Selected Publications

September 2012 – September 2014

Shudong Hao
Artificial Intelligence Laboratory
Beijing Forestry University
shudongh@acm.org
http://shudonghao.com/

Supervisors

Prof. Dengfeng Ke	Prof. Yanyan Xu
Chinese Academy of Sciences	Beijing Forestry University
dengfeng.ke@ia.ac.cn	xuyanyan@bjfu.edu.cn

Contents

Abstract

This document includes three main published papers in the project ACES, from September 2012 to September 2014. Unfortunately, some publications are not included in this file, because they are still under review (*i.e.*, *Contextualized Latent Semantic Indexing: A New Approach Applied to Automated Chinese Essay Scoring*). This document will be updated whenever a new paper has been accepted.

Shudong Hao

2013 12th International Conference on Document Analysis and Recognition

Automated Error Detection and Correction of Chinese Characters in Written Essays Based on Weighted Finite-State Transducer

Shudong Hao[1], Zongtian Gao[1], Mingqing Zhang[1], Yanyan Xu[1], Hengli Peng[2], Kaile Su[3], Dengfeng Ke*[4]

mrholiday@bjfu.edu.cn[1]

1. School of Information Science and Technology, Beijing Forestry University
2. Institute of Educational Measurement, Beijing Language and Culture University
3. College of Mathematics Physics and Information Engineering, Zhejiang Normal University
4. Institute of Automation, Chinese Academy of Sciences

***Abstract*—Chinese text error detection and correction is widely applicable, but the methods so far are not robust enough for industrial use. In this paper, a new method is proposed based on Tri-gram modeled-Weighted Finite-State Transducer (WFST). By integrating confusing-character table, beam search and A* search, we evaluate the performance on real test essays. Various experiments have been conducted to prove that the proposed method is effective with the recall rate of 85.68%, the detection accuracy of 91.22% and the correction accuracy of 87.30%.**

***Keywords*—N-gram language model; Weighted Finite-State Transducer (WFST); Error detection; Error correction.**

I. Introduction

Chinese text error detection and correction can be widely applied into many fields such as document editing, search engines [1], automated essays scoring [2] and so forth. Dating back to 2000s, many efforts have been made to this area [3], but the progress is quite slow. Word segmentation is the main problem that leads to the impreciseness of the performance. Under the help of improved accuracy, automated error detection and correction can satisfy the rapidly growing demand of the industry with reduction in manual proofread [4-5]. Therefore, we focus on this technique in this paper.

Earlier in time, there have been some approaches in order to improve the accuracy. By using lexicon-dictionary, replacing every character for potential errors is a viable method [6]. Tri-gram-based method to detect disperse string [7], multifeature-based algorithm [8] and word-matching featured method [9] have also been implemented for this technique. Their feasibilities have been proved by practice, but because of the static segmentation, the recall-rate and the accuracy remain unsatisfying. Fig.1 shows a general method.

Considering the limitation of the methods mentioned above, we propose a new method to eliminate the inaccuracy caused by static segmentation. We firstly use *N*-gram language model to construct WFST. By replacing potential wrong characters, using beam search and A* search during decoding, we can find the best path which represents the most reasonable segmentation and the correct text line. Experiments on real Chinese test essays demonstrate that our new method is not merely viable but effective.

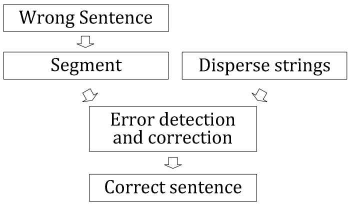


Fig. 1. A general method of Chinese text error detection and correction.

The rest of the paper is organized as follows. In Section II we discuss the proposed method. The experimental results and analysis are represented in Section III. Finally, in Section IV, we conclude our research and point out the directions of future works.

II. The Proposed Method

In earlier researches, segmentation, detection and correction are conducted in sequence. In contrast, we decode the sentence by using beam search [10] and A* to perform dynamic segmentation, sentence scoring, detection and correction simultaneously. The flowchart is illustrated in Fig.2.

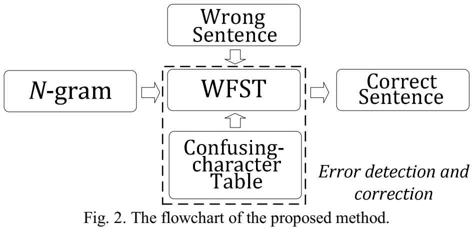


Fig. 2. The flowchart of the proposed method.

* Corresponding author.
Email address: dengfeng.ke@ia.ac.cn

1520-5363/13 $26.00 © 2013 IEEE
DOI 10.1109/ICDAR.2013.156

A. N-gram Language Model

N-gram language model [11] is an *N* tuple of words appeared in a corpus with a conditional probability of the last word, given the previous *N-1* words. In order to balance the performance and the corpus size, we choose the mixed Tri-gram language model, including Uni-gram, Bi-gram and Tri-gram. For convenience, *N*-gram is used to call an *N*-gram language model for short. Fig.3 shows a sample of *N*-gram.

Uni-grams:	-2.486861	**</s>**			
	-99	**<s>**			-0.9356831
	-3.154925	你			-0.9356831
	-3.270324	好			-0.719514
	-4.653283	你好			-0.3645265
Bi-grams:	-1.71277	**<s>**	你		-0.321846
	-4.072419	**<s>**	你好		-0.6793106
	-3.964405	你	好		-0.07850385
	-1.110578	好	**</s>**		
Tri-grams:	-4.84404	**<s>**	你	好	
	-0.6381439	你	好	**</s>**	

Fig. 3. A sample of *N*-gram. In every term, the first number is the log(P) where P stands for the probability and the last is the backoff coefficient (also log(P) and zero as default). The characters are *words*. Note that "<*s*>" and "</*s*>" are regarded as one character respectively. The "<*s*>" means the start of the sentence and "</*s*>" means the end. Both of them are added to the start and the end of a sentene respectively before decoding.

B. *Convertion from N-gram to WFST*

WFST can be regarded as a directed graph $G = (S, A_{forward}, A_{backoff})$, where S, $A_{forward}$ and $A_{backoff}$ are denoted as *States*, *Forward Arcs* and *Backoff Arcs* respectively.

States: We use "ε" (epsilon state) as the very start of the proposed WFST. Each state $S_i \in S$ is defined as:

$$S_i = \{arc_0, arc_1, arc_2 ... arc_n\}$$

where the *arc*s are emerged from the current state *S*. In practice, because every state represents an *N*-gram with its lower order *(N-1)*-gram, we use arc_0 as the backoff arc whose probability is the backoff coefficient from the high order *N*-gram to its lower order (Uni-gram to "ε"). Additionally, each state can be regarded as a breakpoint for a segmentation and a start point for the next segmentation when decoding in WFST.

Forward Arcs: Each arc $A_i \in A_{forward}$ or $A_{backoff}$ represents a word connecting two *N*-grams with a conditional probability of the word. That is:

$$A_i = \{S_{in}, S_{out}, word, score\}$$

where S_{in} records the previous state and S_{out} indicates the next state. The *word* corresponds to the word of a term in *N*-gram and the *score* is the conditional probability.

Backoff Arcs: Each state has and only has one single backoff arc, namely arc_0. By passing through this arc, the system moves from high order *N*-gram state to lower order *(N-1)*-gram state. The structure is similar to the forward arcs, except that the *word* is empty and the *score* is the backoff coefficient. An example is shown in Fig.4 to illustrate the conversion.

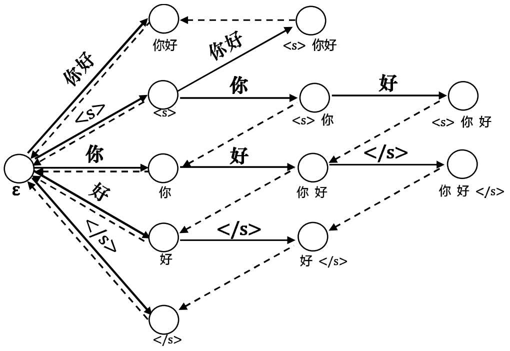


Fig. 4. A conversion of *N*-gram in Fig.3. The solid lines are forward arcs and dashed lines are backoff arcs. The characters on the arcs are *word*s and the *score*s are not shown in this figure.

C. *Confusing-character Table*

It is well known that Chinese characters are ideographic, so there are two kinds of mistakes as to confusing characters. The first one is the characters confused by their similar or even same pronunciations (homophone characters). For instance, "的" and "地" are the most easily confused ones, for their same pronunciations but different functions as conjunctions to indicate the part of speech. The second one is the characters confused by their similar appearances (approximate characters). Typically, "士" (soldier) and "土" (soil) are often confused in handwriting.

It is natural to construct a map, linking the most often confused characters together – whether they are homophone or approximate characters, and we name it "confusing-character table." Fig.5 shows a fragment of the table. Consulting the table and replacing correspondent confusing characters can complete automated correction. The materials we used are from Modern Chinese Dictionary and errors in the test essays.

纷	粉	份	扮	盼
饿	俄	峨	鹅	
乏	泛	眨		
防	仿	访	纺	坊

Fig. 5. A fragment of the confusing-character table. All the characters each line can be considered as confusion to each other.

D. *Decoding Using Beam Search*

By passing through different arcs, a sentence has different forms of segmentation, leading to various scores. The higher the score is, the more possible it is to be a correct answer, and this is the principle for text correction.

We assign each sentence to a *member* to record information about decoding in WFST, denoted as:

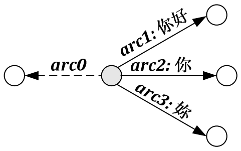
member = {*arc*, *state*, *score*, *scored-string*, *unscored-string*}

and their meanings will be explained below.

Two sets are used during the decoding: *candidate set* and *pre-candidate set*. The *candidate set* is a list that preserves *member*s needed to be examined for the current step. The *pre-candidate set* is a list that preserves the most promising *n* *member*s for the next step, rather than all the new *member*s that may produce abundant useless branches (or *member*s). At the very start, only one *member* whose *unscored-string* is the original input text is added to the *candidate set*.

Decoding in WFST is operated with sentence scoring, dynamic segmenting and error correction simultaneously. When an arc emerging from the *state* of the *member* is identified as passable, a new path as a new *member* will be generated. The *arc* and the *state* of this new *member* record the arc it has passed and the state it arrives at, in preparation for subsequent expansion. The score on the arc is added to the *score* of the *member* which records the sum of the score of its all passed arcs until the current step. Along with scoring, the word in the *unscored-string* corresponding to the word on the arc will be moved to the *scored-string*, and the remaining *unscored-string* is prepared for subsequent segmenting. Thus we complete a dynamic segmentation. The empty *unscored-string* means the end of the decoding, and the *score* is the result of the scoring of the whole sentence.

Three conditions are used to judge whether an arc is passable: *1)* Backoff arc; *2)* the *unscored-string* starts exactly with the word on the arc; *3)* only if after replacing several characters according to the confusing-character table, the second condition is satisfied. The third condition is the automated correction. Moreover, we subtract a proper value (punishment value) per modified character from the score to avoid that a right character is mistakenly modified to a wrong one. Fig. 6 shows this technique.



member	*scored-string*	*unscored-string*
M0		你好吗
M1	你好	吗
M2	你	好吗
M3	妳	好吗

Fig. 6. Assume that the input text is "你好吗 (How are you)." From the current state S_c, arc_i (i = 0, 1, 2, 3) is passable, corresponding to the condition 1 (arc_0), 2 (arc_1, arc_2) and 3 (arc_3) respectively. Note that when passing through the backoff arc, the backoff coefficient is added to the *score* of the *member*, though no segmenting happens. Because all the four arcs are passable, new *member*s as M_i (i = 0, 1, 2, 3) will be generated.

Whenever a new *member* has been generated, it will be sent to the Beam Container immediately. The Beam Container decides which one should be preserved or removed in time in *pre-candidate set* to avoid the drastic expansion of states. The principles for pruning branches are the *score* and the heuristic score of *unscored-string*. If the *pre-candidate set* is larger than the beam width after adding the new *member*, the one whose sum of *score* and heuristic score is minimum in the set will be removed. After all the *member*s in the *candidate set* have been examined, a new *pre-candidate set* is also produced. This new set is the *candidate set* for the next step to be examined.

The algorithm for error detection and correction is illustrated in Fig.7.

```
while at least one member.unscored-string ≠ empty
    pre-candidate set := ∅;
    for each member ∈ candidate set
        do if member.unscored-string ≠ empty
            for each arcs adjacent to member.state
                do if the backoff arc
                        or member.unscored starts with arc.word
                        or member.unscored starts confusingly with arc.word

                   then member.arc := arc;
                        member.state := arc.$S_{out}$;
                        member.score += arc.score;
                        member.scored-string += arc.word;
                        Remove arc.word from member.unscored-string;
                        Send the member to the Beam Container;
        else Send the member to the Beam Container;
    candidate set := pre-candidate set;
```

Fig. 7. The algorithm for error detection and correction. Note that "starts confusingly" is corresponding to the condition 3.

E. A Search*

A* is widely used in path-finding for its best-first search. In our algorithm, it can also be used in Beam Container to predict the score of the *unscored-string* of the *member*. We show three methods to compute the heuristic value. The number of characters in *scored-string* is denoted as *Nr*, and that in *unscored-string* is *Nu*.

1) Predict by scored characters and backoff paths:

$$H1=\frac{score}{Nr}\times Nu \tag{1}$$

2) Predict by scored characters only:

$$H2=\frac{score'}{Nr}\times Nu \tag{2}$$

where the *score'* is the sum of scores without backoff coefficient.

3) Predict by word segmentations:

$$H3=\frac{Nu}{Tr}\times Sa \tag{3}$$

where Tr *is* the *average* number of *characters in a word segmentation and* Sa *is the average score of a segmentation according to the* scored-*string and its* score'.

III. Experiments

A. Performance Indices

Three indices are defined to test the performance [12]: the recall-rate r(R), detection-accuracy a(D) and correction-accuracy a(C). Their computing formulas are listed below:

$$r(R)=\frac{N(W\rightarrow R)+N(W\rightarrow S)}{N(W)} \tag{4}$$

$$a(D)=\frac{N(W\rightarrow R)+N(W\rightarrow S)}{N(W\rightarrow R)+N(W\rightarrow S)+N(R\rightarrow W)} \tag{5}$$

$$a(C)=\frac{N(W\rightarrow R)}{N(W\rightarrow R)+N(W\rightarrow S)+N(R\rightarrow W)} \tag{6}$$

where N(W) is the number of wrong characters in the original text, N(W→R) is the number of characters modified correctly, N(W→S) is the number of characters detected correctly but mistakenly modified and N(R→W) is the number of characters that are originally right but are mistakenly modified to wrong characters.

B. Training Set

The selection of the corpus for the training set is crucial to our experiment because it influences the probability of every term in *N*-gram. In order to recur the daily context, we have three main sources: People's Daily, diverse source (Hodgepodge) and the Awarded Literature in Chinese.

People's Daily is the most formal and major newspaper in China, reporting official news and national affairs. It can be a reliable source because of its most standard Chinese and formal grammar, so we collect the newspapers from 2009 to 2012. The result, however, is not satisfying, comparing to the other two sources. The reason may be that its official tongue is distant from daily conversation and test essays.

The diverse source is a hodgepodge, containing various essays such as microblogs, blogs, publications, newspapers, lyrics, captions of films and so forth. But the nonstandard using of Chinese has impacts on the preciseness of *N*-gram and the performance, so this is also not the best choice.

The third corpus is proved to be the best. We use writings from the awarded literature, including the Mao Dun Literature Awards, the highest level of writing awards for Chinese writers. This kind of corpus integrates the advantages of those two above: standard and everyday used Chinese which is closer to the norm and the style of test essays.

TABLE.I shows the comparison of these three kinds of corpus. According to the initially tentative experimental data, we choose the awarded literature as corpus for the subsequent experiments.

TABLE. I DIFFERENT CORPUSES

Corpus	Recall-rate	Detection-accuracy	Correction-accuracy
People's Daily	75.27%	80.89%	75.29%
Hodgepodge	76.14%	87.75%	83.00%
Awarded Literature	85.68%	91.22%	87.30%

C. Testing Set

We use Minzu Hanyu Kaoshi (MHK), the minorities-oriented Chinese level test [13], as the testing set. By randomly revising 200 essays from 2011 MHK test in Xinjiang, there are four main kinds of writing errors: substitution, deletion, insertion and reversion. TABLE.II shows examples the four kinds of mistakes in Chinese test essay.

TABLE. II FOUR KINDS OF ERRORS

Categories	Erroneous	Correct
Substitution	骆它	骆驼
Deletion	遮挡沙	遮挡风沙
Insertion	虽虽然	虽然
Reversion	赞称	称赞

According to the statistic of these four kinds of mistakes, the substitution is the most common and serious error, and the reversion is the least one. Fig.8 illustrates the proportion in 200 essays. Consequently, in this paper, we focus on the detection and correction for substitution, namely, the erroneous Chinese characters. The answer we used for evaluation is produced by manual correction, according to the specific context in the essays.

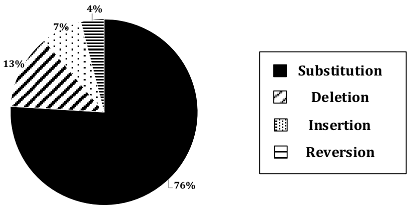


Fig. 8. Proportions of four kinds of errors in 200 MHK test essays.

D. Three Key Values

Heuristic Value: One of the most important problems is the computation method of heuristic value. We do not use heuristic value firstly. Next we compute the value by using formulae (1), (2) and (3) as *H1*, *H2* and *H3* respectively. From Fig.9, it is obvious that *H1* is the best choice.

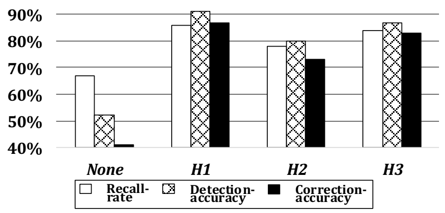


Fig. 9. Different computing methods of the heuristic value.

Punishment Value: We set different punishment values from 0.1 to 3.5 per modified character at 0.1 intervals, as Fig.10 shows. In order to maintain the recall-rate, we choose 0.9 (near the two points of intersection) as the proper punishment value. For this value, the detection-accuracy increases slightly without the recall-rate continuing decreasing.

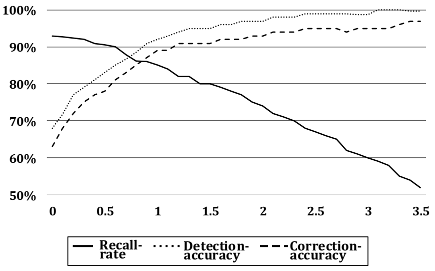


Fig. 10. Different punishment values.

Beam Width: Beam width is used for limit the size of *pre-candidate set* in Beam Container. A proper width can either preserve promising correct answers or avoid useless branches and lower efficiency. So we conduct experiments on the width from 5 to 30 at 5 intervals. From Fig.11, we choose 25 as the proper beam width in order to balance the performance and the efficiency.

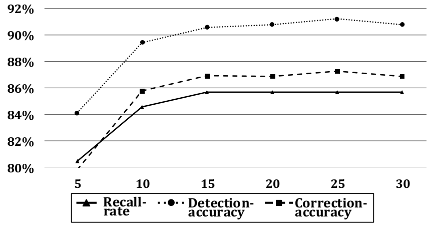


Fig. 11. Different beam widths.

E. Result Analysis

After determine several key factors, we get the results.

Total number of characters: 4761;

Recall-rate: 85.68%;

Detection-accuracy: 91.22%;

Correction-accuracy: 87.30%.

Due to that the recall-rate is relatively low, we examine the errors appeared in the result, and summarize four kinds of ineffectiveness: Disability, Ambiguity, LM and Algorithm. The proportions are shown in Fig.12.

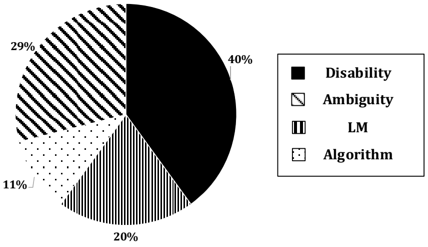


Fig. 12. Errors appeard in the result.

Both the Disability and the Ambiguity depend on the context. The Disability means the error depending on the context. For example, "他 (he)", "她 (she)" and "它 (it)" depend on the gender of the subject. Similarly, "的", "地" and "得" depend on the part of speech of the word they modify. "我们用眼睛来看 (We use eyes to see.)" and "我们用眼镜来看 (We use glasses to see.)" are both viable without specific context but the former is wrong if the context is considered, so we name it the Ambiguity. Some words in the essays never appear in *N*-gram. We call this error the LM due to the impreciseness and limitation of the corpus. The deficiency of Algorithm is caused by some details in our algorithm needed to be optimized or modified.

IV. CONCLUSIONS AND FUTURE WORKS

In this paper, we utilize Tri-gram-modeled WFST to detect and correct the erroneous characters in Chinese test essays. For the sake of improving the accuracy, we construct an algorithm combining confusing-characters table, Beam Search and A* Search based on WFST. As the experiment results show, choosing the awarded literature corpus for training, computing heuristic value by all passed arcs, punishment value of 0.9 and a Beam Width of 25 for the candidate set can improve the recall-rate to 85.68%, the detection-accuracy to 91.22% and the correction-accuracy to 87.30%.

As to deficiencies, the future works are:

1. Utilize the context to identify the gender of the subject.

2. Utilize part of speech to identify "的", "地" and "得".

3. Utilize the tag of name entities to improve the performance.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (No. 61103152) and the National 973 program in China (No. 2010CB328103). We thank the anonymous reviewers for their insightful comments.

REFERENCES

[1] Zhipeng Chen, Yuqin Li, Huasheng Liu, Gang Liu, Hui Tu, "Chinese Spelling Correction in Search Engines Based on *N*-gram Model," Journal of China Academy of Electronics and Information Technology, Vol. 4, No. 3, (2009.6), 323-326.

[2] Xingyuan Peng, Dengfeng Ke, Zhi Zhao, Zhenbiao Chen, Bo Xu, "Automated Chinese Essay Scoring Based on Word Scores," Journal of Chinese Information Processing, Vol. 26, No. 2, (2012.3), 588-595.

[3] Yanan Li, "Automated Essay Scoring for Testing Chinese as a Second Language," Beijing Language and Culture University, Beijing, 2006.

[4] S.Dikli, "An Overview of Automated Scoring of Essays [J]," Journal of Technology, Learning and Assessment, Vol. 5, No. 1, (2006), 1-35.

[5] Li Cai, Xingyuan Peng, Jun Zhao, "Research on Assisted Scoring System for Chinese Proficiency Test for Minorities", Journal of Chinese Information Processing, Vol. 25, No. 5, (2011.9), 120-126.

[6] Zhang Zhaohuang, "A Pilot Study on Automatic Chinese Spelling Error Correction," Communication of COLIPS, Vol. 4, No. 2, (1994),143- 149.

[7] Jinshan Ma, Yu Zhang, Ting Liu, Sheng Li, "Detecting Chinese Text Errors Based on Trigram and Dependency Parsing," Journal of The China Society For Scientific and Technical Information, Vol. 23, No. 6, (2004.12), 723-728.

[8] Lei Zhang, Ming Zhou, Changning Huang, etc., "Multifeature-based Approach to Automatic Error Detection and Correction of Chinese Text," In Proc. Workshop NLPRS. 99, Beijing. 1999.

[9] Yan Wu, Xiukun Li, Ting Liu, Kaizhu Wang, "Research on and Implementation of Chinese Text Proofreading System," Journal of Harbin Institute of Technology, Vol. 33, No. 1, (2001.2), 60-64.

[10] Volker Steinbiss, Bach-Hiep Tran, Hermann Ney, "Improvements in Beam Search," ICSLP 94 Proceedings, (1994), 2143-2416.

[11] Xi Luo, Wataru Ohyama, Tetsushi Wakabayashi, Fumitaka Kimura, "A Study on Automatic Chinese Text Classification," In Proceedings, 11th International Conference on Document Analysis and Recognition, 2011

[12] Hengli Peng, "The minorities-oriented Chinese level test," China Examinations, (2005.10), 57-59.

[13] Lei Zhang, Ming Zhou, Changning Huang, Haihua Pan, "Automatic Detection and Correction of Typed Errors in Chinese Text," Applied Linguistics, No. 1, (2001.2), 19-26.

SCESS: A WFSA-based Automated Simplified Chinese Essay Scoring System with Incremental Latent Semantic Analysis

SHUDONG HAO [1], YANYAN XU [1†], DENGFENG KE [2]
KAILE SU [3] and HENGLI PENG [4]

[1] *School of Information Science and Technology, Beijing Forestry University, Beijing, China*
[2] *Institute of Automation, Chinese Academy of Sciences, Beijing, China*
[3] *Institute for Integrated and Intelligent Systems, Griffith University, Brisbane, Australia*
[4] *Institute of Educational Measurement, Beijing Language and Culture University, Beijing, China*
e-mails: shudongh@acm.org, xuyyxu@gmail.com, dengfeng.ke@ia.ac.cn,
k.su@griffith.edu.au, penghl6402@aliyun.com

(*Received 19 September 2014*)

Abstract

Writing in language tests is regarded as an important indicator for assessing language skills of test takers. As Chinese language tests become popular, scoring a large number of essays becomes a heavy and expensive task for the organizers of these tests. In the past several years, some efforts have been made to develop automated simplified Chinese essay scoring systems, reducing both costs and evaluation time. In this paper, we introduce a system called SCESS (automated Simplified Chinese Essay Scoring System) based on Weighted Finite State Automata (WFSA) and using Incremental Latent Semantic Analysis (ILSA) to deal with a large number of essays. First, SCESS uses an n-gram language model to construct a WFSA to perform text pre-processing. At this stage, the system integrates a Confusing-Character Table, a Part-Of-Speech Table, beam search and heuristic search to perform automated word segmentation and correction of essays. Experimental results show that this pre-processing procedure is effective, with a Recall Rate of 88.50%, a Detection Precision of 92.31% and a Correction Precision of 88.46%. After text pre-processing, SCESS uses ILSA to perform automated essay scoring. We have carried out experiments to compare the ILSA method with the traditional LSA method on the corpora of essays from the MHK test (the Chinese proficiency test for minorities). Experimental results indicate that ILSA has a significant advantage over LSA, in terms of both running time and memory usage. Furthermore, experimental results also show that SCESS is quite effective with a scoring performance of 89.50%.

† Corresponding author.

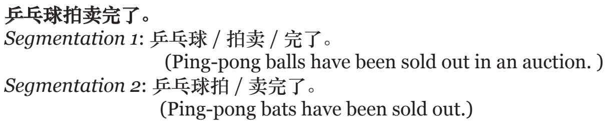
1 Introduction

1.1 Motivation

Writing, which is an important indicator for assessing a test taker's language skill, is an essential part of language tests. In Chinese language tests, test takers are usually required to write an essay according to a given topic, and then human raters will score these essays on the basis of some given educational benchmarks. It often happens that the scores of an identical essay scored by different human raters vary considerably because scoring by human raters is subjective (Peng 2005; Peng, Ke and Xu 2012; Li, Peng and Zhao 2011; Peng and Yu 2013). In addition, as the number of the MHK test takers increases rapidly year by year, it becomes a huge and expensive task for the organizers to score the essays. Therefore, an accurate automated simplified Chinese essay scoring system reducing both costs and evaluation time is urgently needed.

1.2 Research Background

Many automated English essay scoring systems have been developed over the past several decades. Project Essay Grader (PEG) is the earliest automated English scoring system developed by Ellis Batten Page in the 1960s. Page updated PEG and ran some successful trials in the early 1990s (Page 1994; Shermis and Burstein 2003). PEG grades essays predominantly on the basis of writing quality. An educational company called Measurement Incorporated acquired the rights to PEG in 2002 and has continued to develop it. Thomas Landauer has developed a system based on LSA using a scoring engine called Intelligent Essay Assessor (IEA). IEA is an implementation of the Knowledge Analysis Technologies (KAT) engine from Pearson Educational Technologies, which was first used to score essays in 1997 (Landauer, Foltz and Laham 1998; Foltz, Laham and Landauer 1999). IntelliMetric is Vantage Learning's product and was first used commercially to score essays in 1998 (Elliot 2003). Educational Testing Service offers e-rater, an automated essay scoring program which was first used commercially in 1999 and now is used to score the Test of English as a Foreign Language (TOEFL) and Graduate Record Examination (GRE) (Burstein 2003). E-rater is a sophisticated hybrid feature technology that uses syntactic variety, discourse structure (like PEG) and content analysis (like LSA) (Burstein and Chodorow 2010; Attali and Burstein 2006; Ramineni, Trapani, Williamson, Davey and Bridgeman 2012). Bayesian Essay Test Scoring sYstem (BETSY) is based on Bayes' theorem and developed by Lawrence Rudner (Rudner and Liang 2002). Pacific Metrics offers a constructed response automated scoring engine, called CRASE. Currently utilized by several state departments of education and in a U.S. Department of Education-funded Enhanced Assessment Grant, CRASE has been used in large-scale formative and summative assessment since 2007. Numerous researchers have reported that their automated essay scoring systems can, in fact, do better than a human rater. Page made this claim for PEG in 1994 (Page 1994) and Scott Elliot said in 2003 that



乒乓球拍卖完了。

Segmentation 1: 乒乓球 / 拍卖 / 完了。

(Ping-pong balls have been sold out in an auction.)

Segmentation 2: 乒乓球拍 / 卖完了。

(Ping-pong bats have been sold out.)

Fig. 1. Different meanings caused by different segmentations.

IntelliMetric typically outperformed human scorers in speed and consistency (Elliot 2003).

1.3 Related Work

Although many researchers have attached importance to automated English essay scoring and some systems have been applied widely, there has been relatively little research on automated Chinese essay scoring.

For Chinese text processing, segmentation (Teahan, Wen, McNab and Witten 2000; Wang and Liu 2011) is an important first step. Unlike English and other western languages, Chinese does not provide inter-word delimiters, so a sentence may have different meanings with different segmentations. Such an example is shown in Figure 1. Therefore, segmenting reasonably is challenging in automated Chinese essay scoring. In many fields such as search engines and detection and correction for erroneous characters in Chinese texts, several algorithms have been introduced (Pan and Yan 2009; Chang, Chen, Tseng and Zheng 2013). For example, Yan Wu uses regulation-based and count-based methods (Wu, Li, Liu and Wang 2001); Jinshan Ma has proposed a method based on tri-gram and dependency parsing (Ma, Zhang, Liu and Li 2004); Zhipeng Chen uses an n-gram model to correct Chinese spelling in search engines (Chen, Lv, Liu and Tu 2009). These methods are viable to some extent, but they are not very effective when considering all the possible segmentations. This problem becomes more obvious when faced with large-scale tests, such as the MHK test.

As for automated Chinese essay scoring, inspired by the studies of English essay scoring, research work has been done for several years (Li 2006; Chang, Lee, Tsai and Tam 2009; Chang, Lee and Tam 2007). These methods give automated scores from various perspectives. For instance, Latent Semantic Analysis (LSA) focuses on word usage and the content it reflects (Cao and Chen 2007; Zhao 2011). The word level method concentrates on word usage purely, based on a word list trained by human-scored essays (Ke, Peng, Zhao, Chen and Wang 2011). Regularized Latent Semantic Indexing (RLSI) from the field of topic modeling focuses on topic(s) in a dataset (Hao, Xu, Peng, Su and Ke 2014; Wang, Xu, Li and Craswell 2013). Among them, LSA, designed for indexing documents for information retrieval, is the most common technique that has been successfully applied to a wide range of fields (Tonta and Darvish 2010; McInerney, Rogers and Jennings 2012; Wang and Yu 2009; Jin, Gao, Shi, Shang, Wang and Yang 2011), such as bioinformatics (Ismail, Othman and Kasim 2011), language processing (Wang and Wan 2011; Liu, Wang and Liu

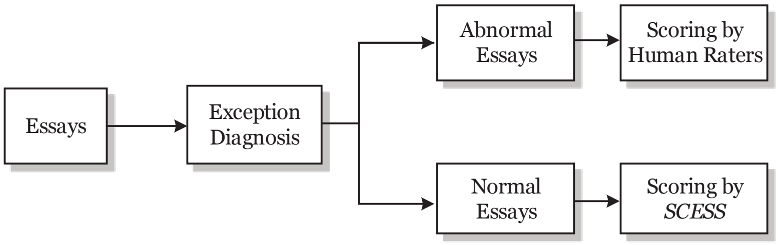


Fig. 2. A complete procedure of automated essay scoring.

2007; Yeh, Ke and Yang 2002; Gorrell 2006; Chang, Sung and Lee 2013) and signal processing (Mesaros, Heittola and Klapuri 2011). The underlying idea is to identify which one of several calibration documents is most similar to the new document based on the most specific (*i.e.*, least frequent) index terms. For essays, the average grade on the most similar calibration documents is assigned as the computer generated score (Landauer, Foltz and Laham 1998). LSA reduces the interference of variety and complex characteristics of natural languages (*i.e.*, ambiguities and synonyms), and represents the relations between terms and documents in a lower-dimension and noise-reduced space. Singular Value Decomposition (SVD) is the fundamental mathematical technique of LSA. By this decomposition, terms from the original matrix can be used to construct an approximate matrix or space under which any document can be represented.

When using LSA to perform essay scoring, SVD requires the entire dataset of essays being loaded into memory for computation, which is impossible if the matrix is huge, not to mention the temporarily stored data during the computation. As more and more people take the MHK test year by year, the number of the essays increases rapidly, so time and memory consumption becomes a big problem. Therefore, an automated simplified Chinese essay scoring system dealing with big datasets is highly desirable.

1.4 Main Contributions

MHK, known as the Chinese proficiency test for minorities, is the most popular test of simplified Chinese as a foreign language. We construct SCESS, a WFSA-based automated simplified Chinese essay scoring system with Incremental Latent Semantic Analysis (ILSA), to score MHK test essays. In general, an automated essay scoring system for MHK tests follows two steps, as shown in Figure 2. The exception diagnosis includes plagiarism detection, empty essay detection and so forth. Those with such problems are identified as abnormal essays and scored by human raters. The remaining essays are normal, and will be sent to SCESS.

When scoring normal essays, the flowchart of SCESS is shown in Figure 3. Text pre-processing is the first step, but current Chinese text processing techniques are not suitable for MHK tests as stated above. In order to provide a more effective method that can be used in this step and a more accurate result for subsequent

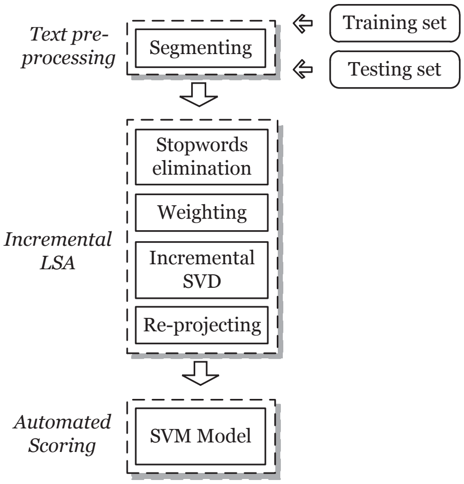


Fig. 3. The flowchart of SCESS.

steps, we propose a WFSA-based algorithm to perform dynamic segmenting, detection and correction for erroneous Chinese characters simultaneously by using beam search (Steinbiss, Tran and Ney 1994) and heuristic search (Xu, Yue and Su 2009). In this algorithm, we use an n-gram language model to construct a WFSA. By replacing possible erroneous characters with the help of a Confusing-Character Table and a Part-Of-Speech Table, we can find the best path which represents the most reasonable segmentation and a correct sentence. When applied in SCESS, WFSA can also be used to obtain an initial assessment from the surface information like character and word usage. In this paper, the performance of error detection and correction is demonstrated, but WFSA is only used to segment sentences for the current version of SCESS.

The second step is to analyse essays using LSA. As to the deficiencies of LSA discussed above, in this paper, we use ILSA in SCESS to solve the problem caused by big datasets. ILSA is introduced by Mattthew Brand (Brand 2002) and has been implemented to the fields of image processing (Chin, Schindler and Suter 2006) and information retrieval such as recommender systems (Sarwar, Karypis, Konstan and Riedl 2002; Brand 2003) and natural language processing (Gorrell 2006). However, there is no related work using ILSA and Incremental Singular Values Decomposition (ISVD) in automated essay scoring. In this paper, we use ISVD as a part of ILSA, to process huge datasets of test essays. Experimental results show that ILSA is very effective. It not only reduces time and memory consumption, but also has a good scoring performance of 89.50%.

The final step is to use several machine learning or pattern recognition strategies to complete automated scoring and a Support Vector Machine (SVM) is used to assist this processing.

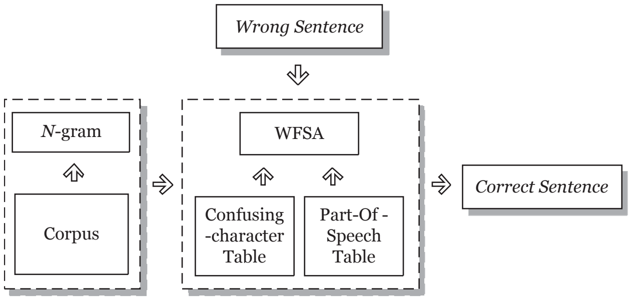


Fig. 4. The flowchart of word segmentation and correction.

1.5 Structure of the Paper

The remainder of this paper is organized as follows. We introduce the automated word segmentation and correction based on WFSA as part of text pre-processing of SCESS in the next section. In Section 3, we present the term-document matrix, weighting, ILSA and scoring performance measurement used in SCESS. Section 4 reports the experimental results and gives discussion about detection and correction for erroneous characters and automated essay scoring using ILSA. Finally, in Section 5, we conclude the paper by summarizing our work and giving some remarks on future directions.

2 Automated Word Segmentation and Correction Based on WFSA

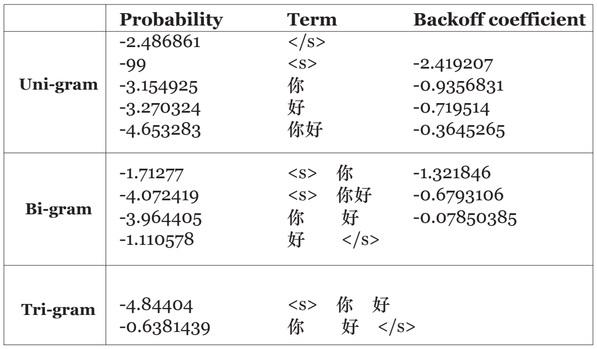
The first step of automated essay scoring is text pre-processing, including word segmentation. WFSA is a powerful tool to detect and correct erroneous Chinese characters, and segment Chinese sentences into words. The flowchart of word segmentation and correction based on WFSA is shown in Figure 4.

2.1 **N***-gram Language Model*

An n-gram language model is an n-tuple of words appearing in a corpus with a conditional probability of the last word, given the previous *n-1* words (Rosenfeld 1994). For convenience, in this paper, an n-gram language model is called an n-gram model. In order to achieve a good balance between practical performance and computational complexity, we choose the tri-gram language model, including uni-grams, bi-grams and tri-grams. This model is constructed from a list including $47,450$ words, which turn out to be the uni-gram terms. Then we construct bi-grams and tri-grams and calculate their probabilities from the corpus, trained by `SRILM` toolkit[1] (Stolcke 2002). Because of the different levels of trimming, the number of uni-grams may be slightly different according to different corpora.

Figure 5 shows some examples of n-grams. For every term, the first value is $\log(P)$

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



	Probability	Term	Backoff coefficient
Uni-gram	-2.486861	</s>	
	-99	<s>	-2.419207
	-3.154925	你	-0.9356831
	-3.270324	好	-0.719514
	-4.653283	你好	-0.3645265
Bi-gram	-1.71277	<s> 你	-1.321846
	-4.072419	<s> 你好	-0.6793106
	-3.964405	你 好	-0.07850385
	-1.110578	好 </s>	
Tri-gram	-4.84404	<s> 你 好	
	-0.6381439	你 好 </s>	

Fig. 5. Examples of n-grams (The sentence in Chinese in this figure means ‘Hello.’). Note that the term $\langle s\rangle$ stands for the start of a sentence. Because it is impossible to start a sentence before $\langle s\rangle$, its probability is set to -99.

where P is the probability of a word, and the last value is the backoff coefficient (calculated by modified KN-discount; 0 as default). The middle characters are words (terms). $\langle s\rangle$ and $\langle /s\rangle$ stand for the start and the end of a sentence respectively, which are added to the sentence before decoding.

2.2 Converting **N**-*gram Model to WFSA*

A WFSA can be regarded as a directed graph $G = (S, A_{forward}, A_{backoff})$, where $S, A_{forward}$ and $A_{backoff}$ denote *States*, *Forward Arcs* and *Backoff Arcs* respectively.

States : We use ϵ (the epsilon state) as the very start of the proposed WFSA. Each state S_i is defined as follows:

$$S_i = \{arc_0, arc_1, arc_2 \ldots arc_n\}$$

where these *arcs* are out-edges of S_i (except arc_0). In practice, because every state represents an n-gram with its lower order *(n-1)*-gram, we use arc_0 as the backoff arc whose probability is the backoff coefficient from the high order n-gram to its lower order (uni-gram to ϵ). Additionally, each state can be regarded as a breakpoint for a segmentation and a start point for the remaining segmentation when decoding in a WFSA.

Forward Arcs : Each forward arc A_i represents a word connecting two n-grams with a conditional probability of the word. That is:

$$A_i = \{S_{in}, S_{out}, word, probability\}$$

where S_{in} records the previous state and S_{out} indicates the next state.

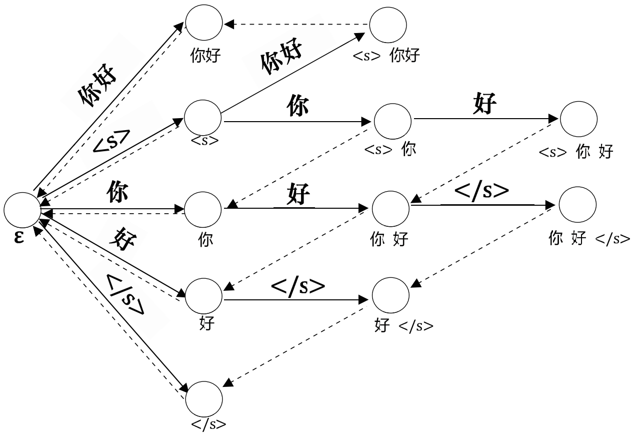


Fig. 6. The WFSA converted from Figure 5.

The *word* corresponds to the word of a term in an *n*-gram model and the *probability* is the probability from the *n*-gram model.

Backoff Arcs : Each state has one backoff arc, namely arc_0. By passing through this arc, the system moves from a higher order *n*-gram state to a lower order *(n-1)*-gram state. The structure of the backoff arc is similar to the forward arc, except that the *word* is empty and the *probability* is the backoff coefficient.

Figure 6 illustrates the WFSA converted from Figure 5. The solid lines are forward arcs and the dashed ones are backoff arcs. The characters on the arcs are *words* and the *probabilities* are not shown for the sake of clarity.

2.3 Confusing-Character Table and Part-Of-Speech Table

2.3.1 The Confusing-Character Table

It is well-known that Chinese characters are ideographic, so there are two kinds of mistakes caused by confusing characters. The first one stems from the confusion of characters with similar or same pronunciation, and the second one is caused by similar appearance.

Our method is to construct a *Confusing-Character Table*, linking the most often confused characters together. Figure 7 shows a fragment of this table. It is hard to translate these Chinese characters in Figure 7 to English. All we need to know is that characters in each line look alike or have similar pronunciations, and are considered as equally likely to be confused with one another. Looking up this table and replacing corresponding confusing characters will complete the automated correction.

We manually did the statistics and constructed the Confusing-Character Table based on *Modern Chinese Dictionary* and previous MHK test essays, which covers $6,674$ confusing characters.

堆	推	淮	谁	难
饿	俄	峨	鹅	
乏	泛	眨		
防	仿	访	纺	坊
纷	份	扮	盼	

Fig. 7. A fragment of the Confusing-Character Table.

层	*n.*
差不多	*adj., adv.*
差点儿	*adv.*
产生	*v.*
部分	*n., adj.*

Fig. 8. A fragment of the Part-Of-Speech Table.

2.3.2 The Part-Of-Speech Table

Many words have different representations caused by different parts of speech. We construct a Part-Of-Speech (POS) Table to deal with these words. During decoding, we check the POS Table as well. If the first several characters have different representations caused by different parts of speech, the arcs whose words are these representations are also passable. Figure 8 illustrates a fragment of the POS Table. The first column shows the words and the second column describes their parts of speech. The materials we used in the POS Table are from *Eight Hundred Words in Modern Chinese* (Lv 1999).

2.4 Decoding Using Beam Search and Heuristic Search

Decoding using a WFSA is performed simultaneously with computing of a sentence's probability, dynamic word segmenting and correction. Passing through different arcs will result in different forms of segmentation and different probabilities of a sentence. The higher the probability is, the more possible it is a correct sentence, and this is the principle for word segmentation and correction.

2.4.1 The Word Segmentation and Correction Algorithm

When decoding a sentence, we need to record the current *state*, the *arc* just passed, the *scored-string*, the *unscored-string* and the *probability* of the *scored-string*. Note that the term *probability* refers to the log probability (the sum of the log probabilities of the arcs). Thus, we associate each sentence with a *member* denoted as:

$$member = \{state, arc, scored\text{-}string, unscored\text{-}string, probability\}.$$

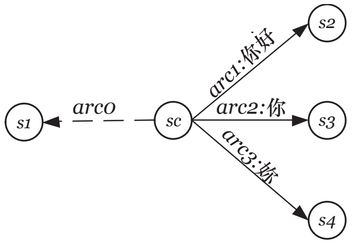
Moreover, two sets are used during the decoding: the *pre-candidate set* and the *candidate set*. The pre-candidate set is a list preserving members that need to be examined in each step, whereas the candidate set preserves the segmentation results. A *Beam Container* is used to perform beam search, selecting the best n members to add to the pre-candidate set. Therefore, the pre-candidate set is a list that preserves

```
    Input: A sentence from test essays
    Output: A correct sentence with spaces as delimiters
 1  $M1.scored\text{-}string = M2.scored\text{-}string = empty$;
 2  $M1.unscored\text{-}string = M2.unscored\text{-}string = input$;
 3  $M1.state = \langle s \rangle.S_{out}$;
 4  $M1.arc = \langle s \rangle$;
 5  $M1.probability = 0$;
 6  $M2.state = \epsilon$;
 7  $M2.arc = \langle s \rangle.S_{out}.A_{backoff}$;
 8  $M2.probability = \langle s \rangle.S_{out}.A_{backoff}.probability$;
 9  Add M1 and M2 to the candidate set;
10  while at least one $member.unscored\text{-}string \neq empty$ (in the candidate set) do
11      pre-candidate set $:= \emptyset$;
12      for each $member \in$ the candidate set do
13          if $member.unscored\text{-}string \neq empty$ then
14              for each arc adjacent to member.state do
15                  if satisfies one of three conditions then
16                      $member.arc := arc$;
17                      $member.state := arc.S_{out}$;
18                      $member.probability \mathrel{+}= arc.probability$;
19                      $member.scored\text{-}string \mathrel{+}= arc.word$;
20                      remove arc.word from member.unscored-string;
21                      send the member to the Beam Container;
22                  end
23              end
24          end
25          else
26              send the member to the Beam Container;
27          end
28      end
29      candidate set := pre-candidate set;
30  end
```

Algorithm 1: The word segmentation and correction algorithm.

the most promising n members for the next step instead of preserving all the new members to avoid producing numerous useless branches (or members).

The word segmentation and correction algorithm is shown in Algorithm 1. At the beginning of the decoding, we initialize two members $M1$ and $M2$ and add them to the candidate set (lines 1-9). If there is an arc starting from the state of the current member (line 15), a new path with a new member will be generated (lines 16-20). The arc and the state of this new member record the arc just passed and the state arrived at respectively, in preparation for subsequent expansions (lines 16-17). The probability on the arc is added to the probability of the member which records the sum of the probabilities of all its past arcs (line 18). Along with scoring, the



member	scored-member	unscored-member
M0		你好吗
M1	你好	吗
M2	你	好吗
M3	妳	好吗

Fig. 9. An example showing three conditions.

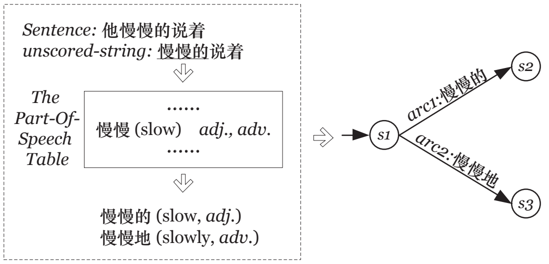


Fig. 10. An example showing how to check the POS Table.

word in the unscored-string corresponding to the word on the arc will be moved to the scored-string (line 19), and the remaining unscored-string is prepared for subsequent segmenting (line 20). The empty unscored-string means the end of the decoding (line 10), and the probability of a member is that of the whole sentence under the corresponding segmenting.

We adopt the following three conditions to judge whether an arc is passable (line 15):

1. The backoff arc;
2. The unscored-string starts exactly with the word on the arc;
3. After replacing several characters according to the Confusing-Character Table or after checking the POS Table, the second condition is satisfied.

The third condition provides automated correction. Moreover, we subtract a proper value (called the *punishment value*) per modified character from the score to avoid that a correct character is mistakenly modified to a wrong one. Figure 9 and Figure 10 show examples of these techniques.

In Figure 9, from the current state S_c, arc_0, arc_1, arc_2 and arc_3 are passable, corresponding to condition 1 (arc_0), condition 2 (arc_1, arc_2) and condition 3 (arc_3) respectively. Note that when passing through the backoff arc, the backoff coefficient is added to the probability of the member, though no segmenting happens. Because all the four arcs are passable, four new members will be generated.

In Figure 10, we show how the POS Table works. First, we check the first several characters of the unscored-string (underlined characters in the figure) in the table.

Since it has two parts of speech, the sentence can pass both arc_1 and arc_2 whose words match two different representations of the unscored-string. Without the POS Table, only arc_1 is passable.

Whenever a new member is generated, it will be sent to the Beam Container (lines 21, 26). The Beam Container decides which one in the pre-candidate set will be preserved or removed immediately to avoid huge expansion of states. The principles for pruning branches are the probability and the heuristic value of the unscored-string. If the pre-candidate set is larger than the beam width after adding the new member, the one whose sum of the probability and the heuristic value is minimum will be removed. After all the members in the candidate set are examined, a new pre-candidate set is generated and this new set will be the candidate set for the next step (line 29). At the end of the decoding, we will get n paths (according to the beam width), and the path with the highest probability is the best one.

2.4.2 Heuristic Search

Heuristic search, which can find applicable paths from a given initial node to a goal node, is widely used in planning and replanning (Xu and Yue 2009; Yue, Xu and Su 2006). In the Beam Container, the pre-candidate set is pruned in order to avoid useless expansion. The criterion is to use the members' probabilities of their scored-strings. To improve the efficiency, we use a heuristic function at the same time. Briefly, a heuristic function is to predict the possible number of erroneous characters in the remaining unsegmented sentence (unscored-string), based on the currently segmented and corrected sentence (scored-string). Therefore, the Beam Container will consider the probability of the scored-string and the heuristic value of the unscored-string of every member, to estimate the probability of the whole sentence and preserve the most promising members for the subsequent expansion.

We propose three heuristic functions for our algorithm. In these functions, the number of the characters in the scored-string is denoted as N_r, and that in the unscored-string is denoted as N_u.

1. Predict by scored characters and backoff paths: $H_1 = \frac{probability}{N_r} \times N_u$, where *probability* means the *scored-string*'s probability.
2. Predict only by scored-string: $H_2 = \frac{probability'}{N_r} \times N_u$, where $probability'$ is the sum of probabilities without backoff coefficients.
3. Predict by word segmentation: $H_3 = \frac{N_u}{T_r} \times S_a$, where T_r is the average number of characters in a word segmentation and S_a is the average probability of a segmentation according to the *scored-string* and its *probability*.

We have conducted an experimental study of these three functions and found that H_1 is the best one for SCESS.

2.4.3 WFSA-based Segmentation in SCESS

In Algorithm 1, we notice that if condition 3 is removed, WFSA can be used as an algorithm for word segmentation. In SCESS, we use WFSA to segment the essays into words without correction. It is inappropriate to use detection and correction in LSA and ILSA, because there exists the risk that it will erroneously correct essays and thus deviate from the original contents. However, since the detection and correction of errors by WFSA has been proved effective, in the near future, we will combine this function with other perspectives, like the word level method (Ke, Peng, Zhao, Chen and Wang 2011), to give a more comprehensive scoring system.

3 Automated Essay Scoring Using Incremental Latent Semantic Analysis

Traditionally, LSA can be performed through four sub-steps, which are matrix construction, weighting, calculating SVD and re-projection (Wild, Stahl, Stermsek, Neumann and Penya 2005). Calculating SVD, however, becomes a hard or even impossible task when faced with a huge dataset. Therefore, we use ILSA to resolve this problem efficiently. At first, all essays are segmented into words using the method described in Section 2, and then constructed into essay vectors which form the original dataset matrix. Next, a frequency weighting function is used to calculate the dataset matrix. Finally, applying the incremental algorithm on that matrix will establish a semantic space where all essay vectors can be re-projected.

3.1 Producing the t-d Matrix

After segmentation, SCESS will produce a *t-d* (term-document) matrix based on the number of words (terms) appearing in the essays (documents). Typically, SCESS needs a training set and a testing set, and the *t-d* matrix of the former is denoted as $\mathbf{D}$ and that of the latter as $\mathbf{Q}$. In these matrices, each column is an *essay vector* $\mathbf{d}_i$ or $\mathbf{q}_i$.

At first, according to the segmentation results, we generate the matrix $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n] \in \mathbb{R}^{m \times n}$ where m is the number of distinct words appearing in the training set, and n is the size of the training set. For each essay vector $\mathbf{d}_i = [d_1, d_2, \ldots, d_m]^\top \in \mathbb{R}^m$ $(i = 0, 1, \ldots, n)$, if the j-th word $(j = 0, 1, \ldots, m)$ appears in other essays but not in this essay, its weight will be 0. Next, we eliminate stop words, such as prepositions and verbal auxiliaries, because they do not have real meanings but appear with high frequencies, and get a set of words $\mathcal{V}$.

Then, we use $\mathcal{V}$ and the segmentation results to generate the matrix $\mathbf{Q}$. For the words appearing in an essay from the testing set, if they are included in $\mathcal{V}$, their weights in essay vectors are their numbers of occurrences. For those out of $\mathcal{V}$, we do not add these words into $\mathcal{V}$ and just eliminate them.

Finally we will get two *t-d* matrices $\mathbf{D}$ and $\mathbf{Q}$, sharing the same set of words $\mathcal{V}$.

3.2 Calculating the TF-IDF Matrix

The TF-IDF matrix, where TF stands for the term frequency and IDF for the inverse document frequency, is a common method for weighting (Nakov, Popova and Mateev 2001). Every element in the *t-d* matrix can be weighted as

$$d_{i,j} = TF_{i,j} \times IDF_{i,j} \tag{1}$$

The term frequency, $TF_{i,j}$ is defined as

$$TF_{i,j} = \frac{num_{i,j}}{\sum_{k=1}^{m} num_{k,j}} \tag{2}$$

where $num_{i,j}$ is the number of the occurrences of the i-th word in the j-th essay, and m is the number of the words in the training set. As for IDF, it can be calculated as

$$IDF_{i,j} = log\frac{n}{1 + DF_i} \tag{3}$$

where n is the size of the training set (or the number of essay vectors), and DF_i is the number of essays which contain the i-th word.

3.3 Incremental Latent Semantic Analysis

ILSA 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 accomplished by ISVD, avoiding the synonyms and ambiguities of words and enabling essay vectors to be projected onto a low-dimension semantic space. The second part is re-projection, in which any dictionary-based essay vector can be re-projected to the semantic space.

3.3.1 Conventional SVD

SVD is the underlying algorithm of LSA, which constructs a semantic space of a given dataset. Given the r-rank matrix $\mathbf{D} \in \mathbb{R}^{m \times n}$ where m is the size of the set of words $\mathcal{V}$ and n is the size of the training set, we apply SVD as follows:

$$\mathbf{D}_{m \times n} = \mathbf{U}_{m \times m} \mathbf{\Sigma}_{m \times n} \mathbf{V}^{\top}_{n \times n} \tag{4}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices, and $\mathbf{\Sigma}$ is a r-rank diagonal matrix where the elements are in descending order. In general, $\mathbf{\Sigma}$ is not a square matrix ($m \neq n$), so it contains extra columns or rows filled with zeros for matrix multiplication.

Specifically, in natural language processing, maintaining only $k \ll r$ will produce a lower dimensionality and better approximation to the original matrix $\mathbf{D}$. By removing (r - k) columns in $\mathbf{U}$, (r - k) rows in $\mathbf{V}$ and (r - k) elements in $\mathbf{\Sigma}$ that are small enough to be considered as trivial, we can multiply the matrices and get an approximation to the original matrix $\mathbf{D}^{'}_{m \times n}$:

$$\mathbf{D}_{m \times n} \approx \mathbf{D}^{'}_{m \times n} = \mathbf{U}_{m \times k} \mathbf{\Sigma}_{k \times k} \mathbf{V}^{\top}_{k \times n}. \tag{5}$$

3.3.2 Incremental SVD

As the dataset grows larger, conventional SVD becomes impractical due to huge memory usage and intolerably long running time. Hence, in SCESS, we use ISVD instead of conventional SVD. ISVD performs as follows. First, given the matrix $\mathbf{D} \in \mathbb{R}^{m \times n}$, we partition it into a small matrix $\mathbf{M} \in \mathbb{R}^{m \times n'}$ and matrices $\mathbf{C}_i \in \mathbb{R}^{m \times q}$ $(i = 1, 2, \ldots, p)$:

$$\mathbf{D} = \left[\begin{array}{ccccc} \mathbf{M} & \mathbf{C}_1 & \mathbf{C}_2 & \ldots & \mathbf{C}_p \end{array} \right] \tag{6}$$

where $p = \lceil \frac{n-n'}{q} \rceil$ and n' is called the *initial value* and q is called the *batch size*.

Next, conventional SVD is used on $\mathbf{M}$. Because $\mathbf{M}$ is small, the computation is fast. Then we update the decomposition result by using $\mathbf{M}$ and $\mathbf{C}_i$:

$$[\mathbf{M}\ \mathbf{C}_i] = [\mathbf{U}\ \mathbf{J}] \left[\begin{array}{cc} \mathbf{\Sigma} & \mathbf{L} \\ \mathbf{0} & \mathbf{K} \end{array} \right] \left[\begin{array}{cc} \mathbf{V}^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{array} \right]^\top = \mathbf{tU} \cdot \mathbf{t\Sigma} \cdot \mathbf{tV} \tag{7}$$

where $\mathbf{L} = \mathbf{U}^\top \mathbf{C}_i$ and $\mathbf{I}$ is an identity matrix. Define $\mathbf{H} = \mathbf{C}_i - \mathbf{UL}$. Applying QR decomposition to $\mathbf{H}$, we get $\mathbf{H} \xrightarrow{\mathtt{QR}} \mathbf{JK}$. Then we decompose $\mathbf{t\Sigma}$ in Equation (7) and it is fast as well:

$$[\mathbf{M}\ \mathbf{C}_i] = [\mathbf{U}\ \mathbf{J}]\mathbf{U}' \cdot \mathbf{\Sigma}' \cdot \left[\begin{array}{cc} \mathbf{V}^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{array} \right]^\top \mathbf{V}'^\top = \mathbf{U}'' \cdot \mathbf{\Sigma}' \cdot \mathbf{V}''^\top. \tag{8}$$

Then, we multiply $\mathbf{U}'' \cdot \mathbf{\Sigma}' \cdot \mathbf{V}''^\top$ as an updated $\mathbf{M}$ and finish an iteration procedure. Iterating this procedure until $\mathbf{C}_p$ has been updated, we get the final result of ISVD.

During the process, an important issue is to maintain necessary semantics (dimensions) to construct the semantic space. Producing too much noise will not only lower the precision of the semantic space, but also has an impact on the computational performance, *i.e.*, larger memory usage and longer computational time. Therefore, we maintain k dimensions, called the *threshold value*, to guarantee the effectiveness of the updating procedure and remove the extra dimensions immediately.

```
Input: The matrix $\mathbf{D} \in \mathbb{R}^{m \times n}$
Output: Matrices $\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}^\top$
$n' \leftarrow$ the initial value;
$q \leftarrow$ the batch size;
$p \leftarrow \lceil \frac{n-n'}{q} \rceil$;
$k \leftarrow$ the threshold value;
$\mathbf{M} \leftarrow$ the first $n'$ essay vectors of $\mathbf{D}$;
$[\mathbf{U}\ \mathbf{\Sigma}\ \mathbf{V}] \leftarrow$ svds($\mathbf{M}, k$);
for each $\mathbf{C}_i$ $(i = 1, 2, \ldots, p)$ do
    $[\mathbf{tU}, \mathbf{t\Sigma}, \mathbf{tV}] \leftarrow$ svds($[\mathbf{M}\ \mathbf{C}_i], k$);
    $[\mathbf{tU}', \mathbf{t\Sigma}', \mathbf{tV}'] \leftarrow$ svds($\mathbf{t\Sigma}, k$);
    $\mathbf{M} \leftarrow \mathbf{tU} \cdot \mathbf{tU}' \cdot \mathbf{t\Sigma}' \cdot \mathbf{tU}'^\top \cdot \mathbf{tU}^\top$;
end
$\mathbf{U} \leftarrow \mathbf{tU} \cdot \mathbf{tU}'$;
$\mathbf{\Sigma} \leftarrow \mathbf{t\Sigma}'$;
$\mathbf{V}^\top \leftarrow \mathbf{tV}'^\top \cdot \mathbf{tU}^\top$;
```

Algorithm 2: Incremental Singular Value Decomposition.

The algorithm of ISVD is presented in Algorithm 2. In this algorithm, the function `svds`$(\mathbf{M}, k)$ is used as conventional SVD where $\mathbf{M}$ is the matrix to be decomposed and k is the threshold value. First, it constructs the matrix $\mathbf{M}$ by the first n' essay vectors according to the initial value, to apply conventional SVD. Then, an intermediate result of decomposition (line 6) is produced.

The next step is to update the intermediate result by adding matrices $\mathbf{C}_i$ from the rest of $\mathbf{D}$. According to the mathematical derivation we introduced previously, it is easier and faster to update the intermediate result, and this updating result has been proved to be approximate to conventional SVD (Brand 2002). This process repeats until all the partitions $\mathbf{C}_i$ in the TF-IDF matrix have been updated to the intermediate result (lines 7-11). Finally, we get the final result of ISVD (lines 12-14).

3.3.3 Re-projection

By applying Equation (5), where k is the threshold value for the dimension retained, we construct a semantic space of the dataset. Any term-based essay vector, $\mathbf{d}_j$, can be re-projected to the space to obtain a uniform and semantic-based representation.

Suppose that $\mathbf{U} \cdot \mathbf{\Sigma} \cdot \mathbf{V}^\top$ is the final result of ISVD performed on the TF-IDF matrix of the training set, and $\mathbf{d}_j$ is an essay vector based on the same set of words $\mathcal{V}$. Then, we will re-project $\mathbf{d}_j$ to the semantic space as :

$$\widehat{\mathbf{d}_j} = \mathbf{\Sigma}^{-1} \cdot \mathbf{U}^\top \cdot \mathbf{d}_j \tag{9}$$

3.4 Scoring Performance Measurement

In this final part, we use a Support Vector Machine to automatically score essays (Peng and Wang 2009; Yannakoudakis, Briscoe, and Medlock 2011). Usually, SVM can be described as an optimization problem:

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \tfrac{1}{2}\mathbf{w}^\top\mathbf{w} + C\textstyle\sum_{i=1}^{l}\xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}^\top\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, i = 1, \ldots, l, \end{aligned}$$

where C is a positive regularization parameter. The input data are pairs of $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i$ is the feature vector, and y_i is the classification label. In SCESS, $\mathbf{x}_i$ and y_i can be regarded as a re-projected essay vector and the human score of this essay respectively. The kernel function $\phi(\mathbf{x}_i)$ maps the input data $\mathbf{x}_i$ to a higher-dimension space where $\mathbf{x}_i$ $(i = 1, \ldots, l)$ are separable according to y_i, so that when new data arrive, they can be classified correctly.

The choice of the kernel function is a nontrivial part of SVM, and it is also an open problem to design an appropriate kernel. In SCESS, the Radial Basis Function (RBF) is used for the kernel, for it allows non-linear relations between the features and the labels:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top\phi(\mathbf{x}_j) = \exp\left(-\gamma||\mathbf{x}_i - \mathbf{x}_j||^2\right), \gamma > 0. \tag{10}$$

Essay vectors from the training set and human scores are used to generate support vectors and establish decision planes. Based on these planes, the essay vectors from the testing set are classified according to support vectors. By this kind of classification, their class labels of essay vectors from the testing set are the automated scores predicted by SVM. Comparing the predicted scores with human scores, we can estimate the performance of automated essay scoring with ILSA.

4 Experimental Results and Discussion

In this section, we show experimental results of detection and correction for erroneous characters, and the experiments on automated essay scoring using ILSA. SCESS is implemented in C++ and all experiments were run on a machine with a 2GHz CPU and 96GB RAM under Linux.

4.1 Introduction to MHK

In MHK tests, test takers are required to write an essay of no more than 350 words based on a given topic, and these essays are restricted to a genre rather than free texts.

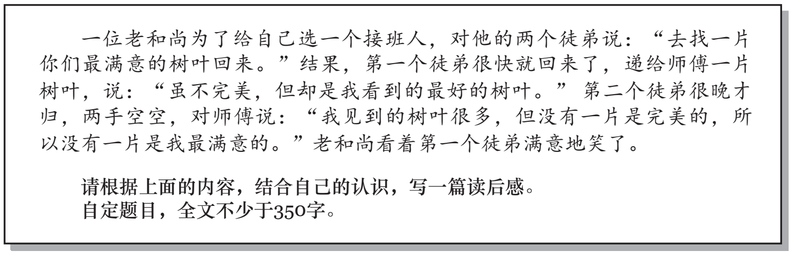
According to the scoring criteria of MHK tests, there are four levels for human raters to evaluate an essay (Peng and Wang 2009): *Character*, *Word*, *Sentence* and *Paragraph*. *Character and Word* are basic requirements including using correct characters, spelling and meaningful words in simplified Chinese; *Sentence* is a higher requirement and test takers should use correct grammar in a sentence and think about the relations between the topic and sentences; *Paragraph* is a consideration about the logical relations among paragraphs and even the whole passage.

A complete system scores essays from those four perspectives, and allocates different weights to them to get comprehensive assessments as overall scores. In consideration of expression and reading comprehension, *Character* and *Word* are basic criteria to assess an essay. Moreover, they are relatively easier to be studied compared with *Sentence* and *Paragraph*. Therefore, in this paper, we implement SCESS based on *Character* and *Word* levels, and we will develop SCESS further by combining all perspectives.

The procedure of human scoring is as follows. First, two human raters give each essay an initial score respectively, from point 1 to point 6 at intervals of 1. If the discrepancy between two scores for a certain essay surpasses 2 points, this essay will be rated by a senior researcher, and the final score will be the average of these three scores; otherwise, the final score will be the average of those two scores. The final score of the essay in MHK tests, ranging from 1 to 6 at intervals of 0.5, will be sent back to the test taker. In our experiments, we use the final score as the annotated score of each essay.

In order to give a first impression about the dataset we use, we give an assignment in the MHK test. All the essays in our dataset are written under the requirement shown in Figure 11. We randomly show two essays in Figure 12, whose scores are 5 and 1 respectively.

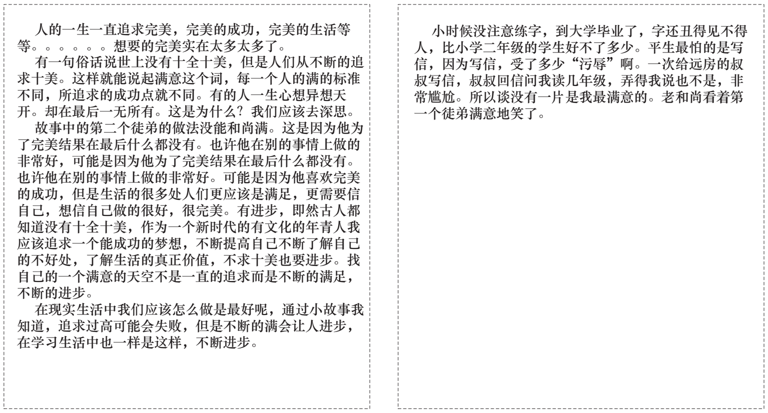
The given assignment varies from year to year, leading to the changes of contents



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。

Fig. 11. The assignment of the essays in our dataset. In this assignment, test takers are required to read a fable talking about the perfect leaf three monks are searching for, and write an essay, rephrasing this fable and expressing their thoughts according to their experiences. An effective response must contain a minimum of 350 words.



人的一生一直追求完美，完美的成功，完美的生活等等。。。。。。想要的完美实在太多太多了。

有一句俗话说世上没有十全十美，但是人们从不断的追求十美。这样就能说起满意这个词，每一个人的满的标准不同，所追求的成功点就不同。有的人一生心想异想天开。却在最后一无所有。这是为什么？我们应该去深思。

故事中的第二个徒弟的做法没能和尚满。这是因为他为了完美结果在最后什么都没有。也许他在别的事情上做的非常好，可能是因为他为了完美结果在最后什么都没有。也许他在别的事情上做的非常好。可能是因为他喜欢完美的成功，但是生活的很多处人们更应该是满足，更需要信自己，想信自己做的很好，很完美。有进步，即然古人都知道没有十全十美，作为一个新时代的有文化的年青人我应该追求一个能成功的梦想，不断提高自己不断了解自己的不好处，了解生活的真正价值，不求十美也要进步。找自己的一个满意的天空不是一直的追求而是不断的满足，不断的进步。

在现实生活中我们应该怎么做是最好呢，通过小故事我知道，追求过高可能会失败，但是不断的满会让人进步，在学习生活中也一样是这样，不断进步。

小时候没注意练字，到大学毕业了，字还丑得见不得人，比小学二年级的学生好不了多少。平生最怕的是写信，因为写信，受了多少“污辱”啊。一次给远房的叔叔写信，叔叔回信问我读几年级，弄得我说也不是，非常尴尬。所以谈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。

Human score: 5

Human score: 1

Fig. 12. Two representative essays of the MHK test (without translation).

of the essays, and thus, the changes of the set of words $\mathcal{V}$. Because LSA and ILSA are both content-oriented methods, for each MHK test, we need to train a different model. In 2009, $190,000$ students in Xinjiang Province in China took the test, and this number has been continuously increasing. Large scale model training requires a system like SCESS which can process big datasets.

4.2 Experiments on Detection and Correction for Erroneous Characters

4.2.1 Introduction to Datasets

We use two datasets in this part. The first one is a large corpus for training the n-gram model, whereas the second one is to test the word segmentation and correction algorithm based on WFSA.

Table 1. *The proportions of four kinds of errors. We have selected* 411 *sentences from* 200 *essays. There are* 607 *character-wise errors in total. These errors are easy for native speakers to correct, so we manually annotated the erroneous characters and corrected them.*

	Substitution	Deletion	Insertion	Transposition
Proportion	**76%**	13%	7%	4%

The selection of the corpus of the training set is important to experimental results because it influences the probability of every term in the N-gram model. We have tested three kinds of corpora: *The People's Daily*, diverse sources (Hodgepodge) and the Literature Corpus in Chinese.

The People's Daily is the most formal and influential newspaper in China, reporting official news and national affairs. By training on this corpus, we get an n-gram model, consisting of $47,493$ uni-grams, $3,716,267$ bi-grams and $954,446$ tri-grams in total.

The diverse source is a hodgepodge, containing various writings such as micro-blog posts, blog posts, publications, news articles, lyrics, subtitles and so forth. This corpus produces $47,494$ uni-grams, $3,246,941$ bi-grams and $908,188$ tri-grams.

The Literature Corpus includes writings in Chinese literature, such as the Mao Dun Literature Awards, which is the highest award for Chinese writers. By training on this corpus, we get $47,489$ uni-grams, $6,557,265$ bi-grams and $7,173,881$ tri-grams.

The testing set used for WFSA comes from the MHK test. We manually collected sentences from 200 test essays, and counted four kinds of character-wise writing errors. They are substitution, deletion, insertion and transposition errors and their proportions are shown in Table 1. From Table 1, we see that the substitution error is the most common and serious one. Therefore, in SCESS, we focus on detection and correction of substitution.

4.2.2 Performance Criteria

There are three performance criteria for estimating the results of detection and correction for erroneous characters, which are *Recall Rate*, *Detection Precision* and *Correction Precision* (Leacock, Chodorow, Gamon and Tetrault 2010):

$$\textit{Recall Rate} = \frac{N(W \to R) + N(W \to S)}{N(W)} \tag{11}$$

$$\textit{Detection Precision} = \frac{N(W \to R) + N(W \to S)}{N(W \to R) + N(W \to S) + N(R \to W)} \tag{12}$$

$$\textit{Correction Precision} = \frac{N(W \to R)}{N(W \to R) + N(W \to S) + N(R \to W)} \tag{13}$$

Table 2. *Comparison of three corpora. The testing set comes from the MHK test (200 essays). We set all the parameters empirically* $(the\ beam\ width = 10,\ the\ punishment\ value = 0.5,\ the\ heuristic\ function = H1)$.

Corpus	Recall Rate	Detection Precision	Correction Precision
People's Daily	86.55%	70.25%	65.14%
Hodgepodge	86.33%	74.67%	70.17%
Literature	**93.93%**	**78.58%**	**74.05%**

where $N(W)$ is the number of wrong characters in the original text; $N(W \rightarrow R)$ is the number of characters modified correctly; $N(W \rightarrow S)$ is the number of characters detected correctly but mistakenly modified and $N(R \rightarrow W)$ is the number of characters that are originally right but are mistakenly modified to wrong characters.

Additionally, we introduce $F-measure$ as an indicator, which is a common index in natural language processing. Adapted to WFSA, we get the equation:

$$F - measure = \frac{2 \cdot Detection\ Precision \cdot Recall\ Rate}{Detection\ Precision + Recall\ Rate}. \tag{14}$$

Equation (14) is used to compare WFSA with other methods.

4.2.3 Experimental Settings

The first step is to choose an appropriate corpus. Table 2 shows the comparison of three kinds of corpora, namely, People's Daily, Hodgepodge and Literature. People's Daily does not perform well, because its official form of usage is distant from daily use and test essays. Hodgepodge uses informal Chinese, so it is not satisfactory, either. Literature is proved to be the best one because it uses not only standard but also daily Chinese which is closest to the style of test essays, so we choose it for subsequent experiments.

After the corpus has been determined, we need to tune the key parameters to achieve best results. There are three parameters in our approach: the heuristic value, the punishment value and the beam width. As described in Section 2.4.2 (Heuristic Search), we have tested three heuristic functions and found H_1 $(= \frac{probability}{N_r} \times N_u)$ is the best one. We have also set different punishment values from 0.1 to 3.5 per modified character at intervals of 0.1 and found that 1.3 is the best choice. The beam width is used to limit the size of the pre-candidate set in the Beam Container. We have tested different beam widths from 5 to 30 at intervals of 5 and decided to choose 25 to balance the performance and the efficiency.

Table 3. *Comparisons of WFSA, Google and Baidu. The testing set comes from the MHK test (200 essays).*

	Recall Rate	Detection Precision	Correction Precision	F-measure
WFSA	**88.50%**	92.31%	88.46%	**0.9036**
Google	45.34%	93.30%	87.50%	0.6102
Baidu	17.79%	94.25%	89.66%	0.2993

4.2.4 Experimental Results and Analysis

Our experimental results are shown in Table 3. The Recall Rate of 88.50%, the Detection Precision of 92.31% and the Correction Precision of 88.46% indicate that our method is quite effective.

In order to further demonstrate that WFSA is effective in detecting and correcting erroneous characters, we compare WFSA with current Chinese text correction systems. They are used by `www.google.com` and `www.baidu.com`, both of which are popular search engines in China and provide detection and correction prompts for erroneous characters.

From Table 3, we see that the Recall Rate of WFSA performs much better than Google and Baidu, though the Detection Precision and the Correction Precision are slightly lower. F-measure demonstrates that WFSA is much more effective.

4.3 Experiments on ILSA

In the experiments on automated essay scoring using ILSA, MATLAB is used to multiply matrices and do re-projection. For the training set, their corresponding human scores (from 0 to 6 at intervals of 0.5) will be paired with the essay vectors in order to train the scoring model based on SVM. `LIBSVM` [2](Chang and Lin 2011) is used to help to complete the training and scoring.

4.3.1 Introduction to Datasets

In order to test ILSA, we use $157,760$ essays and $1,000$ essays with the same assignment from the MHK test as the training set and the testing set respectively. The human scoring distribution is shown in Figure 13.

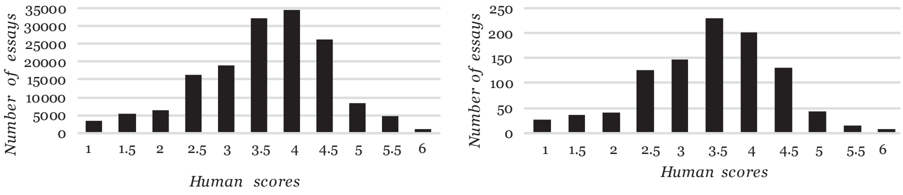


Fig. 13. Human scoring distribution of the training set (left) and the testing set (right).

Table 4. *Results of ILSA with different batch sizes. In order to smooth out biases that may occur in the processing, we run these experiments 10 times on the same dataset and compute average values.*

Batch size	Time(s)	Batch size	Time(s)	Batch size	Time(s)
10	463.7294	20	255.4205	30	180.6784
40	148.7053	50	124.1791	60	108.4457
70	103.2142	80	92.754	90	86.0899
100	82.7572	200	69.81877	**300**	**66.0840**
400	68.7404	500	69.9703	600	79.8691
700	87.6521	800	94.3936	1,000	105.9955

4.3.2 Optimal Batch Size

The batch size of essays is very important to ILSA, so we have conducted a series of experiments to find the optimal batch size. The results of ILSA as the batch size increases are shown in Table 4. In these experiments, we set both the threshold value and the initial value to 100. Then we set the batch size from 10 to 1,000, at intervals of 10 and 100. When the batch size is 0, there is no difference between conventional LSA and ILSA, because it means that we decompose the matrix in only one iteration procedure. That is to say, if batch size is 0, when we are performing conventional SVD on the initial matrix **M** in line 6 in Algorithm 2, we are in fact decomposing the original matrix **D**, and the Algorithm 2 finishes here. Apparently, in this case, no incremental decomposition happens. As the batch size increases,

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

Table 5. *Optimal batch size.*

Batch size	Time(s)	Batch size	Time(s)	Batch size	Time(s)
260	68.84589	270	68.03251	280	67.89831
290	66.77077	300	666.08409	310	65.95101
320	**65.23065**	330	66.52441	340	66.03742
350	66.58706	360	66.34916	370	67.18953
380	67.26448	390	68.7329	400	68.7404

Table 6. *Optimal batch sizes for different sizes of datasets. In order to smooth out biases that may occur in the processing, we run these experiments 10 times on the same dataset and compute average values.*

Size of dataset	Optimal batch size	Size of dataset	Optimal batch size
10, 000	296	20, 000	308
30, 000	320	40, 000	348
50, 000	360	60, 000	432
70, 000	476	80, 000	490

from Table 4, we can see that the time of ILSA decreases sharply at first, and continues increasing gradually.

When we concentrate on the results from 260 to 400 as shown in Table 5, we can see the optimal batch size is 320. Therefore, in the subsequent experiments, the batch size is set to 320.

4.3.3 Relative Update Time

The *relative update time* is used to observe the performance of ILSA based on the size of the dataset and the optimal batch size:

$$relative\ update\ time = \frac{(n - n')}{optimal\ batch\ zize} \times \frac{base}{n} \tag{15}$$

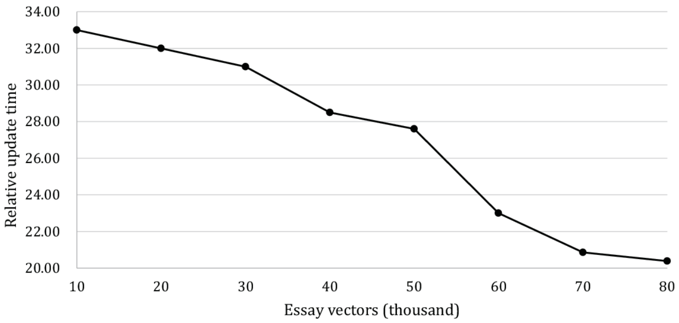


Fig. 14. Relative update time.

where n is the number of essay vectors (the size of the original dataset matrix), and n' is the initial value. *Base* means the size of the dataset as a unit. For example, in our experiment, *base* is $100,000$ because the size of the training set is $157,760$. Thus, the relative update time reflects the ability of ILSA when faced with different sizes of datasets.

To make it clear, we have conducted a series of experiments. In these experiments, we increase n from $10,000$ to $80,000$ at intervals of $10,000$ with $n' = 100$. Fixing n and n', by using various batch sizes on the decomposition and observing the running time, we choose optimal batch sizes for different sizes of datasets in Table 6. From Table 6, we can see that as the size of the dataset grows, the optimal batch size grows as well.

For showing that the performance of ILSA does not become worse, we compute the relative update time and plot it in Figure 14. From Figure 14, we can see that as n grows, the relative update time decrease, meaning that ILSA performs more efficiently as the dataset grows larger. For instance, when $n = 20,000$, ILSA updates decomposition in 32 iteration procedures; when $n = 80,000$, ILSA finishes updating in only about 20 iteration procedures.

4.3.4 Comparison of ILSA and Conventional LSA

The comparison of ILSA and conventional LSA for running time is illustrated in Figure 15. In the experiments, we increase the size of the training set from $31,552$ to $157,760$. Figure 15 shows that when the size grows larger, ILSA performs far more efficiently than conventional LSA. Specifically, when the size grows to $110,432$, the time of conventional LSA is more than two hours $(7,200\text{s})$, as is shown in Figure 15, so it is obvious that ILSA is much better.

In addition to running time, memory usage is another huge advantage of ILSA. Figure 16 shows the comparison of ILSA and conventional LSA for memory usage as the size of the training set grows. From Figure 16, we can see that the maximum memory usage of ILSA is only 492MB, and moreover, it performs relatively stably. In sharp contrast, conventional LSA uses much more memory and

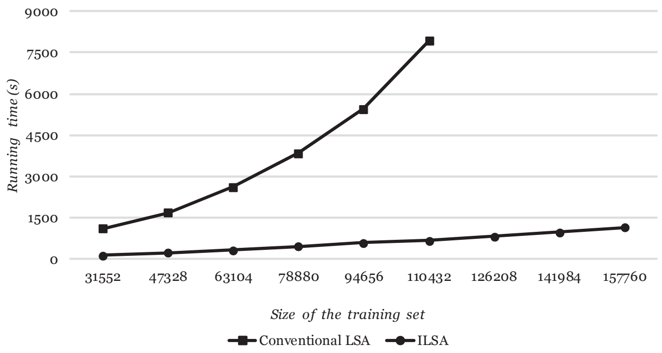


Fig. 15. Comparison of ILSA and conventional LSA for running time.

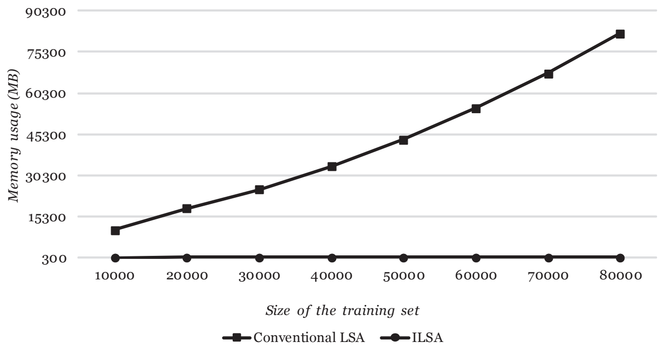


Fig. 16. Comparison of ILSA and conventional LSA for memory usage.

increases distinctly. In practical application, language tests usually produce a big dataset, so, even given huge memory, it is impossible for conventional LSA to finish the decomposition task, but it is viable for ILSA to perform it.

4.3.5 Scoring Performance

From the experimental results, we see that ILSA has great advantages both in time and memory usage. More encouragingly, ILSA does not weaken the scoring performance, compared with conventional LSA. To evaluate ILSA, we use several criteria: *Scoring Accuracy*, *Quadratic Weighted Kappa* and *Spearman's Coefficient*.

Scoring Accuracy : The Scoring Accuracy is calculated as follows:

$$\mathit{Scoring\ Accuracy} = \frac{\sum_{i=1}^{n} t(hs_i, ps_i)}{n} \tag{16}$$

where n is the size of the testing set, and hs_i and ps_i are the human score

Table 7. *Scoring performance of ILSA, conventional LSA, Baseline and Human. The size of the testing set is* $1,000$.

	ILSA	LSA	Baseline	Human
Acceptable Scorings	**895**	889	752	860
Unacceptable Scorings	**105**	111	248	140
Scoring Accuracy	**89.50**%	88.90%	75.20%	86.00%
Quadratic Weighted Kappa	0.56	**0.58**	0.00	0.54
Spearman's Correlation	**0.61**	**0.61**	0.00	0.53

and predicted score of the i-th essay respectively. The function $t(hs_i,\, ps_i)$ is binary, defined as:

$$t(hs_i, ps_i) = \begin{cases} 1 & |hs_i - ps_i| \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

If the difference between a human score and a predicted score is no more than 1 point, it is acceptable.

Quadratic Weighted Kappa : For Quadratic Weighted Kappa, we construct two confusion matrices. The first one shows the human scores and the predicted scores given by conventional LSA, and the second one shows the human scores and the predicted scores given by ILSA. Quadratic Weighted Kappa takes chance agreement into account.

Spearman's Correlation : We calculate the Spearman's Correlation ρ:

$$\rho_{X,Y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2(y_i - \bar{y})^2}} \tag{18}$$

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

In order to give an overall impression of SCESS, we show the results in Table 7. In this table, not only are the performances of LSA and ILSA compared, a further comparison of a baseline classifier (Baseline) and two human raters' performance (Human) is also shown.

The baseline classifier counts the most frequent score in the training set, and thus, gives the predicted scores of the essays in the testing set. In our training set, since the most frequent score is 4, this baseline classifier will give point 4 to all the essays in the testing set. The extreme low values of the Quadratic Weighted Kappa and Spearman's Correlation verify this unreasonable method.

Table 8. *Scoring deviations of ILSA and conventional LSA*

Deviation	ILSA	LSA
$(1, 1.5]$	71	80
$(1.5, 2]$	27	24
$(2, 2.5]$	7	6
$(2.5, 3]$	0	1

From Table 7, we see that ILSA has the best Scoring Accuracy. The Spearman's Correlations of ILSA and LSA are the same and it is better than that of Human. Although the Quadratic Weighted Kappa of ILSA is slightly lower than LSA, ILSA is still effective, because the difference is very small.

Focusing on ILSA and LSA, we use Table 8 to show the scoring deviations. Each cell is the number of the unacceptable scores whose differences with human scores lie in the corresponding deviation interval.

5 Conclusions and Future Work

In this paper, we describe the development of an automated simplified Chinese essay scoring system based on WFSA and ILSA, called SCESS. Combined with the Confusing-Character Table, the Part-Of-Speech Table, beam search and heuristic search, WFSA can effectively segment Chinese sentences into words. In addition, it can detect and correct erroneous simplified Chinese characters. A Recall Rate of 88.50%, a Detection Precision of 92.31% and a Correction Precision of 88.46% show that WFSA is very effective. After segmentation, SCESS uses ILSA to process segmented essays and extract semantic features. Finally, we use SVM to score essays automatically. From the experimental results, we see that ILSA is quite efficient, because it significantly reduces both running time and memory usage. Additionally, it can successfully score essays with the Scoring Accuracy of 89.50%. Overall, SCESS proves to be promising.

In the future, we will test SCESS on more assignments so that the generalizability can be verified. For further improvement of SCESS, we will continue to develop it on *Character* and *Word* levels. For example, we will consider essay contexts to identify the gender of a person and utilize names of entities. For a complete and mature SCESS, WFSA-based detection and correction for erroneous characters will be integrated. Moreover, we will study more methods to assess essays automatically on *Sentence* and *Paragraph* levels. Many novel methods will be tested, including Contextualized Latent Semantic Indexing (CLSI), probabilistic LSA (pLSA), Latent Dirichlet Allocation (LDA), hierarchical LDA (hLDA) and so forth.

Acknowledgements

We would like to thank the anonymous referees for their helpful comments and suggestions on the earlier versions of this paper and Bin Huang, Mingqing Zhang and Zongtian Gao for their help in the experiments. This work is supported by the Beijing Higher Education Young Elite Teacher Project (YETP0768), the Fundamental Research Funds for the Central Universities (YX2014-18) and the National Natural Science Foundation of China (61103152 and 61472369).

References

Attali, Y., and Burstein J. (2006) Automated Essay Scoring With e-rater v.2.0. In *Journal of Technology, Learning, and Assessment*, **4(3)**. Available at `http://www.jtla.org/`.

Brand, M. (2002) Incremental singular value decomposition of uncertain data with missing values. In Anders Heyden, Gunnar Sparr, Mads Nielsen, Peter Johansen (Eds.), *Proceedings of the 2002 European Conference on Computer Vision (ECCV 2002)*, Copenhagen, Denmark. Springer Lecture Notes in Computer Science volume 2350, pp. 707–720. Berlin: Springer Verlag.

Brand, M. (2003) Fast online SVD revisions for lightweight recommender systems. In Daniel Barbar, Chandrika Kamath (Eds.), *Proceedings of the 3rd SIAM International Conference on Data Mining 2003*, San Francisco, CA, USA, pp. 37–46. SIAM.

Burstein, J. (2003) The E-rater Scoring Engine: Automated Essay Scoring with Natural Language Processing. In Shermis, M.D. and Burstein J. (eds.), *Automated Essay Scoring: A Cross-Disciplinary Perspective*, pp. 113–121. Mahwah, NJ: Lawrence Erlbaum Associates.

Burstein, J., and Chodorow, M. (2010) Progress and New Directions in Technology for Automated Essay Evaluation. In Kaplan, R.B. (eds.), *The Oxford Handbook of Applied Linguistics, 2nd Edition*, pp. 487–497. Oxford: Oxford University Press.

Cao, Y.W., and Chen, Y. (2007) Automated Chinese essay scoring with latent semantic analysis. *Examinations Research* **3(1)**: 63-71. Tianjin: Tianjin People's Press.

Chang, C.C., and Lin, C.J. (2011) LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* **2(3)**: 27:1-27:27. New York, NY: Association for Computing Machinery (ACM).

Chang, T.H., Lee, C.H., and Tam, H.P. (2007). On developing techniques for automated Chinese essay scoring: a case in ACES system. Paper presented at *the Forum for Educational Evaluation in East Asia*.

Chang, T.H., Lee, C.H., Tsai, P.Y., and Tam, H.P. (2009) Automated essay scoring using set of literary sememes. *Information: An International Interdisciplinary Journal* **12(2)**: 351-357. Tokyo, Japan: International Information Institute.

Chang, T.H., Chen, H.C., Tseng, Y.H., and Zheng, J.L. (2013) Automatic detection and correction for Chinese misspelled words using phonological and orthographic similarities. In *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing (ACL-SIGHAN 2013)*, Nagoya, Japan, pp. 97–101. Asian Federation of Natural Language Processing.

Chang, T.H., Sung, Y.T., and Lee, Y.T. (2013) Evaluating the difficulty of concepts on domain knowledge using latent semantic analysis. In *Proceedings of International Conference on Asian Language Processing*, Urumqi, China, pp. 193–196. Washington DC: IEEE Computer Society Press.

Chen, Z.P., Lv, Y.Q., Liu, H.S., and Tu, H. (2009) Chinese spelling correction in search engines based on n-gram model. *Journal of China Academy of Electronics and Information Technology* **4(3)**: 323-326. Beijing: China Academy of Electronics and Information Technology.

Chin, T.J., Schindler, K., and Suter, D. (2006) Incremental kernel SVD for face recognition with image sets. In *Proceeding of the 7th International Conference on Automatic Face and Gesture Recognition*, Southampton, UK, pp. 461–466. Washington DC: IEEE Computer Society Press.

Elliot, S. (2003) Intellimetric TM: From Here to Validity. In Shermis, Mark D. and Burstein J. (eds.), *Automated Essay Scoring: A Cross-Disciplinary Perspective*, pp. 71–86. Mahwah, NJ: Lawrence Erlbaum Associates.

Foltz, P. W., Laham D. and Landauer T. K. (1999) The intelligent essay assessor: Applications to educational technology. *Interactive Multimedia Electronic Journal of Computer Enhanced Learning* **1(2)**. Winston-Salem, NC: Wake Forest University.

Gorrell, G. (2006) 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 Computational Linguistics*, Trento, Italy, pp. 97–104. Stroudsburg, PA: Association for Computational Linguistics (ACL).

Hao, S.D., Gao, Z.T., Zhang, M.Q., Xu, Y.Y., Peng, H.L., Ke, D.F., and Su, K.L. (2013) Automated error detection and correction of Chinese characters in written essays based on weighted finite-state transducer. In *Proceedings of the 12th International Conference on Document Analysis and Recognition 2013 (ICDAR 2013)*, Washington DC, USA, pp. 763–767. Washington DC: IEEE Computer Society Press.

Hao, S.D., Xu, Y.Y., Peng, H.L., Su, K.L., and Ke, D.F. (2014) Automated Chinese Essay Scoring From Topic Perspective Using Regularized Latent Semantic Indexing. In *Proceedings of the 22nd International Conference on Pattern Recognition 2014 (ICPR 2014)*, Stockholm, Sweden, pp. 3092–3097. Washington DC: IEEE Computer Society Press.

Ismail, S., Othman, R.M., and Kasim, S. (2011) Pairwise protein substring alignment with latent semantic analysis and support vector machines to detect romote protein homology. In *Ubiquitous Computing and Multimedia Applications*, pp. 526–546. Berlin: Springer Verlag.

Jin, Y., Gao, Y., Shi, Y., Shang, L., Wang, R., and Yang, Y. (2011) P2lsa and p2lsa+: Tow paralleled probabilistic latent semantic analysis algorithms based on the mapreduce model. In Colin Fyfe, Peter Tino, Darryl Charles, Cesar Garca-Osorio, Hujun Yin (Eds.), *Intelligent Data Engineering and Automated Learning*, Springer Lecture Notes in Computer Science, pp. 385–393. Berlin: Springer Verlag.

Ke, D.F., Peng, X.Y., Zhao, Z., Chen, Z.B., and Wang, J.S. (2011) Word-level-based automated Chinese essay scoring method. In *Proceedings of National Conference on Man-Machine Speech Communication*, Xi'an, China, pp. 57–59. Beijing: Chinese Information Processing Society of China.

Landauer, T. K, Foltz, P. W., and Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes* **25(2-3)**: 259–284. London: Routledge.

Leacock, C., Chodorow, M., Gamon, M., and Tetreault, J. (2010) *Automated grammatical error detection for language learners*. Princeton, NJ: Morgan & Claypool Publishers.

Li, C., Peng, X.Y., and Zhao, J. (2011). Research on assisted scoring system for Chinese proficiency test for minority. *Journal Chinese Information Processing* **25(5)**: 120–127. Beijing: Chinese Information Processing Society of China.

Li, Y.N. (2006). Automated essay scoring for testing Chinese as a second language. *PhD Thesis*, Beijing: Beijing Language and Culture University.

Liu, C.H., Wang, Y.C., and Liu, D.R. (2007). Using LSA and text segmentation to improve automatic Chinese dialogue text summarization. *Journal of Zhejiang University Science A* **8(1)**: 79–87. Zhejiang: Zhejiang University.

Lv, S.X. (1999). *Eight hundred words in modern Chinese*. Beijing: The Commercial Press (Beijing) Ltd.

Ma, J.S., Zhang, Y., Liu, T., and Li, S. (2004). Detecting Chinese text errors based on trigram and dependency parsing. *Journal of The China Society For Scientific and Technical Information* **6**. Beijing: Science and Technology Information Society of China.

McInerney, J., Rogers, A., and Jennings N.R. (2012). Improving location prediction services for new users with probabilistic latent semantic analysis. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pp. 906–910. New York, NY: Association for Computing Machinery (ACM).

Mesaros, A., Heittola, T., and Klapuri, A. (2011). Latent semantic analysis in sound event detection. In *Proceedings of the 19th European Signal Processing Conference (EUSIPCO 2011)*, Barcelona, Spain, pp. 1307–1311. The European Association for Signal Processing.

Nakov, P., Popova, A., and Mateev, P. (2001) Weight functions impact on LSA performance. In *Proceedings of EuroConference Recent Advances in NLP (RANLP 2001)*, Tzigov Chark, Bulgaria, pp. 187–193. Stroudsburg, PA: Association for Computational Linguistics (ACL).

Page, E.B. (1994) Computer grading of student prose, using modern concepts and software. *Journal of Experimental Education* **62(2)**: 127–142. London: Taylor & Francis, Ltd. UK.

Pan, H., and Yan, J. (2009) An algorithm of text automatic proofreading based on chinese word segmentation. *Journal of Wuhan University of Technology* **31(3)**: 18–28. Wuhan: Wuhan University of Technology.

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

Peng, X.J., and Wang, Y.F. (2009) CCH-based geometric algorithms for SVM and applications. *Applied Mathematics and Mechanics* **30(1)**: 89–100. Berlin: Springer Verlag.

Peng, H.L., and Yu, Y.Y. (2013) Research on controlling central rating in net-based scoring of subjective test items. *China Examinations* **6**: 3–9. Beijing: China Examinations.

Peng, X.Y., Ke, D.F., and Xu, B. (2012) Automated essay scoring based on finite state transducer: towards ASR transcription of oral English speech. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, Jeju, Korea, pp. 50–59. Stroudsburg, PA: Association for Computational Linguistics (ACL).

Ramineni, C., Trapani, C.S., Williamson, D.M., Davey, T., and Bridgeman, B. (2012) Evaluation of the e-rater Scoring Engine for the TOEFL Independent and Integrated Prompts. *Research Report ETS RR-12-06*, `http://www.ets.org/Media/Research/pdf/RR-12-06.pdf`.

Rosenfeld, R. (1994) Adaptive statistical language modeling a maximum entropy approach. *PhD Thesis*, CMU-CS-94-138, Pittsburgh, PA: Carnegie Mellon Universiy.

Rudner, L.M., and Liang, H. (2002). Automated essay scoring using Bayes theorem. In *Journal of Technology, Learning, and Assessment*, **1(2)**. Available at `http://www.jtla.org/`.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2002) Incremental singular value decomposition algorithms for highly scalable recommender systems. In *Proceedings of the 5th Conference on Computer and Information Science*, Greece, Athens, pp. 27–28. Washington DC: IEEE Computer Society Press.

Shermis, M.D., and Burstein, J.C. (2003) *Automated esssay scoring: a cross-disciplinary perspective*. Mahwah, NJ: Lawrence Erlbaum Associates.

Stolcke, A. (2002) SRILM - An extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing (INTERSPEECH 2002)*, Denver, Colorado, USA, pp. 901–904. Washington DC: IEEE Computer Society Press.

Teahan, W.J., Wen, Y., McNab, R.J., and Witten, I.H. (2000) A compression-based algorithm for Chinese word segmentation. *Computational Linguistics*, **26(3)**: 375–393. Cambridge, MA: MIT Press.

Tonta, Y., and Darvish H. R. (2010) Diffusion of latent semantic analysis as a research tool: a social network analysis approach. *Journal of Informetrics* **4(2)**: 166–174. Philadelphia, PA: Elsevier.

Steinbiss, V., Tran, B., and Ney, H. (1994) Improvements in beam search. In *Proceedings of the 3rd International Conference on Spoken Language Processing (ICSLP 1994)*, Yokohama, Japan. Washington DC: IEEE Computer Society Press.

Wang, D.H., and Liu, C.L. (2011) Dynamic text line segmentation for real-time recognition fo Chinese handwritten sentences. In *Proceedings of the 11th International Conference on Document Analysis and Recognition 2011 (ICDAR 2011)*, Beijing, China, pp. 931–935. Washington DC: IEEE Computer Society Press.

Wang, L., and Wan, Y. (2011) Sentiment classification of documents based on latent semantic analysis. In *Advanced Research on Computer Education, Simulation and Modeling*, pp. 356–361. Berlin: Springer Verlag.

Wang, W., and Yu, B. (2009) Text categorization based on combination of modified back propagation neural network and latent semantic analysis. *Neural Computing and Applications* **18(8)**: 875–881. Berlin: Springer Verlag.

Yue, W.Y., Xu, Y.Y., and Su, K.L. (2006) BDDRPA*: An Efficient BDD-Based Incremental Heuristic Search Algorithm for Replanning. In Abdul Sattar, Byeong Ho Kang (Eds.), *Proceedings of Australian Conference on Artificial Intelligence*, pp. 627–636. Berlin: Springer Verlag.

Wang, Q., Xu, J., Li, H., and Craswell, N. (2013) Regularized latent semantic indexing: A new approach to large-scale topic modeling. *ACM Transactions on Information Systems*, **31(1)**: pp. 5:1–5:44. New York, NY: Association for Computing Machinery (ACM).

Wild, F., Stahl, C., Stermsek, G., Neumann, G., and Penya, Y. (2005). Parameters Driving Effectiveness of Automated Essay Scoring with LSA. In *Proceedings of the 9th Computer Assisted Assessment Conference (CAA Conference 2005)*, pp. 485–494. Loughborough: Loughborough University.

Wu, Y., Li, X.K., Liu, T., and Wang, K.Z. (2001). Research on and implementation of Chinese text proof-reading system. *Journal of Harbin Institute of Technology* **33(1)**. Harbin: Harbin Institute of Technology.

Xu, Y.Y., Yue, W.Y., and Su, K.L. (2009). The BDD-Based Dynamic A* Algorithm for Real-Time Replanning. In Xiaotie Deng, John E. Hopcroft, Jinyun Xue (Eds.), *Proceedings of Frontiers in Algorithmics, Third International Workshop*, pp. 271–282. Berlin: Springer Verlag.

Xu, Y.Y., and Yue, W.Y. (2009) A Generalized Framework for BDD-based Replanning A* Search. In *Proceedings of the 10th International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing*, Daegu, Korea, pp. 133–139. Washington DC: IEEE Computer Society Press.

Yannakoudakis, H., Briscoe, T., and Medlock, B. (2011) A New Dataset and Method for Automatically Grading ESOL Texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, Portland, Oregon, USA, pp. 180–189. Stroudsburg, PA: Association for Computational Linguistics (ACL).

Yeh, J.Y., Ke, H.R., and Yang, W.P. (2002) Chinese text summarization using a trainable summarizer and latent semantic analysis. In Ee-Peng Lim, Schubert Foo, Christopher S. G. Khoo, Hsinchun Chen, Edward A. Fox, Shalini R. Urs, Costantino Thanos (Eds.), *Digital Libraries: People, Knowledge, and Technology*, pp. 76–87. Springer 2002 Lecture Notes in Computer Science. Berlin: Springer Verlag.

Zhao, Y.H. (2011) Application of latent semantic analysis in auto-grading system. *Journal of Yanbian University (Natural Science)* **37(4)**: 345–348. Jilin: Yanbian University.

2014 22nd International Conference on Pattern Recognition

Automated Chinese Essay Scoring From Topic Perspective Using Regularized Latent Semantic Indexing

Shudong Hao, Yanyan Xu
School of Information Science and Technology
Beijing Forestry University
Beijing, China
shudongh@acm.org
xuyanyan@bjfu.edu.cn

Hengli Peng
Institute of Educational Measurement
Beijing Language and Culture University
Beijing, China
penghl6402@aliyun.com

Kaile Su
College of Mathematics Physics and Information Engineering
Zhejiang Normal University
Zhejiang, China
Institute for Integrated and Intelligent Systems
Griffith University
Brisbane, Australia
k.su@griffith.edu.au

Dengfeng Ke*
Institute of Automaton
Chinese Academy of Sciences
Beijing, China
dengfeng.ke@ia.ac.cn

***Abstract*—Finding out an effective way to score Chinese written essays automatically remains challenging for researchers. Several methods have been proposed and developed but limited in the character and word usage levels. As one of the scoring standards, however, content or topic perspective is also an important and necessary indicator to assess an essay. Therefore, in this paper, we propose a novel perspective – topic, and a new method integrating topic modeling strategy called Regularized Latent Semantic Indexing to recognize the latent topics and Support Vector Machines to train the scoring model. Experimental results show that automated Chinese essay scoring from topic perspective is effective which can improve the rating agreement to 89%.**

Keywords—document understanding; automated Chinese essay scoring; topic modeling application; classification application

I. INTRODUCTION

Automated essay scoring is becoming increasingly popular in the field of educational evaluation technology. Comparing to the conventional human rater method, computer-assistant scoring has advantages over three aspects: less human labor, less subjective factors and higher agreement rates [1]. Therefore, since 1970s when Page firstly broke the dawn of the field of automated English scoring, this technique has captured the interests of both the educators and researchers. After that, several successful scoring systems, such as Project Essay Grader (PEG), Intelligent Essay Assessor (IEA), E-rater, IntelliMetric and Bayesian Essay Test Scoring System (BETSY), have been applied into industry widely and successfully [2]. Because of the characteristic of Chinese which leads to the difficulties in processing texts, the research on automated scoring for Chinese essays started later than that for English and is developing slowly [3].

To evaluate an essay written in any languages, the raters often focus on such three aspects: language proficiency, structure arrangement and contents [4]. Since we focus on Chinese written essays, we take MHK (the minorities-oriented Chinese level test) for an example [5]. In MHK, language proficiency is the basic requirement that includes using Chinese characters correctly, choosing words or phrases properly, and writing sentences grammatically; structure arrangement requires test takers to arrange passages logically, in order to support the idea effectively; contents focus on the topic the test taker chooses.

From these three assessment perspectives, researchers have studied several techniques for automated essay scoring for Chinese by virtue of previous that for English. For language proficiency, word-level-based method [6] and Latent Semantic Analysis (LSA) [7] etc., have been introduced. Furthermore, incremental LSA [8] has also been proposed to reduce the computational time and memory usage. The lack of scoring from structure arrangement and contents, however, leads to an incomplete assessment.

Considering these current limitations, we propose the following assumptions: *1)* the topic(s) hidden in an essay is related to the score the writer gets; and *2)* there may exist more than one topic in an essay collection. The first assumption is reasonable, because the content depends on the topic(s), and that is one of the scoring standards. The second assumption leads to that we can apply topic modeling strategies which are popular in text mining, information retrieval and natural language processing [9], into automated Chinese essay scoring, making it possible to score essays from topic perspective.

Latent Dirichlet Allocation (LDA), a representative and successful method in topic modeling, was firstly proposed by David M. Blei [10]. Some other improved approaches such as probabilistic Latent Semantic Indexing (pLSI) and Non-negative Matrix Factorization (NMF) are also studied and applied later as well. Based on some assumptions needed to be maintained during computations, these methods are difficult to be scaled to process big datasets. Recently, a new method is proposed by Quan Wang et al. called Regularized Latent Semantic Indexing (RLSI) which can handle with scalability problems [11-12]. Therefore, we use RLSI learning algorithm to recognize the latent topics in the datasets. This method not

* *corresponding author.*

1051-4651/14 $31.00 © 2014 IEEE
DOI 10.1109/ICPR.2014.533

only performs better according to experimental results, but can be applied into big datasets in the future, as the scale of MHK is becoming larger.

The rest of the paper is organized as follows. In section II, we introduce our proposed method, and we present the experimental results and some discussions in section III. At last, in section VI, we conclude this paper and point out the future work.

II. The Proposed Method

In this paper, we propose a method of automated Chinese essay scoring from topic perspective, whose flowchart is shown in Fig. 1. The whole processing consists of three parts: dataset preparation and text-preprocessing, topic feature extraction, and scoring model training, which will be described as follows.

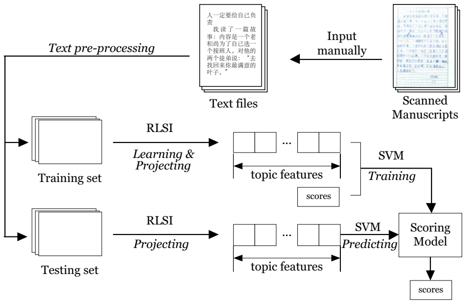


Fig. 1. The flowchart of the proposed method.

A. Dataset preparation and text pre-processing

Raw materials for datasets are scanned manuscripts that are needed to be typed into texts manually.

In text pre-processing, firstly, we use our previous research, Weighted Finite-State Transducer (WFST), to segment the essays to words with relatively high accuracy [13]. WFST is a strategy which segments sentences, and detects and corrects the erroneous characters in Chinese sentences simultaneously. In automated essay scoring, we use WFST for segmentation without corrections for erroneous characters.

Next, with the elimination of stopwords (such as auxiliary verbs) and rare words (occur only one or two times in the whole dataset), we get the term-document matrix **D** where each column is an *M*-dimensional term-based *essay vector* ***d***. *M* is the number of different terms and an essay vector denotes an written essay. The element $d_{i,j}$ is the frequency of the term *i* appeared in the essay ***j***.

For this occurrence-based matrix, we use TF-IDF strategy to weight it in order to achieve a better performance [14]. Generally, TF-IDF weights each element $d_{i,j}$ in the matrix as:

$$d_{i,j} = TF_{i,j} \times IDF_{i,j} \tag{1}$$

where TF is the term frequency and IDF is the inverse document frequency. Therefore, each element $d_{i,j}$ in the matrix **D** will be weighted as:

$$d_{i,j} = \frac{count_{i,j}}{\sum_{m=1}^{M} count_{m,j}} \times log \frac{N}{1 + DF_i} \tag{2}$$

where $count_{i,j}$ denotes the number of term *i* occurred in the essay vector ***j*** and DF_i the number of essay vectors containing the term *i*; *N* is the number of essay vectors in a certain collection, training set or testing set, and it is easy to figure out according to the context.

B. Topic feature extraction

In this subsection, we will introduce how to apply RLSI learning algorithm to automated Chinese essay scoring. Given a term-document matrix **D**, RLSI will generate two matrices, the term-topic matrix **U** and topic-document matrix **V**, that approximate the original matrix **D** and thus minimizes the quadratic loss function:

$$\min_{\boldsymbol{U},\boldsymbol{V}} \left(\|\boldsymbol{D} - \boldsymbol{U}\boldsymbol{V}\|_F^2 + \lambda_1 \|\boldsymbol{U}\|_1 + \lambda_2 \|\boldsymbol{V}\|_F^2 \right) \tag{3}$$

where $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$ regularize **U** and **V**. The dimension of topic is denoted as topic number *K*. Thus, the target can be formalized as an optimization problem where there are two parameters. A common method to solve such problems is to fix one parameter and update the other, and continue this interchange until convergence.

*1) Updating **U***

Initializing randomly and fixing the matrix **V**, we start from updating the matrix **U**. In this step, the optimization problem is transformed into:

$$\min_{\boldsymbol{U}} \left(\|\boldsymbol{D} - \boldsymbol{U}\boldsymbol{V}\|_F^2 + \lambda_1 \|\boldsymbol{U}\|_1 \right). \tag{4}$$

In order to compute iteratively or parallel, we rewrite the equation (3) as below:

$$\min_{\{\boldsymbol{u}_m\}} \sum_{m=1}^{M} \left(\|\boldsymbol{d}_m - \boldsymbol{V}^T \boldsymbol{u}_m\|_2^2 + \lambda_1 \|\boldsymbol{u}_m\|_1 \right) \tag{5}$$

where $\boldsymbol{d}_m = [d_{m1}, d_{m2}, \ldots, d_{mN}]^T$ is the essay vector, and $\boldsymbol{u}_m = [u_{m1}, u_{m2}, \ldots, u_{mK}]^T$ is from matrix **U**. Thus, the updating of **U** will be separated into *M* iterations. Using coordinate descent method with soft-threshold operator [15-17], the solution to each element *u* in this problem is:

$$u_{mk}^{*} = \frac{\left(\left|r_{mk} - \sum_{l \neq k} s_{kl} u_{ml}\right| - \frac{1}{2}\lambda_1\right)_{+} \cdot sign(r_{mk} - \sum_{l \neq k} s_{kl} u_{ml})}{s_{kk}} \tag{6}$$

where s is from $\mathbf{S} = \mathbf{V}\mathbf{V}^T$ and r is from $\mathbf{R} = \mathbf{D}\mathbf{V}^T$, and $(\cdot)_+$ is the hinge function [18].

2) Updating ***V***

Similarly, updating **V** can be formed as:

$$\begin{aligned} &\min_{\boldsymbol{V}} \|\boldsymbol{D} - \boldsymbol{U}\boldsymbol{V}\|_F^2 + \lambda_2 \|\boldsymbol{V}\|_F^2 \\ &= \min_{\{\boldsymbol{v}_n\}} \sum_{n=1}^{N} \left(\|\boldsymbol{d}_n - \boldsymbol{U}\boldsymbol{v}_n\|_2^2 + \lambda_2 \|\boldsymbol{v}_n\|_2^2\right) \end{aligned} \tag{7}$$

and separated into N iterations or parallelized computations.

This is a Ridge Regression problem with $\ell 2$-norm regularization, and the solution is:

$$\boldsymbol{v}_n^{*} = \left(\boldsymbol{U}^T\boldsymbol{U} + \lambda_2 \boldsymbol{I}\right)^{-1} \boldsymbol{U}^T \boldsymbol{d}_n \tag{8}$$

where **I** is the identity matrix with corresponding dimension.

3) Projection

After learning, the matrix **U** constructs a topic space where any term-based essay vector $\boldsymbol{v}_q$ can be projected and represented as a topic-based essay vector $\boldsymbol{v}_q^{*}$. Given an essay vector $\boldsymbol{v}_q$, the target is:

$$\min_{\boldsymbol{v}} \left(\|\boldsymbol{v}_q - \boldsymbol{U}\boldsymbol{v}\|_2^2 + \lambda_2 \|\boldsymbol{v}\|_2^2 \right), \tag{9}$$

and the solution is:

$$\begin{aligned} \boldsymbol{v}_q^{*} &= \arg\min_{\boldsymbol{v}} \|\boldsymbol{v}_q - \boldsymbol{U}\boldsymbol{v}\|_2^2 + \lambda_2 \|\boldsymbol{v}\|_2^2 \\ &= \left(\boldsymbol{U}^T\boldsymbol{U} + \lambda_2 \boldsymbol{I}\right)^{-1} \boldsymbol{U}^T \boldsymbol{v}_q . \end{aligned} \tag{10}$$

4) Algorithm

We present the whole RLSI algorithm for automated Chinese essay scoring in Fig. 2. Note that when updating **U**, we judge the convergence by the difference between two iterations (*deviation*) for each row m:

$$\sqrt{\frac{\sum_{k=1}^{K}\left(u_{mk} - u_{mk}^{*}\right)^2}{\sum_{k=1}^{K} u_{mk}^2}} = \frac{\|\boldsymbol{u}_m - \boldsymbol{u}_m^{*}\|_2}{\sqrt{\sum_{k=1}^{K} u_{mk}^2}} \leq 1 \times 10^{-6} \tag{11}$$

where u_{mk}^{*} is the update value whereas the u_{mk} is the old value. The number of iterations T is an empirical value that can be set by users.

```
Training:
Input:    $\boldsymbol{D} \in \mathbb{R}^{M \times N}$
Output:   $\boldsymbol{U} \in \mathbb{R}^{M \times K}, \boldsymbol{V} \in \mathbb{R}^{K \times N}$
$\mathbf{V} \leftarrow$ random matrix, $\mathbf{U} \leftarrow \mathbf{0}$;
repeat $T$ times:
    $\mathbf{S} \leftarrow \mathbf{V}\mathbf{V}^T$, $\mathbf{R} \leftarrow \mathbf{D}\mathbf{V}^T$;
    for $m$ = 1 to $M$:
        $\boldsymbol{u}_m \leftarrow \mathbf{0}$;
        while deviation > $1 \times 10^{-6}$
            for $k$ = 1 to $K$:
                $u_{mk} \leftarrow \frac{\left(\left|r_{mk} - \sum_{l \neq k} s_{kl} u_{ml}\right| - \frac{1}{2}\lambda_1\right)_{+} sign(r_{mk} - \sum_{l \neq k} s_{kl} u_{ml})}{s_{kk}}$
            end for
        end while
    end for
    $\mathbf{V} \leftarrow (\boldsymbol{U}^T\boldsymbol{U} + \lambda_2 \boldsymbol{I})^{-1}\boldsymbol{U}^T\boldsymbol{D}$;

Projection:
Given:  term-based essay vector $\boldsymbol{v}_q = [v_{q1}, v_{q2}, ..., v_{qM}]$;
Compute the projection as follows:
$\boldsymbol{v}_q^{*} = (\boldsymbol{U}^T\boldsymbol{U} + \lambda_2 \boldsymbol{I})^{-1}\boldsymbol{U}^T\boldsymbol{v}_q$.
```

Fig. 2. The RLSI learning algorithm for automated essay scoring.

C. Scoring model training

After the topic features have been extracted, the scores by human rater (*human scores*) are added to each essay in the training set as classification labels. In our method, the automated scoring is regarded as a multi-class classification problem, and thus, we can use both the human scores and the topic features extracted to train the scoring model.

Since Corinna Cortes and Vladimir Vapnik introduced Support Vector Machines (SVM) [19], it has been a powerful solution to such problems for its accuracy and effectiveness in a wide range, especially for document understanding [20-21]. Therefore, we choose SVM to train the scoring model, and predict the scores of the essays in the testing set. In this paper, we use RBF kernel to map the essay vectors into a higher dimension space:

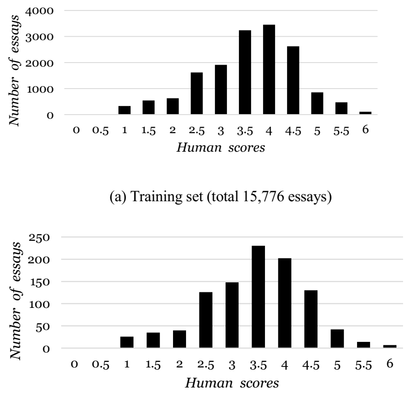
$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = exp\left(-\gamma \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right) \tag{12}$$

where γ is a parameter that needed to be tuned in our experiment, and $\boldsymbol{x}_i$ is the i-th topic-based essay vector.

III. Experiments

A. Experimental Settings

In the experiments, we use the written essays assigned by a same question and their human scores in MHK tests as the datasets. Omitting the empty essays and those without human scores, we obtain a collection with the size of 16,776. We randomly hold out 1,000 essays as the testing set. Fig. 3 shows the human score distribution of the training set and the testing set.



(a) Training set (total 15,776 essays)

(b) Testing set (total 1,000 essays)

Fig. 3. Human scores distribution of the dataset.

In text pre-processing, we generate a term list from the training set, including 12,974 terms. The RLSI learning program for topic feature extraction is written in C programming language. In scoring model training, we utilize LIBSVM [22] to classify the essays as automated essay scoring. All the programs run on a Linux machine with the CPU 2.9GHz Intel Xeon E5-2690 and 256G memory.

B. Performance Index

Automated essay scoring involves two kinds of performance indices between the human raters and the scoring system: *exact agreement* and *adjacent agreement* [2]. A more general method to measure a scoring system is to use adjacent agreement, which means that minor bias between the predicted score and the human score is acceptable. We call that *rating agreement*.

According to the scoring scale (0 to 6) and the scoring standard of MHK, we set 1 to the acceptable bias, because in reality, when the difference of the scores by two raters is larger than one point, the essay will be sent to the third rater for re-scoring. For different tests and scoring scales, this value can be set by the testers. We propose the rating agreement equation as follows:

$$R.A.=\frac{\sum_{n=1}^{N} 1\{|\,hs_i - ps_i\,| \le 1\}}{N} \qquad (13)$$

where *hs* and *ps* are the human score and the predicted score respectively, and *N* is the size of the testing set. The indicator function $1\{\cdot\}$ is used as follows:

$$1\{True\}=1 \;\; and \;\; 1\{False\}=0.$$

All the following experiments are executed in this way: we execute the automated scoring three times and apply the equation (13) to get the rating agreement respectively, using cross-validation to select the parameters C and γ in SVM scoring model to perform best. Then these three rating agreements are averaged as the final results.

C. Parameter Selection

Our target is to construct a scoring model that achieves the best performance, namely, the highest rating agreement. For different datasets, the parameters of models are supposed to be different as well, so they cannot be regarded uniformly. We select the parameters during the RLSI training and the construction of the scoring model respectively.

There are two main parameters in RLSI: the topic number K and the $\ell 1$-norm regularization parameter λ_1. The $\ell 2$-norm regularization parameter λ_2 is trivial since it is imposed on the matrix **V** which is less useful than the matrix **U**, so we set it to 1. Besides, the number of iterations T has little impact on the result, so we set it to 100.

In scoring model training, we use cross-validation to search for the best parameters, C and γ, for each experiment round, according to the SVM optimization equation and equation (12).

1) Topic number

As previously stated, for different applications, the topic number K varies as well. Generally, K is set to 50 to 100 empirically in the field of information retrieval, and the best K is decided by some evaluation formulae. For automated Chinese essay scoring, we determine the best K by rating agreement.

In the experiment, we manipulate the topic number from 25 to 200 at the interval of 25, and Fig. 4 is the experimental results. In Fig. 4, we determine that the rating agreement is best when the topic number is 175. This result validates our assumption that there exist several topics in an essay collection. Especially, the topic number is larger than the empirical value, for there exist more synonyms and ambiguities in Chinese than western languages.

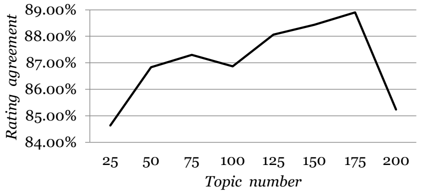


Fig. 4. The selection of topic number.

2) ℓ1-norm regularization parameter λ_1

In RLSI, λ_1 is imposed on the term-topic matrix **U**, which means that it controls the sparsity of the matrix **U**, according to the equation (5) where the hinge function is applied. Additionally, each essay vector is projected to the topic space by the equation (9), meaning that the matrix **U** controls the representative level of the topic space as well.

In our experiments, we set λ_1 from 0.1 to 1.0 at the interval of 0.1. The results are shown in Fig. 5.

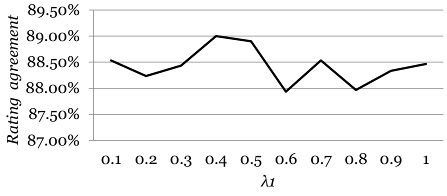


Fig. 5. The selection of λ_1

From Fig. 5, we can see that at the point where λ_1 is 0.4, the rating agreement reaches 89%. This value can be tuned according to different datasets.

D. Results and Discussions

1) Topics discovered by RLSI learning

Firstly, we randomly list five latent topics with their keywords discovered by RLSI learning from 175 topics in the training set, which are shown in Fig. 6.

topic 1	*topic 2*	*topic 3*	*topic 4*	*topic 5*
耐心 **patience**	成功 **success**	爸爸 **dad**	美 **beauty**	朋友 **friends**
老师 **teacher**	了解 **realize**	知足 **happy**	机会 **chance**	优点 **merit**
儿子 **son**	勇敢 **brave**	学会 **learn**	想法 **idea**	缺点 **weakness**
社会 **society**	知足 **happy**	生活 **life**	目标 **target**	选择 **choose**
想法 **idea**	面对 **face with**	信心 **faith**	诚实 **honest**	美 **beauty**

Fig. 6. Topic words discovered by the RLSI learning.

2) Automated scoring distribution

Secondly, we display the automated scoring distribution in Fig. 7, showing that the SVM-based scoring model recognizes the topic features in the written essay collection and predicts the scores successfully. In this result, we average the predicted scores from three round experiments. The concentration in the score 3.5, however, shows the limitation of conventional SVM, though the result is relatively satisfying.

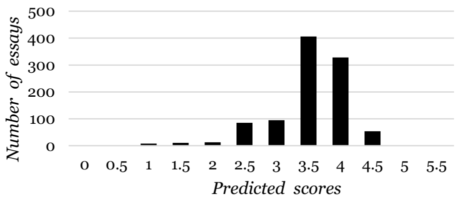


Fig. 7. The automated scoring distribution.

3) Comparison with LDA

Thirdly, in order to make it clearer that RLSI outperforms LDA in the field of automated Chinese essay scoring, we use the LDA tool designed by David M. Blei [1] to extract the topic features.

We initially set α = 1.0, then for each topic number from 25 to 200 at the interval of 25, we use cross-validation to tune the parameters C and γ in the SVM-based scoring model and get the rating agreements. Fig. 8 shows the comparison between LDA and RLSI. From Fig. 8, we can see that RLSI is more suitable than LDA in the field of automated Chinese essay scoring.

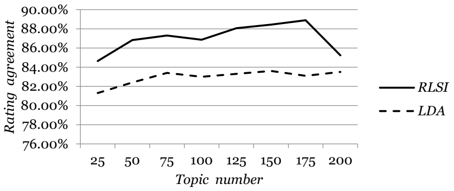


Fig. 8. The comparison between LDA and RLSI.

4) Error Analysis

For the result performed by RLSI, we discuss the possible reasons which lead to erroneous scoring. In the text pre-processing, the written essays are scanned into computer, and someone type the characters into text files manually, as shown in Fig. 1. Then, as previously introduced, we use WFST to segment and eliminate the stopwords. In each of above phases, the errors occur and accumulate, leading to the impreciseness of matrix samples before RLSI learning. Though more advanced technique has been implemented to reduce the errors, they cannot be avoided completely. In addition, Chinese is a language where much more synonyms and ambiguities exist than many western languages do, leading to more noises in feature extraction.

Under the current settings we used in this paper, automated Chinese essay scoring reaches the relatively high

[1] available at `http://cs.princeton.edu/~blei/lda-c/`

level. With the problems above solved, the rating agreement will get higher, and then scoring from topic perspective using RLSI can be applied effectively.

5) Further discussions

Finally, in [12], the author proposed two RLSI algorithms: batch version and online version. Batch version is used to general tasks, whereas online version can be used to handle missing data and to discover the evolution of the topics. In automated essay scoring, the online version is relatively less useful than the batch version. The algorithm we use in this paper is the batch version without parallel computation, since the dataset we use is not too big. Now that the topic modeling in automated essay scoring is proved applicable according to our experiments, this batch version can be applied in large-scale tests, especially for Chinese language tests such as MHK whose scale is increasing rapidly.

IV. Conclusions and Future Work

In this paper, we implement automated Chinese essay scoring from a topic perspective, using Regularized Latent Semantic Indexing, and achieve the performance of rating agreement of 89%, demonstrating that our application is quite effective.

Considering that this is the first attempt in this field, we propose the future work as follows. First, the huge advantage of RLSI is parallel computation or scalability. As the scale of MHK grows larger continually, RLSI can help to reduce the computational time. Second, topic perspective is proved viable, and then finding out a method that scores the essays from a comprehensive view (combining language proficiency, structure, topic and so forth) will enable the scoring system to predict more genuine scores and achieve at a higher rating agreement.

Acknowledgment

This work is supported by the Fundamental Research Funds for the Central Universities (No. XS2014023), the Beijing Higher Education Young Elite Teacher Project (No. YETP0768), the National Natural Science Foundation of China (No. 61103152) and the National 973 program in China (No. 2010CB328103). We thank the anonymous reviewers for their insightful comments.

References

[1] Mark D Shermis and Jill C Burstein, “Automated essay scoring: A cross-disciplinary perspective,” Psychology Press, 2003.

[2] Semire Dikli, “An Overview of Automated Scoring of Essays,” Journal of Technology, Learning, and Assessment, vol. 5, No.1, August 2006.

[3] Yanan Li, “Automated Essay Scoring for Testing Chinese as a Second Language,” Beijing Language and Culture University, Beijing, 2006.

[4] Xingyuan Peng, Dengfeng Ke, Zhi Zhao, Zhenbiao Chen, Bo Xu, “Automated Chinese Essay Scoring Based on Word Scores,” Journal of Chinese Information Processing, Vol. 26, No. 2, pp. 588-595, March 2012.

[5] Hengli Peng, “The minorities-oriented Chinese level test,” China Examinations, pp. 57-59, October 2005.

[6] Dengfeng Ke, Xingyuan Peng, Zhi Zhao, Zhenbiao Chen, and Jinshi Wang, “Word-level-based automated chinese essay scoring method,” In Proceedings of National Conference on Man-Machine Speech Communication, pp. 25-29, Xi’an China, 2011.

[7] Yiwei Cao and Chen Yang, “Automated chinese essay scoring with latent semantic analysis,” Examinations Research, Vol. 3, No. 1, pp. 63-71, 2007.

[8] Mingqing Zhang, Shudong Hao, Dengfeng Ke, Hengli Peng and Yanyan Xu, “Automated Essay Scoring Using Incremental Latent Semantic Analysis,” Journal of Software, accepted.

[9] Ge Xu and Houfeng Wang, “The Development of Topic Models in Natural Language Processing,” Chinese Journals of Computers, Vol. 34, No. 8, pp. 1423-1436, August 2011.

[10] David M. Blei, Andrew Y. Ng and Michael I. Jordan, “Latent Dirichlet Allocation,” Journal of Machine Learning Research, Vol. 3, pp. 993-1022, January 2003.

[11] Quan Wang, Jun Xu, Hang Li and Nick Craswell, “Regularized Latent Semantic Indexing,” In Proceedings of SIGIR’ 11, pp. 685-694, 2011.

[12] Quan Wang, Jun Xu, Hang Li and Nick Craswell, “Regularized latent semantic indexing: A new approach to large-scale topic modeling,” ACM Transactions on Information System, Vol. 31, No. 1, Article 5, 44 pages, January 2013.

[13] Shudong Hao, Zongtian Gao, Mingqing Zhang, Yanyan Xu，Hengli Peng, Kaile Su and Dengfeng Ke, “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, pp. 763-767, 2013.

[14] Preslav Nakov, Antonia Popova, and Plamen Mateev, “Weight Functions Impact on LSA Performance,” In Proceedings of EuroConference RANLP, pp. 187–193, 2001.

[15] Jerome Friedman, Trevor Hastie, Holger Höfling and Robert Tibshirani, “Pathwise Coordinate Optimization,” The Annals of Applied Statistics, Vol. 1, No. 2, pp. 302-332, 2007.

[16] Jerome Friedman, Trevor Hastie and Rob Tibshirani, “Regularization Paths for Generalized Linear Models via Coordinate Descent,” Journal of Statistical Software, Vol. 33, No. 1, pp. 1-22, August 2010.

[17] Tongtong Wu and Kenneth Lange, “Coordinate Descent Algorithms for Lasso Penalized Regression,” The Annals of Applied Statistics, Vol. 2, No. 1, pp. 224-244, 2008.

[18] P. Pucar and J. Sjöberg, “On the Hinge Finding Algorithm for Hinging Hyperplanes,” IEEE Transactions on Information Theory, Vol. 44, No. 3, pp. 1310-1319, May 1998.

[19] Corinna Cortes and Vladimir Vapnik, “Support-Vector Networks,” Machine learning, Vol. 20, No. 3, pp. 273-297, 1995.

[20] Min Li, Youngja Park, Rui Ma and He Yuan Huang, “Business Email Classification Using Incremental Subspace Learning,” In Proceedings of 21st International Conference on Pattern Recognition, pp. 625 - 628, 2012.

[21] Jayant Kumar, Peng Ye and David Doermann, “Learning Document Structure for Retrieval and Classification,” In Proceedings of 21st International Conference on Pattern Recognition, pp. 1558 - 1561, 2012.

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