Computational Linguistics and Chinese Language Processing
Vol. 20, No. 1, June 2015, pp. 1-22

HANSpeller: A Unified Framework for Chinese Spelling Correction

Jinhua Xiong*, Qiao Zhang*,+, Shuiyuan Zhang*,+,

Jianpeng Hou*,+ and Xueqi Cheng*

Abstract

The number of people learning Chinese as a Foreign Language (CFL) has been booming in recent decades. The problem of spelling error correction for CFL learners increasingly is becoming important. Compared to the regular text spelling check task, more error types need to be considered in CFL cases. In this paper, we propose a unified framework for Chinese spelling correction. Instead of conventional methods, which focus on rules or statistics separately, our approach is based on extended HMM and ranker-based models, together with a rule-based model for further polishing, and a final decision-making step is adopted to decide whether to output the corrections or not. Experimental results on the test data of foreigner's Chinese essays provided by the SIGHAN 2014 bake-off illustrate the performance of our approach.

Keywords: Chinese Spelling Correction, HMM, Ranker-Base Model, Rule-based Model, Decision-making.

1. Introduction

Recent studies have shown that Chinese has become a popular choice for a second language among international college students. More and more people are learning Chinese as a Foreign Language (CFL). It is very difficult, however, for CFL learners to master Chinese because of the intrinsic linguistic features of the Chinese language. When CFL learners write Chinese essays, they are prone to generating a greater number and more diversified spelling errors than native language learners. Therefore, spelling correction tools to support such learners in

* Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
E-mail: xjh@ict.ac.cn, {zhangqiao, zhangshuiyuan}@software.ict.ac.cn
+ University of Chinese Academy of Sciences, Beijing, China

correcting and polishing their Chinese essays is valuable and necessary. For the English language, there are many editing tools that provide spelling check functionality, *e.g.* Microsoft Word's spellchecker. For the Chinese language, however, such tools cannot be found until now.

Spelling correction has been studied for many years on regular text and web search queries. Although these two tasks share many common techniques, they have different concerns. Compared to techniques of web search query spelling correction, where corrections should be presented to search engine users in real-time, more complicated techniques can be applied to spelling correction on regular text to improve the performance, as such a situation has a lower real-time requirement.

In spelling correction of Chinese essays of CFL learners, we face more challenges because of the uniqueness of the Chinese language.

1) Chinese corpora for spelling correction, especially publicly available ones, are rare, compared with English corpora. This impedes work on this practical topic.

2) There are no natural delimiters, such as spaces, between Chinese words, which may result in errors in words splitting, which may cause more splitting errors.

3) The number of error types is more than that of other cases, because CFL learners are prone to different kinds of errors that we cannot imagine as native speakers. There are four major error types that confuse people, as illustrated in Table 1.

Table 1. Examples of spelling error types

Error Types	Misspelled	Corrections
Homophone	一筹莫展 年年有鱼 聯合國公布	一愁莫展 年年有余 聯合國公佈
Near-homophone	好码差不多一样	号码差不多一样
Similar shape	列如：家庭會變冷漠 如火如荼	例如：家庭會變冷漠 如火如荼
Other errors	每個禮拜１、３、５	每個禮拜一、三、五
	受了都少苦	受了多少苦
	持续的发展	持续地发展

The first type is the misuse of homophone, which means learners choose the wrong characters with same pronunciation but different meanings. For example, "一筹莫展" may be misspelled as "一愁莫展". Herein, the second character "筹" is misspelled as "愁," both with

the same pronunciation (chóu). Another example is “年年有**鱼**” (There will be **fish** every year), which is homophonous with “年年有**余**” (There will be **surpluses** every year). One should take context into account when judging this type of homophone. A single syllable may also have a range of different meanings. The Cihai dictionary lists 149 Chinese characters representing the syllable "yì".

Second, there is the near-homophone error, which means the pronunciations of chosen words are very similar. For CFL learners, difference in diacritical markings may be not enough to distinguish. For example, there is a problem in discriminating pronunciation of the first character in the following sentences, “好码差不多一样” and “号码差不多一样”.

Besides, some graphically similar Chinese characters are confusing, due to their similar shape. They differ only in subtle aspects. To distinguish between these characters, many aspects, such as sound, meaning, and collocations, should be taken into account. If you do not look carefully, you can hardly distinguish them, *e.g.* “如火如荼” and “如火如茶,” where the first one is correct, and the second one is wrong.

Finally, some error types usually are caused by grammar rules of Chinese, such as the usage of three confusable words “的,” “地,” and “得”. Moreover, the last two words connect with two different pronunciations in different contexts. Therefore, checking correctness of the usage of these three words is difficult.

The direct reason why these error types are always encountered by CFL learners is that Chinese spelling is not phonetic and each word in a Chinese phrase has its specific meaning. Meanwhile, some other error types can be caused by various Chinese input methods.

4) The Chinese language is continuously evolving. Therefore, correction only based on static corpora is not enough. For example, traditional Chinese and simplified Chinese may have different choices for the same word. In some cases, it is very difficult to distinguish them. Thus, web-based high-quality resources should be considered for decision-making on spelling correction.

To address the above challenges, we propose a unified framework, named HANSpeller, for Chinese essay spelling error detection and correction. Our method combines different methods to improve performance. The main contributions are as follows. (1) An HMM-based approach is used to segment sentences and generate candidates for sentence spelling corrections. (2) Under the unified framework, all kinds of error types can be integrated for candidate generation. We collect some error types that can only be found in CFL learner essays and add them into the candidate generation process. (3) In order to address evolving features of the Chinese language, an online high-quality corpus is collected for training and decision-making and online search engine results also are used in the ranking stage of our model, which can also improve the performance significantly.

The rest of the paper is organized as follows. We discuss related works in Section 2, and we introduce our unified framework approach in Section 3, where we focus on the basic processes of our method. In Section 4, we present the detailed setup of the experimental evaluation and the results of the experiment. Finally, in Section 5, we conclude the paper and explore future directions.

2. Related works

The study of spelling correction has a long history (Kukich, 1992). It is aimed at identifying misspellings and choosing optimal words as suggested corrections. In other words, it contains two subtasks that involve spelling error detection and spelling error correction. In early research, the spelling corrections were mainly devoted to solving non-word errors; such errors were often caused by insertion, deletion, substitution, and transposition of letters in a valid word that result in an unknown word. A common strategy at that time was to rely on a word dictionary or some rules like Levenshtein distance (Levenshtein, 1966). Mangu and Brill (1997) proposed a transition-based learning method for spelling correction. Their methods generated three types of rules from training data, which constructed a high performance and concise system for English.

In these methods, however, the dictionaries and rules were always constructed manually, leading to very high cost. Therefore, statistics generative models were introduced for spelling correction, which made spelling correction step into a new stage. The error model and n-gram language model are two important models (Brill & Moore, 2000). Atwell and Elliott (1987) used n-gram and part-of-speech language models for spelling corrections. Mays *et al*. (1991) used word-trigram probabilities for detecting and correcting real word errors. Brill *et al*. (2000) proposed a new channel model for spelling correction, based on generic string to string edits.

With the development of the Internet, the research and technology on query spelling correction for search engines has been studied intensively. The task of web-query spelling correction shares a lot of technology with traditional spelling correction, but it is more difficult. First, the spelling correction task is faced with more error types, as all kinds of errors may occur in a web environment. In addition, search queries consist of some key words rather than sentences, making some sentence-based methods achieve poor performance. Therefore, many novel ideas have been proposed by researchers. Cucerzan and Brill (2004) presented an iterative process for query spelling check, using a query log and trust dictionary. There, the noisy channel model was used to choose the best correction. Ahmad and Kondrak (2005) used the search query logs to learn a spelling error model, which improves the quality of query spelling check. Li *et al*. (2006) applied a distributional similarity based model for query spelling correction. Gao *et al*. (2010) presented a large-scale ranker-based system for search spelling correction, where the ranker uses web-scale language models and many kinds of

features for better performance, including: surface-form similarity, phonetic-form similarity, entity, dictionary, and frequency features. Suzuki and Gao (2012) proposed a transliteration based character method using an approach inspired by the phrase-based statistical machine translation framework and attained good performance in online spelling correction.

Furthermore, Google and Microsoft have developed some application interfaces for checking spelling. Google (2010) has developed a Java API for a Google spelling check service. Microsoft (2010) provides a web n-gram service.

The above works mainly target the task of English spelling correction. As to Chinese spelling correction, the situation is quite different because English words are separated naturally by spaces, while Chinese words are not. This nature of Chinese makes correction much more difficult than that of English. An early work was by Chang (1995), which used a character dictionary of similar shape, pronunciation, meaning, and input-method-code to deal with the spelling correction task. The system replaced each character in the sentence with a similar character in the dictionary and calculated the probability of all modified sentences based on language model. Zhang (2000) introduced a method that can handle not only Chinese character substitution, but also insertion and deletion errors. They distinguished the way of matching between Chinese and English, thereby largely improving the performance over the work of Chang (1995). Hung and Wu (2008) introduced a method that used manually edited error templates to correct errors. Zheng *et al.* (2011) found the fact that, when people type Chinese Pinyin, there are several wrong types. Then, they introduced a method based on a generative model and the input wrong types to correct spelling errors. Liu *et al.* (2011) pointed out that visually and phonologically similar characters are major factors for errors in Chinese text. Thus, by defining appropriate similarity measures that consider extended Cangjie codes, visually similar characters can be quickly identified.

Some Chinese spelling checkers have also incorporated word segmentation techniques. Huang *et al.* (2007) used a word segmentation tool (CKIP) to generate correction candidates before detecting Chinese spelling errors. Hung and Wu (2009) segmented the sentence using a bigram language model. In addition, they combined a confusion set and some error templates to improve the results. Chen and Wu (2010) modified the system on the basis of Huang and Wu (2009) using statistic-based methods and a template matching module.

In addition, a hybrid approach has been applied to Chinese spelling correction. Chang *et al.* (2012) used an inductive learning algorithm in Chinese spelling error classification and got better performance than C4.5, maximum entropy, and Naive Bayes classifiers. Hao *et al.* (2013) proposed a Tri-gram modeled-Weighted Finite-State Transducer method integrating confusing-character table, beam search, and A* to correct Chinese text errors. Jin *et al.* (2014) integrated three models, including an n-gram language model, a pinyin based language model, and a tone based language model, to improve the performance of a Chinese checking spelling

error system.

Chinese essay spelling correction as a special kind of spelling correction research effort has been promoted by efforts, such as the SIGHAN bake-offs (Yu *et al*., 2014; Wu *et al*., 2013). Huang *et al*. (2014) used a tri-gram language model to detect and correct spelling errors. They also employed a dynamic algorithm and smoothing method to improve the efficiency. Chu and Lin (2014) used a word replacement strategy to generate candidates based on the expanded confusion set. Then, a rule-based classifier and SVM-based classifier were used to locate and correct errors. Gu *et al*. (2014) proposed two systems to solve the Chinese spelling check problem. One was built based on a CRF model, and the other was based on 2-Chars and 3-Chars model. Their experimental results showed that the latter model was better.

Chiu *et al*. (2013) divided the correction task into two subtasks to solve. They used word segmentation to find errors and combined machine translation model to translate the wrong sentences into the appropriate ones. Hsieh *et al*. (2013) developed two error detection systems based on CKIP word segmentation tool and Google 1T uni-gram data, respectively. Jia *et al*. (2013) proposed a single source shortest path algorithm based on the graph model to correct spelling errors.

In our system, we need to detect and correct spelling errors on Chinese essays that always are written by CFL learners. It has some different concerns with query text or query spelling correction. Noting that spelling correction methods require lexicons and/or language corpora, we adopt the method based on statistics combined with lexicon and rule-based methods.

3. A Unified Framework for Chinese Spelling Correction

In this section, we present a unified framework, named HANSpeller, for Chinese spelling correction based on extended HMM and ranking models. The major idea of our approach is to model the spelling correction process as a ranking and decision-making problem.

Figure 1 shows the whole outlined architecture of HANSpeller. It separates the Chinese spelling correction system into four major steps. First is to use the extended HMM model to generate the top-k candidates for the sentences being checked. Then, a ranking algorithm is applied to re-rank the correction candidates for later decision. The third step conducts rule-based analysis for a specific correction task, *e.g*. the correction rule of the usage of three confusable words “的,” “地,” and “得”. Finally, the system makes decision whether to output the original sentence directly or correction results based on the previous output and global constrains.

This framework provides a unified approach for spelling correction tasks, which can be regarded as a language independent framework and can be tailored to different scenarios. To

move to another scenario, you need to prepare a language related corpus, but you do not need to be an expert in that language.

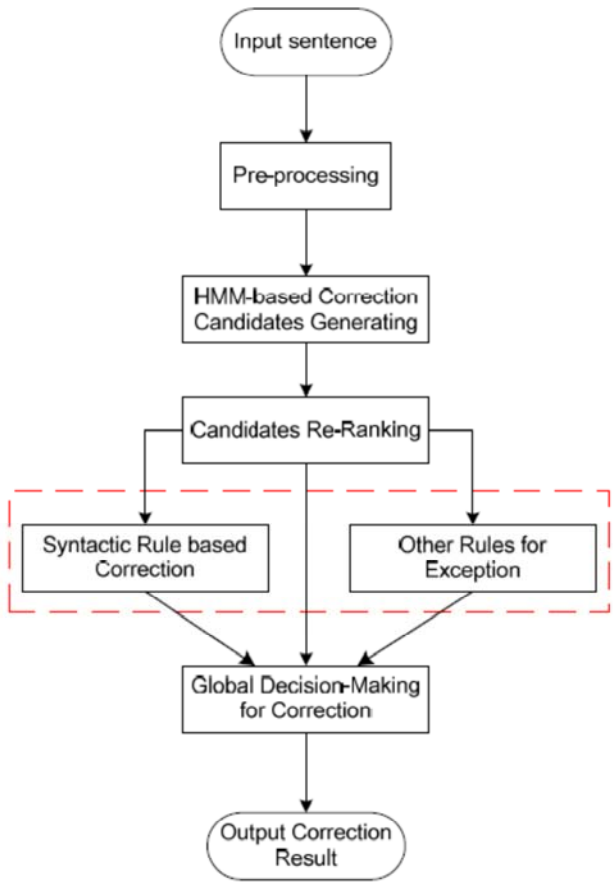

Figure 1. A unified framework (HANSpeller) for Chinese spelling correction.

3.1 Generating Candidates

Generating candidates of spelling correction is the basic part for the whole task, as it determines the upper bound of precision and recall rate of the approach. The HMM method can be used to generate candidates directly, but it faces several challenges when applied to Chinese essay spelling correction. (1) For high-quality spelling correction, the training of HMM is not a trivial task. (2) The long-span dependency in sentences makes a first-order hidden Markov model insufficient to catch contextual information. (3) Too many candidates make the algorithm not efficient enough, and some right corrections may be concealed by the wrong corrections.

To address the above challenges, some extensions have been made to the HMM-based spelling correction approach. First, the HMM-based method is used only for the candidate generation phase, not for final output correction generation. All kinds of possible error transformations will be integrated into the framework of the HMM approach, so as to get a high recall rate. Second, a higher-order hidden Markov model is used to capture long-span context dependency. Third, in order to reduce the number of candidates generated in the process, each word in the sentence only can be replaced with its homophone, near-homophone, or similar-shape word. In addition, a pruning dynamic programming algorithm is adopted to dynamically select the best correction candidates for each round of sentence segmentation and correction.

Figure 2 illustrates the whole process of the candidate generation phase.

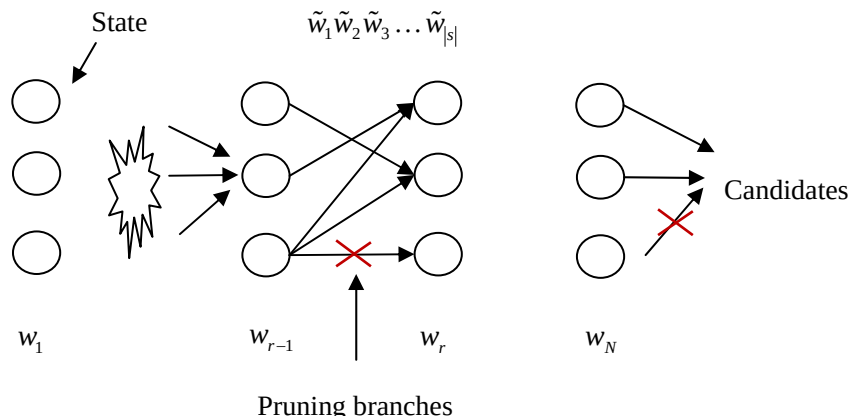

Figure 2. The whole process of generating candidates phase.

During the selection process of state, the edit distance and corrected results are combined to determine the quality of states. Let $S = w_1 w_2 w_3 \ldots w_N$ be a sentence needing correction, where each item w_i is a word. C is a state generated from state transition and segmentation of the S's r-th character, and $\tilde{w}_1\tilde{w}_2\tilde{w}_3\ldots\tilde{w}_{|s|}$ is the current corrected results in C. According to the noisy channel model, the occurrence probability of state C can be expressed as follows:

$$P(C) = P\left(\tilde{w}_1\tilde{w}_2\tilde{w}_3\ldots\tilde{w}_{|s|} | w_1 w_2 w_3 \ldots w_r\right) = \frac{P(w_1 w_2 w_3 \ldots w_r \mid \tilde{w}_1\tilde{w}_2\tilde{w}_3\ldots\tilde{w}_{|s|}) \times P(\tilde{w}_1\tilde{w}_2\tilde{w}_3\ldots\tilde{w}_{|s|})}{P(w_1 w_2 w_3 \ldots w_r)} \quad (1)$$

As $P(w_1 w_2 w_3 \ldots w_r)$ is the same for states in the same level, Equation (1) can be simplified as:

$$P(C) \propto P(w_1 w_2 w_3 \ldots w_r \mid \tilde{w}_1\tilde{w}_2\tilde{w}_3\ldots\tilde{w}_{|s|}) \times P(\tilde{w}_1\tilde{w}_2\tilde{w}_3\ldots\tilde{w}_{|s|}) \quad (2)$$

$$\log P(C) \propto \log P(w_1 w_2 w_3 \ldots w_r \mid \tilde{w}_1 \tilde{w}_2 \tilde{w}_3 \ldots \tilde{w}_{|s|}) + \log P(\tilde{w}_1 \tilde{w}_2 \tilde{w}_3 \ldots \tilde{w}_{|s|}) \quad (3)$$

Conceptually, the above formula can be calculated approximately using edit distance and n-gram language model. Symbolically, it can be represented by:

$$\log P(C) \propto editdis\tan ce(C) + (\log P(\tilde{w}_1) + \log(\tilde{w}_2 \mid \tilde{w}_1) + ...\log P(\tilde{w}_{|s|} \mid \tilde{w}_{|s|-n+1}..\tilde{w}_{|s|-1})) \quad (4)$$

In each round of the state generation stage, the best m states are selected according to the above calculated score. The remaining states are screened out to reduce the states' explosive growth, which improves the performance significantly. Finally, each sentence generates k candidates that represent the most likely correction results.

3.2 Ranking Candidates

In the candidate generation phase, top-k best candidates for a sentence are generated, but the HMM-based framework does not have the flexibility to incorporate a wide variety of features useful for spelling correction, such as online search results. Therefore, it is necessary to re-rank the candidates using more rich features, which can improve the precision of spelling correction significantly.

Given the original sentence, our system first generates a list of candidate sentences based on previous results. Then, the candidates in the list are re-ranked at this stage, based on the confidence score generated by a ranker, herein by an SVM classifier. Finally, we choose the top-2 candidates with the highest score to make the final decision.

The features used in our system can be grouped into five categories. They are listed separately in the table below.

Table 2. Five kinds of different features.

Feature Types	Features
Language Model Features	1.Text probability of candidates 2.Text probability of original sentence
Dictionary Features	1.Number of phrases 2.Number of idioms 2.Proportion of phrases 3.Proportion of idioms 3.Phrases and idioms length
Edit Distance Features	1.The number of homophone edit operations 2.The number of near-homophone edit operations 2.Total number of similar-shape edit operations 3.Total edit cost

Segmentation Features	1.The number of single words 2.The number of segmentations of words using MM 3.The number of segmentations of words using CKIP
Web Based Features	1.The search hits proportion of corrected part in title 2.The search hits proportion of corrected part in snippet

Language model features calculate the n-gram text probability of candidate sentences and the original sentence.

The n-gram language probability for a sentence S can be illustrated as the following equation:

$$P(S) = P(w_1, w_2 \ldots w_n) = P(w_1)P(w_2|w_1)\ldots P(w_n|w_1, w_2 \ldots w_{n-1}) \quad (5)$$

Here, $P(w_1)$ is probability of word w_1 appearing in the corpus and $P(w_n \mid w_1, w_2 \ldots w_{n-1})$ is the condition probability, which means the emergence probability of the word w_n under conditions of words $w_1, w_2 \ldots w_{n-1}$ appearing.

Dictionary features count the number and proportion of phrases and idioms in candidates after segmentation, according to our dictionaries. In addition, some other factors, *e.g*. phrase length, are also taken into account.

Here is an example of a traditional Chinese sentence: 根据/联合国/公布/的/数据. The sub-sentence has 4 phrases and 0 idioms, and the proportion of phrases and idioms are 0.8 and 0.0, respectively, based on dictionaries.

Edit distance features compute the edit number and its weight, from the original sentence to candidate sentences. Here, different edit operations are given different edit weights. For example, in our spelling correction system, we give homophone, near-homophone, and similar shape word different edit weights, which are determined by experience.

Segmentation features use the results of the Maximum Matching Algorithm and the CKIP Parser segmentation. In addition, we count the number of single words. As we know, inappropriate candidates containing spelling errors will tend to have more single words after segmentation.

Web based features use Bing or another search engine's search results when submitting the spelling correction part and the corresponding part of the original sentence to the search engine.

"经济特续增长"and its candidate sentence "经济持续增长" would be an example. When you search "经济特续" or "特续增长" and "经济持续" or "持续增长" using Bing, the search engine will return different hits.

In our framework, the re-ranking phase is a must, because the candidates generated by HMM are ordered only by n-gram language probability and edit distance and the optimal state

of the HMM is not necessarily the best candidate. So, we use more features to reorder the candidates to view the candidate sentences according to the actual quality of candidates as much as possible. This step can help to improve the performance of final spelling correction.

In order to verify the effectiveness of re-ranking, we give the performance, whether adopting re-ranking or not, through experiments in the fourth section of the paper.

3.3 Rule-based Correction for Errors

As illustrated in Figure 1, the third step conducts rule-based analysis for a specific correction task. Some common errors still are difficult to distinguish, such as the usage of three confusable words “的,” “地,” and “得”. In order to correct such errors, syntactic analysis must be developed. The following sentence contains an error of Chinese syntax:

今天/我/穿着/刚/买/地/新/衣服。

Here, the character “地” should be corrected to another character “的”. To deal with these kinds of errors, sentence parsing must be done to check and correct such errors before the syntactic rules are applied. We have summarized three rules of usage for “的,” “地,” and “得” according to Chinese grammar as follows.

The Chinese character “的” is the tag of attributes, which generally is used in front of subjects and objects. Words in front of “的” generally are used to modify or restrict things following “的”.

The Chinese character “地” is adverbial marker, usually used in front of predicates (verbs, adjectives). Words in front of “地” generally are used to describe actions following “地”.

The Chinese character “得” makes the complement and generally is used behind predicates. The part follows “得” generally is used to supplement the previous action.

In addition, some other specific rules are needed to improve the final performance, which can be concluded from the test data and corpus.

3.4 Decision-making on Corrections

Through the aforementioned processing steps, we choose the top-2 candidates for each sub-sentence. To make the final decision on spelling correction, some global constraints should be considered, which can be summarized into four categories.

First, the number of errors in sub-sentence candidates should be considered. If there are more than three errors in a sub-sentence, then we do not correct the sub-sentence. Second, we set different weights for different types of spelling errors by experience. For example,

syntactic errors need to be given more weight than others, as these errors are detected by some strong syntactic rules. Then, if the original sub-sentence is in its candidate set, the sub-sentence has a greater probability of being error-free. Finally, the ratio of corrected sentences to the total amount of checked sentences is also one of the factors to consider. This ratio relates to the average error rate of CFL essays.

Let $Candi_{sentence} = \{candi\_sub_1, candi\_sub_2, ..., candi\_sub_n\}$ be the candidate set of a sentence, and $candi\_sub_i$ be the top-2 candidates of its sub-sentence, $Final\_Candi = \{final\_candi\_sub_p, final\_candi\_sub_{p+1}, ..., final\_candi\_sub_q\}$ be the final candidate list of the sub-sentence in the intermediate process, and $Final\_Correction = \{final\_sub_1, final\_sub_2, ..., final\_sub_n\}$ be the final correction result.

According to the constraints above, our rules are summarized as follows.

1) Scan each element of $Candi_{sentence}$. If the number of errors of top-2 candidates in $candi\_sub_i$ is all more than 3 or the original sub-sentence is in $candi\_sub_i$ and ranked first after re-ranking, store the original sub-sentence in $Final\_Correction$ and continue scanning $candi\_sub_{i+1}$; otherwise, go to Step 2);
2) Compute the scores of the top-2 candidates in $candi\_sub_i$, and store the candidate with higher score in $Final\_Candi$. If the scan is not over, go to Step 1); otherwise, go to Step 3);
3) Provide statistics for the total number of errors in $Final\_Candi$. If the error quantity is less than the threshold value, then output $Final\_Candi$ to $Final\_Correction$ and skip to Step 5); otherwise, go to Step 4);
4) Sort the $Final\_Candi$ according to the score computed in Step 2). Scan $Final\_Candi$, output the front part of $Final\_Candi$ to $Final\_Correction$ according to the global error rate, and the remaining part of $Final\_Candi$ is not corrected, go to step 5);
5) Output the $Final\_Correction$.

In Step 2) above, there is a function to calculate the score of candidate, and the score can be computed as follows:

$$score(candidte) = edit\_weight + original\_weight - edit\_num \quad (6)$$

where $edit\_weight$ is the edit weight of the candidate, $original\_weight$ is the weight of whether the candidate is original sentence or not, and $edit\_num$ is the number of edits in candidate. The weights currently are set by experience. The value of $edit\_weight$ is set according to the error type. If the type is homophone or similar shape, $edit\_weight$ is set to 0.8, otherwise it is set to 0.5. The value of $original\_weight$ is also set by experience. If the candidate is original sentence, it is set to 1, otherwise it is set to 0.75.

On the basis of the above rules, we developed a rule-based classifier to get the final correction result of each sentence.

4. Experiment and Evaluation

4.1 Experimental Setting

In the experiment, 1062 traditional Chinese sentences with/without spelling errors were given, which were from CFL learners' essays. The error types in the sentences mainly resulted from three different categories, being homophone, near-homophone, or similar-shape. The test data was provided by SIGHAN 2014 Bake-off: Chinese Spelling Check Task.

As the test data set was based on traditional Chinese, we must consider building a traditional Chinese corpus to train our model. In our system, we use several corpora, including Taiwan Web as corpus; SogouW dictionary, which is a traditional Chinese dictionary translated from the simplified Chinese dictionary Sogou, a traditional Chinese dictionary of words and idioms; a pinyin table and a cangjie code table of common words; and some Web based resources. The details of the corpora are described below.

(1) Taiwan Web Pages as Corpus

Due to the difference in simplified Chinese and traditional Chinese, although we have a high quality simplified Chinese corpus, we do not translate the simplified corpus into a traditional corpus because the translation process may cause information loss, such as the fact that both"週末" and "周末" in traditional Chinese are translated into "周末" in simplified Chinese. Therefore, we try to find Taiwan webs whose pages contain high quality traditional Chinese text to build the corpus. We gathered pages from the artificially selected pages under the ".tw" domain, containing around 3.2 million web pages, to build the corpus. Then, the content extracted from these pages was used to build a traditional Chinese n-gram model, where n is from 2 to 4.

(2) SogouW Dictionary

SogouW dictionary is built from the statistical analysis of Chinese Internet corpus by Sogou Search Engine. It contains about 150,000 high-frequency words of the Chinese Internet. Nevertheless, words in the corpus are simplified Chinese characters that cannot be used directly. We first translated them into traditional Chinese via Google translation service.

(3) Chinese Words and Idioms Dictionary

As introduced in Chiu *et al.* (2013), we also obtained the Chinese words and Chinese idioms published by the Ministry of Education of Taiwan, which are built from dictionaries and related books. There are 64,326 distinct Chinese words and 48,030 distinct Chinese idioms. We combined these two dictionaries with the SogouW dictionary to build our trie tree dictionary.

(4) Pinyin and Cangjie Code Table

We collected more than 10000 pinyin forms of words commonly used in Taiwan to build

the homophone and near-homophone words table, which will be used in candidate generation phase. In addition, cangjie code can be used to measure the form/shape similarity between Chinese characters. Therefore, we collected cangjie codes to build the table of Similar-form characters.

(5) Web based Resources

We use the online CKIP Parser results to help rank the candidates. For example, the segmentation of "特續下滑" is "特/續/下滑" while "持續下滑" is "持續/下滑". Thus, the segmentation results of a wrong candidate sentence will have more words than the correct one.

In addition, we use the Bing search results as one feature in the candidate ranking phase, which clearly improves the performance. For example, the sentence "根據聯合國公布的數字" has several candidate sentences, one of which may be "根據聯合國公佈的數字". If we use Bing to search the error correction part and the corresponding part of the original sentence "聯合國公佈" and "聯合國公布," the search results will be clear enough to identify the correct candidate sentence, because the first one would be more popular than the second one on the web corpus.

4.2 Evaluation Results and Analysis

To evaluate the method we propose, a Chinese spelling check system was implemented. We have done some experiments to prove the effectiveness of our method for Chinese spelling correction. The task can be divided into two related subtasks. One is error detection and the other one is error correction. Chinese spelling error detection task aims to find out the location of the spelling errors in the sentences. The error correction task aims to correct the error words found in the error detection phase. There are five metrics, used to evaluate the performance of different methods. They are calculated as the following expression:

$$FPR(\text{FalsePositiveRate}) = \frac{FP}{FP+TN} \qquad A(\text{Accuracy}) = \frac{TP+TN}{TP+TN+FP+FN}$$

$$P(\text{Precision}) = \frac{TP}{TP+FP} \qquad R(\text{Re}call) = \frac{TP}{TP+FN}$$

$$F1-Score = \frac{2*P*R}{P+R}$$

where TP, FP, TN, and FN can be obtained from the confusion matrix in Table 3.

Table 3. Confusion Matrix.

Confusion Matrix		System Results	
		Positive (Error)	Negative (No Error)
Gold Standard	Positive	TP	FN
	Negative	FP	TN

11 competing teams joined the SIGHAN Bake-off 2014 and submitted their final results. These submitted methods are used to evaluate the performance of our proposed framework. NCTU & NTUT used a CRF-based parser and scored with a tri-gram LM; NCYU combined E-Hownet and n-gram models to construct the rule induction; NJUPT developed two CSC systems based on CRF model and 2 Chars & 3-Chars model, respectively; NTHU used a channel model and a character-based language model in the noisy model; SinicaCKIP combined the error template rules and n-gram models for Chinese spelling correction; the SJTU proposed an improved graph model based on a graph model for generic errors and two independently trained models for specific errors. The results of the two subtasks are described in detail in Sections 4.2.1 and 4.1.2.

In addition, we will analyze the effects of several features used in the ranking stage on the final results. The comparative results are introduced in Section 4.2.3.

4.2.1 Chinese Spelling Error Detection

The goal of this subtask is to detect whether a Chinese sentence contains errors or not. If the sentence contains errors, the subtask must point out the location of the error word. Table 4 shows the evaluation results of Chinese spelling error detection.

Table 4. Results of error detection subtask for different methods

Methods	A	P	R	F1
Decision-Making Model [CAS]	0.6149	0.7148	0.3823	0.4982
CRF-Model + N-gram model[NCTU& NTUT]	0.5028	0.5138	0.1055	0.175
Rule Induction [NCYU]	0.6008	0.8543	0.2429	0.3783
CRF-Model + N-gram Model [NJUPT]	0.403	0.3344	0.1959	0.247
Noisy Channel Model [NTHU]	0.4228	0.3677	0.2147	0.2711
Error Template Rule + N-gram Model [SinicaCKIP]	0.5367	0.5607	0.339	0.4225
Graph-Model + CRF-Model [SJTU]	0.5471	0.5856	0.322	0.4156

The above results illustrate that our system significantly outperforms other systems with submitted technique reports to the organizer in this subtask. This is due to our method using the extended HMM to guarantee the recall rate and introducing the re-rank phase combined with rich features to improve the precision.

4.2.2 Chinese Spelling Error Correction

The subtask is based on the task of error detection. The main idea is to correct the errors found in the detection phase. In this stage, each sentence will be corrected and compared to the reference answer. Our system showed good performance in this subtask. The error correction results are shown in Table 5.

Table 5. Results of error correction subtask for different methods

Methods	FPR	A	P	R	F1
Decision-Making Model [CAS]	0.1525	0.5829	0.676	0.3183	0.4328
CRF-Model + N-gram model[NCTU& NTUT]	0.0998	0.4925	0.4592	0.0847	0.1431
Rule Induction [NCYU]	0.0414	0.5885	0.8406	0.2185	0.3468
CRF-Model + N-gram Model [NJUPT]	0.3898	0.3964	0.3191	0.1827	0.2323
Noisy Channel Model [NTHU]	0.3691	0.3823	0.2659	0.1337	0.1779
Error Template Rule + N-gram Model [SinicaCKIP]	0.2655	0.5104	0.5188	0.2863	0.3689
Graph-Model + CRF-Model [SJTU]	0.2279	0.5377	0.5709	0.3032	0.3961

The results show that our system also provides good performance in the correction subtask. This is because it achieves good results in the detection subtask, which is the basis of the correction subtask.

4.2.3 The Influence of Different Ranking Features

In this part, we compare the effects of several features used in the ranking step on the final results. As the dictionary features and segmentation features are closely related, we ignore the comparison of segmentation features. In the experiment, we conducted the test over multiple rounds, where we excluded one kind of feature in each round. The test results are shown in Table 6.

Table 6. The effect of difference ranking features

Features (Excluded)	FPR	**Detection-Level**			
		A	P	R	F1
Language Model Features	0.2312	0.548	0.564	0.3153	0.4045
Dictionary Features	0.1523	0.5857	0.7068	0.3418	0.4608
Edit Distance Features	0.1726	0.5574	0.7003	0.3339	0.4522
Web Based Features	0.3663	0.5094	0.4401	0.3558	0.3935
None	0.1525	0.6149	0.7148	0.3823	0.4982
Features (Excluded)	FPR	**Correction-Level**			
		A	P	R	F1
Language Model Features	0.2312	0.5113	0.496	0.2398	0.3233
Dictionary Features	0.1523	0.5584	0.6709	0.2891	0.4041
Edit Distance Features	0.1726	0.5273	0.6612	0.2788	0.3923
Web Based Features	0.3663	0.4586	0.3485	0.2421	0.2857
None	0.1525	0.5829	0.676	0.3183	0.4328

Based on the results above, the language model features and web-based features are the two most important features in the ranking phase on the final results, as the two features mainly reflect the quality of web based corpus.

4.2.4 The Influence of Re-ranking

In this part, we verify the important role of re-ranking in the spelling correction. We correct the sentences in two ways, one only based on HMM and the other adopting re-ranking after generating candidates. Table 7 shows the final results.

Table 7. The correction results of whether adopting re-ranking or not

Error-Detection	A		P	R	F1
With Re-ranking	0.6149		0.7148	0.3823	0.4982
Without Re-ranking	0.4859		0.5156	0.2383	0.3259
Error-Correction	FPR	A	P	R	F1
With Re-ranking	0.1525	0.5829	0.676	0.3183	0.4328
Without Re-ranking	0.2441	0.4407	0.4038	0.1516	0.2205

As illustrated by the above results, re-ranking significantly improves the performance of results both in the error-detection and the error-correction tasks. In the error-detection task, the method with re-ranking outperforms the method without re-ranking with 19.92% improvement in precision and 14.4% improvement in recall rate. In the error-correction task, the precision and recall rate increase by 27.22% and 16.67%, respectively.

5. Conclusion

This paper proposes a unified framework (HANSpeller) for Chinese essay spelling correction based on extended HMM and ranker-based models. An extended HMM is proposed to generate candidate sentences for ranking. A rule-based strategy is used for further correction polishing and for a final decision on whether the output is the correction or not. Our approach was evaluated at the CLP-2014 bake-off on the Chinese spelling correction task, and it displayed good performance, ranking second among 13 teams.

Some interesting future work on Chinese spelling correction would include: (1) collecting and considering more error types in the candidates generating process and (2) how to better deal with the differences between traditional and simplified Chinese.

Acknowledgments

This research was supported by the National High Technology Research and Development Program of China (Grant No. 2014AA015204), the National Basic Research Program of China (Grant No. 2014CB340406), the NSFC for the Youth (Grant No. 61402442) and the Technology Innovation and Transformation Program of Shandong (Grant No.2014CGZH1103).

Reference

Ahmad, F., & Kondrak, G. (2005). Learning a spelling error model from search query logs. In *Proceeding of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 955-962.

Atwell, E. S., & Elliot, S. (1987). Dealing with ill-formed English text. T*he Computational Analysis of English: A Corpus-Based Approach*, 120-138.

Bril, E., & Moore, R. C. (2000). An improved error model for noisy channel spelling correction. In *Proceeding of the 38th Annual Meeting on Association for Computational Linguistics*, 286-293.

Chang, C. H. (1995). A new approach for automatic Chinese spelling correction. In *Proceeding of Natural Language Processing Pacific Rim Symposium*, 278-283.

Chang, R. Y., Wu, C. H., & Prasetyo, P. K. (2012). Error Diagnosis of Chinese Sentences Using Inductive Learning Algorithm and Decomposition-Based Testing Mechanism. *ACM Transactions on Asian Language Information Processing (TALIP)*, *11*(1), 3.

Chen, Y. Z. (2010). *Improve the detection of improperly used Chinese characters with noisy channel model and detection template* (Doctoral dissertation, Master thesis, Chaoyang University of Technology).

Chiu, H. W., Wu, J. C., & Chang, J. S. (2013). Chinese Spelling Checker Based on Statistical Machine Translation. In *Proceeding of the Seventh SIGHAN Workshop on Chinese Language Processing (SIGHAN-7)*, 49-53.

Chu, W. C., & Lin, C. J. (2014). NTOU Chinese Spelling Check System in CLP Bake-off 2014. In *Proceeding of the Third CIPS-SIGHAN Joint Conference on Chinese Language Processing*, 210-215.

Cucerzan, S., & Brill, E. (2004). Spelling Correction as an Iterative Process that Exploits the Collective Knowledge of Web Users. In *Proceeding of EMNLP*, 293-300.

Gao, J., Li, X., Micol, D., Quirk, C., & Sun, X. (2010). A large scale ranker-based system for search query spelling correction. In *Proceeding of the 23rd International Conference on Computational Linguistics*, 358-366.

Google. (2010). *A Java API for Google spelling check service*. http://code.google.com/p/google-api-spellingjava/

Gu, L., Wang, Y., & Liang, X. (2014). Introduction to NJUPT Chinese Spelling Check Systems in CLP-2014 Bakeoff. In *Proceeding of the Third CIPS-SIGHAN Joint Conference on Chinese Language Processing*, 167-172.

Hao, S., Gao, Z., Zhang, M., Xu, Y., Peng, H., Su, K., & Ke, D. (2013). Automated error detection and correction of chinese characters in written essays based on weighted finite-state transducer. In *Proceeding of the 12thInternational Conference on Document Analysis and Recognition (ICDAR)*, 763-767.

Hsieh, Y. M., Bai, M. H., & Chen, K. J. (2013). Introduction to CKIP Spelling Check System for SIGHAN Bakeoff 2013 Evaluation. In *Proceeding of the Seventh SIGHAN Workshop on Chinese Language Processing (SIGHAN-7)*, 59-63.

Huang, Q., Huang, P., Zhang, X., Xie, W. J., Hong, K., Chen, B. Z., & Huang, L. (2014). Chinese Spelling Check System Based on Tri-gram model. In *Proceeding of the Third CIPS-SIGHAN Joint Conference on Chinese Language Processing*, 173-178.

Huang, C. M., Wu, M. C., & Chang, C. C. (2007). Error detection and correction based on Chinese phonemic alphabet in Chinese text. In *Modeling Decisions for Artificial Intelligence*, 463-476.

Hung, T. H. (2009). *Automatic Chinese character error detecting system based on n-gram language model and pragmatics knowledge base* (Doctoral dissertation, Master thesis, Chaoyang University of Technology).

Hung, T. H., & Wu, S. H. (2008). Chinese essay error detection and suggestion system. In *Taiwan E-Learning Forum 2008*.

Jia, Z. Y., Wang, P. L., & Zhao, H. (2013). Graph Model for Chinese Spelling Checking. In *Proceeding of the Seventh SIGHAN Workshop on Chinese Language Processing (SIGHAN-7)*, 88-92

Jin, P., Chen, X., Guo, Z., & Liu, P. (2014). Integrating Pinyin to Improve Spelling Errors Detection for Chinese Language. In *Proceeding of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies*, 455-458.

Kukich, K. (1992). Techniques for automatically correcting words in text. *ACM Computing Surveys (CSUR)*, *24*(4), 377-439.

Levenshtein, V. I. (1966). Binary code capable of correcting detections, insertions, and reversals. *Soviet physics doklady*, *10*(8), 707-710.

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

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 (TALIP)*, *10*(2), 1-39.

Mangu, L., & Brill, E. (1997). Automatic rule acquisition for spelling correction. In *Proceeding of the 14th International Conference on Machine Learning*, 187-194.

Mays, E., Damerau, F. J., & Mercer, R. L. (1991). Context based spelling correction. *Information Processing & Management*, *27*(5), 517-522.

Microsoft Microsoft web n-gram services. (2010). http://research.microsoft.com/web-ngram

Suzuki, H., & Gao, J. (2012). A unified approach to transliteration-based text input with online spelling correction. In *Proceeding of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 609-618.

Wu, S. H., Liu, C. L., & Lee, L. H. (2013). Chinese Spelling Check Evaluation at SIGHAN Bake-off 2013. In *Proceeding of the Sixth International Joint Conference on Natural Language Processing*, 35-42.

Xiong, J., Zhang, Q., Hou, J., Wang, Q., Wang, Y., & Cheng, X. (2014). Extended HMM and Ranking models for Chinese Spelling Correction. *CLP 2014*, 133-138.

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

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

Zheng, Y., Li, C., & Sun, M. (2011). Chime: An efficient error-tolerant Chinese pinyin input method. In *Proceeding of International Joint Conference on Artificial Intelligence(IJCAI), 22*(3), 2551-2256.