Chinese Spelling Check System Based on N-gram Model

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

Abstract

This paper presents our system in the Chinese spelling check (CSC) task of SIGHAN-8 Bake-Off. Given a sentence, our systems are designed to detect and correct the spelling error. As we know, CSC is still a hot topic today and it is an open problem yet. N-gram language modeling (LM) is widely used in CSC, since its simplicity and power. We present a model based on joint bi-gram and tri-gram LM and Chinese word segmentation. Besides, we apply dynamic programming to increase efficiency and employ smoothing technique to address the sparseness of the n-gram in training data. The evaluation results show 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 spelling errors (Wu et al., 2013). Automatic spelling correction began as early as the 1960s (Kukich, 1992). 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) is booming in recent decades. The number of (CFL) learners is expected to become larger for the years to come (Xiong et al., 2014). Automatic Chinese spelling check is becoming a significant task nowadays. For this task, Chinese spelling check (CSC) task are organized at the SIGHAN Bake-offs to provide a platform for comparing and developing automatic Chinese spelling checkers. However, different from English or other alphabetic languages, Chinese is a tonal syllabic and character language, in which each character is pronounced as a tonal syllable (Chen et al., 2013). In Chinese, there is no word delimiters or boundary between words and the length of each Chinese "word" is very short where there may only have two or three characters in most cases. Moreover, types of spelling error are more than other languages, since many Chinese characters resemble in shapes or pronounced the same. Some characters are even similar in both shapes and pronunciations (Wu et al., 2010; Liu et al., 2011).

So much research is under way up to now. For instance, rule-based model (Jiang et al., 2012; Chiu et al., 2013), n-gram model (Wu et al., 2010; Wang et al., 2013; Chen et al., 2013; Huang et al., 2014), graph theory (Bao et al., 2011; Jia et al., 2013; Xin et al., 2014), statistical learning method (Han and Chang, 2013; Xiong et al., 2014), etc, are proposed.

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. Continue to use N-gram LM, this paper proposed a model based on joint bi-gram and tri-gram LM to detect and correct spelling errors. And we try to exploit word segmentation in a pre-processing stage which improves the system performance to a certain extent. In addition, dynamic programming is applied to reduce the running time of our

* Corresponding author

Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing (SIGHAN-8), pages 128–136,
Beijing, China, July 30-31, 2015.

program and additive smoothing is used to solve the data sparseness problem in training set.

The rest of this paper is structured as follows. In Section 2, we briefly present our CSC system, confusion sets and the choice of n-gram order. Section 3 details our Chinese n-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. The system is mainly consists of four parts: Chinese Word Segmentation, Confusion sets, Corpus and Language Model. It performs CSC in the following steps:

Step 1. A given sentence was segmented by CSC system with Chinese words segmentation techniques. Result of Chinese words for segmentation will serve as the basis for the next step.

Step 2. According to the judgment conditions our system finds confusion sets of the corresponding word in the sentence.

Step 3. For each character in this sentence which can be replaced (in accordance with corresponding conditions), 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.

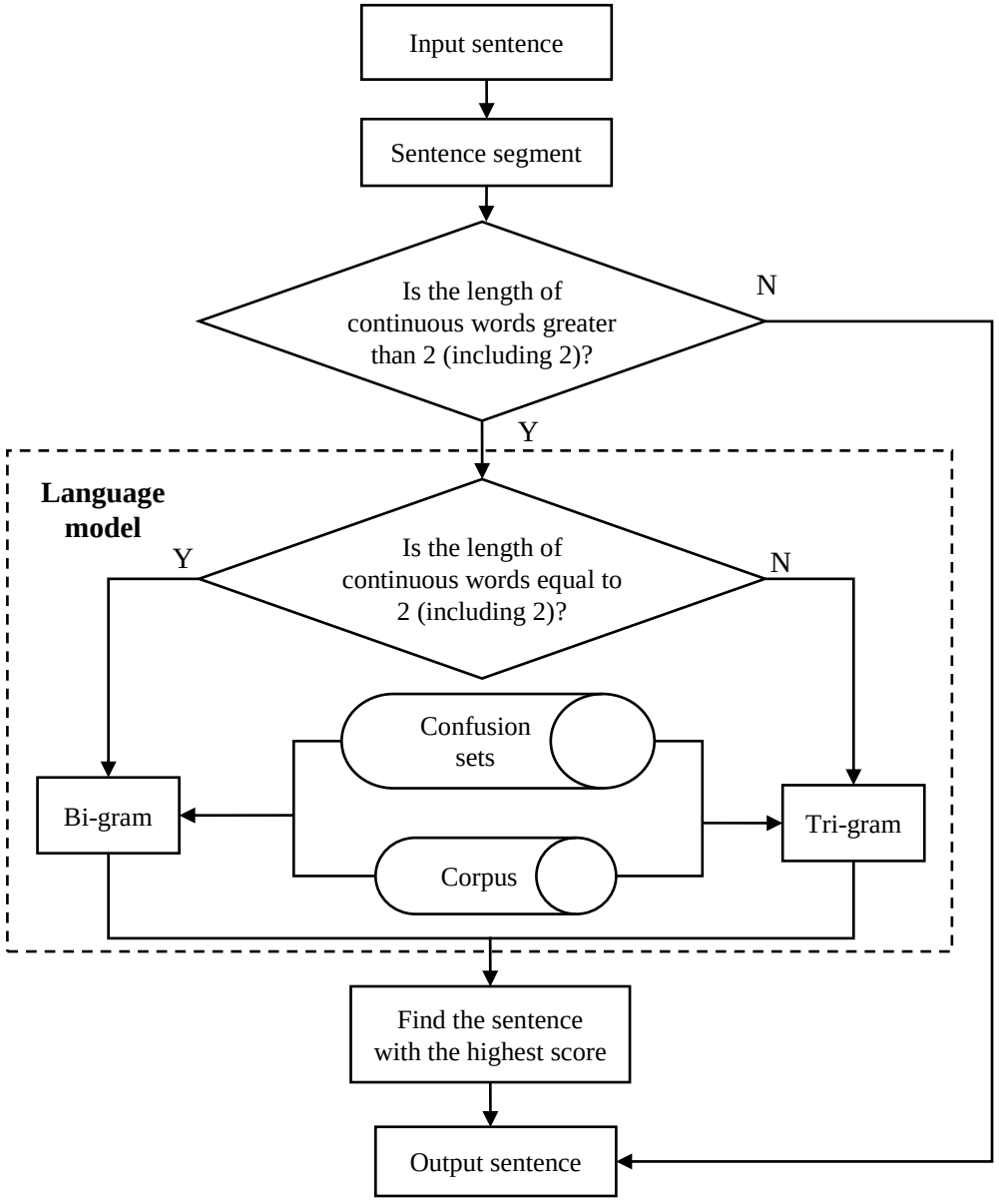

Figure 1. The flowchart of the CSC system.

Step 4. The system will calculate the score of every candidate sentence by using the joint bi-gram and tri-gram LM (using bi-gram and tri-gram based on different conditions). We use the corpus of CCL[1] and SOGOU[2] to generate the frequency of n-gram. Finally, the sentence with the highest score will be chosen as the final output.

In order to decrease the running time in Step 3 and Step 4, we apply dynamic programming to optimize the algorithm.

2.2 Confusion Set

Confusion set, a prepared set which consists of commonly confused characters plays a key role in spelling error detection and correction in texts (Wang et al., 2013). Most Chinese characters have similar characters on shape or pronunciation. Since pinyin input method is currently the most popular Chinese input method, when constructing the confusion sets used in our system, similar pronunciations is predominant. Moreover, characters of similar shapes are not as frequent, but still exist with a significant proportion (Liu et al., 2011). Orthographically similar characters have been also added to the confusion sets of our CSC system. So confusion sets used by the system were created by a number of rules with constraint, including similar pronunciations and similar glyphs.

Some Chinese characters with similar pronunciations, such as the Chinese homonyms ("zi(字)" and "zi(自)"), the nasal ("zang(藏)") and the non-nasal ("zan(赞)"), retroflex ("zhao(找)") and non-retroflex ("zao(早)"), etc.

In addition, it also includes other condition which is easy to confuse (based on statistics) on the pinyin of Chinese character, such as "qi(妻)"-"xi(西)" and "sao(嫂)"-"sou(搜)".

For Chinese characters with similar shape, such as the same radical of Chinese character ("固" and "回") and similar five-stroke input method ("ghnn(丐)" and "ghnv(丐)").

All of these rules are restricted by the strokes of a Chinese character to reduce the size of confusion sets of each character.

2.3 Language Modeling

Lots of previous researchers adopted language modeling to predict which word is correct to replace the possibly erroneous word in sentence, since language modeling can be used to measure the quality of a given word string (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.

Choosing an order of the n-gram in n-gram modeling is of a great importance. The higher order n-gram model such as four-gram or five-gram along with larger corpora tends to increase the quality thus will yield lower perplexity for human-generated text. However, the higher order n-gram models usually suffer from sparseness which leads to some zero conditional probabilities (Chen et al., 2013). For this reason, we use bi-gram and tri-gram with different rules for our system to determine which character is the best choice for correction. In our system, based on the result through Chinese Word Segment, we judge if it has any continuous words whose length is greater than or equal to 2. After that, if the length of unbroken words is equal to 2, we use bi-gram, and if it is greater than 2, we use tri-gram.

3 Chinese N-gram Model

3.1 Bi-gram Model

For given a Chinese character string $C = c_1, c_2, \ldots, c_L$, if the sentence has any errors, error words will appear in a continuous single words which will occur after through Chinese Words Segmentation. Generally speaking, the length of consecutive words is no more than 2 after splitting the sentence which has no mistakes. According to this judge, our system will adopt a bi-gram model to detecting and correcting errors when we find the length of continuous words is equal to 2.

For example, like this sentence "李大年的確是一個問提" will be "李大年/的確/是/一個/問/提" after through Chinese Character Segment. And the "題" is the correction of "提". If there are multiple places where the length of consecutive words is equal to 2, which means the sentence maybe has many spots with typo, then we use the bi-gram words in corresponding places. For example, the sentence "李大年的是的確是一個溫題" will be "李大年/的/是/的確/是/一個/溫/題" after through splitting, where the first "是" is a misspelled character of "事" and the "溫" is a misspelled character of "問".

[1] ccl.pku.edu.cn:8080/ccl_corpus/index.jsp?dir=xiandai
[2] www.sogou.com/labs/dl/c.html

The probability of the character string in the bi-gram model is approximated by the product of a series of conditional probabilities as follows (Jelinek, 1999),

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

In above Bi-gram model, we make the approximation that the probability of a character depends only on the one immediately preceding words.

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

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

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

In our system, bi-gram model used in this way: we utilize the two-tuples word with the maximum score as the correct string to override the old one.

3.2 Tri-gram Model

Based on the above idea of bi-gram, we think it is not suitable to express the sentence's probabilistic model if the length of continuous single words is over 2 after through Chinese splitting. Because there have been three or more consecutive words, we have reason to believe that the sentence appearing in typo may be continuous. So, in this case we use the tri-gram model to detect and correct errors.

Given a Chinese character string $C=c_1,c_2,\dots,c_L$, the probability of the character string in tri-gram model is similar to bi-gram model,

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

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

We estimate the conditional probability in Eq. (3) is used by the maximum likelihood (ML) estimation like bi-gram's method as follows,

$$P(c_l|c_{l-2},c_{l-1})=\frac{N(c_{l-2},c_{l-1},c_l)}{N(c_{l-2},c_{l-1})}, \quad (4)$$

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.

3.3 Getscore Function Definition

We define the candidate sentence as $C'=c'_1,c'_2,\dots,c'_L$, which is the character string derived from the original sentence C by replacing some characters using their confusion sets. The *getscore* function is utilized to select the most suitable candidate sentence. Figure 2 (a) and (b) show the pseudo-code of the *getscore* function by using bi-gram and tri-gram model, respectively.

```
function getscore1(c'_{i-1}, c'_i)
begin
        ret ← N(c'_{i-1},c'_i) / N(c'_{i-1})
        if c'_i = c_i then
        begin
                ret ← ret × λ
        end
end
```

(a) Bi-gram model

```
function getscore2(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
```

(b) Tri-gram model

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.

Figure 3 (a) and (b) show the calculating examples of *getscore* function by using bi-gram and tri-gram model, respectively.

For the example of "問{提,題}", in comparing with other string candidates as shown in Figure 3 (a), we found the string of the highest score "問題". So we detect the error spot and select '題' as the corrected character. Analogously, in "十字路{扣,口}", we detect the error spot and select '口' as the corrected character.

$$getscore(\text{"問提"}) = \frac{N(\text{"問提"})}{N(\text{"問"})} \times \lambda = 0.00022$$

$$getscore(\text{"問題"}) = \frac{N(\text{"問題"})}{N(\text{"問"})} = 0.61963$$

(a) Bi-gram model

$$getscore(\text{"十字路"}) = \frac{N(\text{"十字路"})}{N(\text{"十字"})} \times \lambda = 0.37973$$

$$getscore(\text{"字路扣"}) = \frac{N(\text{"字路扣"})}{N(\text{"字路"})} \times \lambda = 0$$

$$getscore(\text{"十字路"}) = \frac{N(\text{"十字路"})}{N(\text{"十字"})} \times \lambda = 0.37973$$

$$getscore(\text{"字路口"}) = \frac{N(\text{"字路口"})}{N(\text{"字路"})} = 0.91304$$

(b) Tri-gram model

Figure 3. *Getscore* function calculating example.

For the example of "問{提,題}", in comparing with other string candidates as shown in Figure 3 (a), we found the string of the highest score "問題". So we detect the error spot and select '題' as the corrected character. Analogously, in "十字路{扣,口}", we detect the error spot and select '口' as the corrected character.

3.4 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 $0, 1, 2, 3, \ldots$, so the j^{th} element in $V[i]$ will be represented as $V[i][j]$. The score of the candidate sentence with the maximum score is defined as $dp[i][k][l]$, where i is the length. $V[i-1][k]$ is the $i-1^{th}$ character, and $V[i][l]$ is the i^{th} character. Because tri-gram model depends only on the last three characters, we can deduce the state transition equation of the DP algorithm as follows:

$$TupleStr \leftarrow V[i-2][j], V[i-1][k], V[i][l], \quad (5)$$

$$dp[i][k][l] \leftarrow \max(dp[i][k][l], dp[i-1][j][k] * getscore(TupleStr)) \quad (6)$$

The pseudo-code of dynamic programming is shown in Figure 4. The time complexity of the algorithm is reduced to acceptable level as $O(\sum_i^n Len_i N^3)$, where n is the numbers of continuous single words ($C = c_1, c_2, \ldots, c_L$); Len_i, the length of each continuous single words is equivalent to L of c_L; and N is the maximum size of a confusion set.

```
function Trigram_DP(c : string)
begin
   for k ← 0 to V[0].size − 1 do
      for l ← 0 to V[1].size − 1 do
      begin
         if V[0][k] = c_0 then
            dp[1][k][l] ← INIT_Parameter
         else
            dp[1][k][l] ← 1.0
         if V[1][l] = c_1 then
            dp[1][k][l] ← dp[1][k][l] * INIT_Parameter
      end

   for i ← 2 to c.length − 1 do
      for j ← 0 to V[i − 2].size − 1 do
         for k ← 0 to V[i − 1].size − 1 do
            for l ← 0 to V[i].size − 1 do
            begin
               TupleStr ← string(V[i − 2][j], V[i − 1][k], V[i][l])
               dp[i][k][l] ← max(dp[i][k][l], dp[i − 1][j][k] * getscore(TupleStr))
            end
end
```

Figure 4. Pseudo-code of tri-gram dynamic programming.

3.5 Additive Smoothing

In statistics theory, additive smoothing or its alias called Laplace smoothing and Lidstone smoothing, is a technique which is used to smooth categorical data (Chen et al., 1996). For an observation sequence $x = (x_1, x_2, ..., x_d)$ from a multinomial distribution with N trials and parameter $\theta = (\theta_1, \theta_2, ..., \theta_d)$, a "smoothed" version of the data gives the estimator:

$$\hat{\theta} = \frac{x_i + \alpha}{N + \alpha d} \qquad i = 1,2,...,d, \qquad (7)$$

where $\alpha > 0$ is the smoothing parameter ($\alpha = 0$ 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 $1/d$.

In our model, the data make up for the number of occurrences of each string in corpus. Because of the sparsity of training data, which means some Chinese characters do not appear in the training data, we use additive smoothing to alleviate this sparsity problem.

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

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

(a) Bi-gram model

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

(b) Tri-gram model

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

4 Empirical Evaluation

4.1 Task

Chinese Spelling Check task is organized for the SIGHAN-8 bake-off. The goal of this task is to identify the capability of a Chinese spelling checker and hope to produce more advanced Chinese spelling check techniques. A passage, which is consist of several sentences with/without spelling errors i.e., redundant word, missing word, word disorder, and word selection, will be given as the input. Each character or punctuation occupies one position for counting location. The system to be developed should return the locations of the improper characters and the correct ones, if any typo is in this sentence, otherwise no spelling errors. Two training data (CLP-SIGHAN 2014 CSC Datasets[3] and SIGHAN-7 CSC Datasets[4]) are provided as practice. Passages of CFLs' essays selected from the NTNU learner corpus are also provided.

4.2 Metrics

The criteria for judging correctness are:

(1) Detection level: all locations of incorrect characters in a given passage should be completely identical with the gold standard.

(2) Correction level: all locations and corresponding corrections of incorrect characters should be completely identical with the gold standard.

The following metrics are evaluated in both levels with the help of the confusion matrix.

In CSC task of SIGHAN-8 Bake-Off, nine metrics method are used to evaluate the two aspects and 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

SIGHAN-8 Chinese Spelling Check task attracted 9 research teams to participate. 6 participants of 9 submitted their results. For formal testing, each participant has a right to submit at most three runs that use different models or parameter settings. There are 15 runs submitted in the end.

[3] http://ir.itc.ntnu.edu.tw/lre/clp14csc.html
[4] http://ir.itc.ntnu.edu.tw/lre/sighan7csc.html

Three runs of our system

Three runs of our system submitted to the SIGHAN-8 CSC final test are as follows:

Run1 (Tri-gram + word segmentation): This run replaces each word of a sentence with corresponding confusion sets in turn, and then computes new sentence score using tri-gram model. At the same time, we join the sentence segment to as the one of criterions of score calculation. In other words, we think that the less the total number of segments, the higher the score after sentence splitting, that is the numbers of segmentation is inverse proportion to score.

Run2 (Joint bi-gram and tri-gram + word segmentation): This run is the proposed method using joint bi-gram and tri-gram LMs and word segmentation.

Run3 (Tri-gram): This run is the result using the method of Run1 without the step of Chinese word segmentation. This run is the method that we proposed in the Bake-Off 2014 task last year (Huang et al., 2014). We use it as our baseline.

Validation of Run2

Table 1 indicates the top-3 validation scores of Run2, i.e. the proposed method on validation set that using CLP-SIGHAN 2014 CSC Datasets using different INIT_Parameter and λ that both are 30, 35 and 40 respectively. We utilize Test1's method and parameters as our SIGHAN-8 CSC final test Run2.

SIGHAN CSC15 final test

Table 2 shows the evaluation results of the final test. Run1, Run2 and Run3 are the three runs submitted by our system with different methods. The "Best" indicates the high score of each metric achieved in CSC task. The "Average" represents the average of the 15 runs.

According to the result in Table 2, we can see that the result of our system is close to the average level. The recall rate of our system is the major weakness. The reason might be that we do not apply a separate error detection module.

Although comparing with the baseline of tri-gram model, using joint bi-gram and tri-gram models gets improvement. The potential capability of the N-gram method is far from fully leveraged. Some typical errors of our current system will be presented in the next subsection, and some probably improvements are summarized in the Section 5.

4.4 Error Analysis

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

Case 1:
O: 生育嬰兒個數在**特**續下滑。
M: 生育嬰兒個數在**特**續下滑。
Case 2:
O: **或**著是人們有了新的想法。
M: **活**著是人們有了新的想法。
Case 3:
O: 一點**鐘**可不可以跟你見面?
M: 一點**中**可不可以跟你見面?

Figure 6. Error examples.

	FPR	DA	DP	DR	DF	CA	CP	CR	CF
Test1	0.2203	0.4680	0.4150	**0.1563**	**0.2271**	0.4576	0.3810	**0.1356**	**0.2000**
Test2	0.1996	**0.4755**	**0.4301**	0.1507	0.2232	0.4652	**0.3943**	0.1299	0.1955
Test3	**0.1940**	0.4746	0.4246	0.1431	0.2141	**0.4661**	0.3941	0.1262	0.1912

Table 1. Validation Scores of Run 2 on CLP-SIGHAN 2014 CSC Datasets.

	FPR	DA	DP	DR	DF	CA	CP	CR	CF
Run1	0.5327	0.3409	0.2871	0.2145	0.2456	0.3218	0.2487	0.1764	0.2064
Run2	**0.1218**	**0.5464**	**0.6378**	**0.2145**	**0.3211**	**0.5227**	**0.5786**	**0.1673**	**0.2595**
Run3	0.6218	0.3282	0.3091	0.2782	0.2928	0.3018	0.2661	0.2255	0.2441
Average	0.2254	0.5419	0.6148	0.3092	0.3978	0.5213	0.5795	0.268	0.3524
Best	0.0509	0.7009	0.8372	0.5345	0.6404	0.6918	0.8037	0.5145	0.6254

Table 2. Evaluation results of SIGHAN-8 CSC final test.

In the first case, because “持” is not in the confusion set of “特”, our system can't correct the error of “特續” to “持續”.

The second case is an overkill error that belongs to the context problem. Our system didn’t recognize the dependencies of “或著” and context, and “活著” get a highest score in the tri-gram model. So our system select “活” to replace “或”, and leads to error at the same time.

The third case is also an overkill error which is on account of the out of vocabulary (OOV) problem. In this case, the original sentence is in fact correct but unfortunately, our system modifies it to “一點中” and gave it a high score.

5 Conclusions and Future Work

This paper presents the development and evaluation of the system from team of South China Agricultural University (SCAU) that participated in the SIGHAN-8 Chinese Spelling Check task. The proposed joint bi-gram and tri-gram language model is helpful to determine the better character sequence as the results for detection and correction. Chinese word segmentation is performed on the input sentence. Dynamic programming is used to improve the efficiency of the algorithm to solve the high complexity in the computation process of the tri-gram. Additive smoothing is adopted to solve the data sparseness problem in the training set. In addition, we have optimized the Correction Precision by adding orthographically similar characters to the confusion sets.

It is our second attempt on Chinese spelling check, and the evaluation results of SIGHAN-8 CSC final test shows that comparing to the method we proposed in the CSC task of CLP-SIGHAN Bake-Off 2014 last year, we achieve an improvement of 9.7% in DF and 6.3% in CF. 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. Language modeling has been extensively used in our CSC. However, 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. What’s more, we still need to do more research on how to deal with the characters overkill problem to make the CSC more perfect.

Acknowledgments

This work was partially supported by National Natural Science Foundation of China under Grant No. 71472068, Science and Technology Planning Project of Guangdong Province, China under Grant No. 2013B020314013, and the Innovation Training Project for College Students of Guangdong Province under Grant No.201410564294.

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.

Stanley F. Chen, Joshua Goodman. 1996. An Empirical Study of Smoothing Techniques for Language Modeling. *In Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL 1996)*, pp. 310-318.

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.

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 Seventh 7th Workshop on Chinese Language Processing (SIGHAN-7)*, Nagoya, Japan, 14 Oct., 2013, pp. 79–83.

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-7)*, Nagoya, Japan, 14 Oct., 2013, pp. 49-53.

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-7)*, Nagoya, Japan, 14 Oct., 2013, pp. 74-78.

Qiang Huang, Peijie Huang, Xinrui Zhang, et al.. 2014. Chinese spelling check system based on tri-gram model. *In Proceedings of the Third CIPS-SIGHAN Joint Conference on Chinese Language Processing (CLP-2014)*, Wuhan, China, 20-21 Oct., 2014. pp.173-178.

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-7)*, Nagoya, Japan, 14 Oct., 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. 10: 1-10: 39.

Yih-Ru Wang, Yuan-Fu Liao, Yeh-Kuang Wu, et al.. 2013. 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-7)*, Nagoya, Japan, 14 Oct., 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 Proceedings of the First CIPS-SIGHAN Joint Conference on Chinese Language Processing (CLP-2010)*, Beijing, 28-29 Aug., 2010, pp. 54-61.

Yang Xin, Hai Zhao, Yuzhu Wang et al.. 2014. An Improved Graph Model for Chinese Spell Checking. *In Proceedings of the Third CIPS-SIGHAN Joint Conference on Chinese Language Processing (CLP-2014)*, Wuhan, China, 20-21 Oct., 2014. pp.157–166.

Jinhua Xiong, Qiao Zhao, Jianpeng Hou,et al.. 2014. Extended HMM and Ranking Models for Chinese Spelling Correction. *In Proceedings of the Third CIPS-SIGHAN Joint Conference on Chinese Language Processing (CLP 2014)*, Wuhan, China, 20-21 Oct., 2014. pp. 133–138.