Automatic Chinese Confusion Words Extraction Using Conditional Random Fields and the Web

Chun-Hung Wang
Department of Computer Science
National Tsing Hua University
mars@cs.nthu.edu.tw

Jason S. Chang
Department of Computer Science
National Tsing Hua University
jason.jschang@gmail.com

Jian-Cheng Wu
Department of Computer Science
National Tsing Hua University
wujc86@gmail.com

Abstract

A ready set of commonly confused words plays an important role in spelling error detection and correction in texts. In this paper, we present a system named ACE (Automatic Confusion words Extraction), which takes a Chinese word as input (e.g., "不脛而走") and automatically outputs its easily confused words (e.g., "不徑而走", "不逕而走"). The purpose of ACE is similar to web-based *set expansion* – the problem of finding all instances (e.g. "Halloween", "Thanksgiving Day", "Independence Day", etc.) of a set given a small number of class names (e.g. "holidays"). Unlike *set expansion*, our system is used to produce commonly confused words of a given Chinese word. In brief, we use some hand-coded patterns to find a set of sentence fragments from search engine, and then assign an array of tags to each character in each sentence fragment. Finally, these tagged fragments are served as inputs to a pre-learned conditional random fields (CRFs) model. We present experiment results on 3,211 test cases, showing that our system can achieve 95.2% precision rate while maintaining 91.2% recall rate.

1 Introduction

Since many Chinese characters have similar forms and similar or identical pronunciation, improperly used characters in Chinese texts are quite common. Previous works collected these hard-to-distinguish characters to form confusion sets (Ren et al., 1994). Confusion sets are pretty helpful for online detecting and correcting improperly used Chinese characters in precision and speed. Zhang et al. (2000) build a confusion set based on a Chinese input method named Wubi. The basic assumption is that characters that have similar input sequences must have similar forms. Therefore, by replacing one code in the input sequence of a certain character, the system could generate characters with similar forms. Lin et al. (2002) used the Cangjie input method to generate confusion sets under the same assumption in Zhang et al. Another approach is to manually edit the confusion set. Hung manually compiled 6,701 common errors from different sources (Hung and Wu, 2008). These common errors were collected from essays of junior high school students and were used in Chinese character error detection and correction.

Since the cost of manual compilation is high, Chen et al. (2009) proposed an automatic method that can collect these common errors from a corpus. The idea is similar to template generation, which builds a question-answer system (Ravichandran and Hovy, 2001; Sung et al., 2008). The template generation method investigates a large corpus and mines possible question-answer pairs. In this paper, we present ACE system to automatically extract commonly confused words from the Web of a given word. Table 1 shows some examples of ACE's input and output.

input	兵荒馬亂	三令五申	伶牙俐齒
output	兵**慌**馬亂	三令五**伸** 三令五**聲** 三**申**五令	伶牙**利**齒 **靈**牙**利**齒

Table 1: Examples of ACE's input and output.

This paper is organized as follows. Section 2 illustrates the architecture of ACE. Section 3 explains the features we use for training model. Section 4 presents evaluation results. The last section summarizes this paper and describes our future work.

Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing (SIGHAN-7), pages 64–68,
Nagoya, Japan, 14 October 2013.

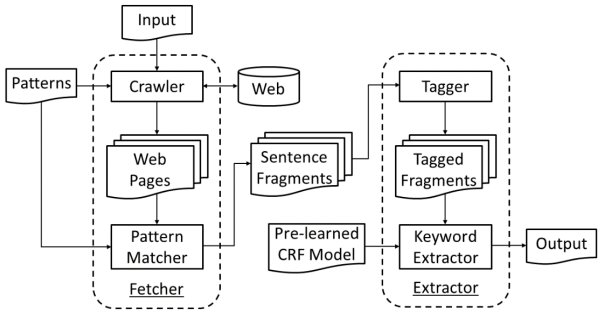

Figure 1: Flow chart of the ACE System.

2 System Architecture

ACE consists of two major components: the Fetcher and the Extractor. Given a Chinese word (assume it is correct), the Fetcher retrieves snippets from Google using hand-coded patterns, and then executes the pattern matching process to produce a set of sentence fragments. The Extractor is responsible for assigning an array of tags to each character in every sentence fragment depends on its features. These tagged fragments are served as inputs to a pre-learned CRFs model (see Section 3) for extracting commonly confused words of the input word. In this section, we will describe the Fetcher and the Extractor in more detail.

2.1 The Fetcher

The Fetcher first constructs a few query strings by using the combination of input word and a set of pre-defined patterns. Table 2 shows our query strings and their English translations.

Type I	<**w**> 誤作	<**w**> be misused as
	<**w**> 寫成	<**w**> be written as
	<**w**> 誤為	<**w**> be misused as
	<**w**> 不是	<**w**> not
Type II	應為 <**w**>	should be <**w**>
	應作 <**w**>	should be <**w**>
	改為 <**w**>	be revised as <**w**>

Table 2: Type I and Type II query strings and their English translations. In each query string, <**w**> is a placeholder for the input word.

There are two types of query strings: Type I are the ones that require the input word **w** to precede the pattern (e.g. "**w**寫成"), and Type II are the opposite ones (e.g. "應作**w**"). For every query, the Fetcher retrieves several Web pages of results from Google where each page contains up to 100 snippets due to Google's restriction. For each snippet, the Fetcher removes its HTML tags and extracts sentence fragments which contain the input word and possibly contain incorrect words with the help of regular expression. These sentence fragments are inputs of the Extractor we will describe later. For Type I query results, sentence fragment is orderly composed by 0 to 6 characters (including Chinese characters, alphanumeric symbols, punctuation marks, etc.), the input word, and 1 to n characters where n is the number of characters of the input word plus 14. For Type II query results, sentence fragment is orderly composed by 1 to n characters, the input word, and 0 to 6 characters. Table 3 shows some examples of extracted sentence fragments of the input word "不脛而走".

Type I
目。復原**不脛而走**"*誤作*"不徑而走"（'97
「**不脛而走**」*寫成*「不徑而走」－Y
Type II
"不徑而走"*應為*"**不脛而走**"big
月 5 日－不徑而走*應作***不脛而走**. 峻工

Table 3: Examples of sentence fragments of the input word "不脛而走". For clarification purposes, we make the input word bold and italicize the pattern.

2.2 The Extractor

The Extractor first assigns an array of tags to each character in every sentence fragment derived from the Fetcher by its features. We may assign up to four tags to each character according to system configurations. Table 4 shows an example of fully tagged fragment. Tag I denotes that this character is in the instance of the input word or not. Tag II and Tag III are pronunciation-related features, indicating pronunciation similarity between this character and any character of the input word. Tag IV is orthographic similarity between this character and any character of the input word. Meanings of tags and how to assign tags to characters will be detailed in Section 4.

After sentence fragments are tagged, these tagged fragments are served as inputs to a pre-learned CRFs model for labeling easily confused words of the input word. Finally, the Extractor combines these labeled characters into words, and then ranks these words based on frequency.

ACE outputs first few ranked words depend on system settings. Let a $= \langle a_1, a_2, \ldots, a_n \rangle$ be the set of ranked words and $f(a_i)$ denotes the frequency of a_i, $f(a_1) \geq f(a_2) \geq \cdots \geq f(a_n)$.

ACE outputs $a' = \langle a_1, \ldots, a_j \rangle$ where $1 \leq j \leq i$ and $a_k \geq C * a_{k-1}$ for each $a_k \in a'$. The default value of C is 0.3 and can be configured in the system. Some example inputs and outputs are listed in Table 1, and Section 6 shows more examples.

characters	Tag I	Tag II	Tag III	Tag IV
“	N	O	O	O
不	N	Y	Y	Y
徑	N	N	N	N
而	N	Y	Y	Y
走	N	Y	Y	Y
”	N	O	O	O
應	N	O	O	O
為	N	O	O	O
“	N	O	O	O
不	Y	Y	Y	Y
脛	Y	Y	Y	Y
而	Y	Y	Y	Y
走	Y	Y	Y	Y
”	N	O	O	O

Table 4: An example of fully tagged fragment.

3 Features Set

One property that makes feature based statistical models like CRFs so attractive is that they reduce the problem to finding an appropriate feature set. This section outlines the four main types of features used in our evaluations.

3.1 Base Feature

One of simplest and most obvious features is the character itself of sentence fragment. Another intuitive feature is that the character is included in the input word (tagged as “Y”) or not (tagged as “N”). More accurately, let $\mathbf{o} = \langle o_1, o_2, \ldots, o_n \rangle$ be a sequence of characters of sentence fragment. Let $\mathbf{w} = \langle w_1, w_2, \ldots, w_m \rangle$ be a sequence of characters of the input word. $\mathbf{w} \subset \mathbf{o}$. For each $o_i \in \mathbf{o}$, we tag o_i as “Y” if $o_i \in \mathbf{w}$, otherwise tag o_i as “N”. In our experiments, we define the combination of those two features as base feature.

3.2 Sound Feature

Liu (2009) previously showed that pronunciation-related errors reach 79.88% among all types of incorrect writings in Chinese. This feature has three tag values: “Y”, “N”, and “O”. We continuously use notations of Section 4.1. Let $U_w = \langle u_{w_1}, u_{w_2}, \ldots, u_{w_m} \rangle$ where u_{w_i} denotes the sound of w_i. Let u_{o_i} denotes the sound of o_i. For each $o_i \in \mathbf{o}$, we tag o_i as “Y” if $o_i \in \mathbf{w}$, else tag o_i as “N” if $u_{o_i} \in U_w$, otherwise tag o_i as “O”.

We build up a look-up table for quickly access a character’s sound. Table 5 is the list of characters grouped by sound. Note that characters in the same group may have different tones. We will consider the feature of same sound and same tone in Section 4.3.

sound	characters
suan	酸 痠 狻 匳 算 蒜 筭
wai	歪 舀 外
zai	哉 災 載 宰 仔 崽 綷 在 再 載

Table 5: Characters grouped by sound.

3.3 Phonetic Alphabet Feature

This feature differentiates two characters with same sound but different tone from each other. Let $H_w = \langle h_{w_1}, h_{w_2}, \ldots, h_{w_m} \rangle$ where h_{w_i} denotes the phonetic symbol of w_i. Let h_{o_i} denotes the phonetic symbol of o_i. For each $o_i \in \mathbf{o}$, we tag o_i as “Y” if $o_i \in \mathbf{w}$, else tag o_i as “N” if $h_{o_i} \in H_w$, otherwise tag o_i as “O”. Table 6 is the list of characters grouped by phonetic alphabet.

phonetic alphabet	characters
suān	酸 痠 狻
suǎn	匳
suàn	算 蒜 筭
wāi	歪
wǎi	舀
wài	外

Table 6: Characters grouped by phonetic alphabet.

3.4 Orthography Feature

In addition to pronunciation-related features, the model could also benefit from orthographical similarity features. We have collected a list of 12,460 Chinese characters accompanied by a group of orthographically similar characters for each from Academic Sinica of Taiwan[1]. Two characters are considered to be orthographically similar according to their forms. In this list, each character may have more than one similar character. Let $\mathbf{r}_{w_i} = \langle r_{w_{i1}}, r_{w_{i2}}, \ldots, r_{w_{ik}} \rangle$ be a set of orthographically similar characters of w_i. Let $\mathbf{R}_w = \langle \mathbf{r}_{w_1}, \mathbf{r}_{w_2}, \ldots, \mathbf{r}_{w_m} \rangle$ be the collection of $\mathbf{r}_{w_i}$.

[1] http://cdp.sinica.edu.tw/cdphanzi/

For each $o_i \in \mathbf{o}$, we tag o_i as "Y" if $o_i \in \mathbf{w}$, else tag o_i as "N" if $o_i \in \mathbf{R}_w$, otherwise tag o_i as "O". Table 7 is the list of characters accompanied by their orthographically similar characters.

character	similar characters
亨	烹 哼 膨 京 享
佐	仜 左 佈 傞 倥 佑
別	捌 咧 唎 喇

Table 7: Characters and their orthographically similar characters.

4 Experiments

In this section, we describe the details of CRFs model training and evaluation. Secondly, we will compare performance of ACE system with two manually compiled confusion sets which can be anonymously accessed online.

4.1 Model Training and Testing

We obtained data set from a document named Terms Unified Usage[2] provided by National Science Council of Taiwan. This document contains 641 correct-and-incorrect word pairs. We randomly selected 577 of them for training and the rest for testing. For each word pair, we constructs query strings to retrieve sentence fragments by using the method described in Section 2.1, and then assigns tags to each character in every sentence fragment by using the method described in Section 2.2. In addition, we tagged target label (e.g. B-I, I-I, O) to each character for the purpose of training and evaluation.

There are 17,019 sentence fragments which containing 126,130 characters in training data, and 1,252 sentence fragments which containing 15,767 characters in testing data. Eight experiments were completed by different combinations of features. Detailed results are presented in table 7 (in next page). In Table 7, characters precision denotes number of correctly labeled characters divided by number of total characters in the testing data. Similarly, sentences precision denotes number of correctly labeled sentences (every character in sentence is correctly labeled) divided by number of total sentence. Since the output of ACE is a ranked list of extracted words, we set 0.3 to constant C (see Section 2.2) to compute precision ratio, recall ratio, and F_1 measure. More precisely, let:

- {A}=incorrect words indicated in Terms Unified Usage
- {B}=incorrect words extracted by ACE

Then, precision ratio P = |{A}∩{B}|/|{B}|*100%, recall ratio R = |{A}∩{B}|/|{A}|*100%, and F_1 measure = 2*P*R/(P+R).

From the result, the CRFs model using the combination of sound and orthography features or using all features performs best, achieving F_1 measure of 94.6%.

4.2 Comparisons to Manually Compiled Confusion Sets

We collected two manually compiled confusion sets for the purpose of comparisons. One is the *Common Error in Chinese Writings*[3] (CECW) provided by Ministry of Education (MOE) of Taiwan, which containing 1,491 correct-and-incorrect word pairs. Another is the *Commonly Misused Characters for Middle School Students*[4] (CMC), which containing 1,720 correct-and-incorrect word pairs. We feed these correct words to ACE system to evaluate the ability of automatic generation of confusion sets. We choose features combinations of "base + S + G" and set constant C to 0.3. Table 8 summarizes the evaluation results, showing that given a Chinese word, ACE system has about 93% chance to produce same result with manually compiled confusion sets.

	Precision	Recall	F_1 measure
CECW	95.2%	91.2%	93.2%
CMC	93.8%	92.0%	92.8%

Table 8: Evaluation results on two confusion sets.

input	output
滄海一粟	滄海一栗
半晌	半餉 半响
發憤圖強	發奮圖強 奮發圖強
掃描	掃瞄
彆扭	蹩扭 憋扭 變扭 辯扭

Table 9: Examples of ACE's input and output.

[2] http://www.nsc.gov.tw/sd/uniword.htm

[3] http://dict.revised.moe.edu.tw/htm/biansz/18a-1.htm

[4] http://kitty.2y.idv.tw/~mars/cset.xlsx

	Characters Precision	Sentences Precision	Extracted Words		
			Precision	Recall	F_1 measure
base only	94.1%	66.1%	89.7%	73.2%	80.6%
base + Sound (S)	97.6%	83.0%	89.7%	77.3%	83.0%
base + Phonetic (P)	97.9%	85.7%	93.1%	87.5%	90.2%
base + Orthography (G)	97.9%	86.6%	89.7%	85.6%	87.6%
base + S + P	97.7%	84.1%	89.7%	89.3%	89.5%
base + S + G	98.8%	92.4%	96.6%	92.7%	94.6%
base + P + G	98.9%	92.8%	96.6%	89.3%	92.8%
base + S + P + G	98.8%	92.9%	96.6%	92.7%	94.6%

Table 7: Test results by different combinations of features.

5 Conclusions and Future Work

In this paper, we present the ACE system which takes a Chinese word as input and automatically outputs its easily confused words. Table 9 shows some real examples of ACE's input and output. We have shown that a CRF-based model with pronunciation- and orthography-related features can achieve performance near that manually compiled confusion sets.

There are several future topics of research that we are currently considering. First, we plan to extend ACE system to support other languages, such as English and Japanese. Secondly, we will investigate another approach without the help of a pre-learned CRFs model. Third, we will look into automatic identification of possible words which can be easily misused as another one, so that we can generate confusion sets without any input. Lastly, we will apply our approach to another application, such as recognizing as many as entity pairs (e.g., <"Tokyo", "Japan">, <"Taipei", "Taiwan">, etc.) of a given semantic relation (e.g. "… is a city of …").

References

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. *Proceedings of the Conference on Natural Language Learning*, 188–191.

Chao-Lin Liu, Kan-Wen Tien, Min-Hua Lai, Yi-Hsuan Chuang, and Shih-Hung Wu. 2009. Phonological and logographic influences on errors in written Chinese words. *Proceedings of the Seventh Workshop on Asian Language Resources, the Forty Seventh Annual Meeting of the Association for Computational Linguistics*, 84-91.

Cheng-Lung Sung, Cheng-Wei, Lee, Hsu0Chun Yen, and Wen-Lian Hsu. 2008. An Alignment-based Surface Pattern for a Question Answering System. *IEEE International Conference on Information Reuse and Integration*, 172-177.

Deepak Ravichandran and Eduard Hovy, E. 2001. Learning surface text patterns for a Question Answering system. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 41-47.

Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference*, 134-141.

Fuji Ren, Hongchi Shi, and Qiang Zhou. 1994. A hybrid approach to automatic Chinese text checking and error correction. In *Proceedings of the ARPA Work shop on Human Language Technology*, 76-81.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning.*

Lei Zhang, Changning Huang, Ming Zhou, and Haihua Pan. 2000. Automatic detecting/correcting errors in Chinese text by an approximate word-matching algorithm. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, 248-254.

Ta-Hung Hung and Shih-Hung Wu. 2008. Chinese essay error detection and suggestion system. *Taiwan E-Learning Forum.*

Yih-Jeng Lin, Feng-Long Huang, and Ming-Shing Yu. 2002. A Chinese spelling error correction System. In *Proceedings of the Seventh Conference on Artificial Intelligence and Applications* (TAAI).

Yong-Zhi Chen, Shih-Hung Wu, Chia-Ching Lu, and Tsun Ku. 2009. Automatic template feneration for Chinese essay spelling error detecting system. *The 13th Global Chinese Conference on Computer in Education*, 402-408.