Multifeature-Based Approach to Automatic Error Detection and Correction of Chinese Text

Lei Zhang[+]，**Ming Zhou**[++]，**Changning Huang**[+++]，**Haihua Pan**[++++]

[+] Dept. of Computer Science, Tsinghua University, Beijing, 100084, zhl@s1000e.cs.tsinghua.edu.cn

[++] Microsoft Research, China, Beijing, 100080, mingzhou@microsoft.com

[+++] Microsoft Research, China, Beijing, 100080, cnhuang@microsoft.com

[++++] Hong Kong City University, Hong Kong

Abstract

Language models adopted by most existing error detection and correction approaches of Chinese text are N-Gram models of character, word or POS tag. Their deficiencies are that only local language constraints are employed and there is no language model unification process. A multifeature-based automatic error detection and correction approach is presented. It uses both local language features and wide-scope semantic features. Winnow is adopted in the learning step. In experiment, this method gets an error detection recall rate of 85.1%, an error detection precision rate of 41.0%, and a correction rate of 51.2%. This approach shows better performance than existing approaches.

Keywords: automatic error detection and correction of Chinese text, natural language processing, Winnow

1. Introduction

Because Chinese is a very flexible language and there are no distinct delimiters between words in Chinese text, the automatic detection and correction of Chinese text errors is a very difficult task. Existing methods achieve error detecting recall rate 60%~70% and error detecting precision rate 20%~30%。Their deficiencies are obvious. (1) N-Gram model adopted by most approaches uses only local language constraints. No long-range constraints are introduced. (2)Many approaches that use POS tag N-Gram model in error checking always introduce an automatic POS tagging step before the checking procedure. This conceals many errors because not only the tagging step tries to find a path with the highest probability, but also the checking procedure uses same basis as the tagging step. (3) Many approaches only use one language model or use several models separately. There should be a language model unification process when multiple models are used.

We present a feature-based approach [4,5,7] to automatic error detection and correction of Chinese text to overcome the shortage. Feature so-called is linguistic pattern in the context of the target word or target character. There are two advantages of feature-based method. First, there are many kinds of features can be extracted from context. Different features can be used in different applications. Second, the unified language model is implemented by treating all features equally. Our method adopts four feature templates: adjoining word, POS class collocations, context semantic class, adjoining characters within a word. Because the feature space is extremely huge and any targets depend only on a small subset of the features in the space, Winnow method which was first used by Golding [6] in English context sensitive spelling check is applied in the learning step.

2. Task Model and Feature Template

"String" is used to represent a Chinese character or a Chinese word in this paper. The task of error detection and correction will be cast to as a string disambiguation task by introducing the concept of confusion set. The confusion set of a string s is $cfs(s)=\{y_1, y_2, ..., y_k\}$. Where y_k is a string. It means that each string y_k in the set is ambiguous with s. When string s appears in a sentence, we take it to be ambiguous among $\{s\} \cup cfs(s)$, the task is to choose the one $\tilde{y}$ that is actually intended upon the context. If $\tilde{y} \in cfs(s)$, we consider the appearance of string s as a mistake, and it should be corrected by $\tilde{y}$. Acquirng confusion sets is an interesting problem. In this paper, a target string's confusion set only contains characters and words whose five strokes input codes are similar as the target string.

Disambiguation is a progress of evaluating and selecting. What to be evaluated is how much the context suggests the target string. In feature-based approaches, a list of active features is used to represent the target string's context. A feature extractor is used to extract feautres from the context. Let the sentence to be checked is $S = W_1 W_2,...,W_n$, where W_j is words; and the target string s is either W_j or a character inside W_j. We use four types of features:

⑴**Adjoining word.** Features of this template record the previous and the next word of target string. They look like $W_{j-1}\underline{\&}$ and $\underline{\&}W_j$. Where $\underline{\&}$ means the target string.

⑵**POS class collocations**. As discussed in section 1, it's necessary to avoid automatic POS tagging process if POS information is used in spelling check task. POS class is therefore introduced. Let T is the set of all possible POS tags. Every subset of T is a POS class. Under this definition, words with same possible tags belong to same POS class. For example, all words that only have tags of Noun and Verb belong to Noun-Verb class. Features of this template look like $C_{j-2}C_{j-1}\underline{\&}, C_{j-1}\underline{\&}C_{j+1}, C_{j-1}\underline{\&}C_{j-2}$, Where C_k is the POS class of word W_k.

⑶**Context semantic class.** One word may also have several semantic tags. So the concept of semantic class is introduced using the similar definition and implementation as that of POS class. The initial semantic tags of words are from "*TongYiCiCiLin*"[1]. There are 6 levels of semantic tags and only the 4th level tags are used in this experiment to generate the semantic classes. The semantic classes of context words inside a window of $[\pm 2, \pm 6]$ are used as features. These features look like M_k. Where M_k is the semantic class of word W_k in the context window.

⑷**Adjoining characters within a word.** In Chinese text, an error word that consists two or more characters may be caused by an error of a single word, such as "他/喝/了/一/碗/粥" changes to "他/喝/了/一/碗/继续". It also may be caused by an error of a single character inside the word, such as "他/要/去/上海" changes to "他/要求/上海". The former kind of mistake can be checked out using above disambiguation model when "粥" is in the confusion set of "继续". Yet, under the later circumstance, the confusion set of string "要求" does not include the string "要 去" due to the definition of the confusion set. If we want to find out this kind of errors, each character inside the word "要求" should be checked individually. But when extracting the previous three kinds of features with the target character "求" inside the word "要求", the result is same as the target is "要求". To distinguish single character "求" and word "要求", adjoining characters within a word are adopted as another feature template. This kind of feature will be extracted only when the target string is a single character and appears in a word that consists two or more characters. Let x_i is the target character in $W_j = x_{i-p},...,x_i,...,x_{i+q}$. This kind of features look like $x_{i-1}\underline{\&}x_{i+1}$. Where the x_{i-1} and x_{i+1} are the previous and next character of x_i inside the word W_j respectively. If x_i is the first character of W_j, then x_{i-1} is set to *nil*. If x_i is the last character of W_j, then x_{i+1} is set to *nil.*

These templates of features capture important but complementary aspects of context. Not only local lexical atmosphere and syntax, but also long-range semantic constraints are included. The 4th kind of feature is designed specially for Chinese text.

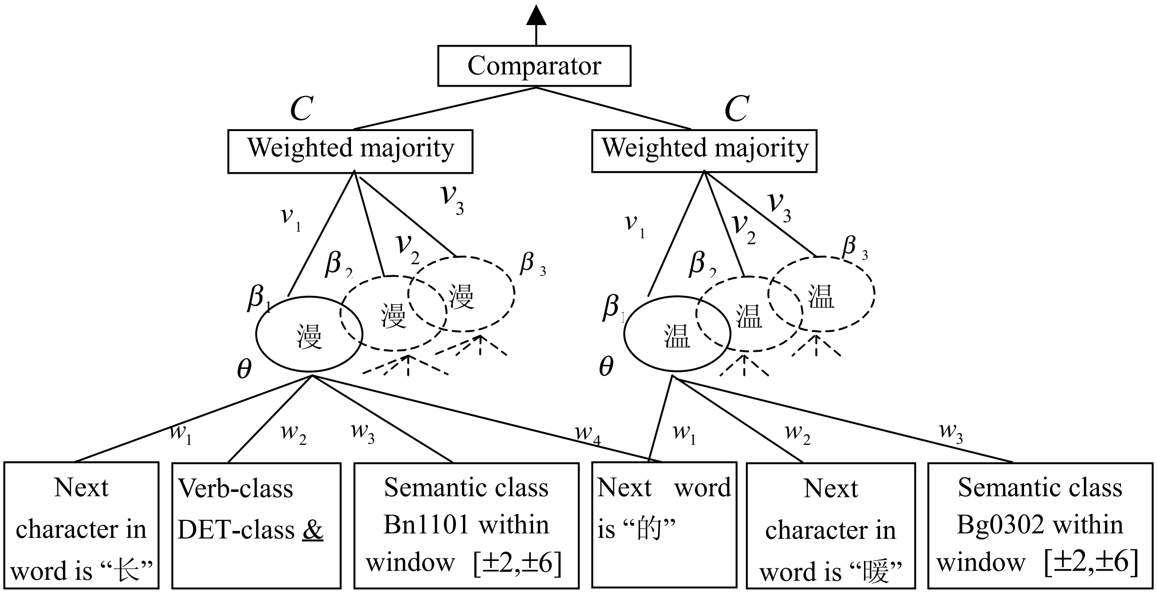

Figure 1. A simple model example

3. Algorithm model

Our algorithm based on Winnow model that is similar to Golding's. Figure 1 shows a simple model example to disambiguate Chinese character “温” and “漫”. In the middle of the figure, overlapping bubbles represent classifier of strings. Every classifier connects some features in the bottom. Each connection has a weight. Let θ_s be the classifier of string s. For a given active feature set F :

$$\theta_s(F) = 1 \Leftrightarrow \sum_{f \in F} w(f, \theta_s) > \varepsilon$$

where $w(f, \theta_s)$ is the weight between classifier θ_s and feature f . ε is a constant and is set to 1.0 in the experiment.

The first step of trainning is to establish the connections. Initially, there is no connection between features and classifiers. Let $S = W_1 W_2, ..., W_n$ is a segmented sentence. Each word W_k is treated as target and its active feature set F is extracted form the sentence, then we connect each features in F and the classifier of W_k with an initial weight. The same process is also used on every character x in W_k to establish connections between the classifier of x and features.

The second step of trainning is to update connection weights. Active features F extracted from the trainning sentence using string s as target is treated as a positive example for the classifier of s, and as a negative example for the classifiers of strings in the confusion set of s. Weights are updated only when classifiers make wrong predictions. If the classifier θ_s predicts 0 for a positive example F, then the weights are promoted:

$$\forall f \in F, w(f, \theta_s) \leftarrow \alpha \cdot w(f, \theta_s)$$

where $\alpha > 1$ is a promotion parameter. If the classifier θ_s predicts 1 for a negative example F, then the weights are demoted:

$$\forall f \in F, w(f, \theta_s) \leftarrow \beta \cdot w(f, \theta_s)$$

where $0 < \beta < 1$ is a demotion parameter.

Using diffirent parameter α, β, we can construct diffirent classifiers of a target. In this experiment, α is set to 1.5, β is set to 0.90, 0.75, 0.65 to get three classifiers of each string. It is resonable to give more trust to the classifier that has better performance in the trainning. So, the *j*th classifier θ^j is assigned a weight $v_j = \gamma^{m_j}$. Where $0 < \gamma < 1$ is a constant, and m_j is the total number of mistakes made by the classifier in trainning. When combining evidences from several classifiers using diffirent parameters, the final score assigned by the weighted majority is:

$$C_s(F) = (\sum_j v_j \theta_s^j(F))/(\sum_j v_j)$$

The checking step is to determine whether an appearance of target string s is suitable or not in a segmented sentence $S = W_1 W_2, ..., W_n$. Where string s is either a word W_k or any character in these words. First, active features F of string s is extracted from the sentence. Then a string $\tilde{y} \in \{s\} \cup cfs(s)$ is selected to satisfy:

$$C_{\tilde{y}}(F) = \underset{y \in \{s\} \cup cfs(s)}{MAX} C_{\tilde{y}}(F)$$

There are two situiations where the appearance of s is treated as a mistake:

(1) $s \neq \tilde{y}$. At this circumstance, $\tilde{y}$ should be suggested as a correction of s.

(2) $s = \tilde{y}$, but

$$\sum_j (v_j \sum_{f \in F} w(f, \theta_s^j))/(\sum_j v_j) < \psi .$$

Where $0 < \psi < \varepsilon$. ψ is set to 0.3 in this paper. In this situation, although s is more suitable for the context than any string in its confusion set, the intension with which the context support it is too little. Yet we can not give the suggestion about how to correct this kind of error at present.

4 Experimental results

The trainning corpus consists *People's daily 93-94*, *Market 94*, *BaiJiaBao 94*. It's about 200M bytes in size. Due to the difficulty for collecting text that contains real errors, our test is based on both real errors and synthesized errors.

In the synthesized error test, five strings are selected at random as the target to check. The target strings are “温”, “罕”, “做”, “模式” and “信念”. For each target, five kinds of error sentences are generated by a comuter program. (1)For a sentence that contains an element of a target's confusion set, use the target string to replace the element. This kind of error is called confusion set substitution error and represented as CS substitution in Table 1. (2)For any sentence, use a target string to replace a character at random position. This is called random substitution error and represented as R substitution in Table 1. (3) For a sentence that contains a target string, delete the left or right character of the target. This is called deletion error. (4)For a sentence that contains a single-character target, duplicate the character. This is called duplication error. (5)For any sentence, insert a single-character target string into a random positon. This is called insertion error. Table 1 shows the check result of these errors. The approach also gives proper corrections to 796 confusion set substitution error sentences. In addition, we try to check 31180 right sentences that contains tagets to be checked and 1905 of these sentences are marked containing error by our approach. Assuming the propotions of each kind of errors and the propotions of the error and no-error occurrences here is similar as that of the real text, the approach gets an error detection recall rate of 85.1%, an error detection precision rate of 41.7% and an correction rate of 51.2%.

In the real error test, we select 20 targets to check. The test paper is an 15000 characters news text input with five strokes input method. In this paper, the 20 targets appear 443 times, including 43 times of error. Our approach marks 32 errors. Among them, 21 places are real errors. The error detection recall rate is 87.5% and the error detection precision rate is 65.5%. The approach also gives proper correction to 16 error appearances of the targets, the correction rate is 65.6%. When only a character trigram model are used to check this paper, the error detection recall rate and precision rate are only 70.8% and 32.1% respectively.

This approach achieves better performance than

Error type	Number of error sentence	Number of Marked sentence	Recall rate(%)
CS substitution	1119	1070	95.6
R substitution	191	170	89.0
Deletion	125	21	16.8
Duplication	60	12	20.0
Insertion	60	50	83.3

Table 1. Check result of synthesized errors

those using simple N-gram model. The improvement comes mainly from the following aspects: ⑴The adoption of context semantic class features makes it possible to find the errors that can not be detected by local language constraints. There are many errors that can not be detected using any local language contraints. For example:

当空气中的[湿]温度稍微大一些时，蜘蛛…

我虽然没有[介入]做过他们的争执，…

In these sentences, the underlined characters or words are wrong and their right forms are inside the square brackets. It is obvious that these errors do not cause any local liguistic abnormity. Yet in our test, many of this kind of errors are correctly detected and corrected. This is the outcome of using context semantic information. ⑵ Multiple features that capture important but complementary aspects of context compensate the deficiencies of each single kind of feature. ⑶The introduction of confusion set makes the task model of automatic Chinese text error detection and correction more explicit. ⑷Winnow method is suitable for the learning step in the spelling check task.

The problems of our approach are data sparseness and space complexity. Insufficient trainning is one aspect of the Data sparseness. Especially when words with high frequency and words with very low frequency both belong to one confusion set. At this circumstance, many correction sugguestion given by our approach prefer words with high freqency even the right correction should be the word with very low frequency.

5. Conclusion

A multifeature-based approach to automatic error detection and correction of Chinese text is implemented. It has the following advantages: ⑴It adopts four feature templates that capture important but complementary aspects of context: adjoining word, POS class collocations, context semantic class, adjoining characters within a word. ⑵The concept of POS class resolves the problem that both the automatic tagging procedure and the checking procedure use the same basis. ⑶The introduction of confusion set makes the task model more explicit. ⑷ Winnow method is suitable for the learning step in the spelling check task. In experiment, the approach shows better performance than existing approaches.

The problems of data sparseness and space complexity should be studied deeply in the future.

References

1 Mei Jiaju, Zhu Yiming, Gao Yunqi, et al. *TongYiCiCiLin*. Shanghai: Shanghai Lexicon Publishing Company, 1983

2 Zhang Zhaohuang. A Pilot Study on Automatic Chinese Spelling Error Correction. Communication of COLIPS,1994,4(2):143-149

3 Sun Cai. Research on Lexical Error Detection and Correction of Chinese Text: [Master's Degree Dissertation]. Beijing: Tsinghua University Computer Science and Technology Department,1997

4 Gale W A, Church K W, Yarowsky D. A method for disambiguating word senses in a large corpus. Computers and the Humanities, 1993,26:415-439

5 Golding A R. A Bayesian hybrid method for context-sensitive spelling correction. In: Proc. 3rd Workshop on Very Large Corpora, Boston, MA:1995

6 Golding A R, Dan R. Applying Winnow to context-sensitive spelling correction. In: Proc. the 13th ICML, Bari, Italy:1996

7 Yarowsky D. Decision list for lexical ambiguity resolution: Application to accent restoration in Spanish and French. In: Proc. 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, NM:1994