# NCTU and NTUT's Entry to CLP-2014 Chinese Spelling Check Evaluation

**Yih-Ru Wang**
National Chiao Tung University
HsinChu, Taiwan
yrwang@mail.nctu.edu.tw

**Yuan-Fu Liao**
National Taipei University of Technology, Taipei, Taiwan
yfliao@ntut.edu.tw

## Abstract

This paper describes our Chinese spelling check system submitted to SIGHAN Bake-off 2014 evaluation. The system's main components are still the conditional random field (CRF)-based word segmentation/part-of-speech (POS) tagger and tri-gram language model (LM) used last year. But we tried to refine the misspelling rules, decision-making threshold and improve LM rescoring speed to reduce false alarm rate and improve rescoring speed. Bake-off 2014 evaluation results show that one of our system (Run2) did achieve reasonable performance with about 0.485/0.468 accuracies and 0.226/0.180 F1 scores in the detection/correction metrics.

## 1 Introduction

Chinese spelling check could be treated as an abnormal word sequence detection problem. Therefore, word segmentation, part-of-speech (POS) parser and language models (LM) are usually adopted to correct the sentence (Bengio 2003).

Therefore, a Chinese spelling checker (Wang 2013) had been built by integrating our conditional random field (CRF)-based parser and a 100K tri-gram LM. Although, these two components are originally designed for automatic speech recognizer (ASR), the system did get some success on Bake-off 2013 evaluation (Wu 2013). These results have confirmed the generalization and sophistication of our parser and LM.

However, there are still many issues in our system. Especially, our system often produces a large amount of false alarms and requires very long processing time on Bake-off 2013 evaluation. Therefore, the focus of this report is on how to reduce the false alarm rate, reduce search space and increase computing speed.

## 2 Summary of the proposed system

The proposed system is an open-set Chinese spelling check system, i.e., no any training data prepared by the Bake-off 2014 evaluation organizers were used in the system.

The block diagram of our system is shown in Fig. 1. There are three main components in the system including (1) a misspelling rules frontend, (2) a CRF-based Chinese parser and (3) a 100k trigram LM.

Basically, our approach is to exchange potential error characters with their confusable ones and rescore the modified sentence using our CRF-based parser and tri-gram LM to see if the modified one could get better word segmentation result and higher LM score or not. By this way, potential spelling error could be detected and corrected.

In this scheme, the input text is first checked and corrected if there are some high frequency misspelled words in the rule-based replacement frontend. The sentence is then segmented into a word sequence using our CRF-based parser and scored with a tri-gram LM. Then each character in short words (less than 3 characters) is considered as a potential error character and is replaced with character that has similar shape or pronunciation. The modified sentence is further re-segmented and re-scored to get a LM score. This process is repeated until the best modification (with maximum LM score) is found.

It could be found that a lot of re-segmentation and re-scoring computations are required by this approach. These steps, especially the LM rescoring, are very time-consumption. Therefore, the computation of LM score should be done as efficient as possible.

In the following subsections, the architecture and performance of the CRF-based parser and LM modules will be further summarized for better understanding our approach.

*Proceedings of the Third CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pages 216–219,
Wuhan, China, 20-21 October 2014

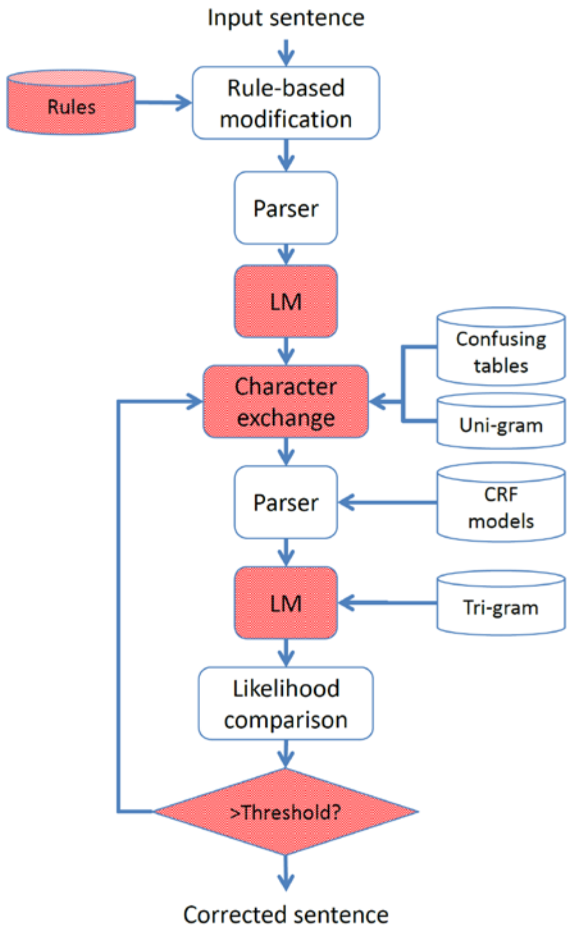

Fig. 1: The schematic diagram of the proposed Chinese spelling checker. Those shaded blocks had been improved for participating Bake-off 2014 evaluation.

### 2.1 CRF-based traditional Chinese parser

The block diagram of traditional Chinese parser is shown in Fig. 2. There are three blocks including (1) text normalization, (2) word segmentation and (3) POS tagging.

Both the word segmentation and POS tagging modules were based on CRF and trained using Sinica Balanced Corpus version 4.0[1]. The corpus had been manually checked and about 1% of inconsist word-segmentations were corrected. The word segmentation is basically implemented following Zhan's work (Zhao 2006), only the radix cues of the characters (in Chinese, "*bushu*") are add as new features (Wang 2013).

The F-measure of the word segmentation is 96.72% for the original database and 97.50% for the manually corrected corpus. The difference between precision and recall rates is less than 0.06%. About the parser, the accuracy of the 47-type POS tagging is 94.22%. According to these evaluation results, it is believed that our traditional Chinese parser is sophisticated enough.

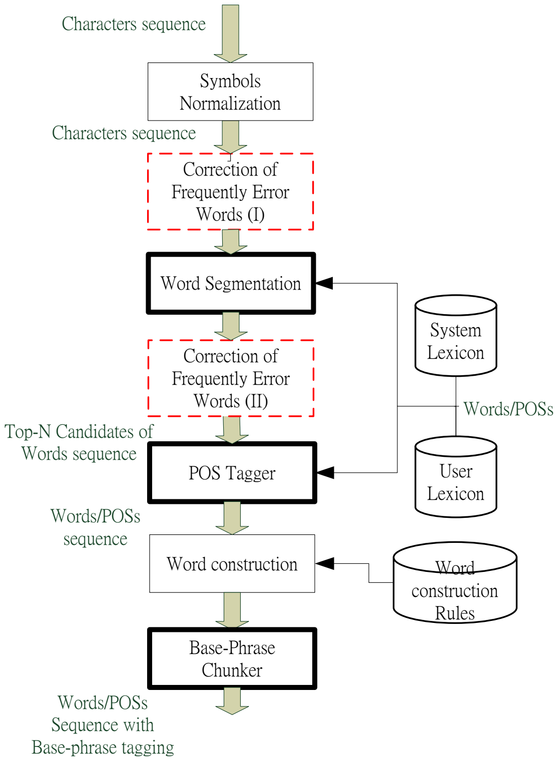

Fig. 2: The schematic diagram of the proposed Chinese parser.

### 2.2 LM construction

Four text corpora, the LDC Chinese Giga-byte[2], Sinica Balanced Corpus, CIRB030[3] (Chinese Information Retrieval Benchmark, version 3.03), the Taiwan Panorama Magazine[4] and context of Wikipedia (zh_tw version) were used to construct a 100k tri-gram LM.

There are in total 440 million words in the corpora. They were first parsed and post-processed (text normalization, word variation replacement, numbers into short-word conversion, etc.). Then, a 100k lexicon with most frequently words (without POS information) that have document frequency (DF) higher than a threshold was established. Finally, SRLIM toolkit (Stolcke 2000) version 1.7.0 was used to build a tri-gram LM for traditional Chinese.

This LM had been adopted to assist ASR and got significant improvement (Chen 2012), it is therefore a well-established LM.

[1] http://www.aclclp.org.tw/use_asbc_c.php
[2] https://catalog.ldc.upenn.edu/LDC2005T14
[3] http://www.aclclp.org.tw/use_cir.php
[4] http://www.aclclp.org.tw/use_gh_c.php (in Chinese)

## 3 System improvement

To speed up the rescoring computation and reduce the false alarm rate, several modifications had been done in this year's system. They are (1) misspelling rule expansion, (2) inline language model computation, (3) decision-making threshold and (4) potential error and exchange candidate selection. They are all shown as shaded blocks in Fig. 1.

### 3.1 Misspelling rule expansion

About 400 more (in total about 1000 now) high frequency error words were added into our misspelling rules. Those words are also collected from Internet. The new rules to replace error words are in general as follows (in Chinese):

腹漲 → 腹脹
行逕 → 行徑
幅射線 → 輻射線
檢查署 → 檢察署
排洩物 → 排泄物
可見一班 → 可見一斑
分道揚鏢 → 分道揚鑣
遺憾終身 → 遺憾終生

Fig. 3: Typical examples of misspelled Chinese word rules used in the frontend module.

### 3.2 Language model computation

The confusing tables used in the system includes many similar shape or pronunciation characters (Liu 2010). There are about 5400 characters in both the similar shape and pronunciation lists. Beside, each character has about 26 and 71 similar shape and pronunciation characters, respectively. The LM rescoring procedure is therefore very time-consuming. In fact, it is the major bottleneck of our system and often requires several days to finish the evaluation.

Two approaches had been tried to alleviate this problem. The first one is to change the format of LM file from an ASCII to a compressed binary one. The other one is to directly call SRILM's libraries instead of the executables in the rescoring program.

To call SRILM's library, three function calls (as shown in Fig. 2) were embedded into our main program to load LM, check word index/out-of-vocabulary (OOV) and compute LM score, respectively. By this way, the 100k tri-gram LM was loaded only once and therefore the LM rescoring time is significantly improved.

```
// srilm headers
#include “Ngram.h”

// srilm library -loolm -ldstruct -lmisc

// global variables
Vocab vocab;

Ngram*ngram;

//function calls
void srilm_init(const char* fname, int order) {
        File file(fname, "r", 0);
        assert(file);
        ngram = new Ngram(vocab, order);
        ngram->read(file, false);
        cerr << "Done\n";
      }

int srilm_getvoc(const char* word) {
        return vocab.getIndex((VocabString)word);
      }

float srilm_wordprob(int w, int* context) {
        return (float)ngram->wordProb(w, (VocabIndex*)context);
      }
```

Fig. 4: Application programming interface (APIs) for initialize SRILM, check word index/OOV and compute LM scores.

### 3.3 Decision-making threshold

In our scheme, each sentence is repeatedly modified, re-segmented and re-scored to find a word sequence with maximum LM score. However, the LM scores for different word segmentations in fact can't be compared fairly.

To alleviate this issue, a high score threshold was added into the decision-making logic. In other words, only those hypotheses that have significant LM score improvement were selected as candidates.

### 3.4 Error and exchange candidate selection

As mentioned in Section 3.2, for each potential error character there are many similar shape or pronunciation confusable ones. However, those tables may be over-completed.

To save some time, two heuristic rules that take advantage of a unigram model are applied. The first one is not to replace those high-frequency characters. The other one is to ignore those very low-frequency candidates. By this way, the search space is dramatically reduced. Bakeoff 2014 Evaluation Results

The goal of the checker is to return the locations of incorrect characters of an input sentence and suggest the correct characters. 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. There are in total 1,062 test sentences in the Bake-off 2014 evaluation.

## 4 Evaluation Results

Four configurations of our system (Run1~4) were tested. Run1 applied only the rule-based frontend. Run2~4 explored different search space and LM score threshold. The settings of the different runs are shown in Table 1. Among them, the search range of Run1~2 is very restricted and Run3~4 are much larger than others.

| Run | Error | Candidate | Log |
|---|---|---|---|
| 1 | - | - | - |
| 2 | 50~2000 | 100~4000 | 3.0 |
| 3 | 1~3000 | 1~5000 | 3.0 |
| 4 | 1~3000 | 1~5000 | 1.5 |

Table 1: Character frequency ranking range and LM score threshold settings for different Runs. Here “Error” and “Candidate” mean the character frequency ranking range to be considered as potential errors and as exchange candidates, respectively.

Table 2 show the all evaluation results. From Table 2, it can be found that Run1 and Run2 do have very low false alarm rate, but higher accuracy in both measures. The reason is that they only modified few errors with high confidence. On the other hand, Run3 and Run4 have higher recall rate and F1 scores but induce more false alarms. In summary, these results show our systems, especially Run1~2, are much conserved.

| | | **Detection Level** | | | | **Correction Level** | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Run** | **F/P Rate** | **Acc.** | **Pre.** | **Rec.** | **F1** | **Acc.** | **Pre.** | **Rec.** | **F1** |
| 1 | **0.038** | **0.513** | **0.630** | 0.064 | 0.116 | **0.509** | **0.600** | 0.057 | 0.103 |
| 2 | 0.181 | 0.485 | 0.455 | 0.150 | 0.226 | 0.468 | 0.392 | 0.117 | 0.180 |
| 3 | 0.281 | 0.461 | 0.420 | 0.203 | 0.274 | 0.435 | 0.349 | 0.151 | 0.211 |
| 4 | 0.642 | 0.313 | 0.294 | **0.267** | **0.280** | 0.276 | 0.232 | **0.194** | **0.211** |

Table 2: Evaluation results of the proposed system on Bake-off 2014 Chinese spelling check task. The table shows the false positive (F/P) rate, accuracy (Acc.), precision (Pre.), recall (Rec.), and F1 score for both the detection and correction levels.

## 5 Conclusions

In this paper, several modifications have been made to improve our Chinese spelling check system. Evaluation results show that our systems have achieved reasonable performance. Especially, Run2 gains about 0.485/0.468 accuracies and 0.226/0.180 F1 scores in the detection/correction levels.

Experimental results also show that a machine learning-based spelling error detector/classifier should be added on top of parser and LM to further improve system’s performance. Finally, our latest traditional Chinese parser is available online at http://parser.speech.cm.nctu.edu.tw.

## Acknowledgments

This work was supported by the Ministry of Science and Technology, Taiwan, under the projects with contract MOST 103-2221-E-027-079 and MOST 103-2221-E-009-125-MY2.

## References

Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin (2003), “A neural probabilistic language model, Journal of Machine Learning Research”, 2003, No. 3(2), pp. 1137–1155.

Sin-Horng Chen, Jyh-Her Yang, Chen-Yu Chiang, Ming-Chieh Liu and Yih-Ru Wang (2012), "A New Prosody-Assisted Mandarin ASR System", IEEE Trans. on Audio, Speech and Language Processing, vol.20, no.6, pp.1669,1684, Aug. 2012.

Chao-Lin Liu, Min-Hua Lai, Kan-Wen Tien, Yi-Hsuan Chuang, Shih-Hung Wu, and Chia-Ying Lee (2011). Visually and phonologically similar characters in incorrect Chinese words: Analyses, identification, and applications, ACM Trans. Asian Lang. Inform. Process. 10, 2, Article 10 (June 2011).

A. Stolcke (2002), SRILM -- An Extensible Language Modeling Toolkit. Proc. Intl. Conf. on Spoken Language Processing, vol. 2, pp. 901-904, Denver.

Yih-Ru Wang, Yuan-Fu Liao, Yeh-Kuang Wu and Liang-Chun Chang (2013). Traditional Chinese Parser and Language Model-Based Chinese Spelling Checker. Proceedings of the 7th SIGHAN Workshop on Chinese Language Processing (SIGHAN'13), Nagoya, Japan, 14 October, 2013, pp. 69-73.

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

H. Zhao, C. N. Huang and M. Li (2006), “An Improved Chinese Word Segmentation System with Conditional Random Field”, the Fifth SIGHAN Workshop on Chinese Language Processing 2006, pp. 108-117.