TREE-GUIDED TRANSFORMATION-BASED HOMOGRAPH DISAMBIGUATION IN MANDARIN TTS SYSTEM

Fangzhou Liu[1], Qin Shi[2], Jianhua Tao[1]

1. National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, 100080, Beijing, China
{fzliu, jhtao}@nlpr.ia.ac.cn
2. IBM China Research Lab, 100083, Beijing, China
shiqin@cn.ibm.com

ABSTRACT

Homograph disambiguation is the core issue of the grapheme-to-phoneme conversion in Mandarin Text-to-Speech system. In this paper, a hybrid algorithm called tree-guided transformation-based learning (TTBL), which combines decision tree with transformation-based learning (TBL), is proposed to resolve homograph ambiguity. It can automatically generate templates, thereby avoiding manually summarizing templates, which is time-consuming and laborious in conventional TBL. In addition, the paper evaluates various keyword selection approaches in different domains. Results of comparative experiments show that, for the task of homograph disambiguation, templates automatically generated by decision tree achieve comparable performance to manually summarized templates, and the TTBL significantly outperforms decision tree.

Index Terms— homograph disambiguation, grapheme-to-phoneme, transformation-based learning, decision tree

1. INTRODUCTION

Grapheme-to-phoneme conversion is an essential component of Text-to-Speech (TTS) system, and directly affects the intelligibility of TTS system. The main problem in Mandarin grapheme-to-phoneme conversion is how to pick out one correct pronunciation from several candidates for polyphones. Most of early TTS system used manual pronunciation rules to disambiguate homograph. However, with the increase of the rule number, the context environment of polyphone may be matched by more than one rule, and thus conflict of rules arises, which is a difficult problem of rule-based approaches. Recently, with the vigorous development of large corpus in speech synthesis, various data-driven approaches, such as log-likelihood based statistical decision lists [1], decision trees [2], extended stochastic complexity (ESC) based stochastic decision lists [3], transformation-based error-driven learning (TBL) [4], have been investigated to solve the homograph disambiguation problem. However, there are still some points that could be improved in these methods.

In this paper, we propose a hybrid algorithm called tree-guided transformation-based learning (TTBL), which combines decision tree (DT) with transformation-based learning (TBL). By converting rules generated by decision trees into TBL templates, TTBL avoids manually summarizing templates which is time-consuming and laborious in conventional TBL. Furthermore, TTBL is able to find some complex pronunciation rules which are difficult to sum up even by linguists. Besides, this paper compares a variety of keyword selection approaches to select the contextual words which are greatly useful in identifying the correct pronunciation of the polyphone. The paper also presents a comparison of TTBL with the conventional TBL and decision tree. Experimental results show that decision tree templates achieve comparable performance to manual templates, and TTBL obviously outperforms decision tree.

This paper is organized as follows: Section 2 describes the TTBL algorithm. Section 3 lists the complete feature set and describes keyword selection in detail. Section 4 presents several comparative experiments and discusses the results. Final conclusions are presented in section 5.

2. TREE-GUIDED TRANSFORMATION-BASED LEARNING

2.1. The conventional TBL framework

Transformation-based error-driven learning [5] is a successful rule-based machine learning algorithm. It has been applied to a variety of tasks, including part of speech tagging [5], noun phrase chunking [6], parsing [7] etc, often achieving state-of-the-art performance with a small and easily-understandable list of rules.

The central idea of transformation-based learning is to learn an ordered list of rules which progressively improve upon the current state of the training set. Figure 1 illustrates the learning process. At first, an initial assignment is made

based on simple statistics, and then rules are greedily learned to correct the mistakes, until no net improvement can be made. During the evaluation phase, the evaluation set is initialized by the same initial-state annotator. Each rule is then applied, in the order it was learned, to the evaluation set. The final classification is the one attained when all rules have been applied.

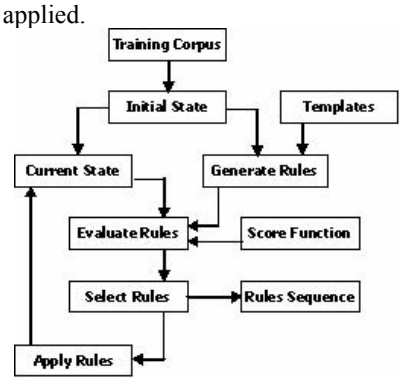

Figure 1: The framework of conventional TBL

2.2. Automatic template generation

A TBL template consists of several features and the relationship among them. For example, in template "POS(X, -1) & POS(X, 1)", "POS" indicates part of speech, "X" indicates the feature value and the number "-1" indicates the offset from the polyphone. Templates determine the predicates of rules, and have the greatest influence on the behavior of TBL system. Conventional TBL needs manual summarization of useful templates. However, that process is time-consuming and laborious, and because of the limitation of knowledge and ability, it is difficult to cover enough pronunciation rules with manually summarized templates.

As is well known, each non-leaf node of decision tree specifies a test about some useful features for classification, and the path from the root to a leaf node gives a decision rule which is composed of those features the path passes. The TTBL presented in this paper attempts to automatically generate templates by converting the rules in leaf nodes of decision tree into TBL templates. As shown in figure 2, template "POS(X, -1) & POS(X, 1)" can be derived from the leaf node A, and template "POS(X, -1) & POS(X, 1) & POS(X, 2)" can be derived from the leaf node B.

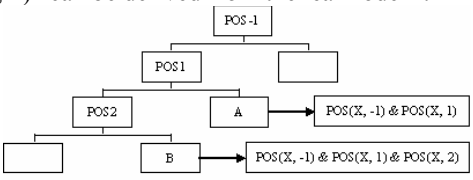

Figure 2: Converting decision rules into TBL templates

For guiding TBL with decision tree, the different characteristics between them should be considered. TBL has an initial-state annotator, and the rules learned by TBL are used to correct the errors made by the initial-state annotator, whereas rules generated by decision tree are aimed at all training set, including both the correct data correctly initialized by the TBL initial-state annotator and the error data the initial-state annotator incorrectly annotates. TBL templates should meet two requirements. First, they should be able to correct errors; second, they should not transform correct data into wrong data. Since the majority of training set are correct data, the decision rules based on all training corpus mainly predict the pronunciations of correct samples, so templates converted from these rules may be weak in error correction. Templates only based on the error data in the training data are just the opposite. They are aimed at the wrong data, and thus meet the first requirement very well, but are easy to make wrong transformation, due to lack of the supervision of correct data. Therefore, both the templates based on all training data and those only based on the error data should be taken into consideration.

3. FEATURE SELECTION

3.1. The feature set

All the features used in our system, including their offset range and value range, are listed in table 1. Note that the semantic class of a notional word is obtained through looking up in a semantic dictionary according to its part of speech. In case of semantic ambiguity, only the most frequent sense is adopted.

Feature type	Offset range	Value range
Character	±2, ±1	-
Word	±2, ±1, 0	-
POS	±2, ±1, 0	39 categories
Semantic class	±2, ±1, 0	67 classes
Length of word	±2, ±1, 0	1~8
Keyword	The entire sentence	-
POS of keyword	The entire sentence	39 categories
Semantic class of keyword	The entire sentence	67 classes
The relative position of the polyphone in the ambiguous word	-	Beginning, middle, end, single word
The relative position of the polyphone in the sentence	-	Beginning, middle, end, single sentence

Table 1: The feature set used in our system

3.2. Keyword selection

A word is considered as a keyword when its presence or absence gives more information than others. There are a variety of keyword selection techniques to measure the amount of this information in various domains. For instance, log-likelihood ratio is used to resolve English homograph ambiguity [1]. Mutual information and information gain, which are both well known as information measures, are used in text classification [8]. Cross entropy [9] is similar to information gain, with the difference that the former ignores the absence of feature. Odds ratio is commonly used in

information retrieval [9]. Inspired by the variation on mutual information [8], we propose two variants of log-likelihood ratio and odds ratio to meet the situation of more than two categories. All 5 feature scoring measures mentioned above are listed in table 2.

$LogLikelihoodRaito(W) = P(W)\sum_i P(P_i)\left\|Log(\frac{P(P_i \mid W)}{P(\overline{P_i} \mid W)})\right\|$
$MutualInformation(W) = \sum_i P(P_i)\log\frac{P(W \mid P_i)}{P(W)}$
$InformationGain(W) = P(W)\sum_i P(P_i \mid W)\log\frac{P(P_i \mid W)}{P(P_i)} + P(\overline{W})\sum_i P(P_i \mid \overline{W})\log\frac{P(P_i \mid \overline{W})}{P(P_i)}$
$CrossEntropy(W) = P(W)\sum_i P(P_i \mid W)\log\frac{P(P_i \mid W)}{P(P_i)}$
$OddsRatio(W) = P(W)\sum_i P(P_i)\left\|\log\frac{P(W \mid P_i)(1-P(W \mid \overline{P_i}))}{(1-P(W \mid P_i))P(W \mid \overline{P_i})}\right\|$

Table 2: Formulas of keyword selection, where $P(W)$ is the probability that word W occurred, $\overline{W}$ means word W does not occur, $P(P_i)$ is the probability of the i-th pronunciation of the target polyphone, $P(P_i \mid W)$ is the conditional probability of the i-th pronunciation given that word W occurred, and $P(W \mid P_i)$ is the conditional probability of word W presence given the i-th pronunciation.

Like automatic template generation presented in section 2.2, the error-driven characteristic of TBL should also be taken into account in keyword selection. Because correct data is in the majority of all training data, keywords selected from all training set mainly identify the pronunciation of correct samples which have already been correctly annotated by the initial-state annotator. Keywords selected only from the error data are aimed at the wrong data, and hence may have a better error correction capability than the ones selected from all training set.

4. EVALUATION AND DISCUSSION

There are 1036 polyphones in modern Chinese characters [3], but most of them have dominating pronunciations or rarely appear in usual articles. We select 33 key polyphones which are most ambiguous and frequently used as study objects. 5000 sentences per polyphone on average are collected from "People's Daily", which have been automatically preprocessed by the front end of our Mandarin TTS system, including word segmentation, POS tagging, and pronunciation labeling. After repeatedly manually proofreading the pronunciations of polyphones, this corpus is divided into training set, development set and test set according to an 8:1:1 ratio. All the experiments described below are based on this corpus.

4.1. Experiments of keyword selection

To compare the performance of keyword selection methods described in section 3.2, 200 keywords per polyphone with the highest score are selected from training set, and experiments are performed on test set using only the keyword feature. Table 3 shows the test result. The initial average precision which achieved by our initial-state annotator based on manual rules is 80.66%.

Method type	Average precision
Log-likelihood ratio	**85.50%**
Mutual information	81.66%
Information gain	84.30%
Cross entropy	84.84%
Odds ratio	85.27%

Table 3: Comparison of keyword selection measures, where

$$\text{average precision} = \frac{\text{total number of the correct test samples of all polyphones}}{\text{total number of all test samples of all polyphones}}$$

Mutual information has inferior performance compared to the other methods due to its bias favoring rare terms. The main reason that information gain performs slightly worse than cross entropy is that TBL hardly uses the template of keyword absence whose information accounts for a large proportion of information gain. The performance of odds ratio is very close to that of log-likelihood ratio which is the simplest but also the most effective approach.

As mentioned in section 3.2, keywords can be selected from either all training data or only the error data. Figure 3 shows a breakdown of their performance on 13 frequent polyphones. Keywords only from the error data outperform those from all training set obviously. The average precision of the former is 2.43% higher than that of the latter.

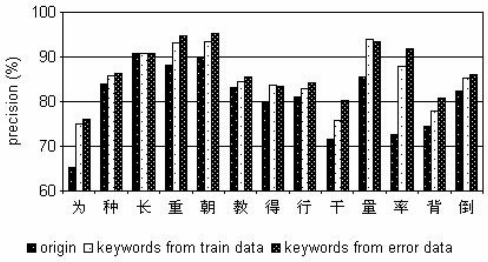

Figure 3: Comparison of keywords from different data

4.2. Experiments of template generation

Figure 4 shows the performance of decision tree templates based on different data. The average precision of decision tree templates based on all training data is 89.38% and that of decision tree templates only based on the error data is 88.31%. Combining the two can achieve the best average precision 90.36%. That is because it combines the advantages of both templates, thus better satisfying those two requirements mentioned in section 2.2.

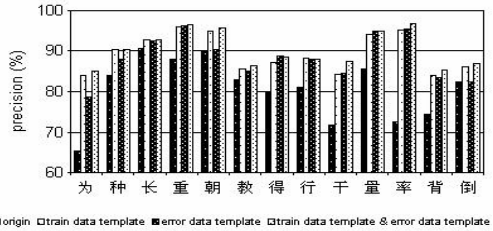

Figure 4: Comparison of DT templates from different data

Figure 5 compares the performance of manual templates with that of decision tree templates. The average precision of decision tree templates achieves 90.36%, while that of manual templates is 90.60%. The performance of decision tree templates is very close to that of manual templates, and even better on some polyphones such as "重", "干", "率" etc. Combining the two can achieve the best average precision 91.13%. Therefore, decision tree templates can be used as good substitutes for manual templates, and are still able to provide beneficial supplement for manual templates even if manual templates have been summarized.

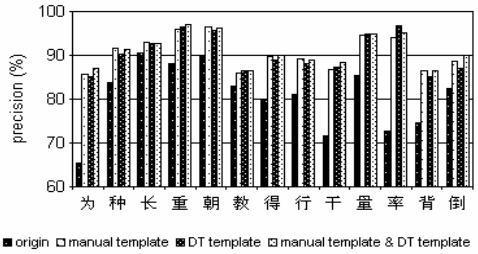

Figure 5: Comparison between DT templates and manual templates

4.3. Comparison between TTBL and decision tree

We also made a comparison of our approach with decision tree, using the identical corpus and feature set. A breakdown of their performance on 13 frequent polyphones is shown in figure 6. When not using the highly lexicalized features such as characters and lexicon words, the average precision of decision tree is 87.82%; when the feature set includes these features, its average precision falls to 85.30%. That is because of the well-known data fragmentation problem of decision tree. The average precision of TTBL achieves 90.36%, greatly exceeding decision tree.

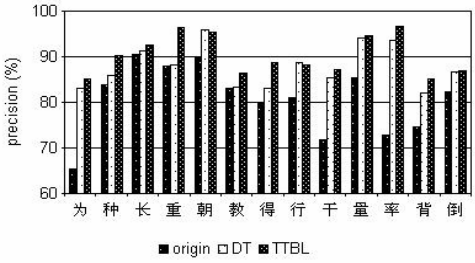

Figure 6: Comparison between decision tree and TTBL, where decision tree does not use lexicalized features

Our approach mainly has the following two advantages over decision tree:

1) Unlike decision tree, transformation-based learning does not recursively split the data, and hence does not suffer from unreliable counts due to data fragmentation.
2) Transformation-based learning can take advantage of the valuable early system based on manual rules by using it as the initial-state annotator.

5. CONCLUSION

In the paper, a hybrid approach is presented to resolve homograph ambiguity, which increases the average precision of 33 key polyphones to 90.36% from 80.66%. It uses decision tree to automatically generate TBL templates, thereby reducing human supervision. The excellent performance of decision tree templates indicates that they can substitute or provide beneficial supplement for manual templates. Moreover, comparative experiments on keyword selection and template generation demonstrated that it is very important to consider the error-driven characteristic of transformation-based learning.

Because of the good performance of our approach on homograph disambiguation, we consider to apply it to some similar tasks of disambiguation, such as text normalization.

6. ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China (No. 60575032, No. 70611120555) and 863 Programs (No. 2006AA01Z138, No. 2006AA01Z194). We thank Yong Qin, Danning Jiang and Zhiwei Shuang for their invaluable advice. In addition, we would like to thank Jian Yu and the anonymous reviewers for their useful comments and suggestions on the paper.

7. REFERENCES

[1] David Yarowsky, "Homograph disambiguation in speech synthesis", *Progress in Speech Synthesis*, Springer-Verlag, pp. 159–175, 1997.

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

[3] Zirong Zhang, Min Chu, "An efficient way to learn rules for grapheme-to-phoneme conversion in Chinese", *ISCSLP2002*, pp. 59.

[4] Min Zheng, Qin Shi et al, "Grapheme-to-phoneme conversion based on TBL algorithm in Mandarin TTS system", *INTERSPEECH2005*, pp. 1897-1900.

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

[6] Lance A. Ramshaw and Mitch P. Marcus, "Text Chunking Using Transformation-based Learning", *Natural Language Processing Using Very Large Corpora*, Kluwer, 1999.

[7] Eric Brill, "Learning to Parse with Transformations", *Recent Advances in Parsing Technology*, Kluwer, 1996.

[8] Yiming Yang, Jan O. Pedersen, "A comparative study on feature selection in text categorization", *ICML97*, pp. 412-420.

[9] Dunja Mladenic, Marko Grobelnik, "Feature selection for unbalanced class distribution and Naive Bayes", *ICML99*, pp. 258-267.