Computational Linguistics and Chinese Language Processing
Vol. 20, No. 1, June 2015, pp. 23-48

A Study on Chinese Spelling Check Using Confusion Sets and *N*-gram Statistics

Chuan-Jie Lin* and Wei-Cheng Chu*

Abstract

This paper proposes an automatic method to build a Chinese spelling check system. Confusion sets were expanded by using two language resources, Shuowen Jiezi and the Four-Corner codes, which improved the coverages of the confusion sets. Nine scoring functions which utilize the frequency data in the Google Ngram Datasets were proposed, where the idea of smoothing was also adopted. Thresholds were also decided in an automatic way. The final system achieved far better than our baseline system in CSC 2013 Evaluation Task.

Keywords: Chinese Spelling Check, Confusion Set Expansion, Google Ngram Scoring Function.

1. Introduction

Automatic spelling check is a basic and important technique in building NLP systems. It has been studied since 1960s as Blair (1960) and Damerau (1964) made the first attempt to solve the spelling error problem in English. Spelling errors in English can be grouped into two classes: non-word spelling errors and real-word spelling errors.

A non-word spelling error occurs when the written string cannot be found in a dictionary, such as in "*fly fron* Paris*". The typical approach is finding a list of candidates from a large dictionary by edit distance or phonetic similarity (Mitton, 1996; Deorowicz & Ciura, 2005; Carlson & Fette, 2007; Chen *et al*., 2007; Mitton, 2008; Whitelaw *et al*., 2009).

A real-word spelling error occurs when one word is mistakenly used for another word, such as in "*fly form* Paris*". Typical approaches include using confusion set (Golding & Roth, 1999; Carlson *et al*., 2001), contextual information (Verberne, 2002; Islam & Inkpen, 2009), and others (Pirinen & Linden, 2010; Amorim & Zampieri, 2013).

Spelling error problem in Chinese is quite different. Because there is no word delimiter

* Department of Computer Science and Engineering, National Taiwan Ocean University
No. 2, Pei-Ning Road, Keelung, 20224 Taiwan
E-mail: (cjlin, wcchu.cse)@ntou.edu.tw

in a Chinese sentence and almost every Chinese character can be considered as a one-character word, most of the errors are real-word errors.

Although that an illegal-character error can happen where writing by hand, i.e. the written symbol is not a legal Chinese character and thus not collected in a dictionary, such an error cannot happen in a digital document because only legal Chinese characters can be typed or shown in computer.

Spelling error problem in Chinese is defined as follows: given a sentence, find the locations of misused characters which result in wrong words, and propose the correct characters.

There have been many attempts to solve the spelling error problem in Chinese (Chang, 1994; Zhang *et al.*, 2000; Cucerzan & Brill, 2004; Li *et al*., 2006; Liu *et al*., 2008). Among them, lists of visually and phonologically similar characters play an important role in Chinese spelling check (Liu *et al*., 2011).

Two Chinese spelling check evaluation projects have been held: Chinese Spelling Check Evaluation at SIGHAN Bake-off 2013 (Wu *et al*., 2013) and CLP-2014 Chinese Spelling Check Evaluation (Yu *et al*., 2014), including error detection and error correction subtasks. The tasks are organized based on some research works (Wu *et al*., 2010; Chen *et al*., 2011; Liu *et al*., 2011). Our baseline system participated in both tasks. This paper describes an extended system based on Chinese Spelling Check (shorten as CSC tasks hereafter) 2013 and 2014 datasets.

This paper is organized as follows. Section 2 introduces our baseline system developed during Chinese Spelling Check Task 2013 and 2014. We sought new resources to expand confusion sets as described in Section 3. New scoring functions and threshold decision using Google Ngram frequencies to estimate the likelihood of passages were defined in Section 4. Section 5 shows experimental results with discussions and Section 6 concludes this paper.

2. Baseline System Description

2.1 System Architecture

Figure 1 shows the architecture of our Chinese spelling checking system. A sentence under consideration is first word-segmented. Candidates of spelling errors are replaced by similar characters one by one. The newly created sentences are word segmented again. They are sorted according to sentence generation probabilities measured by word or POS bigram model. If a replacement results in a better sentence, spelling error is reported.

In CSC tasks, the set of similar characters is called a confusion set. More information about confusion sets is given in Section 2.2.

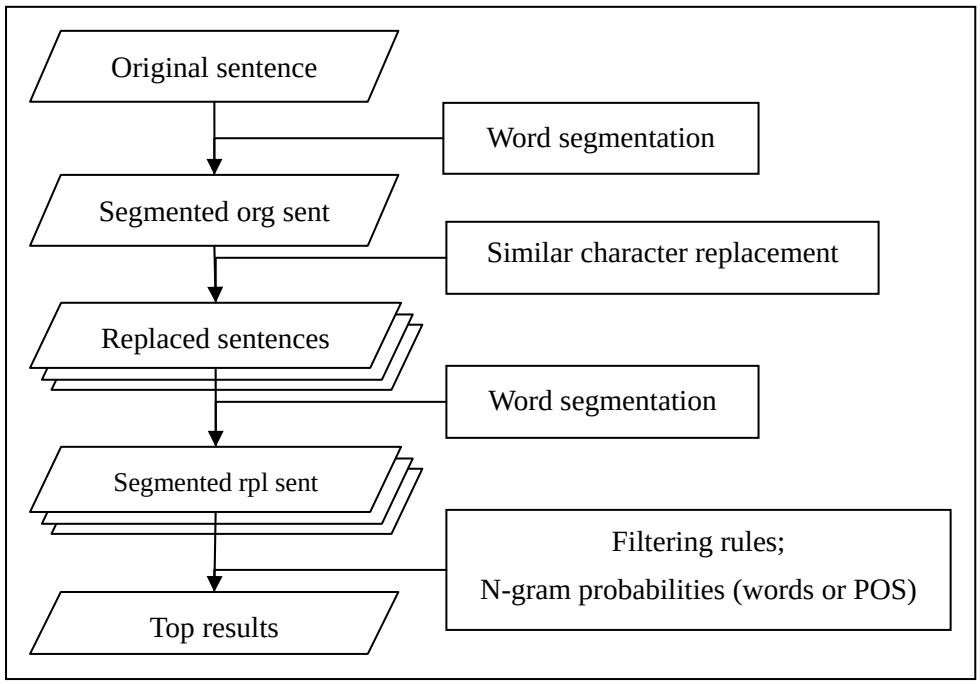

Figure 1. Architecture of NTOU Chinese Spelling Check System

There are two kinds of spelling-error candidates in our system: one-character words and two-character words. Their replacement procedures are different, as described in Section 2.3 and 2.4.

Section 2.5 introduced two rules for filtering out unlikely replacements. *N*-gram probability models in our baseline system are described in Section 2.6. The procedure to decide locations of errors is given in Section 2.7.

2.2 Confusion Sets

In SIGHAN7 Bake-off 2013 Chinese Spelling Check task, the organizers provided six kinds of confusion sets: 4 sets of phonologically similar characters and 2 sets of visually similar characters. The four sets of phonologically similar characters include characters with the same pronunciation in the same tone (同音同調, shorten as SPST hereafter), characters with the same pronunciation but in different tones (同音異調, shorten as SPDT hereafter), characters with similar pronunciations in the same tone (近音同調, shorten as DPST hereafter), and characters with similar pronunciations but in different tones (近音異調, shorten as DPDT hereafter). For example, phonologically similar characters to the character 情 (whose pronunciation is [qing2] and meaning is ‘feeling’) are:

SPST: 檠晴擎[qing2]

SPDT: 青卿蜻傾輕鯖氫清[qing1] 頃請[qing3] 慶罄磬[qing4]

DPST: 擒禽噙琴勤秦芹[qin2]

DPDT: 精經驚睛…京[jing1] 頸景警…井[jing3] 竟靜競徑鏡…敬[jing4]

今筋斤津…金[jin1] 僅儘錦緊…謹 [jin3] 近進勁盡禁…浸[jin4]

親侵欽嶔[qin1] 寢[qin3] 沁撳[qin4]

There are two confusion sets of visually-similar characters. The first one is the set of characters with the same radicals (部首) with the same number of strokes (筆劃) (同部首同筆畫數, shorten as RStrk hereafter). For example, the radical of the character 情 is 心 (shown as 忄 inside the character) with 11 strokes. Characters belonging to the radical 心 with 11 strokes are:

RStrk: 惋您悉惇惆悠患惦惚悼悽惘悸惟惜悻悴悵惠惕

The second visually-similar-character set collects characters with similar Cangjie codes (倉頡碼, shorten as CJie hereafter). Cangjie is a well-known code map of Chinese characters. Each Chinese character is encoded by a combination of at most 5 codes representing basic strokes in its visual structure. Characters who have similar Cangjie codes are likely visually similar. Liu *et al.* (2011) considered the information of surface structure and stroke similarity to create this confusion set. For example, the Cangjie code of the character 情 ([qing2], 'feeling') is PQMB, where "P 忄" denotes its radical part (忄) and "QMB キ一月" denotes its body part (青). So its similar characters are:

CJie:

清[EQMB] 晴[AQMB] 倩[OQMB] 猜[KHQMB] 睛[BUQMB]

靖[YTQMB] 精[FDQMB] 蜻[LIQMB] 鯖[NFQMB] 菁[TQMB]

請[YRQMB] 青[QMB] 債[OQMC] 漬[EQMC] 嘖[RQMC]

磧[MRQMC] 積[HDQMC] 績[VFQMC] 蹟[QMQMC] 責[QMBUC]

2.3 One-Character Word Replacement

After doing word segmentation on the original sentence, every one-character word is considered as candidate where error occurs. These candidates are one-by-one replaced by similar characters in their confusion sets to see if a new sentence is more acceptable.

Taking C1-1701-2 in the test set as an example. The original sentence is

...嬰兒個數卻特續下滑...

and it is segmented as

...嬰兒　個數　卻　特　續　下滑...

“卻”, “特” and “續” are one-character words so they are candidates of spelling errors. The confusion set of the character “卻” includes 腳欲叩卸... and the confusion set of the character “特” includes 持時恃峙侍... Replacing these one-character words with similar characters one-by-one will produce the following new sentences.

...嬰兒個數腳特續下滑...
...嬰兒個數欲特續下滑...
...嬰兒個數卻持續下滑... (*correct*)
...嬰兒個數卻時續下滑...
.....

(English meaning: 嬰兒 infant, 個數 number, 卻 but, 腳 foot, 欲 desire,
特 particular, 續 continue, 持續 keep, 時 time, 下滑 decrease)
(Original sentence: infant number but special continue decrease
‘*but the number of infants particularly continues to decrease*’)
(Correct sentence: 嬰兒個數卻持續下滑 ‘but the number of infants keeps decreasing’)

2.4 Two-Character Word Replacement

Our observation on the training sets finds that some errors occur in two-character words, which means that a string containing an incorrect character is also a legal word. Examples are “身手” ([shen1-shou3], ‘skills’) versus “生手” ([sheng1- shou3], ‘amateur’), and “人員” ([ren2-yuan2], ‘member’) vs. “人緣” ([ren2-yuan2], ‘relation’).

To handle such kinds of spelling errors, we created confusion sets for all known words by the following method. The resource for creating word-level confusion set is Academia Sinica Balanced Corpus (ASBC for short hereafter, cf. Chen *et al*., 1996).

For each word appearing in ASBC, each character in the word is substituted with its similar characters one by one. If a newly created word also appears in ASBC, it is collected into the confusion set of this word. Take the word “人員” as an example. After replacing “人” or “員” with their similar characters, new strings 仁員, 壬員, …, 人緣, and 人韻 are looked up in ASBC. Among them, only 人緣, 人猿, 人文, and 人俑 are legal words thus collected in 人員’s confusion set.

For each two-character word, if it has a confusion set, similar words in the set one-by-one substitute the original word to see if a new sentence is more acceptable.

Take ID=00058 in the Bakeoff 2013 CSC Datasets as an example. The original sentence is

... 在教室裡只要人員好...

and it is segmented as

... 在 教室 裡 只要 人員 好...

where “教室”, “只要”, and “人員” are multi-character words with confusion sets. By replacing 教室 with 教士, 教師…, replacing 只要 with 祇要, 只有, and replacing 人員 with 人緣, 人猿…, the following new sentences will be generated.

... 在教士裡只要人員好...
... 在教師裡只要人員好...
... 在教室裡祇要人員好...
... 在教室裡只要人緣好... (*correct*)
... 在教室裡只要人猿好...

(English meaning: 在 in, 教室 classroom, 教士 priest, 教師 teacher,
裡 inside, 只要 as-long-as, 祇要 as-long-as (*variant*),
人員 member, 人緣 relations, 人猿 ape, 好 good)
(Original Sentence: in classroom inside as-long-as member good
‘*as long as there are good members in the classroom…*’)
(Correct sentence: 在教室裡只要人緣好 ‘in the classroom, as long as you have good relations with the others...’)

2.5 Filtering Rules

Two filter rules are applied before error detection in order to discard apparently incorrect replacements. The rules are defined as follows.

Rule 1: No error in person names

If a replacement results in a person name, discard it. Our word segmentation system performs named entity recognition at the same time. If the replacing similar character can be considered as a Chinese family name, the consequent characters might be merged into a person name. As most of the spelling errors do not occur in personal names, we simply ignore these replacements. Take C1-1701-2 as an example:

...每 位 產 齡 婦女...

(every QF pregnancy age woman '*every woman in the age of pregnancy*')

"魏" is phonologically similar to "位" and is a Chinese family name. The newly created sentence is segmented as

...每 魏產齡(PERSON) 婦女...

(every Chan-Ling Wei woman: *nonsense*)

where "魏產齡" is recognized as a person name so this replacement is discarded.

Rule 2: Stopword filtering

For the one-character replacement, if the replaced (original) character is a personal anaphora (你 'you' 我 'I' 他 'he/she') or numbers from 1 to 10 (一二三四五六七八九十), discard the replacement. We assume that a writer seldom misspell such words. Take B1-0122-2 as an example:

...我 會 在 二 號 出口 等 你...

(I will at two number exit wait you '*I will wait for you at Exit No. 2*')

Although "二" is a one-character word, it is in our stoplist therefore no replacement is performed on this word.

2.6 *N*-Gram Probabilities

A basic hypothesis is that a correct replacement will generate a "better" sentence which has higher probability than the original one.

The likelihood of a passage being understandable can be estimated as sentence generation probability by language models. We tried smoothed word-unigram, word-bigram, and POS-bigram models in our baseline system. The training corpus used to build language models is ASBC. As usual, we use log probabilities instead.

Besides applying rules in which the probabilities were compared directly, we also treated them as features to train a SVM classifier which guessed whether a replacement was correct or not.

2.7 Error Detection

In our system, error detection and correction greatly rely on sentence generation probabilities. Therefore, all the newly created sentences should also be word segmented. If a new sentence results in a better word segmentation, it is very likely that the original character is misused and this replacement is correct. But if no replacement is better than the original sentence, it is reported as "no error".

The detail of our error detection algorithm is delivered here. The original sentence is first divided into several sub-sentences by six sentence-delimiting punctuation marks: comma, period, exclamation, question mark, colon, and semicolon. The following steps are performed on each sub-sentence, referred to as ***original passage*** hereafter.

1. Divide the original sentence into several passages by the sentence-delimiting punctuation marks
2. Perform word segmentation on the original passages
3. Measure the likelihood of the original passages by language models
4. For each one-character word in each original passage
   (1) Skip the word if it is a person name or a stopword (filtering rules)
   (2) Replace the word with its similar characters in the confusion sets to generate un-segmented passages, one new passage for one similar character
   (3) Perform word segmentation on the new passages
5. For each two-character word in each original passage
   (1) If the word appears in the two-character confusion set, replace the word with its similar words in the two-character confusion sets to generate un-segmented passages, one new passage for one similar word
   (2) Perform word segmentation on the new passages

6. Measure the likelihood of the new passages from step 4 and 5 by language models
7. If no new passage has a higher score than its original passage, report “no error” in this original passage
8. Consider only the new passage with the highest score
   (1) If its score comparing to the original one is not higher than a pre-defined threshold, report “no error” in this original passage
   (2) Otherwise, report the location and the similar character (or locations of similar characters in a two-syllable similar word) of the replacement which generates this new passage

3. Confusion Set Expansion

In our experience, the confusion sets provided by the task organizers do not cover all the errors. The error coverage of the confusion sets is depicted in Table 1, where TR means training set and TS means test set. The first 9 rows show the coverage of each confusion set, where set 0 to set 5 have been explained in Section 2.2. We can see that the SPST confusion set alone covers 70% of the errors in CSC 2013 datasets but only about half of the errors in CSC 2014 datasets. The second important confusion set is CJie, which covers 30% to 40% of the errors.

The last 10 rows of Table 1 show the coverage of the unions of confusion sets. The union of set 0~5 covers 94.59% of the errors. The union of set 0~3+5 has the same coverage as the union of set 0~5, which suggests that RStrk can be ignored.

In order to achieve better coverage, we used two resources to expand the confusion sets. One is Shuowen Jiezi and the other is the Four-Corner Encoding System.

Table 1. Error Coverage of Confusion Sets (%)

Confusion Set	TR2013	TS2013	TR2014	TS2014
set0: SPST	70.09	72.13	47.92	47.41
set1: SPDT	15.10	17.50	46.52	47.03
set2: DPST	3.70	4.99	5.15	4.68
set3: DPDT	3.70	4.67	8.41	7.71
set4: RStrk	9.12	3.17	0.38	0.88
set5: CJie	40.46	36.18	29.72	31.10
set6: Cor4	14.81	6.89	1.84	1.52
set7: SWen1	17.09	19.24	11.48	12.64
set8: SWen2	18.23	19.64	11.91	12.90

set0+1	74.93	78.23	71.89	72.57
set0+1+2	78.35	83.06	76.55	76.61
set0+…+3	79.20	83.85	81.55	82.05
set0+…+4	87.75	86.94	81.76	82.30
set0+…+5	94.59	93.27	83.86	84.58
set0+…+6	96.01	93.67	84.22	84.70
set0+…+7	97.15	94.54	84.58	85.59
set0+…+8	97.15	94.54	**84.60**	85.59
set0+1+2+3+5	94.59	93.27	83.86	84.58
set0+1+2+3+5+7	**97.15**	**94.54**	**84.58**	**85.59**

3.1 Confusion Set from Shuowen Jiezi

Shuowen Jiezi[1] (說文解字) is a dictionary of Chinese characters. Xu Shen (許慎), author of this dictionary, analyzed the characters according to the six lexicographical categories (六書). One major category is phono-semantic compound characters (形聲), which were created by combining a radical (形符) with a phonetic component (聲符). Characters with same phonetic components were collected to expand confusion sets, because they are by definition phonologically and visually similar. For example, the following characters share the same phonetic component “寺” ([si4], ‘temple’) thus become confusion candidates (their actual pronunciation are given in brackets):

SWen: 侍[si4]持[chi2]恃[shi4]特[te4]時[shi2]...

It happens a phonetic component might not be atomic, which means it also has its own phonetic component. For example, 潔’s phonetic component is 絜, but 絜’s phonetic component is 丰. We tried two creation methods. The first one was created by collecting characters with the same phonetic component (referred to as SWen1), and the second one was the closure of SWen1 (referred to as SWen2).

Set 7 and 8 in Table 1 represent SWen1 and SWen2. Although they alone do not provide good coverage, unions including SWen sets can cover up to 97.15% errors in CSC 2013 Training set.

Closure set only cover one more error in CSC 2014 Training set. In order not to introduce too much noise, the closure SWen set is not recommended.

[1] http://zh.wikisource.org/wiki/說文解字

3.2 Confusion Set from the Four-Corner System

The Four-Corner System[2] (四角號碼) is an encoding system for Chinese characters. Digits 0~9 represent some typical shapes in character strokes. A Chinese character is encoded into 4 digits which represent the shapes found in its 4 corners. We collect characters in the same Four-Corner codes to expand confusion sets, because they are by definition visually similar. For example, the following characters are all encoded as 6080 in the Four-Corner System (shorten as Cor4 hereafter):

Cor4: 只囚貝足叟是員異買圓圚

Set 6 in Table 1 represents Cor4. Unfortunately unions including Cor4 do not cover more errors than set0~3+5+7. It is hard to say if The Four-Corner System is helpful or not.

3.3 Two-Character Confusion Set Expansion

To make a larger two-character confusion set, unigrams in the Chinese Google Ngram dataset were used instead of ASBC. But some issues should be handles before dataset creation, which are discussed in Section 3.3.1.

3.3.1 Google Ngram Dataset Preprosessing

Chinese Web 5-gram[3] is real data released by Google Inc. who collected from all webpages in the World Wide Web which are unigram to 5-grams. Frequencies of these ngrams are also provided. Some examples from the Chinese Web 5-gram dataset are given here:

Unigram

稀释剂 321928 (‘thinner’ in Simplified Chinese)

稀釋劑 17260 (‘thinner’ in Traditional Chinese)

Bigram

蒸发量 超过 869 (‘the-amount-of-evaporation has-exceeded’ in SC)

蒸發量 超過 69 (‘the-amount-of-evaporation has-exceeded’ in TC)

Trigram

能量 远 低于 727 (‘energy far lower-than’ in SC)

能量 遠 低於 113 (‘energy far lower-than’ in TC)

[2] 四角號碼列表 http://code.web.idv.hk/misc/four.php

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

4-gram

张贴 色情 图片 或 116 (‘posting pornographic images or’ in SC)

張貼 色情 圖片 或 73 (‘posting pornographic images or’ in TC)

5-gram

幸好 我们 发现 得 早 182 (‘fortunately we found-it DE early’ in SC)

幸好 我們 發現 得 早 155 (‘fortunately we found-it DE early’ in TC)

There are several issues with regard to using the Chinese Web 5-gram dataset in this task. First, the Chinese Web 5-gram dataset includes both Traditional and Simplified Chinese ngrams, but our experimental datasets are written in Traditional Chinese. To make full use of this dataset, we decide to translate every Simplified Chinese words into Traditional Chinese. Our translation method was simply table-lookup on the Simplified-to-Traditional Chinese word mappings provided by Wikipedia[4]. Note that the translation may not be perfect.

After translation, some ngrams become identical, such as 電視 and 电视 (‘television’) and all the Chinese Google Ngrams shown in the previous examples. Identical words are combined into one entry and their frequencies are merged.

3.3.2 Confusion Set Expansion by Google Ngram

The two-character confusion set in our baseline system was trained from ASBC. We tried to use unigram set in the Chinese Web 5-gram dataset to create a larger two-character confusion set.

The procedure is the same as in the baseline system development: collect all the two-character words in the Chinese Web unigram set, replace each character by its similar characters, collect all the new strings which also appear in the Chinese Web unigram set as the original word’s two-character confusion set.

In CSC 2014 training data, there are cases that both characters in a two-character word are misused, such as 也是 ([ye3-shi4], ‘also’) vs. 夜市 ([ye4-shi4], ‘night market’). We also performed such kind of replacement and collected legal similar words into the two-character confusion set.

4. Passage Likelihood Scoring

In CSC tasks held in 2013 and 2014, we tried bigram probability model to predict errors in sentences. The language generation model was trained from Academia Sinica Balanced

[4] http://zh.wikipedia.org/wiki/Wikipedia:繁簡處理

Corpus. We found the volume and vocabulary of ASBC was not large enough. So we turn to use Chinese Google Ngram dataset.

4.1 Ngram Scoring Functions

Given a sentence (word-segmented, with or without errors) $S = \{w_1, w_2, \cdots w_m\}$, let $Gram(S, n)$ be the set of all n-grams containing in the sentence S, i.e. $Gram(S, n) = \{(w_i, w_{i+1}, \cdots w_{i+n-1}) | 1 \le i \le m-n+1\}$. We define ***Google Ngram Frequency*** *gnf(g)* of a n-gram to be its frequency count provided in the Chinese Web 5-gram dataset. If it does not appear in that dataset, its value is defined as 0.

Five scoring functions $GS_*(S)$ were used to measure the likelihood of a sentence. Equation 1 is the definition of ***raw frequency score*** $GS_{raw}(S)$ which sums up the frequencies of all n-grams. Equation 2 and 3 give the definitions of ***log frequency score*** $GS_{logn}(S, n)$ and $GS_{log}(S)$ which sums up the logarithm of frequencies of all n-grams. Because large frequency tends to dominate the scores and then leads to bias, hopefully logarithm values can provide a moderate scoring. Note that we skip the ngrams which do not appear in the Chinese Web 5-gram dataset when calculating the log frequency score (or in another word, its log score is set to be 0).

$$GS_{raw}(S) = \sum_{n=2}^{5} \left(\sum_{g \in Gram(S,n)} gnf(g) \right) \tag{1}$$

$$GS_{\log n}(S,n) = \sum_{g \in Gram(S,n)} \log\left(gnf(g)\right) \tag{2}$$

$$GS_{log}(S) = \sum_{n=2}^{5} GS_{\log n}(S,n) \tag{3}$$

It is obvious that matching of a higher gram is more welcome than of a lower gram. To favor higher grams, we define the third scoring function ***length-weighted log frequency score*** $GS_{len}(S)$ which multiplies the log frequency score with n.

$$GS_{len}(S) = \sum_{n=2}^{5} \left(n \times \sum_{g \in Gram(S,n)} \log\left(gnf(g)\right) \right) \tag{4}$$

We further tried two average scores where scores of the same n are averaged before summation. Equation 5 and 6 illustrate the logarithm and length-weighted versions, respectively.

$$GS_{\log av}(S) = \sum_{n=2}^{5}\left(\frac{1}{|Gram(S,n)|} \times \sum_{g \in Gram(S,n)} \log(gnf(g))\right) \quad (5)$$

$$GS_{lenav}(S) = \sum_{n=2}^{5}\left(\frac{n}{|Gram(S,n)|} \times \sum_{g \in Gram(S,n)} \log(gnf(g))\right) \quad (6)$$

We also tried a smoothing-like function to handle zero frequency. If a ngram does not appear in the Chinese Web 5-gram dataset, its log score is set to a negative constant ε. The ***smoothed log frequency score*** *gnf'*(*g*) is defined as Equation 6.

$$gnf'(g) = \begin{cases} \text{if } gnf(g) = 0 & \varepsilon \\ \text{otherwise} & \log(gnf(g)) \end{cases} \quad (7)$$

Figure 1 demonstrates the detailed information and steps of compute the values of two of the scoring functions, log frequency score and length-weighted log frequency score, with or without smoothing, by using the first passage of B1-0143-1 as an example. As we can see, the smoothed length-weighted log frequency score can successfully identify the correct answer.

4.2 Threshold Learning

A replacement is considered to be "correct" if the score of the generated new passage is higher than the original's to a certain degree. As described in Section 2.7, a pre-defined threshold is used to ensure that the new passage is far better than the original passage.

In CSC 2013 and 2014, this threshold was set by consulting classification rules learned by decision tree. In this paper, we try to observe the efficiency of thresholds in a more systematical way as follows.

Two kinds of thresholds were considered. The first one is for the score difference of the scores of the new passage and the original passage. Because the new passage must have a higher score than the original one, this value is always positive. The second one is for the ratio of the score difference to the original passage's score. Because scores may be negative, we take its absolute value instead, i.e.

$$| (score_{new} - score_{org}) / score_{org} |.$$

B1-0143-1 妳還記得我們在高中在已樣的課嗎

Org, Segmented: 妳 還 記得 我們 在 高中 在 已 樣 的 課 嗎

Rpl1, 妳→你, Segmented: 你 還 記得 我們 在 高中 在 已 樣 的 課 嗎

Rpl2, 樣→聽, Segmented: 妳 還 記得 我們 在 高中 在 已 聽 的 課 嗎

Rpl3, 已→一, Segmented: 妳 還 記得 我們 在 高中 在 一樣 的 課 嗎

(English meanings: 妳 you(female), 你 you, 還 still, 記得 remember, 我們 we, 在 in, 高中 high-school, 已 already, 樣 pattern, 聽 listen, 一樣 same, 的 *DE*, 課 class, 嗎 *Qpunc*)

(Org:'*Do you still remember that we were in the patterned class in high school?*')

(Rpl1:'*Do you still remember that we were in the patterned class in high school?*')

(Rpl2: '*Do you still remember that we were in the listened class in high school?*')

(Rpl3: '*Do you still remember that we were in the same class in high school?*')

Google Ngram Information:

Bigram	*gnf*	log	Trigram	*gnf*	log
妳 還	337282	12.729	妳 還 記得	22344	10.014
你 還	27319449	17.123	你 還 記得	1127456	13.935
還 記得	8552177	15.962	還 記得 我們	264628	12.486
記得 我們	756252	13.536	記得 我們 在	40942	10.620
我們 在	24371694	17.009	在 高中 在	843	6.737
在 高中	838050	13.639	在 已 聽	61	4.111
高中 在	100156	11.514	在 一樣 的	19422	9.874
在 已	1193110	13.992	已 聽 的	1991	7.596
在 一樣	41218	10.627	聽 的 課	8342	9.029
已 樣	1025	6.932	Trigram with *gnf*(.)=0		
已 聽	121888	11.710	我們 在 高中, 高中 在 已, 高中 在 一樣, 在 已 樣, 已 樣 的, 樣 的 課, 一樣 的 課, 的 課 嗎		
樣 的	3280256	15.003			
聽 的	5830567	15.579			
一樣 的	35523054	17.386			
的 課	2695074	14.807			
課 嗎	0	---			

4-gram with *gnf*(.) > 0	*gnf*	log	5-gram with *gnf*(.) > 0	*gnf*	log
妳 還 記得 我們	896	6.798	你 還 記得 我們 在	2846	7.954
你 還 記得 我們	43508	10.680	還 記得 我們 在 高中	78	4.357
還 記得 我們 在	16260	9.696			
記得 我們 在 高中	238	5.472			

Figure 1. (a) Examples of Google Ngram Information in Scoring

List of scores

	GS_{log}	GS_{len}	GS'_{log}	GS'_{len}
Org	201.304	499.469	1.304	-290.531
Rpl1	221.456	**575.321**	31.456	-164.679
Rpl2	**227.394**	572.386	**57.394**	-127.614
Rpl3 (*correct*)	203.263	513.261	43.263	**-126.739**

Scoring details:

GS_{Log}(Org) = (log(*gnf*(妳 還)) + log(*gnf*(還 記得)) +…+ log(*gnf*(課 嗎)) +
(log(*gnf*(妳 還 記得)) +…+ log(*gnf*(的 課 嗎)) +
(log(*gnf*(妳 還 記得 我們)) +…+ log(*gnf*(樣 的 課 嗎)) +
(log(*gnf*(妳 還 記得 我們 在)) +…+ log(*gnf*(已 樣 的 課 嗎))
= 12.729 + 15.962 + 13.536 +…+ 15.003 + 14.807 + 0 +
10.014 + 12.486 + 10.620 +…+ 0 + 0 +
6.798 + 9.696 + 5.472 + 0 +…+ 0 + 0 +
0 + 4.357 + 0 +…+ 0
= 135.124+39.857+21.967+4.357 = 201.304

GS_{Log}(Rpl1) = (log(*gnf*(你 還)) + log(*gnf*(還 記得))+…+ log(*gnf*(課 嗎)) +
(log(*gnf*(你 還 記得)) +…+ log(*gnf*(的 課 嗎))+
(log(*gnf*(你 還 記得 我們)) +…+ log(*gnf*(樣 的 課 嗎)) +
(log(*gnf*(你 還 記得 我們 在)) +…+ log(*gnf*(已 樣 的 課 嗎))
= 139.518 + 43.778 + 25.849 + 12.310 = 221.456

GS_{Log}(Rpl2) = 140.477 + 60.594 + 21.967 + 4.358 = 227.394

GS_{Log}(Rpl3) = 127.208 + 49.731 + 21.967 + 4.358 = 203.263

GS_{Len}(Org) = 135.124 × 2 + 39.857 × 3 + 21.967 × 4 + 4.357 × 5 = 499.469

GS_{Len}(Rpl1) = 139.518 × 2 + 43.778 × 3 + 25.849 × 4 + 12.310 × 5 = 575.321

GS_{Len}(Rpl2) = 140.477 × 2 + 60.594 × 3 + 21.967 × 4 + 4.358 × 5 = 572.386

GS_{Len}(Rpl3) = 127.208 × 2 + 49.731 × 3 + 21.967 × 4 + 4.358 × 5 = 513.261

GS'_{Log}(Org) = 135.124 - 10 + 39.857 - 10 × 6 + 21.967 - 10 × 6 + 4.357 - 10 × 7
(1 bigram, 6 trigrams, 6 fourgrams, and 7 fivegrams with *gnf*(.) = 0)
= 125.124 - 20.143 - 38.033 - 65.643 = 1.304

GS'_{Log}(Rpl1) = 139.518 - 10 + 43.778 - 10 × 6 + 25.849 - 10 × 6 + 12.310 - 10 × 6
= 129.518 - 16.222 - 34.151 - 47.690 = 31.456

GS'_{Log}(Rpl2) = 140.477 - 10 + 60.594 - 10 × 3 + 21.967 - 10 × 6 + 4.358 - 10 × 7
= 130.477 + 30.594 - 38.033 - 65.642 = 57.394

GS'_{Log}(Rpl3) = 127.208 - 10 + 49.731 - 10 × 4 + 21.967 - 10 × 5 + 4.358 - 10 × 6
= 117.208 + 9.731 - 28.033 - 55.642 = 43.263

GS'_{Len}(Org) = 125.124 × 2 - 20.143 × 3 - 38.033 × 4 - 65.643 × 5 = -290.531

GS'_{Len}(Rpl1) = 129.518 × 2 - 16.222 × 3 - 34.151 × 4 - 47.690 × 5 = -164.679

GS'_{Len}(Rpl2) = 130.477 × 2 + 30.594 × 3 - 38.033 × 4 - 65.642 × 5 = -127.614

GS'_{Len}(Rpl3) = 117.208 × 2 + 9.731 × 3 - 28.033 × 4 - 55.642 × 5 = -126.739

Figure 1. (b) Details of Scoring Steps

A threshold is trained in the steps as follows. Under a scoring function, all replacements are sorted according to the score difference (or ratio). Largest values are ranked higher. Since each replacement is known to be "correct" or "incorrect", precision, recall, and F-score at each rank can be decided. Choose the difference (or ratio) which achieves the highest F-score as the threshold.

Best F-scores under different scoring functions, smoothing strategies, and training data are shown in Table 2(a) and 2(b), where the first columns represent scoring functions introduced in Section 4.1. Meanings of labels in the second rows are as follows:

OL: no smoothing, at most one error report at one location

OP: no smoothing, at most one error report at one passage

ML: smoothing, at most one error report at one location

MP: smoothing, at most one error report at one passage

Table 2. Best F-Scores Achieved by Threshold Tuning

(a) Threshold Tuning on CSC 2013 Training Set

	Difference				Ratio			
F-score	OL	OP	ML	MP	OL	OP	ML	MP
GS_{raw}	**3.23**	**3.23**	---	---	**3.39**	2.36	---	---
$GS_{logn}(2)$	0.43	0.43	1.11	**1.18**	0.55	0.61	0.76	**0.94**
$GS_{logn}(3)$	10.74	10.27	**22.25**	22.22	6.18	7.49	12.68	**17.09**
$GS_{logn}(4)$	15.16	15.28	**33.81**	33.12	10.85	12.09	17.85	**19.59**
$GS_{logn}(5)$	10.28	9.63	21.38	**21.96**	9.79	9.66	11.50	**13.02**
GS_{log}	6.67	6.74	33.78	**35.78**	3.36	4.19	20.69	**25.87**
GS_{logav}	26.60	28.25	30.92	**33.16**	20.32	25.62	24.58	**30.35**
GS_{len}	9.93	9.86	42.75	**44.06**	4.83	5.50	25.52	**31.34**
GS_{lenav}	27.38	28.34	30.06	**33.74**	19.53	24.51	26.05	**29.34**

(b) Threshold Tuning on CSC 2014 Training Set

	Difference				Ratio			
F-score	OL	OP	ML	MP	OL	OP	ML	MP
GS_{raw}	**3.31**	2.82	---	---	**3.08**	2.73	---	---
$GS_{logn}(2)$	1.50	0.94	**1.62**	1.07	**1.52**	0.85	**1.52**	0.89
$GS_{logn}(3)$	7.17	6.84	**10.61**	9.66	5.81	6.13	7.71	**8.75**
$GS_{logn}(4)$	10.82	10.90	14.31	**14.43**	10.14	**11.65**	10.72	11.41
$GS_{logn}(5)$	7.89	7.73	**8.73**	8.35	**9.44**	9.08	6.32	5.38
GS_{log}	6.20	5.99	**17.19**	16.56	3.99	4.35	12.20	**14.03**
GS_{logav}	13.86	15.13	13.98	**15.79**	13.03	15.12	13.38	**15.65**
GS_{len}	7.98	7.66	**22.07**	21.65	5.04	5.65	14.60	**16.93**
GS_{lenav}	14.03	15.35	14.11	**15.84**	12.91	15.14	13.61	**15.52**

As we can see in Table 2, smoothing and logarithm did improve the performance. Using thresholds of score differences was better than using thresholds of ratios. Among the 9 scoring functions, length-weighted log frequency score GS_{len} outperformed other functions. However, averaging at each *n* level harmed the performance.

To our surprise, bigram model $GS_{logn}(2)$ was not very useful. However, 4-gram model $GS_{logn}(4)$ alone could achieve pretty good performance. Moreover, the characteristics of CSC 2013 training set and CSC 2014 training set are quite different. F-cores on CSC 2014 data sets were much lower.

5. Experiments

5.1 Datasets

Four benchmarks are used to evaluate our systems: the training set and test set in Chinese Spelling Check Evaluation at SIGHAN Bake-off 2013 (Wu *et al*., 2013), and the training set and test set in CLP-2014 Chinese Spelling Check Evaluation (Yu *et al*., 2014). They are referred to as CSC 2013 and 2014 datasets in this paper. Number of topics and errors containing in these datasets are listed in Table 3.

Table 3. Number of Topics and Errors in CSC 2013 and 2014 Datasets

Dataset	#Topics	#Errors
CSC 2013 Training	350	351
CSC 2013 Test	1000	1464
CSC 2014 Training	3434	5280
CSC 2014 Test	531	791

5.2 Evaluation Metrics

There are two subtasks in CSC Task: error detection and error correction. Error detection subtask evaluates the correctness of detected error locations. Error correction subtask evaluates the correctness of locations and proposed corrections.

The metrics are evaluated in both levels by the following metrics:

Accuracy = (TP + TN) / (TP + TN + FP + FN)

Precision = TP / (TP + FP)

Recall = TP / (TP+ FN)

F1-Score = 2 * Precision * Recall / (Precision + Recall)

Note that the unit of "correctness" is topic. It only counts the topics whose errors are all successfully corrected with no false alarm.

5.3 Experimental Results

All combinations of system settings have been evaluated on all the datasets. Table 4 shows the runs achieving the best F1-scores according to each subtask, dataset, and scoring functions. The labels of system settings are defined as follows (cf. Section 3.2):

Ranking and threshold setting

diff: ranking by the score difference

ratio: ranking by the score ratio

Smoothing Strategy

O: no smoothing

M: smoothing

Detection unit

N: at most one error in one topic, no threshold

Q: at most one error in one topic, filtered by threshold

P: at most one error in one passage, filtered by threshold

L: at most one error at each location, filtered by threshold

More precisely, Table 4(a)~4(d) shows the experimental results of ***error detection*** evaluated on CSC 2013 training set, CSC 2013 test set, CSC 2014 training set, and CSC 2014 test set, respectively. Table 4(e)~4(h) shows the experimental results of ***error correction*** evaluated on CSC 2013 training set, CSC 2013 test set, CSC 2014 training set, and CSC 2014 test set, respectively.

Almost all results support similar conclusions as we made in Section 4.2: the best system uses the smoothed length-weighted log frequency score, ranking by score differences without threshold (GS_{len},diff,M,N). Thresholds are not helpful except on CSC 2014 test set.

Table 4. Experimental Results on CSC2013 and 2014 Datasets

(a) Error-Detection, CSC2013 Training Set

Scoring	System	P	R	F	Acc
GS_{raw}	ratio,O,N	100.00	7.71	14.32	7.71
GS_{logn}(2)	diff,M,N	100.00	9.71	17.71	9.71
GS_{logn}(3)	diff,M,N	100.00	30.00	46.15	30.00
GS_{logn}(4)	diff,M,N	100.00	30.00	46.15	30.00
GS_{logn}(5)	diff,M,N	100.00	18.57	31.33	18.57
GS_{log}	diff,M,N	100.00	42.00	59.15	42.00
GS_{logav}	diff,M,N	100.00	37.71	54.77	37.71
GS_{len}	diff,M,N	100.00	46.57	**63.55**	**46.57**
GS_{lenav}	diff,M,N	100.00	36.00	52.94	36.00

(b) Error-Detection, CSC2013 Test Set

Scoring	System	P	R	F	Acc
GS_{raw}	ratio,O,N	100.00	4.80	9.16	4.80
GS_{logn}(2)	diff,M,N	100.00	5.10	9.71	5.10
GS_{logn}(3)	diff,M,N	100.00	18.40	31.08	18.40
GS_{logn}(4)	diff,M,N	100.00	18.20	30.80	18.20
GS_{logn}(5)	diff,M,Q	100.00	11.90	21.27	11.90
GS_{log}	diff,M,N	100.00	25.90	41.14	25.90

GS_{logav}	diff,M,N	100.00	24.80	39.74	24.80
GS_{len}	diff,M,N	100.00	28.80	**44.72**	**28.80**
GS_{lenav}	diff,M,N	100.00	25.00	40.00	25.00

(c) Error-Detection, CSC2014 Training Set

Scoring	System	P	R	F	Acc
GS_{raw}	ratio,M,N	98.21	4.80	9.16	4.80
GS_{logn}(2)	diff,M,N	97.22	3.06	5.93	3.05
GS_{logn}(3)	diff,M,N	99.31	12.64	22.42	12.63
GS_{logn}(4)	diff,M,N	99.38	13.98	24.51	13.97
GS_{logn}(5)	diff,M,N	98.72	6.73	12.60	6.72
GS_{log}	diff,M,N	99.52	18.29	30.90	18.27
GS_{logav}	diff,M,N	99.47	16.37	28.11	16.35
GS_{len}	diff,M,N	99.59	21.40	**35.23**	**21.38**
GS_{lenav}	diff,M,N	99.46	15.96	27.50	15.94

(d) Error-Detection, CSC2014 Test Set

Scoring	System	P	R	F	Acc
GS_{raw}	ratio,M,Q	5.40	5.46	5.43	4.90
GS_{logn}(2)	diff,M,Q	6.45	3.01	4.11	29.66
GS_{logn}(3)	diff,M,Q	17.28	9.79	12.50	31.45
GS_{logn}(4)	diff,M,Q	14.88	14.88	14.88	14.88
GS_{logn}(5)	diff,M,Q	8.85	8.85	8.85	8.85
GS_{log}	diff,M,N	17.94	20.72	19.23	12.99
GS_{logav}	ratio,M,Q	19.21	18.27	18.73	20.72
GS_{len}	diff,M,Q	25.63	19.21	**21.96**	**31.73**
GS_{lenav}	diff,M,Q	19.63	17.89	18.72	22.32

(e) Error-Correction, CSC2013 Training Set

Scoring	System	P	R	F	Acc
GS_{raw}	diff,O,N	100.00	2.86	5.56	2.86
GS_{logn}(2)	diff,M,L	100.00	0.86	1.70	0.86

$GS_{logn}(3)$	diff,M,N	100.00	20.29	33.73	20.29
$GS_{logn}(4)$	diff,M,N	100.00	23.71	38.34	23.71
$GS_{logn}(5)$	diff,M,N	100.00	15.71	27.16	15.71
GS_{log}	diff,M,N	100.00	32.57	49.14	32.57
GS_{logav}	diff,M,N	100.00	30.57	46.83	30.57
GS_{len}	diff,M,N	100.00	41.71	**58.87**	**41.71**
GS_{lenav}	diff,M,N	100.00	30.57	46.83	30.57

(f) Error- Correction, CSC2013 Test Set

Scoring	System	P	R	F	Acc
GS_{raw}	ratio,O,N	100.00	0.90	1.78	0.90
$GS_{logn}(2)$	ratio,M,N	100.00	0.50	1.00	0.50
$GS_{logn}(3)$	diff,M,N	100.00	12.50	22.22	12.50
$GS_{logn}(4)$	diff,M,N	100.00	14.80	25.78	14.80
$GS_{logn}(5)$	diff,M,Q	100.00	10.00	18.18	10.00
GS_{log}	diff,M,N	100.00	19.20	32.21	19.20
GS_{logav}	diff,M,N	100.00	20.10	33.47	20.10
GS_{len}	diff,M,N	100.00	23.60	**38.19**	**23.60**
GS_{lenav}	diff,M,N	100.00	20.70	34.30	20.70

(g) Error- Correction, CSC2014 Training Set

Scoring	System	P	R	F	Acc
GS_{raw}	diff,O,N	95.38	1.81	3.54	1.80
$GS_{logn}(2)$	ratio,M,N	83.33	0.44	0.87	0.44
$GS_{logn}(3)$	diff,M,N	98.68	6.52	12.24	6.52
$GS_{logn}(4)$	diff,M,N	99.10	9.61	17.52	9.60
$GS_{logn}(5)$	diff,M,N	98.13	4.57	8.74	4.57
GS_{log}	diff,M,N	99.26	11.76	21.04	11.75
GS_{logav}	diff,M,N	99.21	11.04	19.86	11.03
GS_{len}	diff,M,N	99.42	15.03	**26.11**	**15.01**
GS_{lenav}	diff,M,N	99.22	11.12	20.01	11.11

(h) Error- Correction, CSC2014 Test Set

Scoring	System	P	R	F	Acc
GS_{raw}	ratio,O,Q	2.90	2.82	2.86	4.05
$GS_{logn}(2)$	diff,M,Q	1.28	0.56	0.78	28.44
$GS_{logn}(3)$	diff,M,Q	11.39	6.03	7.88	29.57
$GS_{logn}(4)$	diff,M,Q	11.55	11.11	11.32	12.99
$GS_{logn}(5)$	diff,M,Q	6.20	6.03	6.11	7.44
GS_{log}	diff,M,P	14.75	8.47	10.77	29.76
GS_{logav}	diff,M,Q	15.03	12.43	13.61	21.09
GS_{len}	diff,M,Q	21.28	15.07	**17.64**	**29.66**
GS_{lenav}	diff,M,Q	15.62	13.56	14.52	20.15

By observing the text in the benchmarks, it seems that the sentences in CSC 2014 datasets were written by non-Chinese-native speakers. It means that (1) even the corrected sentences may not be natural enough, so ngram model cannot predict successfully; (2) some errors are so common that appear in many sentences, so hand-crafted rules may be more successful.

6. Conclusion

In this paper, we proposed two resources to expand confusion sets which improved the error coverage up to 97.17% in CSC training set. We also proposed a method to build a larger two-character confusion set. Nine scoring functions using Google Ngram frequency information were also introduced. Among them, length- weighted log frequency score greatly improved our baseline system on CSC 2013 datasets.

Although that the methods proposed in this paper do not perform well enough on CSC 2014 datasets, we still think that our method can cooperate with hand-crafted rules (as top CSC systems did in CSC 2014), which becomes our future work.

References

de Amorim, R.C., & Zampieri, M. (2013). Effective Spell Checking Methods Using Clustering Algorithms. *Recent Advances in Natural Language Processing*, 7-13.

Blair, C. (1960). A program for correcting spelling errors. *Information and Control*, 3, 60-67.

Carlson, A., & Fette, I. (2007). Memory-Based Context-Sensitive Spelling Correction at Web Scale. In *Proceedings of the 6th International Conference on Machine Learning and Applications*, 166-171.

Carlson, A., Rosen, J., & Roth, D. (2001). Scaling up context-sensitive text correction. In *Proceedings of the 13th Innovative Applications of Artificial Intelligence Conference*, 45-50.

Chang, C.H. (1994). A pilot study on automatic chinese spelling error correction. *Journal of Chinese Language and Computing*, 4, 143-149.

Chen, K.J., Huang, C.R., Chang, L.P., & Hsu, H.L. (1996). Sinica Corpus: Design Methodology for Balanced Corpora. In *Proceeding of the 11th Pacific Asia Conference on Language, Information and Computation*, 167-176.

Chen, Q., Li, M., & Zhou, M. (2007). Improving Query Spelling Correction Using Web Search Results. In *Proceedings of the 2007 Conference on Empirical Methods in Natural Language (EMNLP-2007)*, 181-189.

Chen, Y.Z., Wu, S.H., Yang, P.C., Ku, T., & Chen, G.D. (2011). Improve the detection of improperly used Chinese characters in students' essays with error model. *International Journal of Continuing Engineering Education and Life-Long Learning*, *21*(1), 103-116.

Cucerzan, S., & Brill, E. (2004). Spelling correction as an iterative process that exploits the collective knowledge of web users. In *Proceedings of EMNLP*, 293-300.

Damerau, F. (1964). A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7, 171-176.

Deorowicz, S., & Ciura, M. G. (2005). Correcting Spelling Errors by Modelling Their Causes. *International Journal of Applied Mathematics and Computer Science, 15*(2), 275-285.

Golding, A., & Roth, D. (1999). A winnow-based approach to context-sensitive spelling correction. *Machine Learning*, *34*(1-3), 107-130.

Islam, A., & Inkpen, D. (2009). Real-word spelling correction using googleweb 1t 3-grams. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP-2009)*, 1241-1249.

Li, M., Zhang, Y., Zhu, M.H., & Zhou, M. (2006). Exploring distributional similarity based models for query spelling correction. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, 1025-1032.

Liu, W., Allison, B., & Guthrie, L. (2008). Professor or screaming beast? Detecting words misuse in Chinese. *The 6th edition of the Language Resources and Evaluation Conference.*

Liu, C.L., Lai, M.H., Tien, K.W., Chuang, Y.H., Wu, S.H., & Lee, C.Y. (2011). Visually and phonologically similar characters in incorrect Chinese words: Analyses, identification, and applications. *ACM Transactions on Asian Language Information Processing*, *10*(2), 1-39.

Mitton, R. (1996). *English Spelling and the Computer*. Harlow, Essex: Longman Group.

Mitton, R. (2008). Ordering the Suggestions of a Spellchecker Without Using Context. *Natural Language Engineering*, *15*(2), 173-192.

Pirinen, T., & Linden, K. (2010). Creating and weighting hunspell dictionaries as finite-state automata. *Investigationes Linguisticae*, 21.

Verberne, S. (2002). *Context-sensitive spell checking based on word trigram probabilities*, Master thesis, University of Nijmegen.

Whitelaw, C., Hutchinson, B., Chung, G.Y., & Ellis, G. (2009). Using the Web for Language Independent Spellchecking and Autocorrection. In *Proceedings Of Conference On Empirical Methods In Natural Language Processing (EMNLP-2009)*, 890-899.

Wu, S.H., Chen, Y.Z., Yang, P.C., Ku, T., & Liu, C.L. (2010). 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)*, 54-61.

Wu, S.H., Liu, C.L., & Lee, L.H. (2013). Chinese Spelling Check Evaluation at SIGHAN Bake-off 2013. In *Proceedings of the 7th SIGHAN Workshop on Chinese Language Processing (SIGHAN'13)*, 35-42.

Yu, L.C., Lee, L.H., Tseng, Y.H., & Chen, H.H. (2014). Overview of SIGHAN 2014 Bake-off for Chinese Spelling Check. In *Proceedings of the 3rd CIPS-SIGHAN Joint Conference on Chinese Language Processing (CLP'14)*, 126-132.

Zhang, L., Zhou, M., Huang, C.N., & Pan, H.H. (2000). Automatic detecting/correcting errors in Chinese text by an approximate word-matching algorithm. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, 248-254.