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.

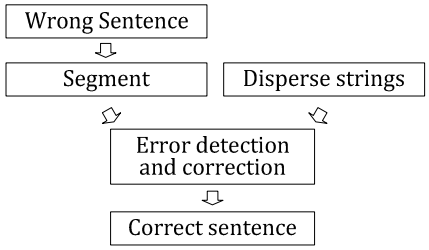


Fig. 1. A general method of Chinese text error detection and correction.

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.

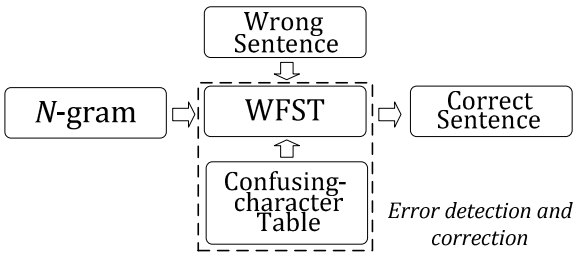


Fig. 2. The flowchart of the proposed method.

* 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:

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

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.

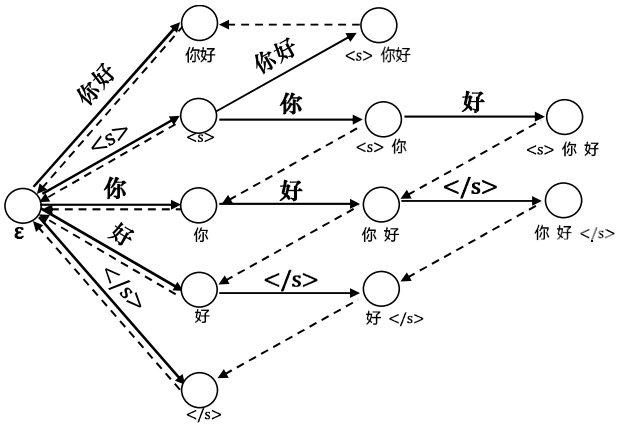


Fig. 4. A conversion of *N*-gram in Fig.3. The solid lines are forward arcs and dashed lines are backoff arcs. The characters on the arcs are *word*s and the *score*s are not shown in this figure.

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:

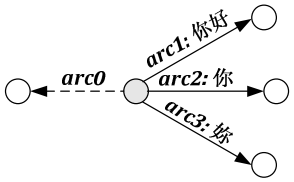
member = {*arc*, *state*, *score*, *scored-string*, *unscored-string*}

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:

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

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

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

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.

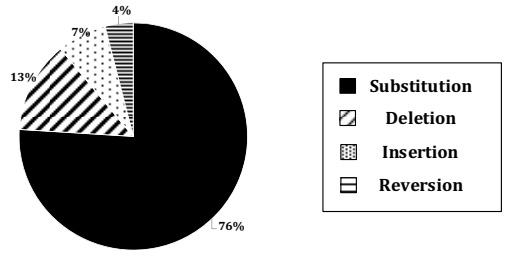


Fig. 8. Proportions of four kinds of errors in 200 MHK test essays.

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.

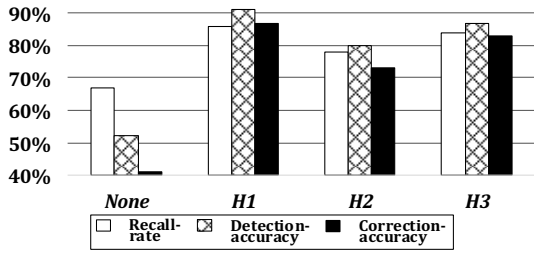


Fig. 9. Different computing methods of the heuristic value.

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.

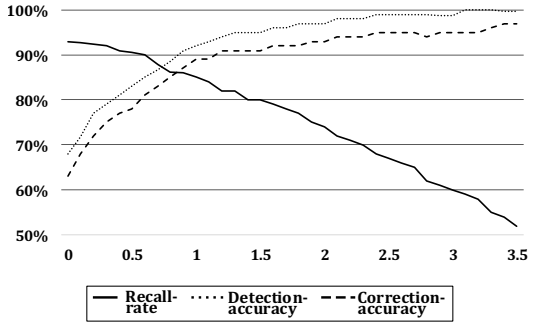


Fig. 10. Different punishment values.

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.

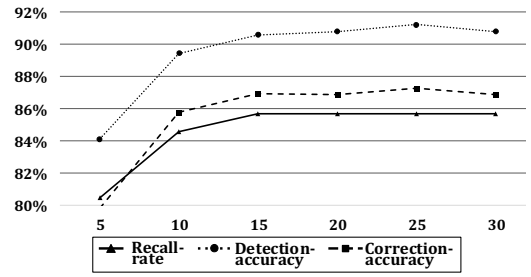


Fig. 11. Different beam widths.

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.

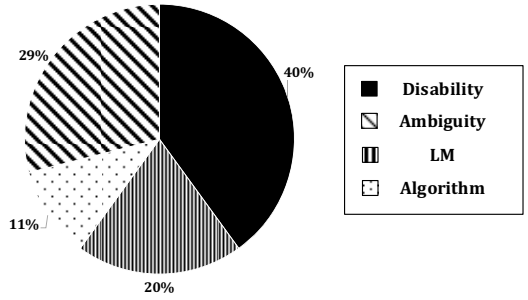


Fig. 12. Errors appeard in the result.

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.