NTOU Chinese Spelling Check System in CLP Bake-off 2014

Wei-Cheng Chu and Chuan-Jie Lin
Department of Computer Science and Engineering
National Taiwan Ocean University
No 2, Pei-Ning Road, Keelung 202, Taiwan R.O.C.
{wcchu.cse, cjlin}@ntou.edu.tw

Abstract

This paper describes details of NTOU Chinese spelling check system participating in CLP-2014 Bakeoff. Confusion sets were expanded by using two language resources, Shuowen and Four-Corner codes. A new method to find spelling errors in legal multi-character words was proposed. Comparison of sentence generation probabilities is the main information for error detection and correction. A rule-based classifier and a SVM-based classifier were trained to identify spelling errors. Two formal runs were submitted, and the rule-based classifier achieved better performance.

1 Introduction

Automatic spell checking 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 (Mitten, 1996; Deorowicz and Ciura, 2005; Carlson and Fette, 2007; Chen *et al.*, 2007; Mitten 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 and Roth, 1999; Carlson *et al.*, 2001), contextual information (Verberne, 2002; Islam and Inkpen, 2009), and others (Pirinen and Linden, 2010; Amorim and Zampieri, 2013).

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

On the other hand, there is also an *illegal-character error* where a hand-written symbol is not a legal Chinese character (thus not collected in a dictionary). Such an error cannot happen in a digital document because all characters in Chinese character sets such as BIG5 or Unicode are legal.

There have been many attempts to solve the spelling error problem in Chinese (Chang, 1994; Zhang *et al.*, 2000; Cucerzan and 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).

This bake-off is the second Chinese spell checking evaluation project. It includes two subtasks: error detection and error correction. The task is organized based on some research works (Wu *et al.*, 2010; Chen *et al.*, 2011; Liu *et al.*, 2011).

2 Replacement and Filtering

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.

Proceedings of the Third CIPS-SIGHAN Joint Conference on Chinese Language Processing, pages 210–215,
Wuhan, China, 20-21 October 2014

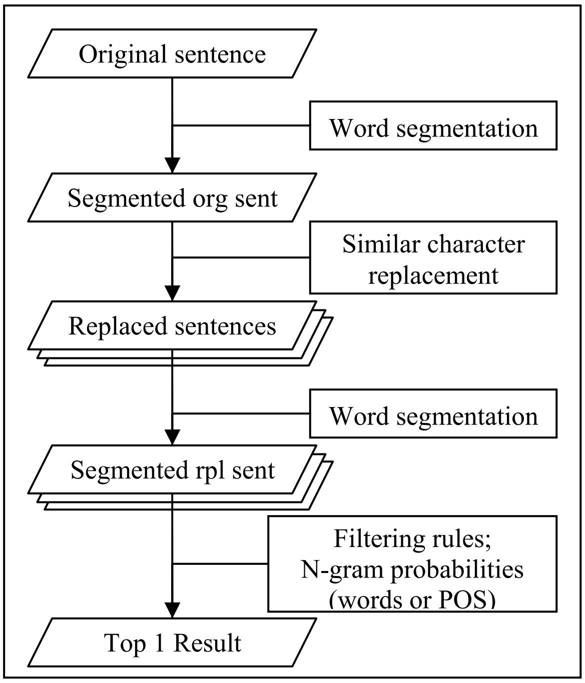

Figure 1. Architecture of NTOU Chinese Spelling Check System

In our experience, the confusion sets provided by the organizers do not cover all the cases in the development set. Two sources used to expend confusion sets are described in Section 2.1.

There are two kinds of spelling-error candidates in our system: one-character words and multi-character words. Their replacement procedures are different, as described in Section 2.2 and 2.3.

2.1 Confusion set expansion

In SIGHAN7 Bake-off 2013 Chinese Spelling Check task (Wu *et al.*, 2013), the organizers provided two kinds of confusion sets, phonologically similar characters and visually similar characters. We adopted all these confusion sets except the one consisting of characters having the same radical and the same number of strokes, because we do not think they are similar.

However, these confusion sets do not cover all the spelling error cases in the training data. We used two resources to expand the confusion sets. One is Showen and the other is the Four-Corner Encoding System.

Shuowen Jieji[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). We collect characters with same phonetic components to expand confusion sets, because they are by definition phonologically and visually similar. For example, the following characters share the same phonetic component "寺" thus become confusion candidates (their actual pronunciation are given in brackets):

侍[si4]持[chi2]恃[shi4]特[te4]時[shi2]...

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:

只囚貝足炅是員異買圓圜

2.2 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...

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

[2] 四腳號碼列表
http://code.web.idv.hk/misc/four.php

...嬰兒個數卻持續下滑...
...嬰兒個數卻時續下滑...
......

2.3 Multi-character word replacement

Our observation on the training sets finds that some errors occur in multi-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, popularity).

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. 馬偉雲 *et al.*, 2001).

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 multi-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 衹要, 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...
... 在教室裡衹要人員好...
... 在教室裡只要人緣好...
... 在教室裡只要人猿好...

2.4 Filtering rules

Two filter rules are applied before error detection in order to discard apparently incorrect cases. 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...

"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...

where "魏產齡" is recognized as a person name. We will discard such a replacement.

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...

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

3 Error Detection and Correction

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 misspelling".

Three language models were used to measure sentence generation probabilities as described in Section 3.1. Two formal runs were output of two different classifiers, SVM-based and rule-based systems, as described in Section 3.2 and 3.3.

3.1 N-gram probabilities

The possibility of a sequence of words can be measured as sentence generation probability by language models. We used smoothed word-unigram, word-bigram and POS-bigram models in our experiments. The training corpus used to build language models is ASBC. As usual, we use log probabilities instead.

A basic hypothesis is that a "better" sentence often has higher probability than the original one. We define *preference scores* to capture such kind of features:

$$pref_M(S_{new}, S_{org}) = \frac{\log(\text{Prob}_M(S_{org}))}{\log(\text{Prob}_M(S_{new}))} - 1 \quad \text{(E1)}$$

where M is the language model (word-unigram model, *etc.*), S_{org} is the original sentence, S_{new} is the new sentence, and Prob(s) is the generation probability of sentence s. By this definition, a new sentence having higher probability than the original one will have a preference score larger than 0, and the higher the better.

3.2 SVM-based classifier

6 features defined in Table 1 were used to train a support vector machine classifier (Chang and Lin, 2011). Besides the preference scores of word-unigram, word-bigram, and POS-bigram probabilities, another kind of features reveals whether a new sentence has the highest preference score among all replacements.

Unfortunately, the developed classifier tends to label all replacements as positive. So we define a threshold so that the replacement is accepted only when SVM thinks the probability of assigning "positive" label is larger than 0.95.

#	Feature definition
1	Preference score of word-unigram prob.
2	Preference score of word-bigram prob.
3	Preference score of POS-bigram prob.
4	Is max of word-unigram prob. preference
5	Is max of word-bigram prob. preference
6	Is max of POS-bigram prob. preference

Table 1. Features for training SVM classifier

3.3 Rule-based classifier

According to our hypothesis of error detection, a correct sentence should have a positive preference score since it has higher generation probability. Moreover, if many replacements have positive preference scores, the correct one should have the highest score.

However, in our observations, sometimes replacing with a frequently-seen word may result in higher preference score, even if the replacement is incorrect. Therefore, we define three thresholds for each n-gram model, respectively, for stricter error detection. Thresholds were trained by using Bakeoff 2013 CSC Datasets (Wu *et al.*, 2013).

The rules of detecting and correcting errors are defined as follows.

1. If no replacement has positive preference scores, report "no error" in both error detection and correction subtasks.
2. Sort the replacements first by their word-bigram preference scores, and then by their word-unigram preference scores, and then by the POS-bigram preference scores.
3. If the top-1 replacement's preference scores are all larger than the thresholds (0.004 for word-unigram, 0.03 for word-bigram, and 0.001 for POS-bigram), report "with error" and output the replacing character and its location in the sentence as correction.

4 Performance

There are two judging correctness in this bakeoff: detection level and correction level.

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

False-Positive Rate = FP / (FP+TN)
Accuracy = (TP+TN) / (TP+TN+FP+FN)
Precision = TP / (TP+FP)
Recall = TP / (TP+FN)
F1-Score= 2* Precision * Recall)/(Precision + Recall)

We submitted 2 formal runs based on two different classifiers. The first run was output by the rule-based classifier and the second run was output by the SVM-based classifier.

Table 2 and 3 illustrate the evaluation results of formal runs. As we can see, using the rule-based classifier performed better than the SVM-based classifier. Unfortunately none of them could achieve acceptable performance.

Run	FPAlarm	Accuracy	Precision	Recall	F1
Formalrun1_NTOU	**0.258**	**0.4652**	**0.4219**	**0.1883**	**0.2604**
Formalrun2_NTOU	0.9925	0.1045	0.1688	0.2015	0.1837

Table 2: Formal run performance in detection level.

Run	FPAlarm	Accuracy	Precision	Recall	F1
Formalrun1_NTOU	**0.258**	**0.4557**	**0.3965**	**0.1695**	**0.2375**
Formalrun2_NTOU	0.9925	0.0678	0.1143	0.1281	0.1208

Table 3: Formal run performance in correction level.

5 Conclusion

In this year, we tried to expand confusion sets in order to obtain larger coverage of similar characters. We also proposed a new method to find spelling errors in legal multi-character words. We submitted 2 formal runs based on the output of a rule-based classifier and a SVM-based classifier, respectively. The evaluation results showed that the rule-based classifier outperformed the SVM-based classifier, but neither of them achieved acceptable performance.

In the future, more features should be investigated and more decision rules should be discovered.

References

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

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

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

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

C.C. Chang and C.J. Lin. 2011. "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, 2:27:1-27.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

S.H. Wu, Y.Z. Chen, P.C. Yang, T. Ku, and C.L. Liu. 2010. "Reducing the False Alarm Rate of Chinese Character Error Detection and Correction," *Proceedings of CIPS-SIGHAN Joint Conference on Chinese Language Processing* (*CLP 2010*), 54-61.

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

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

馬偉雲, 謝佑明, 楊昌樺, 陳克健. 2001. "中文語料庫構建及管理系統設計," *Proceedings of the 14th Conference on Computational Linguistics and Speech Processing* (*ROCLING 14*), 1-17.