Chinese Spelling Check System Based on Tri-gram Model

Qiang Huang, Peijie Huang*, Xinrui Zhang, Weijian Xie, Kaiduo Hong, Bingzhou Chen, Lei Huang
College of Informatics, South China Agricultural University,
Guangzhou 510642, Guangdong, China
kasim0079@qq.com, pjhuang@scau.edu.cn,
nealrichardrui@gmail.com, tsewkviko@gmail.com,
HKDNZ@hotmail.com, cbtpkzm@163.com, hl_mark@163.com

Abstract

This paper describes our system in the Chinese spelling check (CSC) task of CLP-SIGHAN Bake-Off 2014. CSC is still an open problem today. To the best of our knowledge, n-gram language modeling (LM) is widely used in CSC because of its simplicity and fair predictive power. Our work in this paper continues this general line of research by using a tri-gram LM to detect and correct possible spelling errors. In addition, we use dynamic programming to improve the efficiency of the algorithm, and additive smoothing to solve the data sparseness problem in training set. Empirical evaluation results demonstrate the utility of our CSC system.

1 Introduction

Spelling check is a common task in every written language, which is an automatic mechanism to detect and correct human errors (Wu et al., 2013). The problem of devising algorithms and techniques for automatically correcting words in text began as early as the 1960s on computer techniques for automatic spelling correction and automatic text recognition (Kukich, 1992), and it has continued up to the present. A spelling checker should have both capabilities consisting of error detection and error correction. Spelling error detection is to indicate the various types of spelling errors in the text. Spelling error correction is further to suggest the correct characters of detected errors.

Chinese as a foreign language (CFL) have attracted more and more attention, and this trend is continuing. For this purpose, at the SIGHAN Bake-offs, Chinese spelling check (CSC) task are organized to provide an evaluation platform for developing and implementing automatic Chinese spelling checkers. However, spelling check in Chinese is very different from that in English or other alphabetic languages. There are no word delimiters between words and the length of each word is very short. A Chinese "word" usually comprises two or more characters. The difficulty of Chinese processing is that many Chinese characters have similar shapes or similar (or same) pronunciations. Some characters are even similar in both shape and pronunciation (Wu et al., 2010; Liu et al., 2011).

There are many research effort developed for CSC recently, including rule-based model (Jiang et al., 2012; Chiu et al., 2013), n-gram model (Wu et al., 2010; Wang et al., 2013b; Chen et al., 2013), graph theory (Bao et al., 2011; Jia et al., 2013), statistical learning method (Han and Chang, 2013), etc. Some of them are hybrid model.

Language modeling (LM) is widely used in CSC, and the most widely-used and well-practiced language model, by far, is the n-gram LM (Jelinek, 1999), because of its simplicity and fair predictive power. Our work in this paper continues this general line of research by using a tri-gram LM to detect and correct possible spelling errors. In addition, in order to solve the high complexity in the computation process of the tri-gram based CSC, dynamic programming is used to improve the efficiency of the algorithm. Moreover, additive smoothing to solve the data sparseness problem in training set.

The rest of this paper is organized as follows. In Section 2, we briefly present the proposed

* Corresponding author

Proceedings of the Third CIPS-SIGHAN Joint Conference on Chinese Language Processing, pages 173–178,
Wuhan, China, 20-21 October 2014

CSC system, confusion sets and the choice of n-gram order. Section 3 details our Chinese tri-gram model. Evaluation results are presented in Section 4. Finally, the last section summarizes this paper and describes our future work.

2 The Proposed System

2.1 System Overview

Figure 1 shows the flowchart of our CSC system.

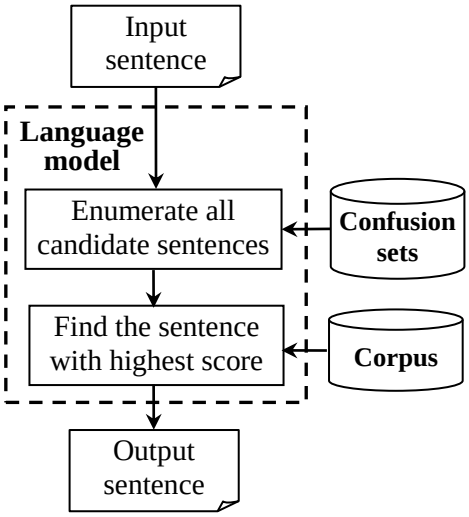

Figure 1. The flowchart of the CSC system.

The system is mainly composed by three components: confusion sets, corpus and language model. It performs CSC in the following steps:

1. Given a test sentence, the CSC system gets the confusion sets of each character in the sentence.

2. For each character in this sentence, the system will enumerate every character of its confusion set to replace the original character. We will get a candidate sentence set after this step.

3. The system will calculate the score of every candidate sentence by using the n-gram model. We use the corpus of CCL [1] and sogou [2] to generate the frequency of n-gram. Finally, the sentence with highest score will be chosen as the final output.

Due to the high complexity of step 2 and step 3, we optimize the algorithm by using dynamic programming.

2.2 Confusion Set

Confusion set is a ready set of commonly confused characters plays an important role in spelling error detection and correction in texts (Wang et al., 2013a). Most Chinese characters have other characters similar to them in either shape or pronunciation. Since pinyin input method is currently the most popular Chinese input method, the confusion sets used in our system is constructed from a homophone dictionary of *qingsongcha* website [3]. Some Chinese characters with similar pronunciation, such as the nasal and the lateral consonants, retroflex and non-retroflex, etc., are also added to the confusion sets in our system.

2.3 Language Modeling

Language modeling can be used to quantify the quality of a given word string, and most previous researches have adopted it as a method to predict which word might be a correct word to replace the possible erroneous word (Chen et al., 2009; Liu et al., 2011; Wu et al., 2010). The most widely-used and well-practiced language model, by far, is the n-gram language model (Jelinek, 1999), because of its simplicity and fair predictive power.

In n-gram modeling, choosing a proper order of the n-gram is important. On the one hand, higher order n-gram models along with larger corpora tend to increase their quality, and thus will yield lower perplexity for human-generated text. On the other hand, the higher order n-gram models, such as four-gram or five-gram, usually suffer from the data sparseness problem, which leads to some zero conditional probabilities (Chen et al., 2013). For these reasons, we have developed a Chinese character tri-gram model to determine the best character sequence as the answers for detection and correction.

3 Chinese Tri-gram Model

3.1 Tri-gram Model

Given a Chinese character string $C = c_1, c_2, ..., c_L$, the probability of the character string in tri-gram model is approximated by the product of a series of conditional probabilities as follows (Jelinek, 1999),

$$P(C) = \prod_{l=3}^{L} P(c_l \mid C^{l-1}) \approx \prod_{l=3}^{L} P(c_l \mid c_{l-2}, c_{l-1}) . \quad (1)$$

In the above tri-gram model, we make the approximation that the probability of a character depends only on the two immediately preceding

[1] ccl.pku.edu.cn:8080/ccl_corpus/index.jsp?dir=xiandai

[2] www.sogou.com/labs/dl/c.html

[3] www.qingsongcha.com/

words.

The easiest way to estimate the conditional probability in Eq. (1) is to use the maximum likelihood (ML) estimation as follows,

$$P(c_l \mid c_{l-2}, c_{l-1}) = \frac{N(c_{l-2}, c_{l-1}, c_l)}{N(c_{l-2}, c_{l-1})} \quad , \qquad (2)$$

where $N(c_{l-2}, c_{l-1}, c_l)$ and $N(c_{l-2}, c_{l-1})$ denote the number of times the character strings "c_{l-2}, c_{l-1}, c_l" and "c_{l-2}, c_{l-1}" occur in a given training corpus, respectively.

3.2 Getscore Function Definition

We define the candidate sentence as $C' = c'_1, c'_2, ..., c'_L$, which is the character string derived from the original sentence C by replacing some characters using their confusion sets. The *getscore* function is used to select the most suitable candidate sentence. Figure 2 shows the pseudo-code of the *getscore* function by using tri-gram model.

```
function getscore(c'_{i-2}, c'_{i-1}, c'_i)
begin
    ret ← N(c'_{i-2}, c'_{i-1}, c'_i) / N(c'_{i-2}, c'_{i-1})
    if c'_i = c_i then
    begin
        ret ← ret × λ
    end
end
```

Figure 2. Pseudo-code of *getscore* function.

Now we add a rule if $c'_i = c_i$, it will get an extra score λ. In the future work, we will add other rules or algorithms to improve the *getscore* function.

For example, in "一心一{億, 意}", in comparing with other string candidates as shown in Figure 3, we found the string of the highest score "一心一意". So we detect the error spot and select '意' as the corrected character.

3.3 Dynamic Programming

Due to the high complexity of enumerating candidate sentences, we use the dynamic programming (DP) to optimize the tri-gram model.

The confusion set of c_i is defined as $V[i]$, and each element in the confusion set is label by 0,1,2,3..., so the jth element in $V[i]$ will be represented as $V[i][j]$. The score of the candidate sentence with the maximum score is defined as $dp[i][j][k]$, where i is the length, $V[i-1][j]$ is the i-1th character, and $V[i][k]$ is the ith character. Because tri-gram model depends only on last three characters, we can deduce the state transition equation of the DP algorithm as follow:

$$strtmp = V[i-1][j], V[i][k], V[i+1][l] \quad , \qquad (3)$$

$$dp[i+1][k][l] = \max(dp[i+1][k][l], dp[i][j][k] * getscore(strtmp)). \qquad (4)$$

Pseudo-code of dynamic programming is shown in Figure 4. The complexity of the algorithm is reduced to acceptable level as $O(MN^3)$, where M is the length of the input sentence, and N is the size of a confusion set.

$$getscore("一心一") = \frac{N("一心一")}{N("一心")} \times \lambda = 0.00248$$

$$getscore("心一億") = \frac{N("心一億")}{N("心一")} \times \lambda = 0$$

$$getscore("一心一") = \frac{N("一心一")}{N("一心")} \times \lambda = 0.00248$$

$$getscore("心一意") = \frac{N("心一意")}{N("心一")} = 0.01574$$

Figure 3. *Getscore* function calculating example.

3.4 Additive Smoothing

In statistics, additive smoothing, which also called Laplace smoothing, or Lidstone smoothing, is a technique used to smooth categorical data. Given an observation $x = (x_1, x_2, ..., x_d)$ from a multinomial distribution with N trials and parameter vector $\theta = (\theta_1, \theta_2, ..., \theta_d)$, a "smoothed" version of the data gives the estimator:

$$\hat{\theta} = \frac{x_i + \alpha}{N + \alpha d} \qquad i = 1,2,..., d \ , \qquad (5)$$

where α > 0 is the smoothing parameter (α = 0 corresponds to no smoothing). Additive smoothing is a type of shrinkage estimator, as the resulting estimate will be between the empirical estimate x_i / N, and the uniform probability $1/d$. Using Laplace's rule of succession, some authors have argued that α should be 1 (in which case the term add-one smoothing is also used), although in practice a smaller value is typically chosen.

In a tri-gram model, the data consists of the number of occurrences of each string in corpus.

```
Procedure DP()
begin
  for i ← 3 to str.length do
    for j ← 0 to V[i -1].size do
      for k ← 0 to V[i].size do
        for l ← 0 to V[i +1].size do
        begin
            strtmp ← V[i −1][ j],V[i][k],V[i +1][l]
            dp[i +1][k][l] ← max(dp[i +1][k][l], dp[i][ j][k] * getscore(strtmp))
        end
end
```

Figure 4. Pseudo-code of dynamic programming.

Additive smoothing allows the assignment of non-zero probabilities to Chinese characters which do not occur in the training set. So we use additive smoothing to process the data sparse problem.

We redefine the new *getscore* function as Figure 5.

```
function getscore(c'i-2, c'i-1, c'i)
begin
    ret ← (N(ci-2, ci-1, ci) + α) / (N(ci-2, ci-1) + αd)
    if c'i = ci then
    begin
        ret ← ret × λ
    end
end
```

Figure 5. Pseudo-code of *getscore* function with additive smoothing.

4 Empirical Evaluation

4.1 Task

The goal of this shared task, i.e. the Chinese spelling check (CSC) task, in CLP-SIGHAN Bake-Off 2014 is developing the computer assisted tools to detect (combining error checking and correction) several kinds of grammatical errors, i.e., redundant word, missing word, word disorder, and word selection. The system should return the locations of the improper characters and must point out the correct characters. Passages of CFL (Chinese as a Foreign Language) learners' essays selected from the National Taiwan Normal University (NTNU) learner corpus are used for training purpose. Two training datas (one consisting of 461 spelling errors and another having 4823 spelling errors) are provided as practice. The final test data set for the evaluation consists of 1062 passages cover different complexities.

4.2 Metrics

The criteria for judging correctness are: (1) Detection level: binary classification of a given sentence, i.e., correct or incorrect should be completely identical with the gold standard. All error types will be regarded as incorrect. (2) Identification level: this level could be considered as a multi-class categorization problem. In addition to correct instances, all error types should be clearly identified.

In CSC task of CLP-SIGHAN Bake-Off 2014, ninth metrics are measured in both levels to score the performance of a CSC system. They are False Positive Rate (FPR), Detection Accuracy (DA), Detection Precision (DP), Detection Recall (DR), Detection F-score (DF), Correction Accuracy (CA), Correction Precision (CP), Correction Recall (CR) and Correction F-score (CF).

4.3 Evaluation Results

The CSC task of CLP-SIGHAN Bake-Off 2014 attracted 19 research teams. Among 19 registered research teams, 13 participants submitted their testing results. For formal testing, each participant can submit at most three runs that use different models or parameter settings. Finally, there are 34 runs submitted in total.

Table 1 shows the evaluation results of the final test. Run1, run2 and run3 are the three runs of our system with different λ in *getscore* function mentioned in Subsection 3.2. We have

	FPR	DA	DP	DR	DF	CA	CP	CR	CF
Run1	0.2034	0.4821	0.4518	0.1676	0.2445	0.4774	0.4375	0.1582	0.2324
Run2	0.6441	0.275	0.2315	0.194	0.2111	0.2627	0.2083	0.1695	0.1869
Run3	0.5009	0.3522	0.2907	0.2053	0.2406	0.3427	0.2712	0.1864	0.221
Average	0.2841	0.4633	0.4958	0.2106	0.2836	0.4485	0.4616	0.1811	0.2498
Best	0.032	0.7194	0.9146	0.484	0.633	0.7081	0.9108	0.4614	0.6125

Table 1. Evaluation results of final test.

chosen three runs with different estimated recall levels as submissions. The "Best" indicates the high score of each metric achieved in CSC task. The "Average" represents the average of the 34 runs.

As we can see from Table 1, we achieve a result close to the average level. The major weakness of our system is its low recall rate, which might be the result of not applying a separate error detection module.

It is our first attempt on Chinese spelling check. The potential of the n-gram method is far from fully exploited. Some typical errors of our current system will be presented in the next subsection, and the corresponding improvements are summarized in the last section.

4.4 Error Analysis

Figure 6 shows some typical error examples of our system ("O" original, "M" modified):

Case1:
O: 我 **戴**著 藍色 的 帽子
M: 我 **帶**著 藍色 的 帽子
Case 2:
O: 我們 在 健**缸** 中心 門口 等
M: 我們 在 健**缸** 中心 門口 等
Case 3:
O: 我們 **十**一點半 在 南門 碰頭
M:我們 **是** 一點半 在 南門 碰頭

Figure 6. Error examples.

The first case is an overkill error that belongs to long distance error correction problem. Our system didn't recognize the dependencies of "戴" and "帽子", and "我帶著" get a highest score in tri-gram model. So our system select "帶" to replace "戴", and leads to error at the same time.

In the second case, because "康" is not in the confusion set of "缸", we can't correct the error of "健缸" to "健康".

The third case is also an overkill error which is due to the out of vocabulary (OOV) problem. In this case, the original sentence is in fact correct but unfortunately, the our system didn't recognize "十一點半" and gave it high penalty.

5 Conclusions and Future Work

This paper presents the development and preliminary evaluation of the system from team of South China Agricultural University (SCAU) that participated in the Bake-Off 2014 task. We have developed a Chinese character tri-gram language model to determine the best character sequence as the answers for detection and correction. It is our first attempt on Chinese spelling check, and tentative experiment shows we achieve a not bad result. However, we still have a long way from the state-of-arts results.

There are many possible and promising research directions for the near future. A separate module for possible spelling error detection will be added to the system to improve the detection accuracy. In addition, although language modeling has been widely used in CSC, the n-gram language models only aim at capturing the local contextual information or the lexical regularity of a language. Future work will explore long-span semantic information for language modeling to further improve the CSC. Moreover, characters of similar shapes are not as frequent, but still exist with a significant proportion (Liu et al., 2011). Orthographically similar characters will be added to the confusion sets of our CSC system.

Acknowledgments
This work was partially supported by the Innovation Training Project for College Students of Guangdong Province under Grant No.1056413096 and No.201410564290.

References

Zhuowei Bao, Benny Kimelfeld, Yunyao Li. 2011. A Graph Approach to Spelling Correction in Domain-Centric Search. *In Proceedings of the 49th*

Annual Meeting of the Association for Computational Linguistics (ACL 2011), pp. 905–914.

Berlin Chen. 2009. Word Topic Models for Spoken Document Retrieval and Transcription. *ACM Transactions on Asian Language Information Processing*, Vol. 8, No. 1, pp. 2:1-2:27.

Hsun-wen Chiu, Jian-cheng Wu and Jason S. Chang. 2013. Chinese Spelling Checker Based on Statistical Machine Translation. *In Proceedings of the 7th SIGHAN Workshop on Chinese Language Processing (SIGHAN'13)*, Nagoya, Japan, 14 October, 2013, pp. 49-53.

Kuan-Yu Chen, Hung-Shin Lee, Chung-Han Lee , et al.. 2013. A Study of Language Modeling for Chinese Spelling Check. *In Proceedings of the 7th SIGHAN Workshop on Chinese Language Processing (SIGHAN'13)*, Nagoya, Japan, 14 October, 2013, pp. 79-83.

Dongxu Han, Baobao Chang. 2013. A Maximum Entropy Approach to Chinese Spelling Check. *In Proceedings of the 7th SIGHAN Workshop on Chinese Language Processing (SIGHAN'13)*, Nagoya, Japan, 14 October, 2013, pp. 74-78.

Frederick Jelinek. 1999. *Statistical Methods for Speech Recognition*. The MIT Press.

Zhongye Jia, Peilu Wang and Hai Zhao. 2013. Graph Model for Chinese Spell Checking. *In Proceedings of the 7th SIGHAN Workshop on Chinese Language Processing (SIGHAN'13)*, Nagoya, Japan, 14 October, 2013, pp. 88-92.

Ying Jiang, Tong Wang, Tao Lin, et al. 2012. A rule based Chinese spelling and grammar detection system utility. *In Proceedings of the 2012 International Conference on System Science and Engineering (ICSSE)*, pp. 437-440.

Karen Kukich. 1992. Techniques for Automatically Correcting Words in Text. *ACM Computing Surveys*, Vol. 24, No.4, pp. 377-439.

Chao-Lin Liu, Min-Hua Lai, Kan-Wen Tien, et al.. 2011. Visually and Phonologically Similar Characters in Incorrect Chinese Words: Analyses, Identification, and Applications. *ACM Transactions on Asian Language Information Processing*, Vol. 10, No. 2, pp. 1-39.

Yih-Ru Wang, Jason S. Chang, Jian-Cheng Wu, et al.. 2013a. Automatic Chinese Confusion Words Extraction Using Conditional Random Fields and the Web. *In Proceedings of the 7th SIGHAN Workshop on Chinese Language Processing (SIGHAN'13)*, Nagoya, Japan, 14 October, 2013, pp. 64-68.

Yih-Ru Wang, Yuan-Fu Liao, Yeh-Kuang Wu, et al.. 2013b. Conditional Random Field-based Parser and Language Model for Traditional Chinese Spelling Checker. *In Proceedings of the 7th SIGHAN Workshop on Chinese Language Processing (SIGHAN'13)*, Nagoya, Japan, 14 October, 2013, pp. 69-73.

Shih-Hung Wu, Yong-Zhi Chen, Ping-Che Yang, et al.. 2010. Reducing the False Alarm Rate of Chinese Character Error Detection and Correction. *In Proceeding of CIPSSIGHAN Joint Conference on Chinese Language Processing (CLP 2010)*, Beijing, 28-29 Aug., 2010, pp. 54-61.

Shih-Hung Wu, Chao-Lin Liu, and Lung-Hao Lee. 2013. Chinese Spelling Check Evaluation at SIGHAN Bake-off 2013. *In Proceedings of the 7th SIGHAN Workshop on Chinese Language Processing (SIGHAN'13)*, Nagoya, Japan, 14 October, 2013, pp. 35-42.