

# An Efficient Way to Learn Rules for Grapheme-to-Phoneme Conversion in Chinese

Zi-rong ZHANG Min CHU Eric CHANG

Microsoft Research Asia, Beijing

[zrzhang@msrchina.research.microsoft.com](mailto:zrzhang@msrchina.research.microsoft.com); [{minchu,echang}@microsoft.com](mailto:{minchu,echang}@microsoft.com)

## ABSTRACT

Grapheme-to-phoneme (G2P) conversion is a very important component in a Text-to-Speech (TTS) system. Determining the pronunciation of polyphone characters is the main problem that the G2P component in a Mandarin TTS system faces. By studying the distribution of polyphones and their characteristics in a large text corpus with corrected pinyin transcriptions, this paper points out that correct G2P conversion for 41 key polyphones and 22 key polyphonic multi-syllabic words will constrain the overall error rate to below 0.068%. In this paper, the Extended Stochastic Complexity based stochastic decision list is used to learn rules for G2P conversion for these key polyphones and polyphonic words. With the generated rules, the error rate for G2P conversion decreased from 0.88% to 0.44%. Tagging corpus with correct pinyin for training and testing rules is a labor consuming and time consuming task. This paper also proposes a semi-automatic approach to do this, which saves almost half of the workload.

## 1. INTRODUCTION

Grapheme-to-phoneme conversion is an important component in TTS systems. In most of the alphabetic languages such as English, the main problem G2P module faces is to generate pronunciations for words that are out of vocabulary (OOV). Many approaches for letter-to-phoneme conversion have been proposed [1][2][3][4]. However, unlike the OOV problem in Western languages, the difficulty in Chinese G2P conversion is to pick out one correct pronunciation from several candidates for all polyphones. This is still not a well-solved problem. The commonly used method in most Mandarin TTS systems is to list as many as possible the words with polyphonic characters into a dictionary [5][6]. Pronunciations for all these words are also listed in the dictionary and they are verified manually. During online G2P conversion, pronunciations are first looking up in the dictionary. When they are not found in the dictionary, the most frequently used pronunciation is chosen. Thus, monosyllabic words, which are polyphones, are often assigned with wrong pronunciations. It is necessary to generate pronunciation rules for these polyphonic monosyllabic words. Summarizing pronunciation rules is a very difficult and time-consuming task for human experts. Furthermore, it is much more difficult to evaluate the validity of these rules. In this paper, the Extended Stochastic Complexity (ESC) based stochastic decision list is used to learn pronunciation rules reliably and efficiently.

The algorithm for generating pronunciation rules with ESC-base stochastic decision list is introduced in Section 2. Section 3 describes the experiments and the results. Final discussion is given in Section 4.

## 2. ALGORITHM

Since the pronunciation for most polyphonic characters and words can be decided from their contexts, pronunciation determination for these characters and words is very similar to the problem of automatic text classification. In this paper, the ESC-based stochastic decision list method [7], which has been used to classify text documents successfully, is used to automatically generate pronunciation rules for polyphonic characters and words.

### 2.1 ESC-based stochastic decision lists

The ESC-based stochastic decision list was proposed by H. Li etc [7]. They have successfully used it in text classification. The main idea of the algorithm is to create a stochastic decision list according to the principle of minimizing ESC [9]. The decision list consists of IF-THEN rules and every rule includes conditions, the result and the probability for correct decision. Since G2P conversion for polyphones can also be treated as a problem of classification, the method is extended to the task of generating pronunciation rules for polyphones in this paper.

We are looking for rules to classify a polyphonic character/word into one of its possible pronunciations, represented by phoneme strings, according to its context. To simplify the problem, we assume that the pronunciation of a polyphone depends only on its nearby context, i.e. words within the same clause. This assumption is applicable for most cases. Then each clause  $d$ , containing a target polyphone, is represented by  $(b_1, b_2, \dots, b_k) \rightarrow c_m$ .  $b_j$  is a Boolean variable. It takes value 1 if  $d$  contains feature  $w_j$ . Otherwise, it takes value 0.  $w_j$  is a feature word that plays an important role in classifying the target polyphone.  $c_m$  represents the adopted pronunciation for the target polyphone in clause  $d$ . Rules for a specific pronunciation  $c_m$  are to be constructed with conditions on presence or absence of certain feature words. In theory, all the words appearing in clauses containing a target polyphone should be considered as candidates for selecting feature words. However, that will require extremely heavy computation load. Thus, in practice, only the most important  $k$  feature words are considered. In our case, feature words are only selected from the left and right adjacent words of the target polyphone. The learning for rules is carried out in three steps: feature selection, growing rules and pruning rules.

### 2.2 Feature Selection

A word is considered as a feature word when its presence or absence gives more information than others. The amount of information is measured in term of Stochastic Complexity (SC)[8].

For a data sequence  $(d_1, c_1), (d_2, c_2), \dots, (d_m, c_m)$ ,  $d_i$  denotes the  $i$ th clause containing a target polyphone and  $c_i$  denotes the pronunciation of the polyphone in this clause.  $c^m = c_1 c_2 \dots c_m$ .  $m$  is the number of the sentences of the data sequence. Then SC of  $c^m$  can be calculated with equation (1).

$$SC(c^m) = mH\left(\frac{m^+}{m}\right) + \frac{1}{2} \log \frac{m}{2\pi} + \log \pi, \quad (1)$$

where  $m^+$  is the number of clauses whose  $c_i$  is 1 in  $c^m$ . When  $I > z > 0$ ,  $H(z) = -z \log(z) - (1-z) \log(1-z)$ ; when  $z = 0$  or  $z = 1$ ,  $H(z) = 0$ .  $\log$  means natural logarithm in this paper.

Let  $c^{m_w}$  denotes the clause set, in which all clauses contain word  $w$ .  $m_w$  is the size  $c^{m_w}$ . Then SC of  $c^{m_w}$  can be calculated with equation (2).

$$SC(c^{m_w}) = m_w H\left(\frac{m_w^+}{m_w}\right) + \frac{1}{2} \log \frac{m_w}{2\pi} + \log \pi, \quad (2)$$

where  $m_w^+$  is the number of clauses whose  $c_i$  is 1 in  $c^{m_w}$ .

Let  $c^{m_{-w}}$  denotes the clause set, in which all clauses do not contain word  $w$ , and  $m_{-w}$  is size of it. Then the SC of it can be calculated with equation (3).

$$SC(c^{m_{-w}}) = m_{-w} H\left(\frac{m_{-w}^+}{m_{-w}}\right) + \frac{1}{2} \log \frac{m_{-w}}{2\pi} + \log \pi, \quad (3)$$

where  $m_{-w}^+$  is the number of clauses whose  $c_i$  is 1 in  $c^{m_{-w}}$ .

Then the information gain caused by presence of word  $w$  can be calculated by (4).

$$\Delta SC(w) = \frac{1}{m} (SC(c^m) - (SC(c^{m_w}) + SC(c^{m_{-w}}))) \quad (4)$$

A large  $SC(w)$  means word  $w$  is an important feature for the target polyphone. Thus, any word, which results  $SC(w)$  larger than a preset threshold, is selected as a feature word.

### 2.3 Growing Rules

Within a set of selected feature words, any combination of the presence or absence the left context word or right context words forms a rule candidate. A rule candidate is valid only when it has at least one corresponding instance in the training set. In the growing stage, only the set of rules, who will result the minimum ESC in the training set, are kept.

Let  $C^{m_t}$  be the clause set, which satisfy a candidate rule  $t$ .  $m_t$  is the size of  $C^{m_t}$ . Let  $C^{m_{-t}}$  represent the clause set, which don't satisfy the candidate rule  $t$ .  $m_{-t}$  is the size of  $C^{m_{-t}}$ . Then the ESC of  $c^m$ ,  $C^{m_t}$  and  $C^{m_{-t}}$  can be calculated with equation (5) – (7).

$$ESC(c^m) = Loss(c^m) + \lambda \sqrt{m \log m}, \quad (5)$$

$$ESC(C^{m_t}) = Loss(C^{m_t}) + \lambda \sqrt{m_t \log m_t}, \quad (6)$$

$$ESC(C^{m_{-t}}) = Loss(C^{m_{-t}}) + \lambda \sqrt{m_{-t} \log m_{-t}}. \quad (7)$$

$Loss(*)$  is the number of clauses who take non-default type. The second item denotes the length of code for describing the model.

is a positive constant and is used to balance the contribution of loss function and length of model.

Then, ESC decrease of  $c^m$  caused by the rule  $t$  can be calculated by (8).

$$\begin{aligned} \Delta ESC(t) &= ESC(c^m) - ESC(C^{m_t}) - ESC(C^{m_{-t}}) \\ &= \{Loss(C^m) - Loss(C^{m_t}) - Loss(C^{m_{-t}})\} \\ &\quad + \left\{ \lambda \left( \sqrt{m \log m} - \sqrt{m_t \log m_t} - \sqrt{m_{-t} \log m_{-t}} \right) \right\} \end{aligned} \quad (8)$$

If rule  $t$  results a positive  $ESC(t)$ , it is kept for next step. Otherwise, it is deleted from the rule list.

### 2.4 Pruning the Rules

During the growing stage, each rule is the best choice at the time it was selected, which means that it is the best, comparing with those haven't been processed. Thus, the whole set of the selected rules do not guarantee the minimum ESC for the whole training set. Some of the rules at the end of the rule list are to be pruned to get the minimum ESC.

The ESC of training set  $c^m$  and decision list  $A$  can be calculated with (9).

$$ESC(c^m : A) = ESC(c^m / A) + \lambda' L(A) = \sum_i ESC(c^{m_i}) + \lambda' L(A) \quad (9)$$

$L(A)$  is the model description length [10] and can be calculated as follows:

$$L(A) = \log |T| + \log(|T|-1) + \log(|T|-2) + \dots + \log(|T|-i+1), \quad (10)$$

where  $|T|$  is the number of rules in rule set  $A$  and  $i$  is a variable between 1 and  $|T|$ .

Let  $A$  and  $A'$  represents the decision list before and after pruning the last rule. The condition for stopping is:

$$ESC(c^m | A') - ESC(c^m | A) \geq \lambda' (L(A) - L(A')). \quad (12)$$

The  $\lambda'$  is a constant, which is used to balance the contribution of the components at each side of the inequation. The smaller the  $\lambda'$  is, the less the rules will be pruned. When it is set to 0, no rule will be deleted.

## 3. ANALYSIS, EXPERIMENT AND EVALUATION

### 3.1 Selection of key polyphones and polyphonic words

After investigating over a 2M-character corpus with correct pinyin transcription annotated, we find several useful characteristics for polyphones in Mandarin.

First, the number of polyphones is large and it is related to the character set used. 830 out of 6763 Chinese characters in “GB2312-80”, which is the most important Chinese character set, have more than one pronunciation. And 1036 polyphones are listed in “Modern Chinese Dictionary” [11].

Second, great discrepancy exists among the usage frequencies for these polyphones. In our 2 million characters corpus, polyphones that are left as monosyllabic words after word segmentation

account for 8.95% of the whole text<sup>1</sup>. If all of them are set to their most frequently used pronunciation, the overall error rate for G2P conversion for the corpus is 0.88%. Furthermore, the cumulative frequency of the top-100 frequently used polyphones account for more than 90% of the overall appearance of all polyphones. And the cumulative frequency of the top-180 polyphones account for more than 95% of all. It's obvious that these 180 high frequency polyphones are the most important ones.

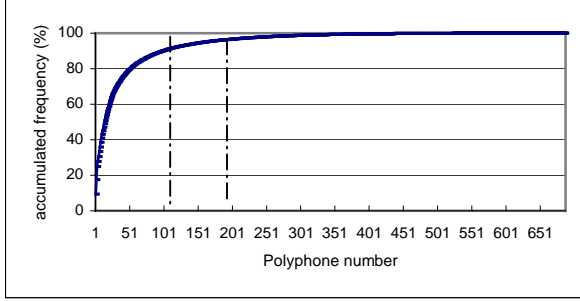


Figure 1: The cumulative frequency of polyphones sorted by their usage frequency.

Third, several pronunciations for a polyphone are not equally used by people. Most polyphones have a dominating pronunciation. For example, the dominating pronunciation (dè and lè) of “的” (dè, dí, dì) and “了” (lè, liǎo) account for more than 99% usage of them. These polyphones are less important. However, there are some polyphones, such as “为” (wéi, wèi) and “长” (cháng, zhǎng), who have no significant dominating pronunciations. They are the key polyphones that should be processed carefully.

Considering the above factors, we generate a list of key polyphones from the top 180 ones with the constraints that the usage frequency of their dominating pronunciation should be smaller than 95%. Only 41 polyphones are left in the list. With the same constraint, 22 polyphonic multi-syllabic words are selected from 580 candidates. If all these key characters and words are processed correctly, the error rate G2P conversion of the 2-M character corpus will be only 0.068%.

### 3.2 Corpus tagging

In order to get stable rules, sufficient data are indispensable. According to the analysis of the last section, we only aim at creating rules for those key polyphones and key polyphonic words. For other polyphones and other polyphonic words, their dominating pronunciations are used in G2P conversion.

Since there is no corpus with pinyin transcriptions available, we do the transcription by ourselves. First, all the polyphones and polyphonic words in a corpus abstracted from “People’s daily”, denoted as Corpus1, are transcribed. There are more than 2 million characters in Corpus1. The statistical analysis in Section 3.1 is done over this corpus. Then, a set of clauses for each key polyphone and polyphonic word are transcribed. There are at least 8000 clauses in each set. These clause sets are denoted as Corpus2.

<sup>1</sup> “的 (dè, dí, dì)” is not included. Though it is used frequently, it seldom takes other pronunciations except “dè”.

Pinyin transcription for text corpus is a labor and time consuming task. A semi-automatic approach is used in our transcription. After transcribing Corpus1, a set of stable rules are generated from it. Then, part of the new sentences can be transcribed with these rules automatically. Only those which can not be processed automatically are reviewed manually. The rule sets are adapted regularly when the new transcribed data increase to a certain amount. When the size of the rule set increases, the portion of automatically transcribed sentences increased too. After finishing Corpus2, we find that the semi-auto method saves about half of the working load.

### 3.3 The relation between the effectiveness of rules and the size of training corpus.

The effectiveness of the rules is related to the size of the training corpus. This is illustrated by an experiment on the polyphone “长”. The training set increases gradually from 1000 to 30000 sentences and the testing set is 1680 sentences disjointed from the training set. Figure 3 shows the results. Between 5000 and 20000, the more the training corpus is, the higher the correct rate is. When the size of training set exceeded 20000, the correct rate increases very slowly. So, when we select corpus for those key polyphones, we must be sure that the training set must be more than certain threshold, in our project, 8000 sentences.

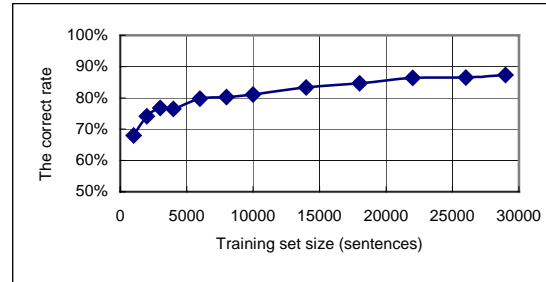


Figure 2: The relation between effectiveness of rules and amount of training corpus.

### 3.4 The relation between the effectiveness of rules and the domain of the training corpus.

To investigate the influence of the domain of training set on the effectiveness of rules, experiment 2 is carried out. In this experiment, two sub-training set is formed according to their sources: newspaper and novel. Each contains 6000 sentences. Two domain dependent testing sets are formed accordingly. Table 1 shows the results. It's obvious that the correct rates are higher when the training and testing set are from the same domains. However, the mismatch of domain causes significant drops in correct rates.

From this experiment, we learn that if we want to get stable pronunciation rules that are effective in more fields, we must have a training set with wide coverage.

Table 1: The correct rates for matching and mismatching conditions.

Testing Set \ Training Set	Novel	
	Novel	Newspaper
Novel	88%	65%
Newspaper	65%	88%

Novel	83.2%	71.9%
Newspaper	78.7%	83.7%

### 3.5 Creating rules

After we have prepared enough data according to the principles mentioned above, we can create rules for those 41 key polyphones and 22 key polyphonic words. In order to minimize the influence of the source of corpus, we divide all training corpus into three parts and use 2/3 of the corpus as training set and the other 1/3 as testing set in turn. Then we can get three sets of rules. Only those rules that appear in all three sets are used.

According to our testing result with 41 key polyphones, the average correct rate is increased from 81.2% to 94.3% after using the final rules. More than half of these polyphones obtained 10% higher correct rates than before. Some polyphones aren't improved obviously, because their high-frequency pronunciation rates are already very high (higher than 90%). Only six polyphones' correct rates are still lower than 90%. This is because that their original correct rates are very low (between 37% and 66%). All these show that the pronunciation rules generated with ESC-based method are very useful. Because of limitation of length of paper, we only list out a part of results in table 2. After using rules, the average correct rate of polyphonic words is also increased from 84.8% to 92.2% and part of the testing results are listed in table 3. When testing the rule sets with continue text, 0.44% error rate is obtained for the G2P conversion. Comparing with the error rate (0.88%) when no rules are used, 50% error reduction is achieved.

Table 2: Testing result of some polyphones.

Pol yphone	Hi gh-fre pī nyī n	Low-fre pī nyī n	Hi gh-fre pī nyī n rate	Correct Rate
传	chuán	zhuàn	88.2%	98.7%
只	zh	zh	73.6%	99.0%
处	chù	ch	76.4%	97.9%
少	shǎo	shào	83.4%	96.9%
间	j i ān	j i àn	96.9%	99.2%
为	wèi	wéi	53.7%	83.3%
藏	zàng	cáng	56.2%	86.2%

Table 3: Testing result of some polyphonic words.

Polyphoni c word	Hi gh-fre pī nyī n	Low-fre pī nyī n	Hi gh-fre pī nyī n rate	Correct rate
合计	hé j i	hé j i	85.4%	97.8%
孙子	sūn z i	sūn z	71.2%	93.3%
朝阳	cháo yáng	zhāo yáng	51.0%	81.6%
地方	dì fāng	dì fang	68.9%	83.1%
得了	dé l e	dé l iǎo	83.9%	93.1%

## 4. DISCUSSION

According to the testing result, ESC-based decision list is very effective for generating rules for those polyphones and polyphonic words and increase the correct rate to 94.3% and 92.2% from 81.2% and 84.8% respectively. 50% error reduction is achieved for the G2P conversion module in Mandarin TTS system.

For further improvement, we can consider the following factors:

The feature words we use at the time is only the left adjacent word and the right adjacent word, while the further words and information of part of speech are not used. However, sometimes they are useful, especially for some propositions and auxiliary words, e.g. 为, 的, 地 etc. For some other polyphones such as “曾”, “仇”, one of their pronunciations only appear in person name. It would be very helpful if we can exactly detect peoples' name or other entities' names.

## 5. ACKNOWLEDGEMENT

First we want to say thanks to Dr. Hang Li for his suggestions and help during our research. We also thank Hongyun Yang for his good tool of word segmentation and pinyin tagging.

## 6. REFERENCE

- [1] M. Davay and A. J. Vitale, “Algorithms for grapheme-phoneme translation for English and French: Applications for database searches and speech synthesis”, Computational Linguistics, 1997, Vol. 23, pp.495-523.
- [2] O. Andersen , R. Kuhn , A. Lazarides , et al . ”Comparison of Two Tree-Structured Approaches for Grapheme-to-Phoneme Conversion”, Proceeding of the 4th International Conference on Spoken Language Processing (ICSLP), Philadelphia, 1996 , pp. 1808-1811.
- [3] K. Torkolla, “An efficient way to learn English grapheme-to-phoneme rules automatically”, Proceedings of the International Conference on Acoustics , Speech and Signal Processing (ICASSP) , Vol.2, Minneapolis, 1993 , pp. 199-202.
- [4] T. J. Sejnowski and C. R. Rosenberg, “Parallel networks that learn to pronounce English text”, Complex Systems, 1987, Vol.1, pp. 145-168.
- [5] Daju Gou and Wanbo Luo, “Processing of Polyphone character in Chinese TTS system”, Chinese Information, No. 1, pp. 33-36
- [6] Yifeng Pan, “Application of Computer in Chinese Pinyin Automatic Transcription”, Shanghai Normal University (Natural Science Version), 1996, Vol.25, No.4, pp.54-58
- [7] H. Li and K. Yamanishi, “Text Classification Using ESC-based Stochastic Decision Lists”, Proceedings of 8<sup>th</sup> International Conference on Information and Knowledge Management (CIKM'99), Kansas City, MO, USA, 1999 , pp. 122-130.
- [8] J. Rissanen, “Fisher information and stochastic complexity”, IEEE Transaction on Information Theory , 1996 , Vol. 42, No.1, pp. 40-47.
- [9] K. Yamanishi, “A decision-theoretic extension of stochastic complexity and its applications to learning”, IEEE

Transactions on Information Theory , 1998 , Vol. 44, No.4,  
pp. 1424-1439.

- [10] K. Yamanishi, "A learning criterion for stochastic rules",  
Machine Learning, 1992, Vol.9, pp. 165-203.
- [11] Dictionary Compile Department, Institute of Language,  
Chinese Academy of Social Science. Modern Chinese  
Dictionary, The third edition, 1996,