Spell Checking for Chinese

Shaohua Yang, Hai Zhao, Xiaolin Wang, Bao-liang Lu

Center for Brain-Like Computing and Machine Intelligence
Department of Computer Science and Engineering,Shanghai Jiao Tong University
Shanghai Jiao Tong University, 800 Dong Chuan Rd., Shanghai 200240, China
shyang.ok@gmail.com, v-xiaow@microsoft.com, {zhaohai,blu }@cs.sjtu.edu.cn

Abstract

This paper presents some novel results on Chinese spell checking. In this paper, a concise algorithm based on minimized-path segmentation is proposed to reduce the cost and suit the needs of current Chinese input systems. The proposed algorithm is actually derived from a simple assumption that spelling errors often make the number of segments larger. The experimental results are quite positive and implicitly verify the effectiveness of the proposed assumption. Finally, all approaches work together to output a result much better than the baseline with 12% performance improvement.

Keywords: spell, checking, chinese

1. Introduction

Spell checking identifies incorrect writing in text, and spell correction gives advice after errors are detected. These techniques assist a writer by identifying incorrect writing and giving useful advice. When only the former works, we say it is spell checking, and when the latter is also involved, we say it is spell correction. The work in this paper will focus on the former task of spell checking as human experience indicates that one can immediately realize what is wrong when the spelling error is clearly marked.

For English spell checking, many studies have been made and quite good results have been obtained. In contrast, for Chinese it is still a challenging work due to some special processing difficulties that arise from Chinese writing, which never occur in English spell checking.

Spelling errors can be roughly put into two main categories. One is a non-word spelling error, in which the input word's form is definitely incorrect and cannot be found in any dictionary. For example, using '*fcrm*' rather than '*farm*'. The other is a real-word spelling error in which the input word's form can be found in the dictionary but is incorrectly used. For example, using '*come form*' not'*come from*'. Different treatments are developed for these two types of spelling errors, context-independent methods for the former type, and context-dependent methods for the latter.

Before analyzing Chinese spelling errors, some background knowledge of Chinese needs to be introduced. It is well known that Chinese is written in a special way like some other East Asian languages. This results in a sophisticated consequence where 'word' is not a natural concept for Chinese text processing, but hanzi is. As words cannot be extracted in a smooth way, most existing spell checking techniques for other languages based on words cannot be adopted for Chinese in a straightforward way. Chinese hanzi sets normally include about 10,000-20,000 characters, but far fewer are really used in everyday life. Typically, the 2,500 most widely characters can cover 97.97% of text, while 3,500 can cover 99.48% of text.

Chinese spelling errors are quite different from English ones, too. In theory, hanzi cannot be spelled incorrectly as all legal characters have been stored in a font lib and a Chinese input system just attempts to build an effective map between Latin letter encoding and the hanzi font. Typically, a Pinyin-based Chinese input system uses 'hao' to represent hanzi '好'(good), where the character shape '好' will be immediately given by the Chinese input system from the build-in font lib. That is, *from a sense of hanzi*, a non-word(hanzi) spelling error is impossible in Chinese. All Chinese spelling errors are effectively real-word(real-hanzi) errors. Thus spell checking for Chinese has to mainly rely on context-dependent methods.

The rest of the paper is organized as follows. The next section discusses the relation between a Chinese input system and the corresponding spelling errors. Section 3. presents related works. Section 4. proposes our approaches for Chinese spell checking. The experiments and results are reported in Section 5.. Section 6. concludes this paper.

2. Chinese Input System and Spelling Errors

Why do spelling errors happen when typing Chinese with a keyboard? This is a question that draws little attention in existing studies. Our empirical study shows that spelling errors in Chinese text mostly depend on the way Chinese is typed into the computer, not on how well users have mastered this language.

We have been aware of that Chinese text is basically written with various characters. A Chinese character is not something like a letter in other language, but a unit that integrates shape, meaning and pronunciation at the same time. We show some hanzi examples in Figure 1.

As there are thousands of characters for modern Chinese, they cannot be directly typed into the computer by a standard Latin-based keyboard. The solution is to assign a Latin-letter based code for each character. Pinyin is a systematic attempt to give Chinese a Romanized representation based on Chinese pronunciation. Usually, a pronunciation-based Chinese input system(CIS) depends on a specific Pinyin scheme for its hanzi encoding. There are several Romanized representations of Chinese. Figure

shape	meaning	pronunciation (pinyin)
爱	love	ai
和	peace/and	he
的	of	de
鸟	bird	niao
游	swim	you

Figure 1: Hanzi is an integrated unit for shape, meaning and pronunciation.

1 gives some examples of Pinyin. Figure 2 demonstrates a typical pronunciation-based CIS, In this figure, when we type the Pinyin sequence ”tongzhi”, the hanzi sequence candidates are shown as follows such as ”通知”.

Figure 2: A Pronunciation-based Chinese input system.

According to the ‘Hanyu Pinyin’ scheme, there are only 398 syllables for thousands of hanzi characters in modern Chinese.[1] This means that 6-20 different characters share the same Pinyin code. That is why an early pronunciation-based CIS was so inefficient, as users at that time had to stop to choose a specific hanzi after each Pinyin sequence was input. Later, researchers observed that the uncertainty with which a Pinyin sequence is translated into a hanzi sequence may be greatly reduced if longer Pinyin sequences are input, thus more and more modern pronunciation-based CIS make use of this fact and handle longer and longer Pinyin sequences. Figure 3 shows the number of hanzi sequence options is reduced as longer and longer syllables are input. There are more than nine options when ‘*tong*’ is input (only the first hanzi is recommended.), there will be five options when ‘*tongyi*’ is input, and there is only one option when ‘*tongyide*’ is input.

The kernel of a modern pronunciation-based CIS is an effective Pinyin-to-hanzi translation and recommendation sub-system. Users either accept the hanzi sequence recommended by the CIS or make another choice (correction) from a list of all possible character sequences corresponding to the Pinyin input. Chinese spelling errors happen when the CIS recommends a wrong decoding of a Pinyin sequence and the user fails to perceive it. For the purpose of faster Chinese input, a modern CIS has to decode longer and longer Pinyin sequences in order to automatically select uncertain characters for users (Lee et al., 1999; Lu et al., 2009). This has the consequence that the user seriously relies on outputs of the CIS and tends to believe what the CIS suggests most of the time. A cutting-edge Pinyin-to-hanzi translation system may achieve an accuracy as high as 90% (Zhang and Yao, 2004; Lu et al., 2009), which is quite a good score for such a translation system but not satisfactory for a CIS. The other 10% of Pinyin sequences that CIS cannot correctly recommend hanzi for surely require correction from user. If user fails to do so, then spelling errors will inevitably occur. We evaluated a spelling error list that we collected for this study and found that 95% of errors are due to the misuse of homophones. This may be reasonably explained as follows, 1) most users adopt a pronunciation-based CIS; 2) most spelling errors are initially caused by the wrong recommendation from the CIS.

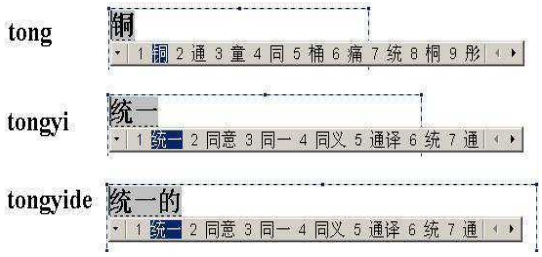

Figure 3: Longer hanzi sequences make the input more deterministic.

3. Related Work

Like many other natural language processing tasks, spell checking was first proposed for English (Peterson, 1980). An early work can be seen in (Pollock and Zamora, 1984), where a simple dictionary-based approximate search strategy was adopted. (Atwell and Elliott, 1987) used n-gram word and part-of-speech language models for this task.

(Golding, 1995) considered spell checking as a disambiguation task from an ambiguous set of correct and incorrect cases. Furthermore, (Golding and Roth, 1996) introduced a Winnow-based approach to handle rich features and achieved quite good results. (Mangu and Brill, 1997) introduced a transition-based learning method for spell checking(correction). Their method can acquire three types of rules from training text and thus constructed a high performance but concise system for English. A recent spell correction work is (Li et al., 2006), where a distributional similarity was introduced to improve the web query spell correction. For other languages, (Athanaselis et al., 2010) used a language model to detect Greek spelling errors as (Atwell and Elliott, 1987).

(QasemiZadeh et al., 2006) presented an adaptive, language independent, and ’built-in error pattern free’ spell checker by implementing a ’Ternary Search Tree’ data structure, but this system needs the user interaction to learn error patterns of media and needs a lot of other resources.

As for Chinese spelling checking, an early work was (Chang, 1994) where character tables for similar shape, pronunciation, meaning, and input-method-code characters are proposed. His approach tries to “guess” all the possible errors, and its limitation is in handling errors other than single character substitution errors.

(Cai, 1997) found a useful fact that most spelling errors cause a Chinese word segmentation abnormality. For ex-

[1]To pronounce a character, the tone must be considered, too. There are five tones in modern Chinese. Thus there are somewhat less than 2,000 syllables with tone considered. However, tone cannot be conveniently typed and is actually excluded by nearly all CIS.

ample, hanzi sequence 'abc' may immediately form a valid Chinese word without spell errors. In the case that character 'b' is incorrectly replaced by 'B' in 'abc', hanzi sequence 'aBc' will be quite probably be segmented into three single-hanzi words, 'a', 'B' and 'c'. (Lee et al., 1999) proposed a Block-of-Combinations (BOC) segmentation method based on single-character word occurrence frequency.

(Zhang et al., 2000b) proposed an effective approach to detect and correct Chinese spelling errors. The largest improvement over (Chang, 1994) is that their method can handle not only hanzi substitution, but also insertion and deletion errors. They adopted an "approximate word match method used in English spelling error detection and correction to find all words in a dictionary whose minimum edit distance to a given string is less than a threshold" ((Zhang et al., 2000b): pp. 249). Due to a significant difference between English and Chinese, they carefully refined the original word match method. Finally, (Zhang et al., 2000b) developed a more complicated approximate match algorithm in order to handle as many errors as possible, and achieved better performance than (Chang, 1994).

Due to the significant difference between English and Chinese, the approach in (Zhang et al., 2000b) has been carefully improved. To handle many errors as many as possible, a complicated approximate match algorithm was developed. They did report better results than those by (Chang, 1994).

(Zhang et al., 2000a) also adopted a Winnow-based Chinese spell error detection and correction approach that had been introduced by (Golding and Roth, 1996). Both local language features and wide scope semantic features were considered in their approach, which is evaluated on errors caused by a typical shape-based CIS, the "five-stroke" input.

(Cucerzan and Brill, 2004) made use of logs of user queries to deal with the spelling correction of search queries through an iterative transformation of the input query strings into other likely queries. This method relies heavily on the statistics of the Internet search query logs.

(Zheng et al., 2011) proposed a method which is based on a generative model to study how to correct people's mistakes when people typed the Chinese Pinyins, however, the difference between our work and theirs is that our input assumption is Chinese words rather than Pinyin. Our method is also effective in applications such as automatic correction for the OCR text.

Note that all spell checking or correction tasks require some supporting linguistic resources. Typically, a lexicon and a very large text corpus are necessary. When a labeled training corpus is exploited, we say that it is a 'supervised' method, otherwise, it is an 'unsupervised' one. Among the above approaches, (Golding and Roth, 1996; Mangu and Brill, 1997; Zhang et al., 2000a; Li et al., 2006) can be regarded as supervised, and the others as unsupervised. Generally speaking, the performance of these two categories are not directly comparable.

4. Our Approaches

In this section, two approaches are introduced; one is based on minimized-path segmentation, and the other adopts two statistical criteria. Note that our approach is basically an unsupervised one as it assumes no labeled corpus is available.

However, additional linguistic resources are required by the proposed method, including a Chinese word lexicon, a hanzi-Pinyin dictionary[2] and a large sample of plain Chinese text.

4.1. Approach using Minimized-Path Segmentation

In this subsection, a rule-based method (mini-path) is proposed.

Our algorithm for spell checking is partially motivated from some observations found by (Cai, 1997), that is, spelling errors make word segmentation abnormal. However, the word is not a natural processing unit in Chinese[3]. Words are obtained after some word segmentation task is performed. As words are usually required before error checking, and meanwhile, errors make word segmentation abnormal. Our solution is to take advantage of the basic idea of the minimized-path algorithm, one of the traditional rule-based word segmentation methods for Chinese. Let $W = \{\{w_i\}_{i=1,\ldots,n}\}$ be a word list, where each item w_i is a legal word. This is a Viterbi-style algorithm to search for the best segmentation S^* for a given text T, as follows,

$$S^* = \underset{w_1 \cdots w_i \cdots w_n = T}{\operatorname{argmin}} n, \tag{1}$$

with all $\{w_i\} \in W$. n is called path length for the segmentation. The idea behind the minimized-path segmentation algorithm is that a reasonable segmentation should maximize the lengths of all segments or minimize the path length. Without any misused hanzi, the segmentation can be performed successfully and output a segment sequence, $S^* = w_1 \cdots w_i \cdots w_n$. Assuming that hanzi h'_k is used instead of the correct one h_k in the word $w_i = h_1 h_2 ... h_m$, w_i will not be successfully matched from the given word list any more. Usually, the correct segment w_i has to be split into more than two parts $h_1 h_2 ... h_{m'}$, $h_{m'+1} ... h_m$ for a correct matching. Thus, the path length with misused characters is larger.

The algorithm first takes the output of a minimized-path word segmentation which aims to minimize the number of words based on a dictionary. Then we look for some similar word replacement from the given word list so that the path length can be reduced. If such a replacement is found, then it will be shown that some misused characters do exist. The idea leads to the algorithm in Figure 4, where $Edit\_Dist(u_{ij}, v_{ij})$ means edit distance of hanzi sequences u_{ij} and v_{ij}, and $dist(u_{ij}, v_{ij})$ means their Pinyin

[2] A list which gives each character its corresponding Chinese pronunciation (Pinyin).

[3] Chinese word definition is problematic; there are several different Chinese word segmentation conventions in computational linguistics (Sproat and Emerson, 2003). So, to be more careful, we sometimes call outputs of a word segmenter *segments* or *segmented units* instead of words in Chinese text processing.

Input: A given legal word list W
An input segmented sequence, $s = w_1 w_2 ... w_n$ by using minimized-path segmentation algorithm;
The maximum length of a merging word sequence, $K = 5$
two dimensional array graph $g[][]$ to represent the candidates' score
Initialization:
$i = 1$
$g[\,][\,] = -\infty$
Algorithm:
while $i < n$
  $j = i$
  while $j \leq min(i + K, n)$
    let $u_{ij} = w_i ... w_j$
    find set s' which includes v_{ij} of W so that $Edit\_Dist(u_{ij}, v_{ij}) \leq 1$
    if s' not empty
      for v_{ij} in s'
        $g[i-1][j] = max(g[i-1][j], length(v_{ij}) - dist(u_{ij}, v_{ij}))$
    else if $j == 1$
      $g[i-1][j] = length(u_{ij})$
    $j = j + 1$
  $i = i + 1$
Output: Revised segmented sentence, $s^* = \arg\max_s distance(1, n)$

Figure 4: Algorithm 1

edit distance. To effectively get the merged words' candidates, we adopt the state-of-the-art index structure and search algorithm in (Ji et al., 2009). The main idea is to build a trie structure to index the words in the dictionary by which we can quickly find the candidates within the specified edit distance. after the graph is built, we can get the revised segmented sentences by using the dynamic programming. The algorithm greedily tries to concatenate neighboring words by making a change within edit distance 1,[4] i.e. changing only one character, so that the modified word sequence can find a match in the legal word list. Then all word sequences with modifications are evaluated by a metric $Score_{ij} = length(v_{ij}) - dist(u_{ij}, v_{ij})$. includes two items, the first indicates how many characters the matched word sequence consists of, and the second indicates that the Pinyin edit distance between the original word and the concatenated word sequence. Following the definition of the metric, the algorithm will actually give a segmentation of which each include as many words and as few revisions as possible, which still follows the idea of minimized-path segmentation. Comparing the output segmentation and the original one, the difference will be where spelling errors are located. Note that this algorithm also allows outputting spell correction according to the corresponding correction form in the word list.

4.2. Approach using Statistical Criteria

We consider two statistical criteria to detect Chinese spelling errors in this subsection. To realize our algorithm ,we need a large text sample T to calculate the logarithmic probabilities. In addition, we also need a confusion set to get candidate errors. We build the confusion set by utilizing the Pinyin. If the Pinyin is the same for two hanzi characters, they are in the same confusion set.

The first ways is based on a language model(LM), which was used in some existing work (Zhang et al., 2000b; Zhang and Yu, 2006; Liu et al., 2008). The difference is that a bigram language model is used here rather than trigram as before. The idea is still simple enough. If a character incorrectly occurs, then it should significantly change bigram probabilities (which can be estimated well by a language model) in which it is involved. For example, $h_1 h_2 h_3$ is a hanzi string. Bigrams $h_1 h_2$ and $h_2 h_3$ should appear with normal probabilities. If h_2 is incorrectly replaced by h'_2, then $h_1 h'_2$ and $h'_2 h_3$ should not be so frequent as $h_1 h_2$ and $h_2 h_3$, respectively. This leads to in the following spelling error checking method.

First, calculate the logarithmic probabilities $h_{-1}h$ and hh_{+1} for hanzi h in the text U, L_1 and L_2, respectively, where h_{-1} and h_{+1} are the previous and next hanzi of h. Then we calculate the logarithmic probabilities $h_{-1}h_1$ and $h_2 h_{+1}$ in U of which the h_1 and h_2 are h's candidates to get the maximum value of the logarithmic probabilities. the max values are respectively L'_1 and L'_2. If both $L'_1 - L_1 > \alpha$ and $L'_2 - L_2 > \alpha$ hold, then h will be probably a misused character in V, where α is a predefined threshold.[5]

The second is mutual information, which has been suggested by (Zhang and Yu, 2006) for this task. However, we will use hanzi-based mutual information rather than the word-based one in that work. Mutual information of two characters is defined as

$$MI(h_1, h_2) = log \frac{p(h_1 h_2)}{p(h_1)p(h_2)}$$

MI indicates how possible two characters are collocated together. So, our criterion is that if both $MI(h_{-1}, h_1) -$

[4] We set the edit distance to one because spelling errors seldom occur in sequence according to empirical observation.

[5] Our empirical attempts with trigram or four-gram have been shown unsuccessful, thus only bigram criterion is reported in this paper.

$MI(h_{-1}, h) > \beta$ and $MI(h_2, h_{+1}) - MI(h, h_{+1}) > \beta$ for hanzi sequence $h_{-1}hh_{+1}$ in which h_1 and h_2 are the values to get the largest mutual information's value, then we say h may be a spelling error candidate.

4.3. The Hybrid Model

To effectively incorporate the statistical information into the rule-base algorithm. we add the statistical features into the score function as stated in the rule-based method. In the mini-path+LM method, we add the score function lmScore(v_{ij})-lmScore(u_{ij}) of which lmScore(u) is the language model score of the sequence of words u. And the results proves the effectiveness of the statistical features.

Our algorithm earn three merits. First, it is an efficient segmentation algorithm. Existing work, for example, (Zhang et al., 2000b), has to greedily generate too many words or word sequences to match all possible error cases. In our approach, this difficulty is avoided as we only consider combining neighboring words. As our search space is greatly reduced, the final decoding algorithm can be also efficiently executed in a Viterbi style like finding the longest path in a DAG. Second, our algorithm effectively separates the knowledge source for spelling errors from the processing procedure. All knowledge is from the function $dist()$ and the word list W (We will give all necessary details about the list in the experimental section.). The former defines error bias from the correct cases, and the latter defines how the correct cases look. Such separation makes the algorithm easier to adapt for spelling errors caused by other types of CIS, for example, the five-stroke input (the most popular shape-based CIS and focused on by (Zhang et al., 2000b; Zhang et al., 2000a)). Third, as the edit distance is used, all three types of spelling errors, hanzi substitution, insertion and deletion errors can be detected in a unified way.

5. Experiments

5.1. Linguistic Resources

In the proposed approach, three types of linguistic resources are required,

(a) a legal word list

(b) a hanzi-Pinyin dictionary

(c) a large sample of plain text

For (a), we adopt an online standard lexicon[6] which has been frequently used for Chinese input methods and contains about 213,662 words. For (b), we choose a list that contains 7,809 characters and their corresponding Pinyin.[7] For (c), the 1998 China Daily newspaper corpus[8] is adopted.

As for the language model implementation, SRILM with Good-Turing algorithm as smoothing strategy is adopted.[9] and the berkeleylm[10] software is used to support the arpa file format.

To evaluate our approach, a text sample with natural spelling errors is collected in various ways and errors are annotated by hand. The corpus is collected from real human-input text and is quite different from those adopted in some existing work that consist of man-made spell errors or confusion sets. This corpus has 8765 characters with 439 spelling errors at this time. In addition, we especially annotate a development set with 3018 characters and 157 spelling errors to determine some necessary parameters. checking for Chinese has suffered a lot from the lack of a standard evaluation corpus, the corpus will be released for research purpose.

5.2. Experimental Results

Spell checking performance is evaluated by F-score $F = 2RP/(R + P)$ which is a common way to measure the spelling checking system's performance. The recall R is the ratio of the correctly identified spelling errors of the checker's output to all spelling errors in the gold-standard and the precision P refers to the ratio of the correctly identified spelling errors of the checker's output to all identified errors of the checker's output.

Table 1: Results of all methods

Approach	R	P	F
mini-path	0.421	0.287	0.337
LM	0.456	0.429	0.442
MI	0.492	0.303	0.374
mini-path+LM	0.712	0.364	0.482
mini-path+LM+MI	0.720	0.367	0.486
MS-Word 2007(SC)	0.554	0.269	0.363

The best results that modern Chinese spell checking systems ever achieved are less than 0.7 for recall and less than 0.4 for precision according to (Chang, 1994; Cai, 1997; Zhang et al., 2000b; hua Li and Wang, 2002). Due to the absence of all evaluation corpora in the existing work, we cannot give a direct comparison with them. The best results that were ever reported are from (Zhang et al., 2000b), but we found that a key function in it is not disclosed, which makes it impossible to re-implement their work. Hence, the spell checking output of Microsoft Word 2007 Simplified Chinese version is taken as our baseline[11].

From table 1, we see that the proposed method gives a group of satisfactory results, and achieves 12% F-score improvement over the baseline. For LM criterion, α is tuned to 1.5. Note that although the mini-path's F-Score is lower than the base-line method MS-Word, after we incorporate the element of Language Model, the score is much better than than MS-Word both in Precision and Recall. As seen from the table, the mini-path+LM method improves 15% compared to the rule-based method and 12% compared to the baseline method. It fully represents that the statistical information brings big improvements because it incorporates probability differences between the target phrase and

[6] http://download.csdn.net/detail/daxuea/2144016

[7] This list is at http://download.csdn.net/source/1992252

[8] http://ccl.pku.edu.cn:8080/ccl corpus/jsearch/index.jsp

[9] http://www-speech.sri.com/projects/srilm/

[10] http://code.google.com/p/berkeleylm/

[11] Another reason that we take MS-Word as the baseline is that it was unofficially reported that the work of (Zhang et al., 2000b) has been successfully transferred into MS-Word.

the original phrase. It coincides with our impression that the LM is an important criterion to estimate a sentence's fluency. In addition, we find that incorporating the element of MI into the final result gives little improvement.

5.3. Detailed Analysis

First, we give the result curve according to language model criterion with different α. The results in Figure 5 show that peak performance is given with $\alpha = 1.5$. In addition, the shape of this curve means a language model alone is not so good a criterion for spell checking as it doesn't provide stable performance.

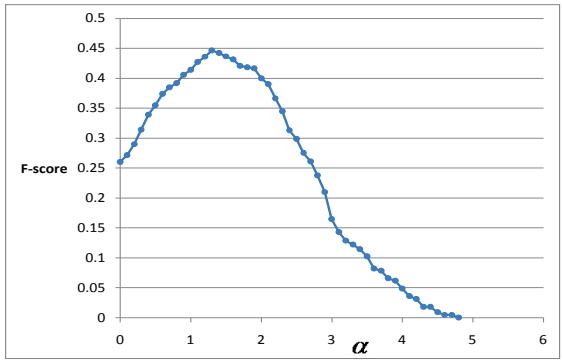

Figure 5: Choose a better α for language model criterion.

We also investigate how the size of the legal word list affects the performance of the proposed rule-based method (mini-path). Choosing 1/10, 2/10, ..., of the original word list uniformly, the results of spell checking are given in Figure 6 and Figure 7 where the solid red line represents the actual performance, and the dashed green line represents the corresponding linear fit. This figure shows a stable performance improvement as the word list is continuously enlarged. The results also demonstrate that a word list may be a good knowledge source for the spell checking task and the size of the word list is an important factor in the final performance. They also show that the proposed mini-path approach relies on a large high-quality lexicon.

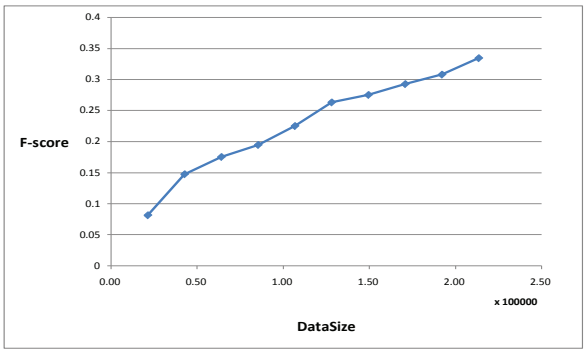

Figure 6: mini-path:the relation between datasize and f-score.

5.4. Summary

Today, the most popular CIS are pronunciation-based and nearly everybody uses a pronunciation-based CIS for their work or entertainment. This is where our work is mostly motivated, that is, to catch up with the recent changes in this field. In addition, we develop a new evaluation corpus for the task of spell checking. This corpus is collected from real-world applications so that it can effectively reflect the current state of CIS.

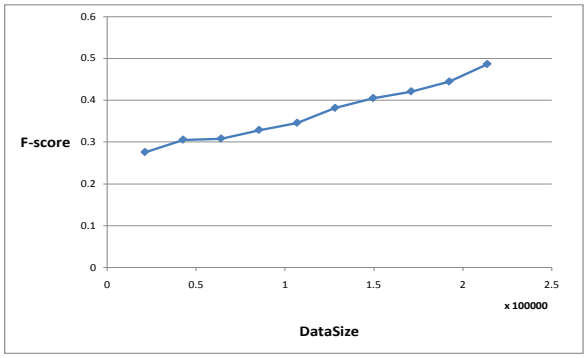

Figure 7: mini-path+LM:the relation between datasize and f-score.

6. Conclusion

In this paper, we investigate the relationship between Chinese input systems and spell errors, and report some novel results about Chinese spelling checking. Chinese spell checking is presented in this paper. As modern Chinese input systems have generally turned to Pinyin, spell checking tasks for Chinese should tune itself to fit the new challenge. Besides, existing work often requires high computational cost to solve the problem due to some special characteristics of Chinese. This work aims to give a positive response to all these challenges.

The contributions of this paper can be summarized as follows,

(1) An assumption for Chinese spell checking that **spell errors in Chinese usually lead to a larger number of segmented units** is proposed and its effectiveness is verified. An efficient novel spell checking algorithm for Chinese based on the above assumption is proposed.

(2) Two statistics-based approaches for spell errors are also evaluated and compared and are effectively combined into the rule-based method which proves promising results.

(3) We empirically show that a legal word list plays an important role in spell checking. Precisely, the accuracy of the proposed approach increases with increases in the size of the word list.

7. Acknowledgements

This work was partially supported by the National Natural Science Foundation of China (Grant No. 60903119 and

Grant No. 61170114), the National Research Foundation for the Doctoral Program of Higher Education of China under Grant No. 20110073120022, the National Basic Research Program of China (Grant No. 2009CB320901), the Science and Technology Commission of Shanghai Municipality (Grant No. 09511502400), and the European Union Seventh Framework Programme (Grant No. 247619).

8. References

Theologos Athanaselis, Konstantinos Mamouras, Stelios Bakamidis, and Ioannis Dologlou. 2010. A corpus based technique for repairing ill-formed sentences with word order errors using co-occurences of n-grams. *International Journal on Artificial Intelligence Tools (IJAIT) (in press)*.

Eric Atwell and Stephen Elliott. 1987. Dealing with ill-formed english text. In *The Computational Analysis of English:A Coprus-Based Approach*, London.

Sun Cai. 1997. Research on lexical error detection and correction of chinese text. Master's thesis, Tsinghua University, Beijing, China.

Chao-Huang Chang. 1994. A pilot study on automatic chinese spelling error correction. *Journal of Chinese Language and Computing*, 4:143–149.

S. Cucerzan and E. Brill. 2004. Spelling correction as an iterative process that exploits the collective knowledge of web users. In *Proceedings of EMNLP*, volume 4, pages 293–300.

Andrew Golding and Dan Roth. 1996. Applying winnow to context-sensitive spelling correction. In *Proceeding of the 13thInternational Conference on Machine Learning*, pages 182–190, San Francisco, CA.

Andrew Golding. 1995. A bayesian hybrid method for context-sentive spelling correction. In *Proceeding of the Third Workshop on Very Large Corpora*, pages 39–53, Boston, MA.

Jian hua Li and Xiao-Long Wang. 2002. Combining trigram and automatic weight distribution in chinese spelling error correction. *JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY*, 17:915–923.

S. Ji, G. Li, C. Li, and J. Feng. 2009. Efficient interactive fuzzy keyword search. In *Proceedings of the 18th international conference on World wide web*, pages 371–380. ACM.

Kin Hong Lee, Mau Kit Michael Ng, and Qin Lu. 1999. Text segmentation for chinese spell checking. *Journal of the American Society for Information Science*, 5:751 - 759.

Mu Li, Yang Zhang, Mu-Hua Zhu, and Ming Zhou. 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*, page 1025 - 1032, Sydney, Australia.

Wei Liu, Ben Allison, and Louise Guthrie. 2008. Professor or screaming beast? detectingwords misuse in chinese. In *The 6th edition of the Language Resources and Evaluation Conference*, Marrakech, Morocco.

Li Lu, Xuan Wang, Xiao-Long Wang, and Yan-Bing Yu. 2009. A conditional random fields approach to chinese pinyin-to-character conversion. *Journal of Communication and Computer*, 6:25–31.

Lidia Mangu and Eric Brill. 1997. Automatic rule acquisition for spelling correction. In *Proceeding of the 14thInternational Conference on Machine Learning*, pages 187–194, San Francisco, CA.

James L. Peterson. 1980. Computer programs for detecting and correcting spelling errors. *Communications of the ACM*, 23:676–687.

Joseph J. Pollock and Antonio Zamora. 1984. Automatic spelling correction in scientific and scholarly text. *Communications of the ACM*, 27:358–368.

B. QasemiZadeh, A. Ilkhani, and A. Ganjeii. 2006. Adaptive language independent spell checking using intelligent traverse on a tree. In *Cybernetics and Intelligent Systems, 2006 IEEE Conference on*, pages 1–6. IEEE.

Richard Sproat and Thomas Emerson. 2003. The first international Chinese word segmentation bakeoff. In *The Second SIGHAN Workshop on Chinese Language Processing*, pages 133–143, Sapporo, Japan.

Le Zhang and Tian-Shun Yao. 2004. Improving pinyin to chinese conversion with a whole sentence maximum entropy model. In *The First International Joint Conference on Natural Language Processing (IJCNLP-04)*, Hainan Island, China.

Yang-Sen Zhang and Shi-Wen Yu. 2006. Summary of text automatic proofreading technology. *Research on Computer Applications*, 6:8–12.

Lei Zhang, Ming Zhou, Chang-Ning Huang, and Ming-Yu Lu. 2000a. Approach in automatic detection and correction of errors in chinese text based on feature and learning. In *The Third Chinese World Congress on Intelligent Control and Automation*, pages 2744–2748, Heifei, China.

Lei Zhang, Ming Zhou, Chang-Ning Huang, and Hai-Hua Pan. 2000b. 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*, pages 248–254, Morristwon, NJ.

Y. Zheng, C. Li, and M. Sun. 2011. Chime: An efficient error-tolerant chinese pinyin input method. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (accepted)*.