A Study of Language Modeling for Chinese Spelling Check

Kuan-Yu Chen[†*]**, Hung-Shin Lee,**
Chung-Han Lee, Hsin-Min Wang
[†]Academia Sinica, Taiwan
{kychen, hslee, chlee012, whm}@iis.sinica.edu.tw

Hsin-Hsi Chen
[*]National Taiwan University, Taiwan
hhchen@ntu.edu.tw

Abstract

Chinese spelling check (CSC) is still an open problem today. To the best of our knowledge, language modeling is widely used in CSC because of its simplicity and fair predictive power, but most systems only use the conventional *n*-gram models. Our work in this paper continues this general line of research by further exploring different ways to glean extra semantic clues and Web resources to enhance the CSC performance in an unsupervised fashion. Empirical results demonstrate the utility of our CSC system.

1 Introduction

Chinese is a tonal syllabic and character (symbol) language, in which each character is pronounced as a tonal syllable. A Chinese "word" usually comprises two or more characters. The difficulty of Chinese processing is that many Chinese characters have similar shapes or similar (or same) pronunciations. Some characters are even similar in both shape and pronunciation (Wu *et al.*, 2010; Liu *et al.*, 2011). However, the meanings of these characters (or words composed of the characters) may be widely divergent. Due to this reason, all the students in elementary school in Taiwan or the foreign Chinese learners need to practice to identify and correct "erroneous words" in a Chinese sentence, which is called the Incorrect Character Correction (ICC) test. In fact, the ICC test is not a simple task even for some adult native speakers in Taiwan.

Since most Chinese characters have other characters similar to them in either shape or pronunciation, an intuitive idea for CSC is to construct a confusion set for each character. Currently, many CSC systems use the confusion sets (Zhang *et al.*, 2000; Wu *et al.*, 2010; Liu *et al.*, 2011) to recursively substitute characters and find an optimal result to detect and correct erroneous words. Moreover, many researches have been focusing on automatically constructing the confusion sets from various knowledge sources, such as the Cangjie code (Liu *et al.*, 2011), psycholinguistic experimental results (Kuo *et al.*, 2004; Lee *et al.*, 2006; Tsai *et al.*, 2006), and templates generated from a large corpus (Chen *et al.*, 2009). Language modeling can be used to quantify the quality of a given word string, and most previous researches have adopted it as a method to predict which word might be a correct word to replace the possible erroneous word.

Although language modeling has been widely used in CSC, most researches only use the conventional *n*-gram models. To the best of our knowledge, the *n*-gram language models, aiming at capturing the local contextual information or the lexical regularity of a language, are inevitably faced with two fundamental problems. On one hand, it is brittle across domains, and the performance of the model is sensitive to changes in the genre or topic of the text on which it is trained. On the other hand, it fails to capture the information (either semantic or syntactic information) conveyed beyond the *n*-1 immediately preceding words. In view of these problems, this paper focuses on exploring the long-span semantic information for language modeling for CSC. Moreover, we make a step forward to incorporate a search engine to provide extra information from the Web resources to make a more robust system.

The rest of this paper is organized as follows. In Section 2, we briefly review the *n*-gram and topic language models. Section 3 details our proposed CSC system. A series of experiments are presented in Section 4. Finally, conclusions and future work are given in Section 5.

2 Language Modeling

2.1 *N*-gram Language Modeling

From the early 20th century, statistical language modeling has been successfully applied to various applications related to natural language processing (NLP), such as speech recognition (Chen and Goodman, 1999; Chen and Chen, 2011), information retrieval (Ponte and Croft, 1998; Lavrenko and Croft, 2001; Lavrenko, 2009), document summarization (Lin and Chen, 2010), and spelling correction (Chen *et al.*, 2009; Liu *et al.*, 2011; Wu *et al.*, 2010). The most widely-used and well-practiced language model, by far, is the *n*-gram language model (Jelinek, 1999), because of its simplicity and fair predictive power. Quantifying the quality of a word string in a natural language is the most commonly executed task. Take the tri-gram model for example, when given a word string $W_1^L = w_1, w_2, \ldots, w_L$, the probability of the word string is approximated by the product of a series of conditional probabilities as follows (Jelinek, 1999),

$$\begin{aligned} P(W_1^L) &= P(w_1)\prod_{l=2}^{L} P(w_l \mid W_1^{l-1}) \\ &\approx P(w_1)P(w_2 \mid w_1)\prod_{l=3}^{L} P(w_l \mid w_{l-2}, w_{l-1}). \end{aligned} \tag{1}$$

In the tri-gram model, we make the approximation (or assumption) that the probability of a word depends only on the two immediately preceding words.

The easiest way to estimate the conditional probability in Eq. (1) is to use the maximum likelihood (ML) estimation as follows,

$$P(w_l \mid w_{l-2}, w_{l-1}) = \frac{c(w_{l-2}, w_{l-1}, w_l)}{c(w_{l-2}, w_{l-1})}, \tag{2}$$

where $c(w_{l-2}, w_{l-1}, w_l)$ and $c(w_{l-2}, w_{l-1})$ denote the number of times the word strings “ w_{l-2}, w_{l-1}, w_l ” and “ w_{l-2}, w_{l-1} ” occur in a given training corpus, respectively. Without loss of generality, the tri-gram model can be extended to higher order models, such as the four-gram model and the five-gram model, but the high-order *n*-gram models usually suffer from the data sparseness problem, which leads to some zero conditional probabilities. Various language model smoothing techniques have been proposed to deal with the zero probability problem. For example, Good-Turing (Chen and Goodman, 1999), Kneser-Ney (Chen and Goodman, 1999), and Pitman-Yor (Huang and Renals, 2007) are well-known state-of-the-art smoothing approaches. The general formulation of these approaches is (Chen and Goodman, 1999):

$$\begin{aligned} &P(w_l \mid w_{l-n+1}, \ldots, w_{l-1}) \\ &= \begin{cases} f(c(w_{l-n+1}, \ldots, w_l)) & , \ \mathit{if}\ c(w_{l-n+1}, \ldots, w_l) \neq 0 \\ \beta(w_{l-n+1}, \ldots, w_{l-1})\, f(c(w_{l-n+1}, \ldots, w_l)), & \mathit{if}\ c(w_{l-n+1}, \ldots, w_l) = 0 \end{cases} \end{aligned} \tag{3}$$

where $f(\cdot)$ denotes a discounting probability function and $\beta(\cdot)$ denotes a back-off weighting factor that makes the distribution sum to 1.

2.2 Topic Modeling

The *n*-gram language model, aiming at capturing only the local contextual information or the lexical regularity of a language, is inevitably faced with the problem of missing the information (either semantic or syntactic information) conveyed by the words before the *n*-1 immediately preceding words. To mitigate the weakness of the *n*-gram model, various topic models have been proposed and widely used in many NLP tasks. We can roughly organize these topic models into two categories (Chen *et al.*, 2010): document topic models and word topic models.

2.2.1 Document Topic Modeling (DTM)

DTM introduces a set of latent topic variables to describe the “word-document” co-occurrence characteristics. The dependence between a word and its preceding words (regarded as a document) is not computed directly based on the frequency counts as in the conventional *n*-gram model, but instead based on the frequency of the word in the latent topics as well as the likelihood that the preceding words together generate the respective topics. Probabilistic latent semantic analysis (PLSA) (Hofmann, 1999) and latent Dirichlet allocation (LDA) (Blei *et al.*, 2003; Griffiths and Steyvers, 2004) are two representatives of this category. Take PLSA for example, we can interpret the preceding words, $W_1^{L-1} = w_1, w_2, \ldots, w_{L-1}$, as a document topic model used for predicting the occurrence probability of w_L :

$$\begin{aligned} &P_{\text{PLSA}}(w_L \mid W_1^{L-1}) \\ &\quad = \Sigma_{k=1}^{K} P(w_L \mid T_k) P(T_k \mid W_1^{L-1}), \end{aligned} \tag{4}$$

where T_k is the *k*-th latent topic; $P(w_L \mid T_k)$ and $P(T_k \mid W_1^{L-1})$ are respectively the probability that the word w_L occurs in T_k and the probability of T_k conditioned on the preceding word string W_1^{L-1}. The latent topic distribution $P(w_L \mid T_k)$ can be estimated beforehand by maximizing the total log-likelihood of the training corpus. However, the preceding word string varies with context, and thus the corresponding topic mixture weight $P(T_k \mid W_1^{L-1})$ has to be estimated on the fly using inference algorithms like the expectation-maximization (EM) algorithm.

On the other hand, LDA, having a formula analogous to PLSA, is regarded as an extension to PLSA and has enjoyed much success for various NLP tasks. LDA differs from PLSA mainly in the inference of model parameters (Chen *et al.*, 2010). PLSA assumes that the model parameters are fixed and unknown while LDA places additional a priori constraints on the model parameters by thinking of them as random variables that follow some Dirichlet distributions. Since LDA has a more complex form for model optimization, which is hardly to be solved by exact inference, several approximate inference algorithms, such as the variational approximation algorithm, the expectation propagation method (Blei *et al.*, 2003), and the Gibbs sampling algorithm (Griffiths and Steyvers, 2004), have been proposed for estimating the parameters of LDA.

2.2.2 Word Topic Modeling (WTM)

Instead of treating the preceding word string as a document topic model, we can regard each word w_l of the language as a word topic model (WTM) (Chen, 2009; Chen *et al.*, 2010). To crystalize this idea, all words are assumed to share the same set of latent topic distributions but have different weights over the topics. The WTM model of each word w_l in W_1^{L-1} for predicting the occurrence of a particular word w_L can be expressed by:

$$\begin{aligned} &P_{\text{WTM}}(w_L \mid \mathrm{M}_{w_l}) \\ &\quad = \Sigma_{k=1}^{K} P(w_L \mid T_k) P(T_k \mid \mathrm{M}_{w_l}). \end{aligned} \tag{5}$$

Each WTM model M_{w_l} can be trained in a data-driven manner by concatenating those words occurring within the vicinity of each occurrence of w_l in a training corpus, which are postulated to be relevant to w_l. To this end, a sliding window with a size of *S* words is placed on each occurrence of w_l, and a pseudo-document associated with such vicinity information of w_l is aggregated consequently. The WTM model of each word can be estimated by maximizing the total log-likelihood of words occurring in their associated "vicinity documents" using the EM algorithm. Notice that the words in such a document are assumed to be independent of each other (the so-called "*bag-of-words*" assumption). When we calculate the conditional probability $P(w_L \mid W_1^{L-1})$, we can linearly combine the associated WTM models of the words occurring in W_1^{L-1} to form a composite WTM model for predicting w_L:

$$\begin{aligned} &P_{\mathrm{WTM}}(w_L \mid W_1^{L-1}) \\ &\quad = \Sigma_{l=1}^{L-1} \alpha_l \cdot P_{\mathrm{WTM}}(w_L \mid \mathrm{M}_{w_l}), \end{aligned} \tag{6}$$

where the values of the nonnegative weighting coefficients α_l are empirically set to decay exponentially with *L-l* and sum to one (Chen, 2009).

Word vicinity model (WVM) (Chen *et al.*, 2010) bears a certain similarity to WTM in its motivation of modeling the "word-word" co-occurrences, but has a more concise parameterization. WVM explores the word vicinity information by directly modeling the joint probability of any word pair in the language, rather than modeling the conditional probability of one word given the other word as in WTM. In this regard, the joint probability of any word pair that describes the associated word vicinity information can be expressed by the following equation, using a set of latent topics:

$$\begin{aligned} &P_{\mathrm{WVM}}(w_i, w_j) \\ &\quad = \Sigma_{k=1}^{K} P(w_i \mid T_k) P(T_k) P(w_j \mid T_k), \end{aligned} \tag{7}$$

where $P(T_k)$ is the prior probability of a given topic T_k. Notice that the relationship between words, originally expressed in a high-dimensional probability space, are now projected into a low-dimensional probability space characterized by the shared set of topic distributions. Along a similar vein, WVM is trained by maximizing the probabilities of all word pairs, respectively, co-occurring within a sliding window of *S* words in the training corpus, using the EM algorithm. To calculate the conditional probability $P(w_L \mid W_1^{L-1})$, we first obtain the conditional probability $P(w_L \mid w_l)$ from the joint probability $P(w_L, w_l)$ by,

$$\begin{aligned} &P_{\mathrm{WVM}}(w_L | w_l) \\ &\quad = \frac{\Sigma_{k=1}^{K} P(w_L \mid T_k) P(T_k) P(w_l \mid T_k)}{\Sigma_{k=1}^{K} P(w_l \mid T_k) P(T_k)}. \end{aligned} \tag{8}$$

Then, a composite WVM model $P_{\mathrm{WVM}}(w_L \mid W_1^{L-1})$ is obtained by linearly combining $P_{\mathrm{WVM}}(w_L \mid w_l)$, as in WTM.

2.3 Other Language Models

In addition to topic models, many other language modeling techniques have been proposed to complement the *n*-gram model in different ways, such as recurrent neural network language modeling (RNNLM) (Tomáš *et al.*, 2010), discriminative training language modeling (DLM) (Roark *et al.*, 2007; Chen et al., 2012), and relevance modeling (RM) (Lavrenko and Croft, 2001; Chen and Chen, 2011; Chen and Chen, 2013). RNNLM tries to project W_1^{L-1} and w_L into a continuous space, and estimate the conditional probability in a recursive way by incorporating the full information about W_1^{L-1}. DLM takes an objective function corresponding to minimizing the word error rate for speech recognition or maximizing the ROUGE score for summarization as a holy grail and updates the language model parameters to achieve the goal. RM assumes that each word sequence W_1^L is associated with a relevance class *R*, and all the words in W_1^L are samples drawn from *R*. It usually employs a local feedback-like procedure to obtain a set of pseudo-relevant documents to approximate *R* in the practical implementation.

3 The Proposed CSC System

3.1 System Overview

Figure 1 shows the flowchart of our CSC system. The system is mainly composed by three components: text segmenters, confusion sets, and language models. It performs CSC in the following steps:

1. Given a test word string, the CSC system treats the string as a query and posts it to a search engine to obtain a set of query suggestions.
2. Both the original word string and query suggestions will be segmented by using the maximum matching algorithm.
3. After segmentation, we assume that only the single-character words can be erroneous, so the system will iteratively substitute these words with possible characters by referring to the confusion sets.
4. Finally, the system will calculate the probability for each possible word string (by using the *n*-gram model, topic models, or both), and the most likely word string will be chosen as the final output.

3.2 Word Segmentation

Although the CKIP Chinese word segmentation system (Ma, 2003) is a famous and widely-used tool for the NLP community in Taiwan, we are aware that it has implemented an automatically merging algorithm, which might merge some error characters to a new word. To avoid the unexpected result, we have implemented our own forward and backward word segmentation tools based on the maximum matching algorithm. Given a word string, the CSC system will perform both forward and backward word segmentation, and then both forward

Table 1. Results of our CSC system.

	Sub-task1				Sub-task 2		
	Detection Accuracy	Detection F-score	Error Location F-score	False-Alarm Rate	Location Accuracy	Correction Accuracy	Correction Precision
Tri-gram	0.654	0.607	0.368	0.447	0.507	0.467	0.467
Tri-gram + Search Engine	0.835	0.739	0.458	0.141	0.489	0.445	0.445
Tri-gram + Search Engine + PLSA	0.836	0.741	0.467	0.141	0.494	0.450	0.450

and backward language models are applied to calculate the probabilities of the string.

3.3 Confusion Sets

The confusion sets are constructed from a pre-defined confusion corpus (Wu *et al.*, 2010; Liu *et al.*, 2011) and augmented by referring to the basic units of Chinese characters. We calculate the Levenshtein distance between any pair of Chinese characters based on their Cangjie codes. If the distance is smaller than a pre-defined threshold, the character pair is added to the confusion sets.

3.4 Language Modeling

Although language modeling has been widely used in the CSC task, most researches only use the conventional *n*-gram models. In this work, we evaluate the tri-gram language model as well as various topic models in our CSC system. The *n*-gram model and topic model are combined by a simple linear interpolation. Our lexicon consists of 97 thousand words. The tri-gram language model was estimated from a background text corpus consisting of over 170 million Chinese characters collected from Central News Agency (CNA) in 2001 and 2002 (the Chinese Gigaword Corpus released by LDC) and Sinica Corpus using the SRI Language Modeling Toolkit (Stolcke, 2000) with the Good-Turing smoothing technique. The topic models were also trained by using the same text corpus with 32 latent topics. Due to the space limitation, only the results with the PLSA topic model will be reported in the paper. Our preliminary experiments show that all the topic models discussed in Section 2 achieve similar performance.

3.5 Search Engine

In addition to topic models, we have also incorporated Web information in our CSC system by using a search engine. Given a test word string, our system treats the string as a query and posts it to a search engine to obtain a set of query suggestions. These query suggestions will also be treated as candidates. We use Baidu (http://www.baidu.com/) as the search engine.

4 Experimental Results

The experiments include two sub-tasks: error detection and error correction. All the experimental materials are collected from students' written essays. The first sub-task focuses on the evaluation of error detection. The input word string might consist of no error to evaluate the false-alarm rate of a system. The evaluation metrics include the detection accuracy, detection F-score, error location F-score, and false-alarm rate. As can be seen from the left part of Table 1, the tri-gram language model (denoted as "Tri-gram") can achieve a certain level of performance. Incorporating the suggestions from a search engine (denoted as "Tri-gram+Search Engine") in the CSC system yields significant improvements over Tri-gram in all evaluation metrics. Further incorporating topic modeling (denoted as "Tri-gram+Search Engine+PLSA") can slightly improve the detection F-score and error location F-score. The results demonstrate that the Web information is an indispensable reference for error detection, and the topic models can further improve the precision and recall rate without increasing the false alarm rate.

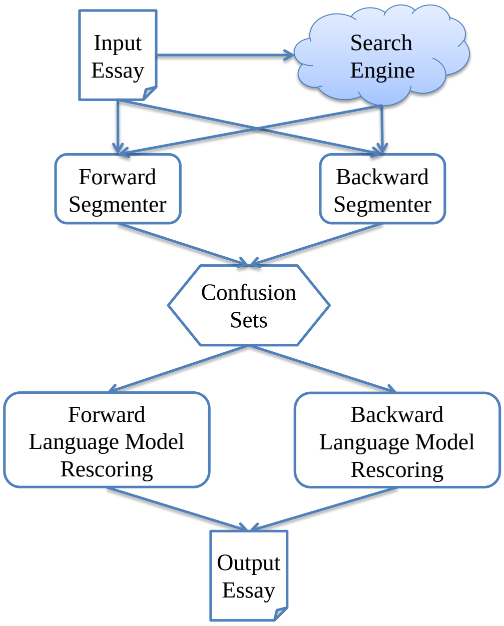

Figure 1. The flowchart of the CSC system.

The second sub-task focuses on the evaluation of error correction. Each sentence includes at least one error. The evaluation metrics are the location accuracy, correction accuracy, and correction precision. The experimental results are listed in the right part of Table 1. To our surprise, Web information and the PLSA topic model cannot complement the conventional tri-gram model to achieve better performance. The reasons could be two-fold. First, we do not have a sufficient set of development documents to select a reasonable interpolation weight between the tri-gram model and the topic model. Second, the confusion sets should be further modified by some unsupervised or supervised methods to separate the wheat from the chaff.

5 Conclusions & Future Work

This paper has proposed a systematic way to render the semantic clues and Web resources to improve the performance of Chinese spelling check. The experimental results have demonstrated that our proposed system can improve error detection in terms of detection accuracy, detection F-score, error location F-score, and false-alarm rate. Our future research directions include: 1) investigating more elaborate language models for CSC, 2) seeking the use of discriminative training algorithms for training language models to directly optimize the detection and correction performance, and 3) applying and exploring unsupervised or supervised methods to construct the confusion sets.

References

Andreas Stolcke. 2000. SRI Language Modeling Toolkit (http://www.speech.sri.com/projects/srilm/).

Berlin Chen. 2009. Word Topic Models for Spoken Document Retrieval and Transcription. *ACM Transactions on Asian Language Information Processing,* Vol. 8, No. 1, pp. 2:1-2:27.

Berlin Chen, and Kuan-Yu Chen. 2013. Leveraging Relevance Cues for Language Modeling in Speech Recognition. *Information Processing & Management,* Vol. 49, No. 4, pp. 807-816.

Brian Roark, Murat Saraclar, and Michael Collins. 2007. Discriminative *N*-gram Language Modeling. *Computer Speech and Language*, Vol. 21, No. 2, pp. 373-392.

Chao-Lin Liu, Min-Hua Lai, Kan-Wen Tien, Yi-Hsuan Chuang, Shih-Hung Wu, and Chia-Ying Lee. 2011. Visually and Phonologically Similar Characters in Incorrect Chinese Words: Analyses, Identification, and Applications. *ACM Transactions on Asian Language Information Processing,* Vol. 10, No. 2, pp. 1-39.

Chia-Ying Lee, Jie-Li Tsai, Hsu-Wen Huang, Daisy L. Hung, and Ovid J.L. Tzeng. 2006. The Temporal Signatures of Semantic and Phonological Activations for Chinese Sublexical Processing: An Event-Related Potential Study. *Brain Research,* Vol. 1121, No. 1, pp. 150-159.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research,* Vol. 3, pp. 993-1022.

Frederick Jelinek. 1999. *Statistical Methods for Speech Recognition*. The MIT Press.

Jay M. Ponte and W. Bruce Croft. 1998. A Language Modeling Approach to Information Retrieval. In Proceedings of SIGIR.

Jie-Li Tsai, Chia-Ying Lee, Ying-Chun Lin, Ovid J. L. Tzeng, and Daisy L. Hung. 2006. Neighborhood Size Effects of Chinese Words in Lexical Decision and Reading. *Language & Linguistics,* Vol. 7, No. 3, pp. 659-675.

Kuan-Yu Chen, Hsin-Min Wang, and Berlin Chen. 2012. Spoken Document Retrieval Leveraging Unsupervised and Supervised Topic Modeling Techniques. *IEICE Transactions on Information and Systems,* Vol. E95-D, No. 5, pp. 1195-1205.

Kuan-Yu Chen, and Berlin Chen. 2011. Relevance Language Modeling for Speech Recognition. In Proceedings of ICASSP.

Kuan-Yu Chen, Hsuan-Sheng Chiu, and Berlin Chen. 2010. Latent Topic Modeling of Word Vicinity Information for Speech Recognition. In Proceedings of ICASSP.

Lei Zhang, Ming Zhou, Changning Huang, and Mingyu Lu. 2000. Approach in Automatic Detection and Correction of Errors in Chinese Text based on Feature and Learning. In Proceedings of WCICA.

Mikolov Tomáš, Karafiát Martin, Burget Lukáš, Černocký Jan and Khudanpur Sanjeev. 2010. Recurrent Neural Network based Language Model. In Proceedings of INTERSPEECH.

Shih-Hsiang Lin and Berlin Chen. 2010. A Risk Minimization Framework for Extractive Speech Summarization. In Proceedings of ACL.

Shih-Hung Wu, Yong-Zhi Chen, Ping-che Yang, Tsun Ku, and Chao-Lin Liu. 2010. Reducing the False Alarm Rate of Chinese Character Error Detection and Correction. In Proceedings of SIGHAN.

Song-Fang Huang and Steve Renals. 2007. Hierarchical Pitman-Yor Language Models for ASR in Meetings. In Proceedings of ASRU.

Stanley F. Chen and Joshua Goodman. 1999. An Empirical Study of Smoothing Techniques for Language Modeling. *Computer Speech and Language,* Vol. 13, No. 4, pp. 359-393.

Thomas L. Griffiths and Mark Steyvers. 2004. Finding Scientific Topics. In Proceedings of PNAS.

Thomas Hofmann. 1999. Probabilistic Latent Semantic Analysis. In Proceedings of UAI.

Victor Lavrenko and W. Bruce Croft. 2001. Relevance-based Language Models. In Proceedings of SIGIR.

Wei-Yun Ma and Keh-Jiann Chen. 2003. Introduction to CKIP Chinese Word Segmentation System for the First International Chinese Word Segmentation Bakeoff. In Proceedings of SIGHAN.

W.-J. Kuo, T.-C. Yeh, J.-R. Lee, L.-F. Chen, P.-L. Lee, S.-S. Chen, L.-T. Ho, D.-L. Hung, O.J.-L. Tzeng, and J.-C. Hsieh. 2004. Orthographic and Phonological Processing of Chinese Characters: An fMRI Study. In Proceedings of NeuroImage.

Yong-Zhi Chen, Shih-Hung Wu, Chia-Ching Lu, and Tsun Ku. 2009. Chinese Confusion Word Set for Automatic Generation of Spelling Error Detecting Template. In Proceedings of ROCLING.