Combining Active and Semi-supervised Learning for Homograph Disambiguation in Mandarin Text-to-Speech Synthesis

Binbin Shen[1], *Zhiyong Wu*[1], *Yongxin Wang*[2], *Lianhong Cai*[1,2]

[1] Tsinghua-CUHK Joint Research Center for Media Sciences, Technologies and Systems
Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China
[2] Tsinghua National Laboratory for Information Science and Technology (TNList)
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

cbb09@mails.tsinghua.edu.cn, zywu@sz.tsinghua.edu.cn,
wangyongxin@mails.tsinghua.edu.cn, clh-dcs@tsinghua.edu.cn

10.21437/Interspeech.2011-567

Abstract

Grapheme-to-phoneme conversion (G2P) is a crucial step for Mandarin text-to-speech (TTS) synthesis, where homograph disambiguation is the core issue. Several machine learning algorithms have been proposed to solve the issue by building models from well annotated training corpus. However, the preparation of such well annotated corpus is very laboring and time-consuming which requires lots of manual hand-label work to validate the proper pronunciations of the homographs. This work tries to cover this problem by introducing the active learning (AL) and semi-supervised learning (SSL) algorithms for the homograph disambiguation task using unlabeled data. Experiments show that the proposed framework can greatly reduce the cost of manual hand-label work while preserving the performance of the trained model.

Index Terms: text-to-speech (TTS) synthesis, homograph disambiguation, active learning (AL), Yarowsky algorithm, semi-supervised learning (SSL)

1. Introduction

Grapheme-to-phoneme (G2P) is of great importance for text-to-speech (TTS) synthesis. Unlike the alphabetic languages such as English where the main problem of G2P is to generate correct pronunciations for out of vocabulary (OOV) words [1], the core issue in Chinese G2P conversion is homograph disambiguation [2]-[5]. Homograph disambiguation aims to get correct pronunciation from several candidates according to the context information such as part-of-speech, word and position information of the homographs in a word or sentence.

Conventionally, homograph disambiguation is solved by a set of decision rules that are generated manually by experts. However, the generation of decision rules is solely dependent on the knowledge of experts and can be very time consuming. Several machine learning algorithms have also been proposed and applied to solve the homograph disambiguation problem, including maximum entropy (ME) models with Gaussian or inequality smoothing [2], transformation-based error-driven learning (TBL) with different methods to generate templates [3][4], stochastic decision list [5], etc. Most of these methods have proposed to build models from well annotated training corpus and put focus on minimizing the error rate on the test set of the used corpus. However, the preparation of such well annotated training corpus is very laboring and time-consuming as it requires lots of manual work to get correct pronunciation.

This work tries to cover the above problem by introducing an active and semi-supervised learning framework for the homograph disambiguation task using unlabeled data. In the framework, active learning (AL) is used to select a small set of most informative samples from the unlabeled data for manual labeling and semi-supervised learning (SSL) is then used to train the model with a large set of unlabeled data and much reduced amount of manual labor.

Section 2 gives a brief introduction to the TBL algorithm, which serves as the base model of our work. The proposed active and semi-supervised learning framework for homograph disambiguation is then detailed in Section 3. The corpus and the basic setup for TBL are described in Section 4, followed by detailed discussion of experiments in Section 5. Some final remarks are given in Section 6.

2. Transformation-based learning (TBL)

Since Transformation-based error-driven learning (TBL) was first introduced to solve part-of-speech tagging problem [6], it has been one of the most successful rule-based learning algorithms in natural language processing. The idea of TBL is to learn an ordered list of transformation rules from the candidates according to their contribution to the training data.

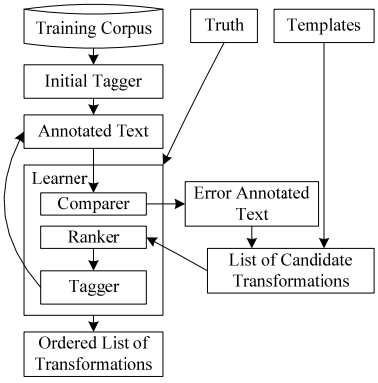


Figure 1: *Learning process of transformation-based error-driven learning (TBL) algorithm.*

Figure 1 shows the learning process of TBL algorithm. The unlabeled samples from training corpus will be first tagged with an initial tagger. The tagging results will then be compared with the truths given in the annotated corpus. All the error-annotated samples will be used to generate a list of candidate transformations according to manually prepared templates, where transformation is used to transform original tagging results to new ones. The score of each candidate transformation is measured by its contribution (the decrement of error numbers after and before applying the transformation). The transformation with the highest score is then added to the ordered transformation list and the training corpus is updated by applying the learnt transformation. Iterations will continue until no more improvements can be made or the highest score is below a predefined threshold.

Copyright © 2011 ISCA

3. Active and semi-supervised learning for homograph disambiguation

3.1. Proposed framework

Figure 2 illustrates the diagram of the proposed framework. A large scale unlabeled corpus has been collected (unlabeled texts in Figure 2). Some samples are selected and labeled manually as the seed pool for training the base TBL model after a thorough examination of characteristics of different homographs. The trained model is then used to tag the samples from the unlabeled pool. Uncertainty derived from entropy is evaluated from the tagging results of these examples. Active learning algorithm [7] is used to find most informative samples from the unlabeled pool. The most uncertain samples (i.e. the samples with the biggest uncertainty) are believed to demonstrate new patterns that have not been found in the current labeled pool, and are marked as the most informative samples. They are automatically selected, manually labeled or confirmed by experts, and then added to the labeled pool. Meanwhile, Yarowsky algorithm [8] (a semi-supervised learning method) is carried out by choosing most confident samples (i.e. the samples with the lowest uncertainty) with current estimated labels, and adding them to the labeled pool. The model is retrained using the updated labeled pool so that it can learn the new patterns from the newly added samples.

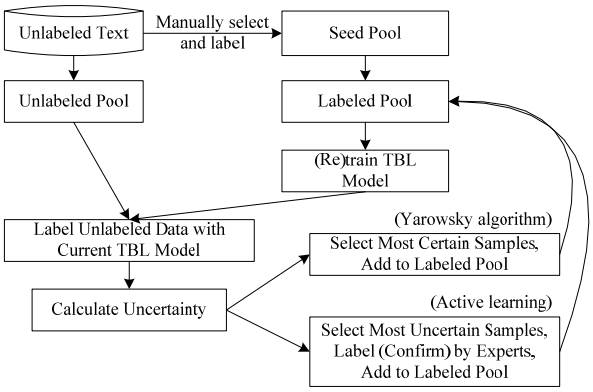


Figure 2: *Diagram of the proposed framework that combines Yarowsky algorithm and active learning for homograph disambiguation.*

3.2. Active learning

As illustrated in Table 1, the active learning for homograph disambiguation starts with an unlabeled pool U_0, and a seed pool of manually labeled samples $A_0=\{x_i,\ x_i \in U_0\}$. TBL training algorithm is performed on the seed pool A_0 to get the base model M_0. N iterations continue to retrain the model by selecting samples from the unlabeled pool. The selecting strategy considers *the uncertainty* of the tagging results of the samples while applying current model to them. For iteration n, the current model M_n is used to label samples in the unlabeled pool U_n and calculate the uncertainty for each sample. Thereafter, a set of maximum K_{AL} samples ($S_{AL,n}$) with highest uncertainty are selected by the strategy, where the uncertainty of each sample must be greater than T_{AL}. These samples are then manually labeled or confirmed by the experts. Meanwhile, another set of maximum K_{SSL} samples ($S_{SSL,n}$) with lowest uncertainty are selected, where the uncertainty of each sample must be less than T_{SSL}. The automatically labeled most confident samples $S_{SSL,n}$ together with the manually labeled most uncertain samples $S_{AL,n}$ are then added to the labeled pool A_n to produce a new labeled pool A_{n+1} for semi-supervised learning. The new model M_{n+1} is then retrained over A_{n+1}. Iterations will terminate if no more improvements can be made.

Table 1. *Active learning for homograph disambiguation.*

1: Unlabeled pool: U_0
2: Seed pool with manual labels: $A_0=\{x_i, x_i \in U_0\}$
3: Train base model using: $A_0 \rightarrow M_0$
4: **for** $n = 0$ to N **do**
5: $S_{AL,n}$ = UncertainSampleSelection(U_n, A_n, M_n, K_{AL}, T_{AL})
6: Label or confirm by experts for $S_{AL,n}$
7: $S_{SSL,n}$ = CertainSampleSelection(U_n, A_n, M_n, K_{SSL}, T_{SSL})
8: $A_{n+1} = A_n + S_{AL,n} + S_{SSL,n}$
9: $U_{n+1} = U_n - S_{AL,n} - S_{SSL,n}$
10: Retrain model using: $A_{n+1} \rightarrow M_{n+1}$
11: End iteration if no more improvements can made
12: **end for**

3.3. Uncertainty sampling

In the above framework, the most uncertain samples $S_{AL,n}$ and the most confident samples $S_{SSL,n}$ should be automatically selected to update the labeled pool and to retrain the model. In this work, entropy is used as the query strategy in active learning and confidence measurement in self-training.

3.3.1. Uncertain sample selection

For each sample in the unlabeled data pool, the homographs contained in the current sample are first disambiguated by the current trained model; and a score is computed to measure the uncertainty of the current trained model in disambiguating the homograph in the current sample.

A scoring criterion is designed to select at most K_{AL} samples that have the *top highest uncertainty scores* under current trained model, where the uncertainty score of each selected sample must be greater than a predefined threshold T_{AL}. These samples are believed to contain homographs that are difficult to be disambiguated by the current automatically trained model. Incorrect pronunciations might be produced if the model cannot deal well with such samples, and will thus hurt the performance of the G2P module in getting correct pronunciation for the TTS system.

Manual correction of the labeling results for homographs in such most uncertain samples will produce the maximum benefit to the model in the next iteration of training procedure.

3.3.2. Confident sample selection

On the other hand, the least samples of manual labeling or corrections are preferred to reduce the cost of data acquisition.

Another criterion is used to select at most K_{SSL} samples with the *top least uncertainty scores* under current trained model; and the score of each selected sample is less than the threshold T_{SSL}. The homographs contained in these samples could be easily disambiguated by the current trained model, which means that the current model is quite confident about the labeling results of the homographs. Thus, the estimated labels by current model of the homographs in these samples are treated as the correct labels directly.

The samples with most certainty under current model are automatically labeled, selected and added to the labeled pool for retraining model in next iteration. The manual labeling is not required for such samples. In this way, the cost of manual work in preparing the training data set can be greatly reduced.

3.3.3. Uncertainty measurement

Entropy is often used as a measurement of uncertainty or impurity in machine learning, and it can be easily generalized to probabilistic multi-label classifiers. In this work, entropy is adopted as the uncertainty measurement for *sample x under the current model θ*:

$$H_x = -\sum_i P_\theta(y_i \mid x) \log P_\theta(y_i \mid x) \quad (1)$$

Where y_i ranges over all possible pronunciations of sample x, $P_\theta(y_i|x)$ is the probability of pronunciation y_i given sample x.

This work uses Radu's method [9] to compute probability distribution $P_\theta(y_i|x)$, as implemented in fnTBL toolkit. The method defines equivalence class as a set of samples that share the same characteristics: all samples in the class are applied the same TBL rule sequence. The training set is then divided into several equivalence classes, and the distribution for each class is estimated by using maximum likelihood estimation:

$$P_\theta(y_i \mid x) = \frac{count(c, y_i)}{count(c)} \quad (2)$$

Where c is the equivalence class that sample x belongs to, $count(c)$ is the number of all samples in class c, and $count(c,y_i)$ is the number of samples in class c with pronunciation y_i.

4. Corpus collection and basic TBL setup

4.1. Corpus collection

Out of 1036 homographs in Chinese characters [5], 108 key homographs have been selected as the target characters for the study of homograph disambiguation. These key homographs are most ambiguous and frequently used in our daily life, and hence are of great importance for homograph disambiguation. Detailed procedures on key homograph character selection can be found in [3].

A text corpus for homograph disambiguation study is collected from the "People's Daily" newspaper. At least 2000 samples (i.e. sentences) are collected for each key homograph character. These sentences are automatically processed by the text analysis module of our homegrown Mandarin TTS system [10], including word tokenization, part-of-speech tagging, and grapheme-to-phoneme conversion. The pronunciations of the homograph characters are further manually checked to ensure the accuracy of the corpus.

4.2. Setup for TBL algorithm

In this work, TBL algorithm is used not only as the benchmark but also as the base model in the active learning framework.

For the TBL algorithm, the default G2P results based on pronunciation dictionary are used as the initial states and the manually checked results are used as the target states. Feature selection and template design are two essential components for the usage of TBL.

4.2.1. Features

Features used in TBL algorithm include adjacent characters (from previous 2 to next 2 characters), words (from previous 2 to next 2 words), part-of-speech (POS) of the words (the POS categories use the specification from [11]), the words that appear before the homograph in the sentence, the words that appear after the homograph in the sentence, and the last characters of the right words. The features are summarized in Table 2.

Table 2. *Summarization of the features used in this work. The POS categories use the specification from [11].*

Features	Feature description	Offset
LC	Character	±2, ±1
LW	Lexical word	±2, ±1, 0
POS	Part-of-speech of the word	±2, ±1, 0
LWS	Words that appear before the homograph in the sentence	-
RWS	Words that appear after the homograph in the sentence	-
RWES	Last characters of words in RWS	-

4.2.2. TBL templates

Simplified templates are defined to avoid over-fitting problem. For the homograph character LC_i, and the pronunciation tag Y_i, the defined TBL templates might include the templates for the above different single features as defined in Equation (3-8).

$$(LC_{i+j} \rightarrow Y_i \mid j = \pm 2, \pm 1) \quad (3)$$

$$(LW_{i+j} \rightarrow Y_i \mid j = \pm 2, \pm 1, 0) \quad (4)$$

$$(POS_{i+j} \rightarrow Y_i \mid j = \pm 2, \pm 1, 0) \quad (5)$$

$$(LWS_i \rightarrow Y_i), LWS_i \in \{LW_{i+j} \mid j = -10, \ldots, -1\} \quad (6)$$

$$(RWS_i \rightarrow Y_i), RWS_i \in \{LW_{i+j} \mid j = 1, \ldots, 10\} \quad (7)$$

$$(RWES_i \rightarrow Y_i), RWES_i = WordEnd(RWS_i) \quad (8)$$

5. Experiments

5.1.1. Experimental setup

10-fold cross-validation is used to perform the experiments. For each homograph character, all the samples are randomly partitioned into 10 subsets. Out of the 10 subsets, a single subset is retrained as the test set for model evaluation, and the other 9 subsets are used as the training set. Cross-validation process is repeated 10 times, with each of the 10 subsets used exactly once as the test set. The accuracy of the model for homograph disambiguation is computed, and the 10 results from all the folds are then averaged to produce the final result of accuracy.

For conventional TBL, TBL transformations are learned from all the samples in the training set for each homograph.

As a comparison, in active and semi-supervised learning framework, the training set are treated as unlabeled data; and the manual labeling procedure of the most uncertain samples $S_{AL,n}$ for each iteration is simulated by providing the annotation result in the training set directly.

5.1.2. Experiments

Experiments were conducted to evaluate whether the proposed active and semi-supervised learning framework can reduce the cost of manual labeling while preserving the performance of the trained TBL model.

Different parameter settings of (K_{AL}, T_{AL}, K_{SSL}, T_{SSL}) may affect the performance of the proposed framework. Two groups of experiments were conducted. The first group (A) performed experiments on K_{SSL} and T_{SSL} by fixing K_{AL} and T_{AL}. Another group (B) performed experiments on K_{AL} and T_{AL} by fixing K_{SSL} and T_{SSL}. Table 3 shows the experimental results for homograph '背' (pronounced as 'bei1' or 'bei4').

As can be seen from group (A) in Table 3, the accuracy of the proposed framework degrades when K_{SSL} increases. This is because K_{SSL} and T_{SSL} control the number of the most

Table 3. *Results of parameter setting experiments for homograph '背' (with pronunciation 'bei1' or 'bei4').*

Exp. Group	K_{AL}	T_{AL}	K_{SSL}	T_{SSL}	# *iterations*	# manually labeled samples		% accuracy of estimated labels in S_{SSL}	% accuracy of the proposed framework (# learnt TBL rules)
						# seeds	# S_{AL}		
A	10	0.5	∞	0.1	4	100	29	83.42%	80.67% (7)
			1000	0.1	6		43	84.04%	80.67% (6)
			100	0.1	24		162	86.61%	84.76% (11)
B	10	0.5	10	0.1	100	100	620	97.70%	96.65% (25)
	20	0.5	20	0.1	91		624	98.48%	97.77% (23)
	30	0.5	30	0.1	59		673	97.70%	97.03% (24)
	40	0.5	40	0.1	46		618	97.93%	97.77% (23)
	50	0.5	50	0.1	37		680	97.86%	97.77% (23)
	70	0.5	70	0.1	26		685	97.76%	97.77% (24)
	100	0.5	100	0.1	18		754	97.20%	97.03% (23)

1) The number of the samples in training set and test set is 2413 and 269 respectively.
2) The accuracy of the conventional TBL algorithm is 97.77%, and the number of learnt rules is 23.

confident samples selected for semi-supervised learning; and the accuracy of the estimated labels of these samples will affect the performance of the trained model. On the other hand, K_{AL} and T_{AL} parameters control the number of the most uncertain samples selected by active learning, and decide the number of samples to be manually labeled. As indicated by group (B) in Table 3, the numbers of the manually labeled samples are significantly reduced by more than 64%.

Differences between the automatically selected samples by the framework and the original labeled samples in data set are further investigated. Results indicate the difference mainly lies in the samples that appear very few times in the data set.

Out of different combination of parameter settings of (K_{AL}, T_{AL}, K_{SSL}, T_{SSL}), (50, 0.5, 50, 0.1) is selected as the parameters by considering the balance between the accuracy performance of the framework and the number of manually labeled samples for each iteration.

Table 4 shows the results for five key homographs. As can be seen, the proposed framework can significantly reduce the manual effort to label the training data that are required for the model training; while the performance of the TBL trained in the proposed framework is similar to the conventional TBL model trained on all the samples.

Table 4. *Performance of the conventional TBL and the proposed framework for homograph disambiguation, where "% decrease of manual work" is computed by "(# samples - # manually labeled samples) / (# samples)".*

Homograph characters	# samples	# manually labeled samples	% decrease of manual work	% accuracy	
				TBL	Proposed framework
调	29534	4800	83.75%	95.92%	95.06%
倒	1992	649	67.42%	95.95%	96.40%
当	25002	3833	84.67%	97.23%	96.29%
散	3267	866	73.49%	98.90%	97.25%
背	2413	780	67.68%	97.77%	97.77%

6. Conclusions

Homograph disambiguation is the core issue for grapheme-to-phoneme conversion in Mandarin text-to-speech synthesis. A large scale well annotated corpus is always required to learn models from the corpus by machine learning algorithms. The preparation of such well annotated training corpus is very difficult and time-consuming as it will require lots of manual hand-label work to validate of the proper pronunciations of the homographs. This work proposes a framework by combining the active learning and semi-supervised learning algorithms for homograph disambiguation using unlabeled corpus. In the framework, active learning is used to select a small set of most informative samples from the unlabeled data for manual labeling and semi-supervised learning is used to train model with both labeled and unlabeled data. Experimental results indicate that the proposed framework can greatly reduce the manual effort in labeling data; while the performance of the framework is similar to the conventional method.

7. Acknowledgements

This work is supported by the National Natural Science Foundation of China (60805008, 60928005, 60931160443 and 61003094), and the Ph.D. Programs Foundation of Ministry of Education of China (200800031015).

8. References

[1] Andersen, O., Kuhn, R., Lazarides, A., Dalsgaard, P., Haas, J. and Noth, E., "Comparison of two tree-structured approaches for grapheme-to-phoneme conversion", in *Proc. SLP*, 3: 1700-1703, 1996.

[2] Mao, X., Dong, Y., Han, J. and Wang, H., "A comparative study of diverse knowledge sources and smoothing techniques via Maximum Entropy for polyphone disambiguation in Mandarin TTS systems", in *Int. Conf. Natural Language Processing and Knowledge Engineering*, 162-169, 2007.

[3] Zheng, M., Shi, Q., Zhang, W. and Cai, L., "Grapheme-to-phoneme conversion based on TBL algorithm in Mandarin TTS system", in *Proc. InterSpeech*, 1897-1900, 2005.

[4] Liu, F., Shi, Q. and Tao, J., "Tree-guided transformation-based homograph disambiguation in Mandarin TTS system", in *Proc. ICASSP*, 4657-4660, 2008.

[5] Zhang, Z., Chu, M. and Chang E., "An efficient way to learn rules for grapheme-to-phoneme conversion in Chinese", in *Proc. ISCSLP*, paper 59, 2002.

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

[7] Settles, B, "Active learning literature survey", *Computer Sciences TR 1648*, University of Wisconsin—Madison, 2009.

[8] Yarowsky, D., "Unsupervised word sense disambiguation rivaling supervised methods", in *Proc. ACL*, 189-199, 1995.

[9] Florian, R., Henderson, J. and Ngai, G., "Coaxing confidences from an old friend: probabilistic classifications from transformation rule lists", in *Proc. EMNLP*, 26-34, 2000.

[10] Wu, Z., Cao, G., Meng, H. and Cai, L., "A unified framework for multilingual text-to-speech synthesis with SSML specification as interface", *Tsinghua Science and Technology*, 14(5): 623-630, 2009.

[11] Yu, S., Duan, H., Zhu, X. and Sun, B., "The basic processing of contemporary Chinese corpus at Peking university specification", *Journal of Chinese Information Processing*, 16(5): 49-64, 2002.