A New Statistical Approach to Chinese Pinyin Input

Zheng Chen
Microsoft Research China
No. 49 Zhichun Road Haidian District
100080, China,
zhengc@microsoft.com

Kai-Fu Lee
Microsoft Research China
No. 49 Zhichun Road Haidian District
100080, China,
kfl@microsoft.com

Abstract

Chinese input is one of the key challenges for Chinese PC users. This paper proposes a statistical approach to Pinyin-based Chinese input. This approach uses a trigram-based language model and a statistically based segmentation. Also, to deal with real input, it also includes a typing model which enables spelling correction in sentence-based Pinyin input, and a spelling model for English which enables modeless Pinyin input.

1. Introduction

Chinese input method is one of the most difficult problems for Chinese PC users. There are two main categories of Chinese input method. One is shape-based input method, such as "wu bi zi xing", the other is Pinyin, or pronunciation-based input method, such as "Chinese CStar", "MSPY", etc. Because of its facility to learn and to use, Pinyin is the most popular Chinese input method. Over 97% of the users in China use Pinyin for input (Chen Yuan 1997). Although Pinyin input method has so many advantages, it also suffers from several problems, including Pinyin-to-characters conversion errors, user typing errors, and UI problem such as the need of two separate mode while typing Chinese and English, etc.

Pinyin-based method automatically converts Pinyin to Chinese characters. But, there are only about 406 syllables; they correspond to over 6000 common Chinese characters. So it is very difficult for system to select the correct corresponding Chinese characters automatically. A higher accuracy may be achieved using a sentence-based input. Sentence-based input method chooses character by using a language model base on context. So its accuracy is higher than word-based input method. In this paper, all the technology is based on sentence-based input method, but it can easily adapted to word-input method.

In our approach we use statistical language model to achieve very high accuracy. We design a unified approach to Chinese statistical language modelling. This unified approach enhances trigram-based statistical language modelling with automatic, maximum-likelihood-based methods to segment words, select the lexicon, and filter the training data. Compared to the commercial product, our system is up to 50% lower in error rate at the same memory size, and about 76% better without memory limits at all (Jianfeng etc. 2000).

However, sentence-based input methods also have their own problems. One is that the system assumes that users' input is perfect. In reality there are many typing errors in users' input. Typing errors will cause many system errors. Another problem is that in order to type both English and Chinese, the user has to switch between two modes. This is cumbersome for the user. In this paper, a new typing model is proposed to solve these problems. The system will accept correct typing, but also tolerate common typing errors. Furthermore, the typing model is also combined with a probabilistic spelling model for English, which measures how likely the input sequence is an English word. Both models can run in parallel, guided by a Chinese language model to output the most likely sequence of Chinese and/or English characters.

The organization of this paper is as follows. In the second section, we briefly discuss the

Chinese language model which is used by sentence-based input method. In the third section, we introduce a typing model to deal with typing errors made by the user. In the fourth section, we propose a spelling model for English, which discriminates between Pinyin and English. Finally, we give some conclusions.

2. Chinese Language Model

Pinyin input is the most popular form of text input in Chinese. Basically, the user types a phonetic spelling with optional spaces, like:

woshiyigezhongguoren

And the system converts this string into a string of Chinese characters, like:

我是一个中国人 (I am a Chinese)

A sentence-based input method chooses the probable Chinese word according to the context. In our system, statistical language model is used to provide adequate information to predict the probabilities of hypothesized Chinese word sequences.

In the conversion of Pinyin to Chinese character, for the given Pinyin P, the goal is to find the most probable Chinese character H, so as to maximize $\Pr(H \mid P)$. Using Bayes law, we have:

$$\hat{H} = \arg\max_{H} \Pr(H \mid P) = \arg\max_{H} \frac{\Pr(P \mid H)\Pr(H)}{\Pr(P)} \qquad (2.1)$$

The problem is divided into two parts, typing model $\Pr(P \mid H)$ and language model $\Pr(H)$.

Conceptually, all H 's are enumerated, and the one that gives the largest $\Pr(H, P)$ is selected as the best Chinese character sequence. In practice, some efficient methods, such as Viterbi Beam Search (Kai-Fu Lee 1989; Chin-hui Lee 1996), will be used.

The Chinese language model in equation 2.1, $\Pr(H)$ measures the a priori probability of a Chinese word sequence. Usually, it is determined by a statistical language model (SLM), such as Trigram LM. $\Pr(P \mid H)$, called typing model, measures the probability that a Chinese word H is typed as Pinyin P.

Usually, H is the combination of Chinese words, it can decomposed into $w_1, w_2, \cdots, w_n$, where w_i can be Chinese word or Chinese character. So typing model can be rewritten as equation 2.2.

$$\Pr(P \mid H) \approx \prod_{i=1}^{n} \Pr(P_{f(i)} \mid w_i), \qquad (2.2)$$

where, $P_{f(i)}$ is the Pinyin of w_i.

The most widely used statistical language model is the so-called n-gram Markov models (Frederick 1997). Sometimes bigram or trigram is used as SLM. For English, trigram is widely used. With a large training corpus trigram also works well for Chinese. Many articles from newspapers and web are collected for training. And some new filtering methods are used to select balanced corpus to build the trigram model. Finally, a powerful language model is obtained. In practice, perplexity (Kai-Fu Lee 1989; Frederick 1997) is used to evaluate the SLM, as equation 2.3.

$$PP = 2^{-\frac{1}{N}\sum_{i=1}^{N}\log P(w_i \mid w_{i-1})} \qquad (2.3)$$

where N is the length of the testing data. The perplexity can be roughly interpreted as the geometric mean of the branching factor of the document when presented to the language model. Clearly, lower perplexities are better.

We build a system for cross-domain general trigram word SLM for Chinese. We trained the system from 1.6 billion characters of training data. We evaluated the perplexity of this system, and found that across seven different domains, the average per-character perplexity was 34.4. We also evaluated the system for Pinyin-to-character conversion. Compared to the commercial product, our system is up to 50% lower in error rate at the same memory size, and about 76% better without memory limits at all. (JianFeng etc. 2000)

3. Spelling Correction

3.1 Typing Errors

The sentence-based approach converts Pinyin into Chinese words. But this approach assumes correct Pinyin input. Erroneous input will cause errors to propagate in the conversion. This problem is serious for Chinese users because:

1. Chinese users do not type Pinyin as frequently as American users type English.
2. There are many dialects in China. Many people do not speak the standard Mandarin Chinese dialect, which is the origin of Pinyin. For example people in the southern area of China do not distinguish ‘zh’-‘z’, ‘sh’-‘s’, ‘ch’-‘c’, ‘ng’-‘n’, etc.
3. It is more difficult to check for errors while typing Pinyin for Chinese, because Pinyin typing is not WYSIWYG. Preview experiments showed that people usually do not check Pinyin for errors, but wait until the Chinese characters start to show up.

3.2 Spelling Correction

In traditional statistical Pinyin-to-characters conversion systems, $\Pr(P_{f(i)} \mid w_i)$, as mentioned in equation 2.2, is usually set to 1 if $P_{f(i)}$ is an acceptable spelling of word w_i, and 0 if it is not. Thus, these systems rely exclusively on the language model to carry out the conversion, and have no tolerance for any variability in Pinyin input. Some systems have the “southern confused pronunciation” feature to deal with this problem. But this can only address a small fraction of typing errors because it is not data-driven (learned from real typing errors). Our solution trains the probability of $\Pr(P_{f(i)} \mid w_i)$ from a real corpus.

There are many ways to build typing models. In theory, we can train all possible $\Pr(P_{f(i)} \mid w_i)$, but there are too many parameters to train. In order to reduce the number of parameters that we need to train, we consider only single-character words and map all characters with equivalent pronunciation into a single syllable. There are about 406 syllables in Chinese, so this is essentially training: $\Pr(\mathit{Pinyin\ String} \mid \mathit{Syllable})$, and then mapping each character to its corresponding syllable.

According to the statistical data from psychology (William 1983), most frequently errors made by users can be classified into the following types:

1. Substitution error: The user types one key instead of another key. This error is mainly caused by layout of the keyboard. The correct character was replaced by a character immediately adjacent and in the same row. 43% of the typing errors are of this type. Substitutions of a neighbouring letter from the same column (column errors) accounted for 15%. And the substitution of the homologous (mirror-image) letter typed by the same finger in the same position but the wrong hand, accounted for 10% of the errors overall (William 1983).
2. Insertion errors: The typist inserts some keys into the typing letter sequence. One reason of this error is the layout of the keyboard. Different dialects also can result in insertion errors.
3. Deletion errors: some keys are omitted while typing.
4. Other typing errors, all errors except the errors mentioned before. For example, transposition errors which means the reversal of two adjacent letters.

We use models learned from psychology, but train the model parameters from real data, similar to training acoustic model for speech recognition (Kai-Fu Lee 1989). In speech recognition, each syllable can be represented as a hidden Markov model (HMM). The pronunciation sample of each syllable is mapped to a sequence of states in HMM. Then the transition probability between states can be trained from the real training data. Similarly, in Pinyin input each input key can be seen as a state, then we can align the correct input and actual input to find out the transition probability of each state. Finally, different HMMs can be used to model typists with different skill levels.

In order to train all 406 syllables in Chinese, a lot of data are needed. We reduce this data requirement by tying the same letter in different syllable or same syllable as one state. Then the number of states can be reduced to 27 (26 different letters from ‘a’ to ‘z’, plus one to represent the unknown letter which appears in the typing letters). This model could be integrated into a Viterbi beam search that utilizes a trigram language model.

3.3 Experiments

Typing model is trained from the real user input. We collected actual typing data from 100 users, with about 8 hours of typing data from each user. 90% of this data are used for training and remaining 10% data are used for testing. The character perplexity for testing corpus is 66.69, and the word perplexity is 653.71.

We first, tested the baseline system without spelling correction. There are two groups of input: one with perfect input (which means instead of using user input); the other is actual input, which contains real typing errors. The error rate of Pinyin to Hanzi conversion is shown as table 3.1.

	Error Rate
Perfect Input	6.82%
Actual Input	20.84%

Table 3.1 system without spelling correction

In the actual input data, approximately 4.6% Chinese characters are typed incorrectly. This 4.6% error will cause more errors through propagation. In the whole system, we found that it results in tripling increase of the error rate from table 3.1. It shows that error tolerance is very important for typist while using sentence-based input method. For example, user types the Pinyin like: wisiyigezhonguoren (我是一个中国人), system without error tolerance will convert it into Chinese character like: wi肆意各种u哦人.

Another experiment is carried out to validate the concept of adaptive spelling correction. The motivation of adaptive spelling correction is that we want to apply more correction to less skilled typists. This level of correction can be controlled by the “language model weight”(LM weight) (Frederick 1997; Bahl etc. 1980; X. Huang etc. 1993). The LM weight is applied as in equation 3.1.

$$\hat{H} = \arg\max_{H} \Pr(H \mid P) = \arg\max_{H} \Pr(P \mid H)\Pr(H)^{a} \text{ ,}$$

where a is the LM weight. (3.1)

Using the same data as last experiment, but applying the typing model and varying the LM weight, results are shown as Figure 3.1.

As can be seen from Figure 3.1, different LM weight will affect the system performance. For a fixed LM weight of 0.5, the error rate of conversion is reduced by approximately 30%. For example, the conversion of “wisiyigezhonguoren” is now correct.

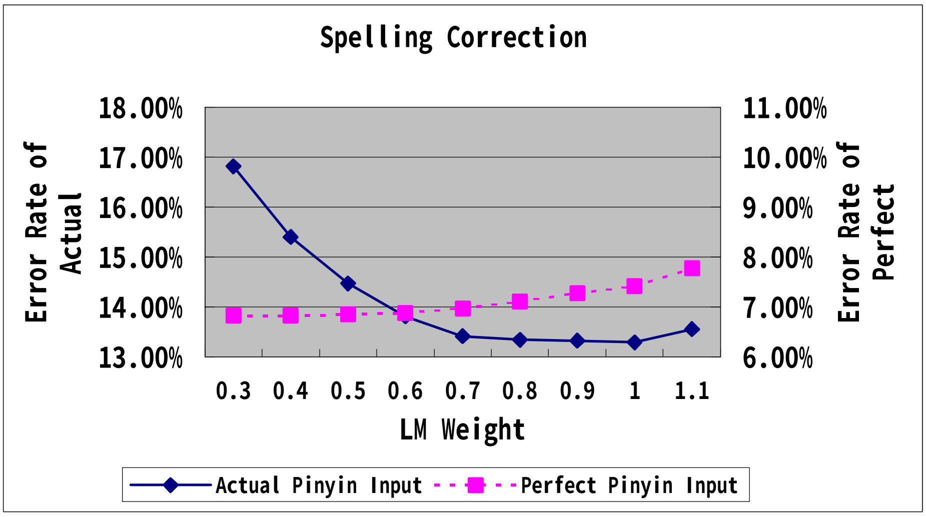

Figure 3.1 effect of LM weight

If we apply adaptive LM weight depending on the typing skill of the user, we can obtain further error reduction. To verify this, we select 3 users from the testing data, adding one ideal user (suppose input including no errors), we test the error rate of system with different LM weight, and result is as table 3.2.

	a_1	a_2	a_3	*Dynamic a*
User 0	6.85%	7.11%	7.77%	6.85%
User 1	8.15%	8.23%	8.66%	8.15%
User 2	13.90%	12.86%	12.91%	12.86%
User 3	19.15%	18.19%	17.77%	17.77%
Average	12.01%	11.6%	11.78%	10.16%

Table 3.2 user adaptation

The average input error rates of User 1,2,3 are 0.77%, 4.41% and 5.73% respectively.

As can be seen from table 3.2, the best weight for each user is different. In a real system, skilled typist could be assigned lower LM weight, and the skill of typist can be determined by:

1. the number of modification during typing.
2. the difficulty of the text typed distribution of typing time can also be estimated. It can be applied to judge the skill of the typist.

4. Modeless Input

Another annoying UI problem of Pinyin input is the language mode switch. The mode switch is needed while typing English words in a Chinese document. It is easy for users to forget to do this switch. In our work, a new spelling model is proposed to let system automatically detect which word is Chinese, and which word is English. We call it modeless Pinyin input method. This is not as easy as it may seem to be, because many legal English words are also legal Pinyin strings. And because no spaces are typed between Chinese characters, and between Chinese and English words, we obtain even more ambiguities in the input. The way to solve this problem is analogous to speech recognition. Bayes rule is used to divided the objective function (as equation 4.1) into two parts, one is the spelling model for English, the other is the Chinese language model, as shown in equation 4.2.

Goal: $$\hat{H} = \arg\max_{H} \Pr(H \mid P) \qquad (4.1)$$

Bayes Rule: $$\hat{H} = \arg\max_{H} \frac{\Pr(P \mid H)\Pr(H)}{\Pr(P)} \qquad (4.2)$$

One of the common methods is to consider the English word as one single category, called <English>. We then train into our Chinese language model (Trigram) by treating <English> like a single Chinese word. We also train an English spelling model which could be a combination of:

1. A unigram language model trained on real English inserted in Chinese language texts. It can deal with many frequently used English words, but it cannot predict the unseen English words.
2. An "English spelling model" of tri-syllable probabilities – this model should have non-zero probabilities for every 3-syllable sequence, but also should emit a higher probability for words that are likely to be English-like. This can be trained from real English words also, and can deal with unseen English words.

This English spelling models should, in general, return very high probabilities for real English word string, high probabilities for letter strings that look like English words, and low probabilities for non-English words. In the actual recognition, this English model will run in parallel to (and thus compete with) the Chinese spelling model. We will have the following situations:

1. If a sequence is clearly Pinyin, Pinyin models will have much higher score.
2. If a sequence is clearly English, English models will have much higher score.
3. If a sequence is ambiguous, the two models will both survive in the search until further context disambiguates.
4. If a sequence does not look like Pinyin, nor an English word, then Pinyin model should be less tolerant than the English tri-syllable model, and the string is likely to remain as English, as it may be a proper name or an acronym (such as "IEEE").

During training, we choose some frequently used English syllables, including 26 upper-case, 26 lower-case letters, English word begin, word end and unknown into the English syllable list.

Then the English words or Pinyin in the training corpus are segmented by these syllables. We trained the probability for every three syllable. Thus the syllable model can be applied to search to measure how likely the input sequence is an English word or a Chinese word. The probability can be combined with Chinese language model to find the most probable Chinese and/or English words.

Some experiments are conducted to test the modeless Pinyin input methods. First, we tell the system the boundary between English word and Chinese word, then test the error of system; Second, we let system automatically judge the boundary of English and Chinese word, then test the error rate again. The result is as table 4.1.

	Total Error Rate	English Error Rate
Perfect Separation	4.19%	0%
Mixed Language Search (TriLetter English Spelling Model)	4.28%	3.6%
Mixed Language Search + Spelling Correction (TriLetter English Spelling Model)	4.31%	4.5%

Table 4.1 Modeless Pinyin input method (Only choose 52 English letters into the English syllable list)

In our modeless approach, only 52 English letters are added into English syllable list, and a tri-letter spelling model is trained based on corpus. If we let system automatically judge the boundary of English word and Chinese word, we found the error rate is approximate 3.6% (which means system make some mistake in judging the boundary). And we found that spelling model for English can be run with spelling correction, with only a small error increase.

Another experiment is done with an increased English syllable list. 1000 frequently used English syllables are selected into English syllable list. Then we train a tri-syllable model base on corpus. The result is shown in table 4.2.

	Total Error Rate	English Error Rate
Perfect Separation	4.19%	0%
Tri Letter English Spelling Model	4.28%	3.6%
Tri Syllable English Spelling Model	4.26%	2.77%

Table 4.2 Modeless Pinyin input method (1000 frequently used English syllables + 52 English letters + 1 Unknown)

As can be seen from table 4.2, increasing the complexity of spelling model adequately will help system a little.

5. Conclusion

This paper proposed a statistical approach to Pinyin input using a Chinese SLM. We obtained conversion accuracy of 95%, which is 50% better than commercial systems. Furthermore, to make the system usable in the real world, we proposed the spelling model, which allows the user to enter Chinese and English without language mode switch, and the typing model, which makes the system resident to typing errors. Compared to the baseline of system, our system gets approximate 30% error reduction.

Acknowledgements

Our thanks to ChangNing Huang, JianYun Nie and Mingjing Li for their suggestions on this paper.

References

Chen Yuan. 1997.12. Chinese Language Processing. Shang Hai education publishing company.

Jianfeng Gao, Hai-Feng Wang, Mingjing Li, Kai-Fu Lee. 2000. A Unified Approach to Statistical Language Modeling for Chinese. IEEE, ICASSP 2000.

Kai-Fu Lee. 1989. Automatic Speech Recognition, Kluwer Academic Publishers.

Chin-Hui Lee, Frank K. Soong, Kuldip K. Paliwal. 1996. Automatic Speech and Speaker Recognition -- Advanced Topics, Kluwer Academic Publishers.

Frederick Jelinek. 1997. Statistical Methods for Speech Recognition, The MIT Press, Cambridge, Massachusetts.

William E. Cooper. 1983. Cognitive Aspects of Skilled Typewriting, Springer-Verlag New York Inc..

Bahl,L., Bakis, R., Jelinek, F., and Mercer, R. 1980. Language Model / Accoustic Channel Balabnce Mechanism. IBM Technical Disclosure Bulletin, vol.23, pp. 3464-3465.

X. Huang, M. Belin, F. Alleva, and M. Hwang. 1993. Unified Stochastic Engine (USE) for Speech Recognition, ICASSP-93., vol.2, pp. 636-639.