Grapheme-to-Phoneme Conversion Based on TBL Algorithm in Mandarin TTS System

Min Zheng, Qin Shi, Wei Zhang and Lianhong Cai

Computer Science Department in Tsinghua University, Beijing, 100084
IBM China Research Lab, Beijing, 100083
Email: kristy99@mails.tsinghua.edu.cn, {shiqin; zhangzw}@cn.ibm.com, clh-dcs@tsinghua.edu.cn

Abstract

Grapheme-to-phoneme (G2P) conversion is an important component in a Text-to-Speech (TTS) system. The difficulty in Chinese G2P conversion is to pick out one correct pronunciation from several candidates according to the context information. By evaluating the distribution of polyphones in a corpus with manually corrected pinyin transcriptions, this paper pointed out that the overall error rate of G2P conversion was greatly decreased after processing 78 key polyphones. This paper proposed a transformation-based error-driven learning (TBL) algorithm to solve G2P conversion for polyphones. The correct rates of G2P for polyphones, which originally have high accuracy or low accuracy, are both improved. Besides, two additional experiments show that the capacity of the TBL algorithm has great relationships with initial status and TBL algorithm is more suitable than decision tree to solve polyphones' G2P problem.

Keyword: Grapheme-to-Phoneme (G2P); Transformation-Based Learning (TBL); Decision Tree (DT); polyphone

1 Introduction

Grapheme-to-phoneme (G2P) conversion is an important component in mandarin Text-to-Speech (TTS) system. In most of the alphabetic languages such as English, the main problem G2P module is to generate correct pronunciations for words that are out of vocabulary (OOV). However, unlike the OOV problem, the difficulty in Chinese G2P conversion is to pick out one correct pronunciation from several candidates according to the context information such as Part-Of-Speech, lexical words or position of the polyphone in a word or sentence. Traditionally, the commonly used method is to list polyphonic words and characters with correct pronunciations into a dictionary. But such dictionary can not completely solve G2P problem, pronunciation rules according to the context are needed to handle more complicated problem. Recently, various data-driven methods including neutral network[1], decision trees[2][3], pronunciation-by-analogy models[4] and extended stochastic complexity methods[5] have been tried to solve G2P problem.

In this paper, TBL algorithm is proposed to solve G2P problem in mandarin TTS system. As an automatic rule learning methods, it is proved to be efficient and is widely used in numerous tasks, including learning rules for part-of-speech tagging[6], prepositional phrase attachment[7] and grammatical relation extraction etc; Now we leverage it to solve the polyphone problem and receive great improvements. Besides, two comparative experiments are implemented by using the same features and corpus: One experiment based on different initial status is inferred that a better experiment result of TBL algorithm depends on a better initial status assignment. Another is to compare TBL with decision tree(DT) algorithm, which is successfully dealt with many similar problems like parsing[8], prosody labelling[9] and phrase break prediction[10] etc. Comparing the results, TBL algorithm shows better performance to solve polyphones' G2P problem.

The paper is organized as follows: The TBL algorithm is explained in Section 2. The polyphone candidates' selection and corpus preparation are shown in Section 3. Section 4 describes the experiment, including features selection, template design and algorithm implementation. Two comparative experiments are given in Section 5. Final discussions and conclusions are shown in Section 6.

2 Transformation-Based Learning Algorithm

Transformation-based learning (TBL) is one of the most successful rule-based machine learning algorithms. The central idea of transformation-based learning (TBL) is to learn a group of ordered rules from all the candidates according to their contribution to the training corpus.

There are two kinds of states in the training corpus: one is initial state that is annotated automatically by the current system. The other is target state which is corrected manually. The task of the algorithm is to select a set of ordered rules from candidates to transfer the wrong initial states into the target states with minim errors. In order to generate a set of rules candidates automatically, templates are needed. A template is composed of several features and the relationship among them. The rules generation space is limited by these templates. Every difference between the initial and target state will derive a set of rules according to the templates. The process of the algorithm is illustrated in Figure1.

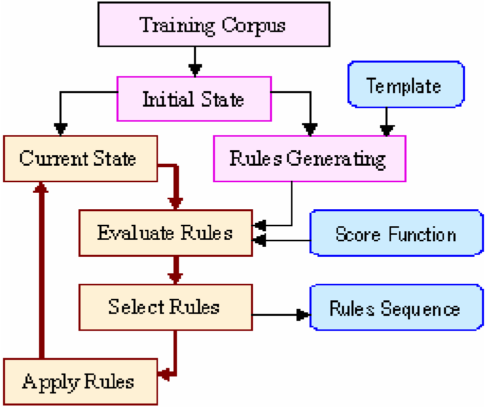

Fig.1: Processing of TBL algorithm

In the every iteration for training, the score for every rule is calculated according to the difference between the updated states and target states. The rule which has the highest gain score is chosen and appended to the rules sequence. The current states of corpus are updated according to the selected rule. Iterations will continue until no more improvement can be made.

3. Polyphone Analysis

There are 682 polyphones in the Mandarin characters. But many of them have dominating pronunciations or rarely appear in usual articles, which can be solved by dictionary. It's unnecessary to generate rules for all 682 polyphones. The widely used polyphones, which also have high error rate, should be analyzed first.

3.1 Key Polyphones Selection

There are 12903 sentences in the corpus for polyphones analysis, including 271720 Chinese characters. It is randomly selected from newspapers, novels and oral talk. The corpus is manually checked with correct information of word segmentation, pos tagging (91 kinds) and phonetic notation by listening to the record speech and reading the text transcriptions. Three factors for selecting polyphones are considered:

1) Discrepancy among the occurrence frequency of polyphones

There are 682 polyphones defined in the homograph dictionary, but the occurrence of polyphones in the corpus is quite different:

Table1 Occurrence number of some polyphones

polyphone	Occurrence numbers in the corpus
一	2333
为	775
地	582
冠	38
铺	31

The homograph dictionary is sorted according to the usage frequency of polyphones in descending order. The results of the coverage ratio of the whole polyphones are shown in the following table 2:

Table2 Coverage ratio of Polyphones

Number of the top polyphones	Coverage ratio of the whole polyphones
50	0.59628
100	0.78404
150	0.88235
200	0.93939
220	0.95573

From the table2, the cumulative frequency of the top-220 frequently used polyphones takes up more than 95% of all polyphones appearance. Obviously, it is important to generating pronunciation rules for these 220 polyphones first.

2) Accuracy rate

The pronunciations of some polyphones have already reached a very high accuracy in the current system, such as the right ratio of the pronunciation for the polyphone "会" has already reached 100%. Obviously, it has no use to generate pronunciation rules for these polyphones.

3) Dominating pronunciation rate

Many polyphones have dominating pronunciation. For example, the pronunciation "de0" for the polyphone "的" (de0, di2, di4) takes up 99% ratio, the pronunciation "le0" for the polyphone "了" (le0, liao3) also takes up 98% ratio and special cases could be handled by the homograph dictionary. However, other polyphones like "为" (wei2, wei4) and "长" (chang2, zhang3), who have no significant dominating pronunciations, are the key polyphones that should be processed carefully.

According to the above analysis, a list of key polyphones is generated from the top-220 ones according to the following two criteria:

1) The usage frequency of its dominating pronunciation is lower than 98%;
2) The right ratio of the original pronunciation is lower than 98%.

Only 78 polyphones are left in the list. The detailed statistic analysis results are shown in the following table3:

Table3 Selected polyphones under different situations

Ratio of the dominating pronunciation	Right Ratio of the original pronunciation	Number of Selected polyphones
0.93	0.94	40
0.95	0.96	53
0.96	0.96	60
0.98	0.98	78
0.98	0.99	98

3.2 Polyphone Corpus Design

Although the corpus with corrected pinyin scripts is ready, whether it is suitable and enough for learning rules should be studied. Since TBL algorithm is based on error information, the effectiveness of the rules is related both with the size and the error ratio of the corpus.

This is illustrated by two experiments on the polyphone "为" who occurs in 5000 sentences.

1) *First experiment*: Increasing the training set gradually from 1000 to 3000 sentences and testing it on 500 sentences disjointed from the training set.
2) *Second experiment*: Increasing the error ratio of the training set gradually from 5% to 20% and testing it on 500 sentences disjointed from the training set.

Table4 Correct ratio with different corpus capacity

Number of sentences in training set	Error Ratio of the training set	Correct ratio of the testing set
1000	5%	68%
1000	10%	73%
2000	10%	73%
2000	20%	81%
2500	10%	75%
2500	20%	86%
3000	10%	78%
3000	20%	88%

From the result shown in table4, larger training corpus and higher error ratio will result in higher correct rate. Besides this, the error ratio is discovered has more effect than the size of the training corpus. According to the study, more sentences are collected. The following is the description of TBL training corpus

1) Corpus for rules generating: Averagely, 500 sentences for each polyphone, which have error pronunciations for polyphones from the rules learning corpus,
2) Corpus for rules learning: Averagely, 2500 sentences for each polyphone;
3) Corpus for rules testing: Averagely, 1500 sentences for each polyphone;

4 Experiment on TBL algorithm

In the TBL experiment, the G2P result of the current TTS system is used as the initial states and manually checked pinyin scripts is used as the target states. Feature selection, template design and rules learning process are described in the following. At last, the experiment result is showed.

4.1 **Feature Selection**

Linguistic information that could be provided by the current TTS system is used here:

1) Lexical features: *LC*(character), *LW*(lexical word);
2) Syntactic feature: *POS*(part of speech)*;*
3) Other features: *POSITION*(position of the polyphone in a word), *LEN*(length of lexical word), *BEGIN*(whether at the beginning of the sentence);
4) Special feature for the polyphone “一” and “不”: *TONE*(tone type of character);

4.2 Template Design

The template items consist of several features and the relationship among them. Such as “POS(Y,-1) & POS(Y,1) & LEN(n,0): A->B”, in which “Y” indicates the feature value, the number “-1” indicates the offset from the polyphone and the letter “A” and “B” indicate the original and standard pronunciation of the polyphone respectively. In the template, the offset is in the range of {-2,2}.There are two kinds of template set:

1) One template set is especially designed for “一” and “不”: There are 13 templates which consist of 5 features: “*TONE*”, “*POSITION*”, “*LEN*”, “*POS*” and “*LC*”;
2) Another template set is designed for other polyphones: 19 templates are designed including 6 features: “*POSITION*”, “*LEN*”, “*POS*”, “*LC*”. “*LW*” and “*BEGIN*”;

4.3 Rules Generation

Rules are generated by traversing all the incorrect samples according to the rules templates. Such as the sentence

“招牌(ng) 为[wei4->wei2](vi) ：(wl) 北京(npr) 中医药(ng)”

It can produce a rule “*POS(ng,-1) & POS(w1,1) & LEN(1,0): wei2->wei4*” according to the template item: “*POS(Y,-1) & POS(Y,1) & LEN(n,0): A->B*”.

4.4 Rules Learning

The learning process is a greedy search. During each iteration, the rule that results in the greatest reduction in errors in the training data is selected. A rule candidate is presented as r_i. The error reduction function for r_i is defined as $f(r_i)$. The process is described as follows:

```
Assign initial values to create training data $C$.
Repeat
    Find the rule candidate $r_i$ that gives the best reduction in errors in $C$
    If ( $f(r_i) \geq$ Minimum)
        Add $r_i$ to the ordered list of rules, $R$
        Apply $r_i$ to all relevant cases in $C$
    End if
Until ( $f(r_i) <$ Minimum)
```

The experiment result of TBL for polyphones is shown in the Table5. From the result, the correct rates of polyphones G2P conversion which originally have high accuracy or low accuracy are all improved.

Table5 Accuracy of some polyphones using TBL

Polyphone	Original accuracy	Accuracy with TBL
长	0.973054	0.978346
为	0.779476	0.971616
倒	0.731405	0.828512
朝	0.812963	0.914815
重	0.831081	0.908072
教	0.836336	0.962462
冠	0.389744	0.851282
不	0.867704	0.954270
还	0.987952	0.988956
着	0.944862	0.960526

5 Comparative Experiments

5.1 Experiments with different initial status

As TBL framework described in section 2, rules are generated based on each erroneous tag in the initial state. Therefore, initial status influences final performance. The initial status of experiment shown in section 4 is based on a polyphone dictionary and a set of polyphone pronunciation rules, which is considered as a good initial status. In order to test the influence of initial status, a comparative bad initial status is generated by shielding the dictionary and the rules set, which means giving each polyphone an uniform pronunciation first (such as the initial pronunciation of polyphone “为” is “wei2”). The results with bad initial status, including original and final accuracy, are shown in table 6:

Table6 Results of some polyphones with bad initial status

Polyphone	Initial pronunciation	Original accuracy	Final Accuracy
长	Zhang3	0.446108	0.855280
为	wei2	0.490175	0.817686
倒	dao3	0.599174	0.774793
朝	chao2	0.862963	0.988889
重	Chong2	0.195946	0.878378

Comparing the results shown in table 5 and 6, good initial status will results in better result. Although TBL is an automatic learning algorithm, good initial assignment is very important.

5.2 Comparative experiments with decision tree

Decision tree (DT) is a decision-making mechanism which assigns a probability to each possible choice based on the context: *P(f|h),* where f is an element of the feature vocabulary (the set of choices) and h is a history (the context of the decision). This probability *P(f|h)* is determined by asking a sequence of questions Ql Q2 ... Qn about the context, where the ith question asked is uniquely determined by the answers to the previous i-1 questions.

Each question asked by the decision tree is represented by a tree node and the possible answers to this question are associated with branches emanating from the node. The best question at a node is the question which maximizes the likelihood of the training data at that node after applying the question (shown in Fig.2).

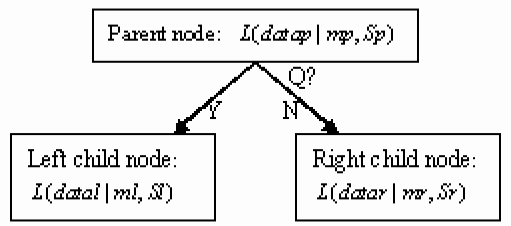
Parent node: L(datap | mp, Sp)
Q?
Y
N
Left child node: L(datal | ml, Sl)
Right child node: L(datar | mr, Sr)

Fig.2: Splitting node by evaluating question process

Each node defines a probability distribution on the space of possible decisions. A node at which the decision tree stops asking questions is a leaf node. The leaf nodes represent the unique states in the decision-making problem, i.e. all contexts which lead to the same leaf node have the same probability distribution for the decision.

The experiment base on decision tree (DT) is also used the same corpus and features. Comparing with TBL with good initial states, the result is shown in the following table.

Table7 Comparative results using TBL and DT

Polyphone	Origin accuracy	TBL	DT
长	0.973054	0.978346	0.892495
为	0.779476	0.971616	0.842623
倒	0.731405	0.828512	0.807453
朝	0.812963	0.914815	0.795796
重	0.831081	0.908072	0.746575

6 Conclusions

According to the results, the transformation-based error-driven algorithm is very effective for generating rules for polyphones. It improves the performances both for the polyphones which have low original accuracy (such as increases the correct rate to 97.2% from 77.9% for polyphone "为") and for the polyphones which already have high original accuracy (such as increases the correct rate to 97.0% from 97.8% for polyphone "应").

From the comparative experiments with different initial status, it is inferred that a better result of TBL algorithm depends on a good initial assignment. From another comparative experiment with decision tree, the differences and advantages of TBL algorithm is analyzed below:

1) The TBL algorithm creates a relatively small number of rules that are linguistically motivated and understandable by both humans and machines;
2) Each time the depth of the decision tree is increased, the average amount of training material available per node at that new depth is halved (for a binary tree). In TBL, the entire training corpus is used for finding all transformations, and therefore this method is more resistant to sparse data problems;

Acknowledgements

Special Thanks to Dr. Liqin Shen for offering this great opportunity to do the research work in IBM China Research Lab, also thanks to Xijun Ma, Yong Qin, Haiping Li, Yi Liu etc for giving many useful advices and help.

7 References

[1] T.J. Sejnowski and C.R. Rosenberg, "*Parallel networks that learn to pronounce english text*" Complex systems, vol. 1, pp. 145–168, 1987.

[2] Andersen, R. Kuhn, A. Lazarides et al. "*Comparison of Two Tree-Structured Approaches for Grapheme-to-Phoneme Conversion*", Proc. ICSLP'96, pp. 1808-1811.

[3] Wern-Jun Wang, Shaw-Hwa Hwang, Sin-Horng Chen, "*The broad study of homograph disambiguity for mandarin speech synthesis*" ICSLP96(3) pp:1389-1392

[4] F. Yvon, "*Grapheme-to-phoneme conversion using multiple unbounded overlapping chunks*" in Proceedings of Conference on New Methods in Natural Language Processing (NeMLaP), Turkey,1996, pp. 218–228.

[5] Zi-Rong Zhang, Min Chu, "*An efficient way to learn rules for grapheme-to-phoneme conversion in Chinese*", (ISCSLP) 2002, pp:59

[6] E.Brill. 1995, "*Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging*", Computational Linguistics, 21(4):543-565

[7] Ferro, L., Vilain, M., & Yeh, A, "*Learning transformation rules to find grammatical relations*", Workshop on Computational Natural Language Learning (GNLL-99) (1999), pp:43-52

[8] David M. Magerman, "*Statistical decision-tree models for parsing*", Proceedings of the 33rd Annual Meeting of the ACL, 1995, pp: 276 - 283

[9] X.J. Ma, W. Zhang, "*Automatic Prosody Labeling Using both Text and Acoustic Information*," Proc. ICASSP, HongKong, 2003.

[10] Byeongchang Kim, Gary Geunbae Lee, "*Decision-Tree based Error Correction for Statistical Phrase Break Prediction in Korean*". Proc. 17th on Computational linguistic, 2000: 1051-1055