# Chinese Spell Checking Based on Noisy Channel Model
# 雜訊通道模型爲本的中文拼字改錯系統

Hsun-Wen Chiu 邱絢紋 (101065506)
(chiuhsunwen@gmail.com)

Advisor: Jason S. Chang 張俊盛
(jason.jschang@gmail.com)

Natural Language Processing Lab

Institute of Information Systems and Applications

National Tsing Hua University

Hsinchu, Taiwan 30013.

July, 2014

# 摘要

中文自動更正拼字或打字錯誤在文書處理、網路搜尋及自動作文評分都是很重要的議題。然而，中文改錯不同於一般拼音語言的拼寫改錯，中文沒有詞間的分隔符號，而且不同的中文輸入法可能會產生不同的錯字類型，所以使得中文改錯更加困難。本篇論文針對音似形似的錯誤提出了一個利用雜訊通道模型（Noisy Channel Model）改錯，首先利用網路語料庫產生混淆字集（Confusion Set）和對應的機率生成通道模型（Channel Model），接著透過雜訊通道模型中的通道模型和語言模型（Language Model）改錯。本系統的組成包含訓練階段和執行階段，在訓練階段我們利用網路語料中 n 連詞（ngrams）的頻率估計每一個字對應混淆字的機率，在執行階段，系統會根據輸入的句子產生多個候選字，最後利用通道模型和語言模型選出最合適的字。實驗結果顯示，本論文提出的方法所製作的雛形系統，有不錯的改錯精確率與召回率。

**關鍵詞:** 雜訊通道模型、語言模型、網路語料、混淆字集。

# Abstract

Chinese spell checking is an important component of many Chinese NLP applications, including word processors, search engines, and automatic essay rating. Compared to English, Chinese has no word boundaries, and there are various Chinese input methods that cause different kinds of typos. Therefore, it is more difficult to develop a spell checker for Chinese. In this paper, we introduce a novel method for correcting Chinese errors based on sound or shape similarity. In our approach, potential typos in a given sentence are then corrected using a channel model and a character-based language model in the noisy channel model. In the training phase, we estimate the channel probabilities for each character based on ngrams in Web corpus. At run-time, the system generates correction candidates for each character in the given sentence and selects the appropriate correction using the channel model and the language model. The experimental results show that the proposed method achieves significantly better accuracy and recall than more complicated methods in the previous work.

**Keywords:** A Noisy Channel Model, Character-based Language Model, Web Corpus, Confusion Set.

# Acknowledgments

The author thanks Professor Jason S. Chang and Dr. Jian-cheng Wu for their guidance and inputs. This work was derived from a system participating in the Bake-off 2013: Chinese Spelling Check in Seventh SIGHAN Workshop on Chinese Language Processing (SIGHAN-7).

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Spell checking is a necessary task for text processing of every written language, which involves automatically detecting and correcting typographical errors. However, compared to spell checkers for alphabetical languages (e.g., English or French), Chinese spell checkers are more difficult to develop because there are no word boundaries in Chinese writing system and errors may be caused by various Chinese input methods. In this thesis, we define typos as Chinese characters that are misused due to sound or shape similarity. Liu et al. (2011) show that people tend to unintentionally generate typos due to sound similarity (e.g., *索定 (*suo ding*) instead of 鎖定 (*suo ding*)) or shape similarity (e.g., *銷定 (*xiao ding*) instead of 鎖定 (*suo ding*)). On the other hand, some typos found on the Web (e.g., forums or blogs) are used deliberately for the purpose of speedy typing or just for fun. Therefore, spell checking is an important component for many applications, including computer-aided writing, search engines, and social media text normalization.

Relatively little work has been done on the task of Chinese spell checking. The methods proposed in the literature can be classified into two types: rule-based methods and statistical methods. Rule-based methods use knowledge resources, for example, dictio-

Table 1.1: Example trigrams with corresponding frequency and probability.

| Trigrams | Frequency | LM probability(log) |
|---|---|---|
| 所定的 (*suo ding de*) | 5 | -0.70 |
| 鎖定的 (*suo ding de*) | 2 | -1.49 |

naries, confusion sets, and segmentation systems. Simple rule-based methods, however, have their limitations. The following sentence is a snippet collected from students' written essays which is correct .

> 爲什麼你要如此地用功呢？如果我不用功，那以後我將趕不上自己所定的目標。(*wei she me ni yao ru ci di yong gong ne* ？*ru guo wo bu yong gong* ，*na yi hou wo jiang gan bu shang zi ji suo ding de mu biao* 。)

Unfortunately, based on simple rules the two characters 所 (*suo*) and 定 (*ding*) are likely to be regarded as typos of the dictionary word 鎖定 (*suo ding*) with identical pronunciation.

The data-driven, statistical spell checking approach appears to be more robust and perform better. Statistical methods typically use a large corpus to create a language model to validate the correction hypotheses. Intuitively, by using 自己所定的目標 (*zi ji suo ding de mu biao*), the three characters 所定的 (*suo ding de*) are a trigram with high probability in a monolingual corpus, we therefore may determine the 所定 (*suo ding*) is not a typo after all. Table 1.1 shows the frequency and probability of 所定的 (*suo ding de*) and 鎖定的 (*suo ding de*).

In this thesis, we propose a model using statistical approaches and model generates the most appropriate corrections in a given sentence. In the training phase, we automatically generate the channel model (confusion set). We use a Chinese spell checker to correct

Table 1.2: The three correction candidates of the given sentence.

| Hypotheses |
|---|
| 爲什麼你要如此地用功呢？如果我不用功，<br>那以後我將趕不上自己**所**定的目標。 |
| 爲什麼你要如此地用功呢？如果我不用功，<br>那以後我將趕不上自己**瑣**定的目標。 |
| 爲什麼你要如此地用功呢？如果我不用功，<br>那以後我將趕不上自己**鎖**定的目標。 |

instances in the training data and estimate the channel probability of a typo condition on a correct character , then re-estimate the probability, and iterate until convergence.

At run-time, the checker corrects typos using a noisy channel model. Consider the following sentence.

> 爲什麼你要如此地用功呢？如果我不用功，那以後我將趕不上自己**鎖**定的目標。(*wei she me ni yao ru ci di yong gong ne* ？*ru guo wo bu yong gong* ， *na yi hou wo jiang gan bu shang zi ji suo ding de mu biao* 。)

The checker generates correction candidates by the replacements of each character and confusable characters with channel probabilities in a beam search algorithm, then calculates the probability of correction hypotheses according to the language model and the channel model. Three correction candidates are shown in Table 1.2. Finally, the checker returns the correction with the highest score, e.g., the follow sentence:

> 爲什麼你要如此地用功呢？如果我不用功，那以後我將趕不上自己**所**定的目標。(*wei she me ni yao ru ci di yong gong ne* ？*ru guo wo bu yong gong* ， *na yi hou wo jiang gan bu shang zi ji suo ding de mu biao* 。)

The rest of the paper is organized as follows. We present the related work in the next section 2. Then, we describe the proposed model for automatically correcting the spelling typos in section 3. Section 4 presents the experimental data, results, and performance analysis. We conclude in Section 5.

# Chapter 2

# Related Work

Chinese spell checking is a task involving automatically detecting and correcting typos in a given Chinese sentence. Previous work typically takes the approach of combining rule-based and statistical approaches. A rule-based approach depends on dictionary knowledge and a confusion set, i.e., a collection set of certain characters consisting of visually and phonologically similar characters. On the other hand, statistical-based methods usually use a channel model and a language model, which is generated from a reference corpus. A statistical language model assigns a probability to a sentence of words by means of ngram probability to compute the likelihood of a sentence, or it correction candidates (derived from the channel model).

Chang (1995) proposed a system that replaces each character in the sentence based on the confusion set and estimates the probability of all modified sentences according to a bigram language model built from a newspaper corpus, and produce a corrected sentence with higher probability than the given sentence. They used a confusion set consisting of pairs of characters with similar shape that were collected by comparing the original text and its OCR results. Similarly, Zhuang et al. (2004) proposed an effective approach using OCR to develop a shape-based confusion set. In addition, Zhuang et al. (2004) also used a language model based on multiple resources, and Latent Semantic Analysis. Their experiments seem to show that a simple ngram model performs the best. In contrast, we automatically generate the confusion set with probability from the web corpus and correct the given sentence with character-based ngram model.

In recent years, Chinese spell checkers have incorporated word segmentation. For example, Huang et al. (2007) proposed a method that use the Sinica Word Segmentation System (Ma and Chen, 2003) as a preprocessing step for detecting typos. With a character-based bigram language model and the rule-based methods of using dictionary and confusion sets, the method determines whether the word is a typo or not using a set of rules based on word segmentation results. There are many more systems that use word segmentation to detect typos. In Hung and Wu (2009), the given sentence is segmented using a bigram language model before typo detection. In addition, the method also uses a confusion set and handcrafted common patterns of typos based on data provided by the Ministry of Education (*MOE*) in Taiwan. Chen and Wu (2010) modified the system proposed by Hung and Wu (2009) by combining statistic-based methods and a pattern-matching module generated automatically to detect and correct typos based on a language model.

More recently, Wu et al. (2010) adopted the noise channel model, a framework used in many NLP tasks including spell checkers and machine translation systems. The system combined a statistical method and template matching with the help of a dictionary and a confusion set. They also used word segmentation to detect typos, but they did not use existing word segmentation, as Huang et al. (2007) did, because that might identify a typo as a new word. Instead, a backward longest first approach is used to segment sentences using an online dictionary provided by MOE, and a set of error templates with a confusion set. The system then corrects potential typos as a kind of translation by using ngram language model and generating correction candidates from confusion sets. Contrary to their approach, we directly corrects typos using a channel model with character-based language model.

In SIGHAN-7 Bake-offs Chinese Spelling Check, Yeh et al. (2013) proposed a learning-based method to estimate probability of confusable characters (sound similarity and shape similarity). The similar characters they used is consisting of the confusing matrix constructed by the Hidden Markov Model Toolkit (HTK, used for speech recognition) and length based Cangjie code similarity measure. Then, they used Sinica Word Segmentation System (Ma and Chen, 2003), E-HowNet, a typo-correction list, and word-based language model to correct typos. Jia et al. (2013) also used word segmentation to detect typos, but they modified shortest path word segmenter and transformed to a single source shortest path problem on directed acyclic graph. For correction model, they used language model with single source shortest path algorithm built on Sinica corpus and used mutual information to choose the best correction.

In contrast to work on Chinese spelling checking in the literature, we introduce a novel method for correcting typos, focusing on sound and shape similarity. We automatically augment the confusion set with probability using Web corpus, and correct typos using the channel model and the character-based language model. In the training phase, we estimate the channel model using the Expectation-maximization algorithm based on the confusion set and the character ngram in the web corpus. Our system generate the correction candidates using the channel model, and find the best correction using the language model. The experiment results show that our method achieves significantly better performance in a simple, systematic approach.

# Chapter 3

# Method

Using fixed rule to correct typos in a given Chinese sentence (e.g., 自己鎖定的目標 *(zi ji suo ding de mu biao)*) does not work very well. Methods in previous work typically correct typos based on a set of detection rules. Unfortunately, the detection rules depend on a lot of resources, and can be at times unreliable. Typo positions usually are detected using heuristic rules based on Chinese dictionary, word segmentation, and the frequency of the ngram. However, dictionaries, and word segmentation have their limitations. For example, the segmentation result of the sentence "自己鎖定的目標" is "自己/鎖定/的/目標", the two characters 鎖 and 定 may or may not be considered as a word or a typo, depend on the segmentation system. To avoid the limitations of the rule-based method, a promising approach for Chinese spell checking is to train a noisy channel model based on unannotated data.

Figure 3.1: Outline of the training and run-time phase.

1. Training phase (Section 3.2)
   (1) Limit Confusable Characters (Section 3.2.1)
   (2) Retrieve Ngrams (Section 3.2.2)
   (3) Correct Ngrams and Train Channel Model (Section 3.2.3)
   (4) Output the Channel Model
2. Run-time phase (Section 3.3)

## 3.1 Problem Statement

We focus on the task of correcting typos in a given sentence. Therefore, the purpose is to find the typos in a given sentence which has the highest score of the probability combining the channel model and the language model. In the noisy channel model. To train the channel model, we use a set of predetermined confusable characters, iteratively correct the confusable ngrams in the Web corpus using the existing checker, and estimate the probability of a typo conditioned on a confusable character. Then, the confusable characters in the sentence are replaced by a member in the confusion set as correction candidates, the system calculates the score for each correction candidate using the channel model and the language model, and returns the correction with highest score as the output of the system. Our goal is to find corrections of the typos accurately. The correction results can be used in a on-line Chinese essay writing system, or in a word processor. We now formally state the problem that we are addressing.

***Problem Statement***: We are given a sentence $S$ with $n$ characters $s_1$, $s_2$, ..., $s_\mathrm{n}$, the character-based language model *LM*: $P(c_1, c_2, ..., c_\mathrm{n})$, and the channel model *CM*: $P(c_\mathrm{i}, s_\mathrm{i})$. Our goal is to find $c_1$, $c_2$, ..., $c_\mathrm{n}$ to correct $S$. For this, we train *LM* using the Chinese corpus and generate *CM* using the ngrams (with typos) $N$ using confusable characters *CS*, and a existing Chinese spell checker *CSC*.

Table 3.1: The full confusion set and the limited confusion set of 鎖.

| Type | Sound | Shape |
|---|---|---|
| Full | 所瑣索梭娑嗦縮莎 | 瑣銷鋇鏜鐺鑽貝貧 |
| | 唆蓑簑數碩勺鑠蟀 | 賞員賄煩鈔貼敗財 |
| | 説朔爍帥率妁鎗鎔 | 狽盼賸賤賊損貽則 |
| | 鎰鎳鎢鎬鎮鎊鎚鎘 | 貞負頁賽贊圓 |
| Limited | 索瑣鎖所 | 賸鏜鎖 |

In the rest of this section, we describe our solution to the problem of Chinese spell checking (see Figure 3.1). We describe the process of training the channel model in Section 3.2. More specifically, we describe the method for limiting confusable characters in Section 3.2.1, and the use of ngrams in Section 3.2.2. We will also describe an Expectation-Maximization (*EM*) algorithms for estimating channel probabilities in Section 3.2.3. This algorithm relies on a set of confusable characters and ngrams. Finally in Section 3.3, we describe how to correct typos using the trained noisy channel model at run-time by combining channel model and language model.

## 3.2 Training Channel Model

We attempt to learn to develop a channel model from the ngrams of Web corpus for correcting Chinese spell typos. The training process is shown in Figure 3.1.

### 3.2.1 Limiting Confusable Characters

In the first stage of training the channel model (Step (1) in Figure 3.1), we limit the confusable characters in the full confusion set based on the sound and shape similar characters, which containing unlikely confusable characters. For example, the full confusion set of 鎖 *(suo)* is shown in Table 3.1. The goal of this method is to reduce the sizes of the confusion sets and improve the accuracy.

The input to this stage is a set of confusable characters. We generate potential confusable characters by reducing some unlikely confusable characters, and expanding the confusable characters slightly.

The output of this stage is confusion sets that can be used to correct ngrams (retrieving and correcting steps in Figure 3.1) for training channel model. Limited confusion set of 鎖 *(suo)*, automatically generated from the full confusion set is shown in Table 3.1. We can see that the limited confusion set minimizes the confusable characters and retains more likely characters. The limited confusion set is used to accurately find typos in ngrams and reduce the computational complexity.

Our method for limiting confusable characters can generate many characters, potentially including a significant number of characters that are not useful in correcting typos. We also remove some loosely similarly relations and expand the confusable characters slightly. For example, we remove all relations based on non-identical phonologically similarity. After that, we add the similarly sounding characters based on nasal consonant in Chinese phonetics (e.g., "ㄣ，ㄥ" *(en, eng)* and "ㄢ，ㄤ" *(an, ang)*), and retroflex consonant (e.g., "ㄙ，ㄕ" *(shi, si)* and "ㄔ，ㄘ" *(chi, chi)*). We also modify the shape similarity by comparing the characters in Cangjie codes *(倉頡碼)* to filter out confusable characters with low similarity. We retain character pairs differing from each other by at most one symbol in Cangjie codes that tend to be highly similar in shape. For example, the code of 徵 *(zheng)* and 微 *(wei)* are highly similar in shape, and their corresponding codes "竹人山土大" and "竹人山山大", differ only in one place.

Note that we do not attempt to estimate the channel probabilities of typos of a character at this point. Instead, we only use sound or shape similarity to limit confusable characters, leading to more effective confusion set as the basis for subsequent probability estimation.

### 3.2.2 Retrieving Ngrams

In the second stage of the training phase (Step (2) in Figure 3.1), we retrieve ngrams (e.g., 所定目標 *(suo ding mu biao)*) possibly containing a typo characters (e.g., 所 *(suo)*) that can be corrected using the confusable characters (e.g., 所 *(suo)*, 鎖 *(suo)*, or 索 *(suo)*). In order to estimate channel probabilities, we use an existing Chinese spell checker *CSC* to correct typos in the ngrams to a parallel corpus with typos annotated. We extract ngrams based on collocates of high frequency words containing the confusable character. The procedure for retrieving and correcting ngrams consist of a number of steps, namely, generating collocates for words containing a specific character, filtering these collocates by frequency, producing the ngrams for the remaining collocates, and correcting these ngrams using *CSC*. Each step is described below in detail.

For this stage of the learning process, we use a collection of <*Word, Collocate*> pairs (e.g., <目標, 鎖定> *(<mu biao, suo ding>)*, <版面, 鎖定> *(<ban mian, suo ding>)*). We generate the word from the corpus using word frequency and find corresponding collocates using Dice coefficient, which is a statistic association value used for comparing the relation of words and collocates. The collocates of each word are sorted according to the Dice coefficient. We retain at most *K* collocates per word to reduce the computational cost. We compute Dice coefficient using the following equation:

$$Dice(word, collocate) = \frac{2 \cdot freq_{\text{(word)}} \cdot freq_{\text{(collocate)}}}{freq_{\text{(word)}} + freq_{\text{(collocate)}}} \tag{3.1}$$

Table 3.2: Two sample collocates of 鎖定 and 封鎖.

| Words | Collocates | Dice |
|---|---|---|
| 鎖定 | 版面 | .025 |
| | 單擊 | .021 |
| | 防偷 | .004 |
| | 目標 | .004 |
| | 移動 | .004 |
| | 已經 | .002 |
| | 敬請 | .001 |
| | 解除 | .001 |

| Words | Collocates | Dice |
|---|---|---|
| 封鎖 | 衝出 | .019 |
| | 長城 | .017 |
| | 突破 | .015 |
| | 嚴密 | .007 |
| | 網絡 | .002 |
| | 大陸 | .001 |

Table 3.3: Sample texts of typo 所 and 索 of 鎖 from the corpus.

| Typos | Texts | Count |
|---|---|---|
| 所 | 中 所 定 目標 | 86 |
| | 依 所 定 目標 | 83 |
| | 達到 所 定 目標 | 44 |
| | 我們 所 定 的 目標 | 42 |
| 索 | 索 定 海珠 收 | 66 |
| | 索 定 起息日 | 93 |
| | 索 定 高清 | 40 |

where $freq_{(word)}$ is the frequency of the word, and $freq_{(collocate)}$ is the frequency of the collocate. Take 鎖 *(suo)* for instance, the words (e.g., 鎖定 *(suo ding)* and 封鎖 *(feng suo)*) and their corresponding collocates of words are shown in Table 3.2. The word 鎖定 *(suo ding)* has the highest Dice coefficient of 0.025 with the collocate 版面 *(ban mian)*, while 封鎖 *(feng suo)*) has the highest Dice coefficient of 0.019 with the collocate 衝出 *(chong chu)*.

Table 3.4: A sample of instances containing character 鎖 and potentially confusable characters.

| Words | Collocates | Confusable Characters | Confusable Collocates |
|---|---|---|---|
| 鎖定 | 目標 | **所** | 目標**所**定 |
| 封鎖 | 突破 | **索** | 突破封**索** |
| 深鎖 | 眉頭 | **瑣** | 眉頭深**瑣** |

For each <*Word, Collocate*> pair, we generate all possible potential ngrams *N* containing *Word* and *Collocate*. This stage of the learning process operates over a dataset of word ngrams. The sample texts of the typos (所, 索, and 瑣) of 鎖 found in a corpus is shown in Table 3.3. We find the ngrams in the corpus with identical collocates and *Word* containing confusable characters (e.g., <所定, 目標>). Sample confusable collocates of character 鎖 is shown in Table 3.4. In this sample, we can find that 鎖 may be misused as confusable characters (e.g., 所, 索, 瑣) in the corpus with such information in the ngrams, we can generate typo pairs (e.g., [所, 鎖], [索, 鎖], [瑣, 鎖]). Finally, we correct the typos in these ngrams by using existing Chinese spell checker (In Section 3.2.3). With the typos and corrections, we can estimate the channel probabilities.

### 3.2.3 Correcting Ngrams and Training Channel Model

In the third and final stage of training, we correct the ngrams and train the channel model for supporting correction candidates. Figure 3.2 shows the algorithm for correcting ngrams using a Chinese spell checker and estimating the channel probabilities related to typo pairs. The procedure is repeated for all ngrams obtained in the previous stage until the channel probabilities converge.

We are given a set of ngrams as training data (described in Section 3.2.2). Recall that our goal is to estimate the channel model for each character, in the form of [original, correction, log channel probability] (e.g., [所, 鎖, -4.284] and [索, 鎖, -5.264]). In order to generate a parallel corpus, we need to provide representative ngrams to the training algorithm. The training set is created by retrieving the ngrams from *Word*s of each character and the corresponding *Collocate*s in the corpus.

We apply a previously developed Chinese spell checker(*CSC*) to correct ngrams. In this checker, we adopt the confusion set generated in Stage (1) to reduce the unlikely confusable characters and improve the accuracy for generating typo pairs. We combine the global error rate and local error probability to reliably estimate the channel probabilities using following equation:

$$ChannelProbability(O,C) = W_{\mathrm{GL}} \cdot GlobalErrorProb + (1 - W_{\mathrm{GL}}) \cdot LocalErrorProb(O,C) \tag{3.2}$$

where $O$ is original character, $C$ is corrected character, and $W_{\mathrm{GL}}$ is a weight for probability. The global error probability is a prior probability calculated from a development data set, which can instead the detection and avoid data sparse. The global error probability calculated by the following equation.

$$GlobalErrorProbability(OriginalSet, CorrectionSet) = \left\{ \begin{array}{c} \frac{count(nochange)}{count(char)} \\ \\ \frac{count(typos)}{count(char)} \end{array} \right\} \tag{3.3}$$

where *count(nochange)* is the count of corrected characters, *count(typos)* is the count of typos, and *count(char)* is the count of characters. The *Original Set* and *Correction Set* are the development data.

Figure 3.2: Automatically correcting ngrams and estimating the channel model.

```
Procedure BuildChannelmodel(ngrams N):

     for each ngram in N:
(1)     correctedngram = CSC(ngram)
        for each character i in ngram and correctedngram:
(2)        typopairs += [ngram(i), correctedngram(i), frep(ngram)]
(3)  global = globalerrorprobability(development data)
(4)  typopairs sorted by ngram(i), i = 0
     for each [Original, Correction, Freq] in typopairs:
(5a)     local = localerrorprobability(Original, Correction)
(5b)     channel = channelprobability(local, global)
(5c)     channelmodel += [Original, Correction, channel]

(6) Output the channel model.
```

Table 3.5: A sample of the typo pairs for 鎖.

| Ngrams | Corrected Ngrams | Typo Pairs |
|---|---|---|
| 目標所定 | 目標所定 | [目, 目], [標, 標] [所, 所], [定, 定] |
| 突破封索 | 突破封鎖 | [突, 突], [破, 破] [封, 封], [索, 鎖] |
| 眉頭深瑣 | 眉頭深鎖 | [眉, 眉], [頭, 頭] [深, 深], [瑣, 鎖] |

We use the Expectation-maximization algorithm to estimate the local error probabilities related to the confusion set. We initialize the confusion set with uniformed probability in the E-step and re-estimate the probability of each character in M-step until the local error probability converge. For each of the potentially confused ngram (e.g., 所定目標 *(suo ding mu biao)*, we attempt to find typos and corrections using *CSC* (Step (1)) and produce the typo pairs (Step (2)). The typo pairs are in the form of [Original, Correction]. The frequency is the count of how many times of the ngram occurs in the corpus. We estimate the local error probability based on nochange pair (e.g., [所, 所] *([suo, suo])*), and correction pair (e.g., [所, 鎖] *([suo, suo])*). In Table 3.5, we show a sample of the typo pairs in the ngrams of the character 鎖 *(suo)*.

Table 3.6: Sample of the typo pairs with frequency.

| Original | Correction | Typo Pairs | Frequency |
|---|---|---|---|
| 所 | 所 | [所, 所] | 6,799,532 |
| 所 | 匠 | [所, 匠] | 529 |
| 所 | 索 | [所, 索] | 235 |
| Total Frequency | | | 6,800,296 |

Then we calculate the global error probability using the development data (Step (3)). In Step (4), the typo pairs are sorted according to the *Original*. For each [*Original, Correction*] pair, we calculate the local error probability of the *Original* conditioned on *Correction* (Step (5a)). The probability is calculated as follows:

$$LocalErrorProbability(Original, Correction_{\text{Original}}) = \frac{count(Original, Correction_{\text{Original}})}{count(Original)} \tag{3.4}$$

As shown in Table 3.6, the total *count* of 所 *(suo)* is 6799532 + 529 + 235 = 6800296, the *count* of (所, 索) is 235, and the *LocalErrorProbability(*所*,* 索*)* is calculated as follows:

LocalErrorProbability(所, 索) = Count(所, 索)) / Count(所) = 235/6800296 =0.0000346

However, we can not reliably estimate that 所 *(suo)* as a typo of 瑣 *(suo)*, if *CSC* does not find [所, 瑣] *([suo, suo])*. In that case, we use smoothing algorithm to adjust the frequency count. If a confusable character does not has a certain typo pair, we use add-one smoothing algorithm to deal with the unseen problem. For example, confusable characters (e.g., 瑣, 鎖) of 所 *(suo)* are not found in the corpus, so we add count one for them. Table 3.7 shows a confusion set of 所 *(suo)* and the corresponding smoothed local error probability.

Table 3.7: The result of the local error probability with smoothing.

| Original | Correction | Frequency | Local Error Probability$_{\log}$ |
|---|---|---:|---:|
| 所 | 所 | 6799532 | -0.0001 |
| 所 | 匠 | 529 | -9.4614 |
| 所 | 索 | 235 | -10.2728 |
| 所 | 瑣 | 1 | -15.7324 |
| 所 | 鎖 | 1 | -15.7324 |

Table 3.8: A sample of the channel model for 所 *(suo)*.

| Original | Correction | Frequency | Channel Probability$_{\log}$ |
|---|---|---:|---:|
| 所 | 所 | 6799532 | -0.1416 |
| 所 | 匠 | 529 | -2.2111 |
| 所 | 索 | 235 | -4.4357 |
| 所 | 瑣 | 1 | -10.4947 |
| 所 | 鎖 | 1 | -10.4947 |

We combine the global error probability and the local error probability to estimate the channel probabilities in Step (5b), and save the *Original*, *Correction*, and their channel probability in the channel model in Step (5c). Steps (1) through (5) are repeated to re-estimate the local error probability until the probabilities converge. The output of this stage of training is a channel model with reliable probabilities, automatically estimated using the confusable characters and ngrams based on collocates. A samples of the channel model for 所 *(suo)* is shown in Table 3.8.

## 3.3 Run-time Typo Correction

Once the channel model is automatically trained for each character, we store the model as a confusion set with probability. We then correct a given sentence using the procedure shown in Figure 3.3 with the character-based language model and the channel model.

For each character in the given sentence of *n* characters (e.g., 自己鎖定的目標 *(zi ji suo ding de mu biao)*), we correct typos as follows. In Step (1), the system initializes *n*

Figure 3.3: Pseudocode of the runtime phase.

```
Procedure CorrectTypos(sentence S of length n, LM, CM):

    for each char s in S:
(1a)  initial n+1 stacks(S, correction, score), stack(i), i = 0...n
(1b)  place null char & score = 1, (Null, Null, 1) in stack(0)

      for all stack(i), i = 0...n-1:
        for (s, c, p) in stack(i):
          for each confusable character c' of $s_i$' with $p_{CM}$':
            if s does not overlap with s' or s' = Null

(2)             Create new hypothesis c''=c+c' with p''=p*$p_{CM}$'*$p_{LM}$(c,c')
(3)             Place (s'', c'', p'') in stack(k), s''=s or s', k=|s''|
                  for all (Q, U, $P_U$) in stack(k):
(4)                 Combine(c'', U) if s''=Q & Compatible(c'', U)
(5)             Prune stack(k) if stack(k) is to big

(6) Output correction c* with maximal score in stack(n)
```

stacks for the channel model, [*Character, Ngram, Score*]. In Step (2), the system replaces each character with the confusable characters (e.g., 所,索,瑣,鎖 *(suo, suo, suo, suo)*) in the channel model as the correction candidates. For each confusable characters, we create new hypotheses with a score, character ngram state, character, and correction candidates. In order to reduce computational complexity, we use beam search algorithm to replace each and calculate the score of sentences. The score in a hypothesis is calculated based on the channel model and the language model as follows.

$$Score(hypothesis) = log(LanguageProbability^{W_{\text{LC}}} \cdot ChannelProbability^{(1-W_{\text{LC}})}) \quad (3.5)$$

$$= W_{\text{LC}} \cdot log(LanguageProbability) + (1 - W_{\text{LC}}) \cdot log(ChannelProbability) \quad (3.6)$$

where $W_{\text{LC}}$ is a weight parameter in channel model and language model. A sample hypothesis is shown in Table 3.9. In Step (3), the new hypothesis are stored in the stack and

Table 3.9: A sample of the hypotheses.

| Originals | Corrections | Ngrams | Score |
|---|---|---|---:|
| <s> | <s> | None | 1 |
| 自 | 自 | (<s>,) | 0.0 |
| 己 | 己 | (<s>,自) | -2.6049 |
| 鎖 | 所 | (自,己) | -2.6756 |
| 定 | 定 | (己,所) | -5.1145 |
| 的 | 的 | (所,定) | -6.3698 |
| 目 | 目 | (定,的) | -5.1627 |
| 標 | 標 | (的,目) | -5.7875 |
| </s> | </s> | (目,標) | -10.2282 |

Table 3.10: A sample of the given sentences and corrections.

| | Sentences | Corrections |
|---|---|---|
| Given Sent. | 遇到逆竟時，我們必須勇於面對。 | 竟->境 |
| Corrected Sent. | 遇到逆境時，我們必須勇於面對。 | |
| Given Sent. | 或許我們會在挫折中有令人不同凡想的成就呢！ | 想->響 |
| Corrected Sent. | 或許我們會在挫折中有令人不同凡響的成就呢！ | |
| Given Sent. | 人生難免會碰到的一些錯折。 | 錯->挫 |
| Corrected Sent. | 人生難免會碰到的一些挫折。 | |

combined with the existing hypothesis in Step (4). If the stack has too many hypotheses, we prune the stack down to a fixed size in Step (5).

Finally in Step (6), we compare the score of all the hypotheses in the last stack, and output the correction candidate with the highest score as output. When there is no correction candidates with the highest score (e.g., score(自己所定的目標) = -10.2282), we output the given sentence. Table 3.10 shows three input sentences and the corresponding corrected sentences output. For example, 竟 *(jing)* is corrected as 境 *(jing)*, because 境 *(jing)* is the most appropriate for the context of 遇到逆 * 時 *(yu dao ni jing shi)*.

# Chapter 4

# Experiment and Discussion

We describe the resources we used and the systems compared in our experiments in Section 4.1, and the results of the experiments and the limitations of our system show in in Section 4.2.

## 4.1 Experiment Setting

Our systems were designed to provide wide-coverage spell checking for Chinese texts. As such, we trained our systems using the confusion set, a compiled corpus, Web-scale ngrams, and an existing Chinese spell checker. These resources are used for different purposes: The confusion sets provide the correction candidates; the compiled corpus provide the training data for the language model. And finally, Web-scale ngrams and the existing Chinese spell checker are used for training the channel model. We evaluate our systems on the sentence level. In this section, we first present the details of data used in training(Section 4.1.1 to Section 4.1.4). Then, Section 4.1.5 describes the test data, while Section 4.1.6 describes the systems compared. Finally, the evaluation metrics for the performance of the systems are reported in Section 4.1.7.

Table 4.1: Sample sound-similar characters from SIGHAN 7 Bake-off 2013: Chinese Spelling Check Shared Task.

| | for 已 | for 勇 | for 胡 |
|---|---|---|---|
| *II* | 尾椅以蟻乙矣旖偯迤迆倚 | 擁泳踴永湧甬蛹臃俑詠湧壅恿 | 瑚壺鬍鶘蝴糊湖葫囫弧斛狐 |
| *ID* | 漪藝屹肄軼刈誼囈迤移縊逸彝溢佾夷沂宜劓醫疫頤噫疑懿鎰毅揖義訑蛇邑億液佚役儀易羿裔姨咦弋蜴壹驛飴弈譯帟翌異益栘臆遺亦艾腋曳咿議詣意伊繹怡貽憶掖浥翳圯胰奕一翼抑衣痍依 | 擁鄘傭臃邕癰慵用庸壅傭 | 瓠互琥虎扈許惚唬乎戲呼户忽滬楛沍護 |
| *SI* | | 允吻蓊隕穩刎 | 氟縛匐夫蝠紼扶芾幅孚弗輻紱袱福浮伏符菔拂涪怫彿服芙俘 |
| *SD* | | 員翁玟耘嗡文聞孕瘟熨溫慍芸塭雲韻雯運紋筠紊蚊甕昀暈醞抆蘊氳汶紜匀問雲免 | 富馥覆附膚俯鈇複府夫腑腐頫甫阜腹賻副傅復釜敷付撫斧孵赴婦莆趺輔咐父麩脯賦負駙佚訃 |
| *RS* | 己巳 | 勉勁勃 | 冑胥胖胤胃胝背胞胎胚胛 |
| Count | 94 | 68 | 106 |

Table 4.2: Sample of shape-similar characters from SIGHAN 7 Bake-off 2013: Chinese Spelling Check Shared Task.

| | for 已 | for 勇 | for 胡 |
|---|---|---|---|
| Shape Similarity | 己忌厄氾妃改杞凹卮犯危扼呃卷范尼屉居紀苑桅倦脆捲圈 | 湧奐肋踴男另筋脅瘓舶喚換渙煥努劣勢勞力助叻勵 | 古湖葫瑚糊故蝴枯沽姑咕估鈷苦居固鶘辜鬍牯涓捐娟楛硼個涸錮舌箇 |
| Count | 25 | 22 | 30 |

### 4.1.1 Confusion Set

The confusion sets we used were the same as those provided for SIGHAN 7 Bake-off 2013: Chinese Spelling Check Shared Task (Liu et al., 2011). The confusion sets represent sound similarity and shape similarity between a typo and potential corrections.

There four categories of phonological similarity between two characters: identical sound and tone (*II*), identical sound but different tone (*ID*), similar sound and identical tone (*SI*), similar sound and different tone (*SD*), and identical radical and number of strokes (*RS*). A sample of sound-related confusion sets from SIGHAN 7 Bake-off 2013 is shown in Table 4.1. In this sample, the confusion sets of 已 *(yi)*, 勇 *(yong)*, and 胡 *(hu)* contain a lot of unlikely confusable characters. Examples of unlikely pairs include 已 *(yi)* and 肄 *(yi)* in *ID*, 勇 *(yong)* and 穩 *(wen)* in *SI*, 胡 *(hu)* and 馥 *(fu)* in *SD*. In Table 4.2, we show the shape-related confusion sets of 已 *(yi)*, 勇 *(yong)*, and 胡 *(hu)*. The confusion sets also contain loosely similarly relations, for instance, 已 *(yi)* and 圈 *(quan)* are not very similar visually.

In our work, we expand the sets slightly, while also remove some unlikely confusable characters in order to improve the performance. We modify the confusion set using the pronunciation and Cangjie codes (倉頡碼). The process is described in detail in Section 3.2.1.

### 4.1.2 Google Chinese Web 5-gram

In 2010, *Google* published a Chinese Web 5-gram dataset based on public webpages through Linguistics Data Consortium (LDC).[1] Chinese Web 5-gram consists of Chinese

[1] https://catalog.ldc.upenn.edu/LDC2010T06

Table 4.3: The information of n-grams in Google Chinese 5-gram and Traditional Chinese 5-gram.

| N-gram Types | Google Chinese 5-gram | Traditional Chinese 5-gram |
|---|---:|---:|
| Unigram | 1,616,150 | 527,694 |
| Bigram | 281,107,315 | 102,092,428 |
| Trigram | 1,024,642,142 | 237,599,483 |
| Fourgram | 1,348,990,533 | 201,500,549 |
| Fivegram | 1,256,043,325 | 126,959,922 |

word n-grams and their observed frequency counts generated from approximately 883 billion word tokens of text in publicly accessible Web pages. The Google Chinese Web 5-gram contains 30 GB (gzip compressed) of text files with n-grams ranging from unigrams (single words) to fivegrams. In this work, we used only the traditional Chinese 5-grams. Table 4.3 shows the information of 5-grams in Google Chinese Web 5-gram and traditional Chinese Web 5-gram. We use the traditional Chinese Web 5-gram to retrieve ngrams (at most ten *Words*) in the training phase for estimate channel model probabilities. The advantage of using the Web ngram is that unlike a compiled corpus, it contains many typos.

### 4.1.3 Existing Chinese Spell Checker

We use an existing Chinese spell checker (*CSC*) we previously developed in 2013 (Chiu et al., 2013). This *CSC* is based on a novel method for detecting and correcting Chinese typographical typos. The approach involves word segmentation, detection rules, and phrase-based machine translation. The error detection module detects typos by segmenting words and checking word and phrase frequency based on compiled and Web corpora. The phonological or morphological typographical typos found then are corrected by running a decoder based on the statistical machine translation model. The language model is trained using the word-based corpus using the SRILM (Stolcke et al., 2011) toolkit. The

Table 4.4: The information of the word, character, article, and percentage in the area of sinica corpus.

| Areas | Word Token | Character | Article | Percentage |
|---|---:|---:|---:|---:|
| Literature | 777,050 | 1,169,801 | 1,385 | 15% |
| Life | 858,750 | 1,398,791 | 2,301 | 25% |
| Society | 1,610,997 | 2,711,720 | 3,246 | 35% |
| Science | 629,838 | 1,054,738 | 994 | 10% |
| Philosophy | 439,955 | 673,080 | 695 | 8% |
| Art | 474,340 | 781,415 | 518 | 6% |
| Others | 101,394 | 160,306 | 89 | 1 % |
| Total Count | 4,892,324 | 7,949,851 | 9228 | 100% |

translation model is trained using the frequency of the word containing typos and the corrected word. The results show that the proposed system achieves high accuracy in error detecting and correcting. We use this Chinese spell checker to train the channel model and as a system to compared with the proposed method.

### 4.1.4 Sinica Corpus

"Academia Sinica Balanced Corpus of Modern Chinese", or "Sinica Corpus", is the first balanced Chinese corpus with part-of-speech tags. The size of the corpus we used is about 5 million words. The corpus is segmented according to the word segmentation standard proposed by the ROC Computational Linguistic Society. Each segmented word is manually tagged with a part of speech. Texts in the corpus are collected from different areas: Literature, Life, Society, Science, Philosophy, and Art. Table 4.4 shows the information about numbers of word, character, article, and percentage by area. We use Sinica Corpus (ignoring word segmentation) to train a character-based n-gram language model running the SRILM (Stolcke et al., 2011) toolkit. The sizes of the ngrams of the character-based language model is shown in Table 4.5

Table 4.5: The information of n-grams in character-based language model.

| Ngram Types | Ngram Count |
|---|---:|
| Unigram | 17,201 |
| Bigram | 741,739 |
| Trigram | 859,442 |
| Fourgram | 791,846 |
| Fivegram | 588,200 |

Table 4.6: The information of test dataset from SIGHAN 7 Bake-off 2013: Chinese Spelling Check Shared Task.

| Dataset | Subtask 1 | Subtask 2 |
|---|---:|---:|
| Number of sentences with typos | 300 | 1,000 |
| Number of typos | 376 | 1265 |
| Average length of sentences | 69 | 74 |
| Sentence-level typo percentage | 30% | 100% |
| Character-level typo percentage | 0.55% | 1.70% |

### 4.1.5 Test Data

We used the test dataset from SIGHAN 7 Bake-off 2013 to evaluate our systems. This dataset is used for two sub-tasks: error detection and error correction. For the error detection (Subtask 1), there are 1,000 sentences with/without spelling errors. And for the error correction (Subtask 2), there are also 1,000 sentences but all sentences have typos. These sentences are extracted from student essays containing various common typos. The information of the dataset is shown in Table 4.6.

In Subtask 2, we observe that all sentences contain from one to five typos and that most typos are confusable either because of pronunciation or shape. Therefore, the confusion set was suitable for error correction. The dataset was released in sentence format with the information of sentences NID. Several sample sentences in Subtask 2 are shown in Figure 4.1.

Figure 4.1: Several sample sentences in subtask2 from SIGHAN 7 Bake-off 2013: Chinese Spelling Check Shared Task.

(NID=00401) 遇到逆**竟**時，我們必須勇於面對，而且要愈挫愈勇，這樣我們才能朝著成功之路前進。

(NID=00402) 大自然也一樣的，無法天天都是晴天，天天都很順利，但生活，就是如此這般，有失才有得，只是每個人是如何去看**侍**的。

...

...

(NID=00881) 哥哥不但給我了目標，也給了我學習的榜樣，他常告**訢**我讀書不能一直苦讀，而是要去融會**灌**通，其實方向就是爲人所設定目標。

(NID=00882) 我想人生最重要的指引應該就是方向了吧！不管是在開車時、寫字時或者是隨便一個動**做**都需要方向，但你可曾想過，人生也需要方向嗎？

...

...

(NID=01481) 再來，如果生活上沒有壓力的阻礙，但一整天下來也沒有觀察到什麼特別的事物，那麼，可能就要改變一下自己的**息**慣了。

(NID=01865) 儘管沒有頻繁的聯絡，可是在**交**潔明月下相信友人一定還記得我！晚風不急不**涂**的吹過，我腦海裡閃過曾經共享樂共患難的快活時光，雖然不是全然的無憂無慮，可是卻是我視如珍寶的回憶。

(NID=01866) 秋天的風徐徐吹來，葉子落下，這代表著什麼？大家就像那樹梢上的葉片，生命中的第一年過去了，就像準備踏上國中之路。橘黃色的葉**辨**，陣陣的微風吹過，就像在耳旁提醒著你將要向大家告別。

### 4.1.6 Systems Compared

Recall that we propose a system to correct typos in Chinese based broadly on a noisy channel model. The systems starts with a given sentence, and use the language model and the channel model generated in training phase. The output of the system is a corrected sentence, which can either be returned to the user directly. Hence, we propose to use *CSC* based on statistical machine translation in 2013 described in Section 4.1.3. We found that the translation probability (in the channel) is not reasonable and the system rely on mostly the detection model and the language model. In order to solve these problem, we propose a novel method for training a channel model in this paper. The most important point is that we calculate the channel model more accurately and reasonably. We propose a method using Expectation-Maximization (*EM*) algorithms to learn to estimate the channel probability form the training data of ngrams. We also combine the global error probability and local error probability in the channel model. Finally, we correct typos using a noisy channel model based on channel probability and character-based language model.

We compare four systems in our experimental evaluation. We experimented with different probability in statistical machine translation and noisy channel model in first experiment. We compare our system with the system proposed by Chiu et al. (2013), which detect and correct typos using the result of the rule-based method and the machine translation method. The second experiment is designed to compare our system and one to one weight for the language model and channel model. The last, we compare the system with one to one weight in global error probability and local error probability. The four systems compared are:

- The proposed system: The system proposed in this paper (**NCM**): Correct typos used the noisy channel model with accurate and reasonable channel probability. The local error probability is trained using EM algorithm and the system also using the global error probability and the weight, $W_{\mathrm{GL}}$ and $W_{\mathrm{LC}}$.

- The system compared with Chiu et al. (2013).
  - The system proposed by Chiu et al. (2013) (**CSC**): Correct typos used the machine translation method with inaccurate translation probability.
- One to one weight use in the language model and the channel model in the noisy channel model.
  - One to one weight in the noisy channel model (**NCM**$_{\mathrm{LMCMWt}}$): Correct typos use the noisy channel model with one to one weight of the language model and the channel model.
- One to one weight use in the global error probability and the local error probability in the channel model.
  - One to one weight in the channel model (**NCM**$_{\mathrm{GLWt}}$): Correct typos use the channel model with one to one weight of the global error probability and the local error probability

### 4.1.7 Evaluation Metrics

Spell checking is usually compared based on the evaluation of the systems. This evaluation calculate using two metrics, recall and precision We used the evaluation tool provided by the SIGHAN 7 Bake-off 2013: Chinese Spelling Check Shared Task. For error correction subtask, SIGHAN adopt sentence-level metrics for performance evaluation, because the number of typos is too small comparing to all the characters so that it is not suitable to use characters as the unit of performance metrics. In order to assess the effectiveness of the proposed system, we use the test data and the gold standard which are also proposed by SIGHAN to experiment with our system. We also exploit different confusion set and channel probability in the proposed system and compare the system in this paper to the system we developed in 2013 (Chiu et al., 2013). To evaluate our system, we use the SIGHAN evaluation metrics are follows:

$$LocationAccuracy(LA) = \frac{L}{T} \tag{4.1}$$

$$CorrectionAccuracy(CA) = \frac{A}{T} \tag{4.2}$$

$$CorrectionPrecision(CP) = \frac{A}{R} \tag{4.3}$$

where $L$ is the number of sentences correctly detected the error location, $T$ is the number of sentences in the test data, $A$ is the number of sentences correctly corrected the error, and $R$ is the number of sentences the system returns corrections. The standard data format for determining corrections includes NId, location of the typo, and the correction of the output from the system, which is used to match with the gold standard.

For example, give 6 test sentences with gold standard shown in Table 4.7. If the output the results of our system:

- Output:

  1. "00403, 24, 非"

Table 4.7: The given test sentences with gold standard.

| Sentence | Gold Standard |
|---|---|
| (NID=00403) 劉墉在三歲過年時，全家陷入火海，把家燒得面目全飛、體無完膚。 | 00403, 24, 非 |
| (NID=00408) 或許我們會在挫折中有令人不同凡想的成就呢！ | 00408, 16, 響 |
| (NID=00415) 先苦後甘的美好滋味，才應是人們必生去追尋的目標，沒有努力的收獲反而使你乏味，人生的順境和逆境，好好的選則你將來要踏上的旅程吧。 | 00415, 16, 畢, 31, 穫, 52, 擇 |
| (NID=00436) 當我從溫暖的被窩爬起來時，外面的戰火己停止了，天空也放晴了，再度露出那耀眼的笑容。 | 00436, 19, 已, 28, 晴 |
| (NID=01122) 自己從今天就要開始跟時間爭鬥，不放過忍何一秒，這才是我眞正的潛能，自己的潛能也要慢慢的出來，不然你還是比人家爛，但是如果有努力的話，老天爺不會匱待你的。俗話説：「黃天不復苦心人。」這就是這個義思。 | 01122, 19, 任, 72, 虧, 82, 皇, 85, 負, 96, 意 |
| (NID=01441) 多了快樂，少了憂愁，拋去煩腦，可不是生活一大樂事嗎？ | 01441, 14, 惱 |

2. "00408, 16, 饗"

3. "00415, 16, 嘩, 52, 擇"

4. "01122, 72, 虧"

5. "01441, 14, 惱"

The evaluated tool will yield the follows:

- Location Accuracy (LA) = 0.50 (=3/6)

  L = 3 sentences ("00403, 24", "00408, 16", "01441, 14"), T = 6 sentences ("00403, 24, 非", "00408, 16, 饗", "00415, 16, 嘩, 52, 擇", "01122, 72, 虧", "01441, 14, 惱")

- Correction Accuracy (CA) = 0.33 (=2/6)

  A = 2 sentences ("00403, 24, 非", "01441, 14, 惱"), T = 6 sentences ("00403, 24, 非", "00408, 16, 饗", "00415, 16, 嘩, 52, 擇", "01122, 72, 虧", "01441, 14, 惱")

- Correction Precision (CP) = 0.40 (=2/5)

  A = 2 sentences ("00403, 24, 非", "01441, 14, 惱"), R = 5 sentences ("00403, 24, 非", "00408, 16, 饗", "00415, 16, 嘩, 52, 擇", "01122, 72, 虧" , "01441, 14, 惱")

## 4.2 Evaluation

In this section, we report the results of the experimental evaluation using the methodology described in the previous section. First, we report the results of Comparison 1, which compared the two systems of Chiu et al. (2013) (**CSC**) and our system (**NCM**). For this comparison, we set the parameters as follows according to the some experimentation with the different values of these parameters in our system.

Table 4.8: The evaluation results of comparison systems.

| Systems | Location Accuracy | Correction Accuracy | Correction Precision |
|---|---:|---:|---:|
| NCM | **.49** | **.44** | **.60** |
| CSC | .40 | .38 | .58 |
| $\text{NCM}_{\text{LMCMWt}}$ | .10 | .09 | .55 |
| $\text{NCM}_{\text{GLWt}}$ | .48 | .40 | .43 |

- $W_{\text{GL}} = 0.1$

- $W_{\text{LC}} = 0.7$

- Local error probability = EM

Second, we show the results of Comparison 2 for the system with one to one weight (**$\text{NCM}_{\text{LMCMWt}}$**) for the language model and the channel model. Finally, we describe the results of Comparison 3, which compared the system with one to one weight (**$\text{NCM}_{\text{GLWt}}$**) for the global error probability and local error probability. During this evaluation, we tested our systems on 1,000 sentences containing at least one typo, provided in SIGHAN Bake-off 2013: Chinese Spelling Check. The confusion set with identical sounds and strongly similar shape relations described in Section 3.2.1. In this section, we discuss our evaluation results and analyze the cause of these errors. Table 4.8 shows the results of *NCM* and comparing with three systems.

As we can see, the performance of noisy channel model **NCM** is significantly better than **CSC**. Because **CSC** use a lot of unreliable rules to find typos position during error detection. In contrast, we skip the detection process and correct the typos systematically and without heuristic rules. **CSC** adopts a statistical machine translation to correct typos, but uses inaccurate probabilities for the confusion sets. On the other hand, we use the global error probability and local error probability based on EM algorithm to reliably

Table 4.9: The comparison systems with weight for the language model and channel model.

| Language model | Channel model | Location | Correction | Correction |
|---:|---:|---:|---:|---:|
| ($W_{LC}$) | ($1-W_{LC}$) | Accuracy | Accuracy | Precision |
| 1.0 | 0.0 | .589 | .489 | .491 |
| 0.9 | 0.1 | .558 | .471 | .410 |
| 0.8 | 0.2 | .549 | .478 | .575 |
| 0.7 | 0.3 | **.493** | **.443** | **.600** |
| 0.6 | 0.4 | .419 | .381 | .594 |
| 0.5 | 0.5 | .341 | .309 | .566 |
| 0.4 | 0.6 | .268 | .243 | .533 |
| 0.3 | 0.7 | .231 | .202 | .414 |
| 0.2 | 0.8 | .164 | .137 | .228 |
| 0.1 | 0.9 | .089 | .068 | .073 |
| 0.0 | 1.0 | .031 | .018 | .025 |
| One to one weight | | .098 | .089 | .546 |

estimate probabilities, leading to better results.

We investigate the weight for the language model and channel model. The results (Table 4.9) show that setting the weight 0.7 for the language model and 0.3 for the channel model performs significantly better than other weights in terms of *CP*. So the balance weight between language model and channel model is important in spell checking. We also show that the channel model we estimate is useful. We compare **NCM** with **NCM**$_{\text{LMCMWt}}$ (Comparison 2) in Table 4.8. *LA* ranges from 0.10 to 0.49, and *CA* from 0.09 to 0.44, which shows that the weight can be tuned for the language model and channel model to increase the recall.

In the last comparison, we use different weights in global error probability and local error probability. We compare the system with different and with one to one weight **NCM**$_{\text{GLWt}}$, although **NCM**$_{\text{GLWt}}$ has the acceptable performance but we can enhance the performance using the global error probability. As can be seen in Table 4.10, the weights

Table 4.10: The comparison systems with weight for the global error rate and local error rate.

| Global Error Prob. | Local Error Prob. | Location | Correction | Correction |
|---|---|---|---|---|
| ($W_{GL}$) | (1-$W_{GL}$) | Accuracy | Accuracy | Precision |
| 1.0 | 0.0 | .479 | .400 | .430 |
| 0.9 | 0.1 | .509 | .427 | .471 |
| 0.8 | 0.2 | .522 | .444 | .509 |
| 0.7 | 0.3 | .518 | .446 | .528 |
| 0.6 | 0.4 | .522 | .449 | .544 |
| 0.5 | 0.5 | .525 | .456 | .563 |
| 0.4 | 0.6 | .516 | .456 | .582 |
| 0.3 | 0.7 | .508 | .453 | .589 |
| 0.2 | 0.8 | .498 | .447 | .596 |
| 0.1 | 0.9 | **.493** | **.443** | **.600** |
| 0.0 | 1.0 | .483 | .433 | .599 |
| One to one weight | | .479 | .400 | .430 |

0.1 and 0.9 in global probability and local probability leads to the highest precision. But if we do not consider the global probability, the performance decrease by 1%. The results show that with the global error probability added in channel model, we can improve the *LA* and *CA*, while decrease precision. In Table 4.8, we can see that the *CP* ranges from 0.43 to 0.60, so that the weight for the global error probability and local error probability can be tuned to improve performance.

Since a Chinese spell checker is often relied on the manually compiled confusion sets and data, the correction performance of our system is limited by the confusion set that we used to correct typos. There are three situations where our system can not correct typos successfully. The first situation is when the correction is not in the confusion set of a typo. The second is the correction fail to result in the highest score. The last is due to semantic errors not covered by the current confusable sets. Examples of these situations are shown in Table 4.11. To deal with these errors, we may have to use Web-based character n-grams to automatically generate confusion sets, which are more likely to cope with such

Table 4.11: A sample of three situations.

| Type | Sentence | Correction |
|---|---|---|
| Confusion Set | 我一直深深地相信著人定勝天這句話，我不是不認同那些命運、占卜、算命之類的東西，而是我認爲凡事有心最重要。就像一場百米競賽，認眞、付出苦心練習的人就算飲**悢**敗北了，我仍然爲他拍手鼓掌。 | 76, 悢->恨 |
|  | 如果你堅持對別人是生氣的，那別人的**歡**說都是沒辦法改變你的，可以改變的只有自己。所以，在任何壓力大或不高興的時候，就去試著改自己的想法，去尋找或是創造屬於自己的快樂。 | 18, 歡->勸 |
| Score | 劉墉在三歲過年時，全家陷入火海，把家燒得面目全**飛**、體無完膚。 | 24, 飛->非 |
|  | 這個佈告欄是本班的標**制**，是本班的心情留言區，更是本班的成品區。 | 11, 制->誌 |
| Semantic | 「**雜**草不除根，春風吹又生」，爲什麼只是小小的一枝草，生命力卻如此旺盛？不是因爲它容易長大，而是它有活下去的希望。 | 02, 雜->斬 |

situations.

# Chapter 5

# Conclusion and Future Work

Many avenues exist for future research and improvement of our system. For example, confusion sets can be automatically generated using Web-based character n-grams to improve correction performance. Part of speech tagging can be performed to provide more information for the noisy channel model. Named entities can be recognized in order to avoid false alarms. A supervised statistical classifier can be used to model channel probability more accurately. Additionally, an interesting direction to explore is using a Web corpus in addition to a compiled corpus for correcting typos. Yet another direction of research would be to consider errors related to a missing or redundant character, or collect data from user to update channel probabilities dynamically.

In summary, we have introduced a novel method for Chinese spell checking. In our approach, the channel model is trained based the sound and shape similarity using Web corpus, and the potential typos in a given sentence is corrected using a noisy channel model. In the training phase, we limit the confusable characters, retrieve the ngrams from the Web corpus, and correct ngrams and estimate the channel probability. At run-time, our system generate the correction candidates and calculate their probabilities using the language model and channel model from a given sentence. The results show that the proposed system achieves significantly better accuracy than our previous system. The experimental results prove that the channel probability we estimate for the noisy channel model are useful in Chinese spell checking.

# References

Chao-Huang Chang. A new approach for automatic chinese spelling correction. In *Proceedings of Natural Language Processing Pacific Rim Symposium*, volume 95, pages 278–283. Citeseer, 1995.

Yong-Zhi Chen and Shih-Hung Wu. Improve the detection of improperly used chinese characters with noisy channel model and detection template. Master's thesis, Chaoyang University of Technology, 2010.

Hsun-wen Chiu, Jian-cheng Wu, and Jason S Chang. Chinese spelling checker based on statistical machine translation. In *Sixth International Joint Conference on Natural Language Processing*, page 49, 2013.

Chuen-Min Huang, Mei-Chen Wu, and Ching-Che Chang. Error detection and correction based on chinese phonemic alphabet in chinese text. In *Modeling Decisions for Artificial Intelligence*, pages 463–476. Springer, 2007.

Ta-Hung Hung and Shih-Hung Wu. Automatic chinese character error detecting system based on n-gram language model and pragmatics knowledge base. Master's thesis, Chaoyang University of Technology, 2009.

Zhongye Jia, Peilu Wang, and Hai Zhao. Graph model for chinese spell checking. In *Sixth International Joint Conference on Natural Language Processing*, page 88, 2013.

C-L Liu, M-H Lai, K-W Tien, Y-H Chuang, S-H Wu, and C-Y Lee. Visually and phonologically similar characters in incorrect chinese words: Analyses, identification, and applications. *ACM Transactions on Asian Language Information Processing (TALIP)*, 10(2):10, 2011.

Wei-Yun Ma and Keh-Jiann Chen. Introduction to ckip chinese word segmentation system for the first international chinese word segmentation bakeoff. In *Proceedings of the second SIGHAN workshop on Chinese language processing-Volume 17*, pages 168–171. Association for Computational Linguistics, 2003.

Andreas Stolcke, Jing Zheng, Wen Wang, and Victor Abrash. Srilm at sixteen: Update and outlook. In *Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop*, page 5, 2011.

Shih-Hung Wu, Yong-Zhi Chen, Ping-che Yang, Tsun Ku, and Chao-Lin Liu. Reducing the false alarm rate of chinese character error detection and correction. In *Proceedings of CIPS-SIGHAN Joint Conference on Chinese Language Processing (CLP 2010)*, pages 54–61, 2010.

Jui-Feng Yeh, Sheng-Feng Li, Mei-Rong Wu, Wen-Yi Chen, and Mao-Chuan Su. Chinese word spelling correction based on n-gram ranked inverted index list. In *Sixth International Joint Conference on Natural Language Processing*, page 43, 2013.

Li Zhuang, Ta Bao, Xiaoyan Zhu, Chunheng Wang, and Satoshi Naoi. A chinese ocr spelling check approach based on statistical language models. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 5, pages 4727–4732. IEEE, 2004.