 25 (1980), 164–176.

Thomas K. Landauer, Darrell Laham, and Peter W. Foltz. 2001. The intelligent essay assessor: Putting know ledge to the test. In *The Association of Test Publishers Computer-Based Testing : Emerging Technologies and Opportunities for Diverse Applications conference*.

Thomas K. Landauer, Darrell Laham, and Peter W. Foltz. 2003. Automated essay scoring: A cross-disciplinary perspective. In *Automated scoring and annotation of essays with the Intelligent Essay Assessor*. Lawrence Erlbaum Associates, INC., Mahwah, NJ, 87–112.

Vantage Learning. 2001. About IntelliMetric. Vantage Learning, Newtown, PA.

Vantage Learning. 2003. A true score study of 11th grade student writing responses using IntelliMetric Version 9.0. Vantage Learning, Newtown, PA.

Yanan Li. 2006. *Automated Essay Scoring for Testing Chinese as a Second Language*. PhD thesis.

Charles F. Van Loan. 1976. Generalizing the Singular Value Decomposition. *SIAM J. Numer. Anal.* 13 (1976), 76–83.

Mehryar Mohri. 2004. Weighted Finite-State Transducer Algorithms: An Overview. *Formal Languages and Applications* 148 (March 2004), 551–564.

Preslav Nakov, Antonia Popova, and Plamen Mateev. 2001. Weight Functions Impact on LSA Performance. In *Proceedings of EuroConference RANLP*. 187–193.

Ellis Batten Page. 1994. Computer grading of student prose, using modern concepts and software. *The Journal of experimental education* 62 (1994), 127–142.

Hengli Peng. 2005. The minorities-oriented Chinese level test. *China Examinations* 10 (2005), 57–59.

Xingyuan Peng, Dengfeng Ke, Zhi Zhao, Zhenbiao Chen, and Bo Xu. 2012. Automated Chinese Essay Scoring Based on Word Scores. *Journal of Chinese Information Processing* 2 (March 2012), 102–108.

Chaitanya Ramineni, Catherine S. Trapani, David M. Williamson, Tim Davey, and Brent Bridgeman. 2012. Evaluation of the e-rater Scoring Engine for the TOEFL Independent and Integrated Prompts. *ETS Research Report* (2012).

Lawrence M. Rudner and Tahung Liang. 2002. Automated Essay Scoring Using Bayes' Theorem. *The Journal of Technology, Learning and Assessment* 1 (2002).

Mark D Shermis and Jill C Burstein. 2003. *Automated essay scoring: A cross-disciplinary perspective*. Psychology Press.

Andreas Stolcke. 2002. SRILM – An Extensible Language Modeling Toolkit. In *Proceedings of the International Conference on Spoken Language Processing*. 901–904.

K. Wang, C. Zong, and K.-Y. Su. 2012. Integrating generative and discriminative character-based models for Chinese word segmentation. *ACM Trans. Asian Lang. Inform. Process* 31, Article 7 (2012), 41 pages. Issue 2. http://doi.acm.org/10.1145/2184436.2184440

Q. Wang, J. Xu, H. Li, and N. Craswell. 2013. Regularized latent semantic indexing: A new approach to large-scale topic modeling. *ACM Trans. Inf. Syst.* 31, Article 5 (2013), 44 pages. http://dx.doi.org/10.1145/2414782.2414787

Helen Yannakoudakis, Ted Briscoe, and Ben 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*. 180–189.

Lei Zhang, Ming Zhou, Changning Huang, and Haihua Pan. 2001. Automatic Detection and Correction of Typed Errors in Chinese Text. *Applied Linguistics* (2001), 19–26.

Yahui Zhao. 2011. Application of latent semantic analysis in auto-grading system. *Journal of Yanbian University (Natural Science)* 37 (2011), 345–348.