IMPROVING MISPRONUNCIATION DETECTION USING MACHINE LEARNING

Yuqiang Chen[1*], *Chao Huang*[2], *Frank Soong*[2]

[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
[2]Microsoft Research Asia, Beijing, China
kentcyq@sjtu.edu.cn, {chaoh, frankkps}@microsoft.com

ABSTRACT

In this paper, we investigate the problem of mispronunciation detection by considering the influence of speaker and syllables. Machine learning techniques are used to make our method more convenient and flexible for new features, such as syllables normalization. The experimental results on our database, consisting of 9898 syllables pronounced by 100 speakers, show the effectiveness of our method by reducing the average false acceptance rate (FAR) by 42.5% using data set generated by model without adaptation to observation set and reducing average FAR by 32.5% using data set generated by model with adaptation to observation set.

Index Terms— Computer Aided Language Learning (CALL), Automatic Mispronunciation Detection (AMD), Machine Learning

1. INTRODUCTION

Computer Assisted Language Learning (CALL) has received a considerable attention in recent years. As a part of CALL system, automatic mispronunciation detection (AMD) [1, 2] is regarded as an important tool since it provides realtime feedback on pronunciations, which is very helpful for language learners. Our work is focused on the AMD in Mandarin which has roughly 2000 tonal syllables. Each syllable normally consists of three parts: an initial (mainly a consonant), a final (mainly a vowel) and a tone. Pronunciation problem on any of the parts is regarded as a mispronunciation.

Much progress has been made in AMD in Mandarin. In the previous works, Franco et al. [2] used posterior probability scores based on Hidden Markov Models (HMM) and log-likelihood ratio score given by Gaussian mixture models for pronunciation error detection. Zhang et al. [3] proposed scaled log-posterior probability (SLPP) as an improvement. In addition to SLPP, some other features were also reported effective. For example the speaker normalization [4] measures the average proficiency level of a speaker and provides useful information for giving a final score. Different from all these previous works, we propose in this paper to add a new feature, the phone normalization, which will be shown later in the paper to be very useful as well. However with more and more features, currently there is not a ready-to-use probabilistic framework to put theses features in, at least to our best of knowledge. To handle the problem, some previous work [4] took the way of using heuristical weighting factor to utilize features. Unfortunately when given too many features, constructing and tuning a good weighting factor function not only very difficult but also very tricky, and furthermore, it even cannot be determined whether a heuristic weighting factor function is good enough or not. Thus in this paper, we try a novel way, which is to use machine learning methods to learn the decision function since machine learning algorithms, such as support vector machine [5], are theoretically sound, relatively mature and always achieves optimal solutions with respect to certain criteria, for example Naive Bayes achieves maximum a posterior solution and SVM achieve maximum margin solution. Experimental results demonstrate the effectiveness of applying machine learning methods in AMD.

[*]joined in the work as an intern at Microsoft Research Asia.

2. IMPROVING MISPRONUNCIATION DETECTION

In this section, we will describe our methods that improve automatic mispronunciation detection (AMD) from two aspects. One aspect is to use more normalization techniques. In specific, the syllable normalization is introduced in this paper. The other aspect is that we explicitly regard the AMD as a classification problem, and use machine learning methods to solve the problem.

2.1. Notations

To formally discuss the problem, we first introduce some notations used in this paper. As a classification problem, we have $\mathcal{C} = \{c_1, c_2\}$ be the target class that a syllable is pronounced correctly (c_1) or incorrectly (c_2). We also have the syllable set $\mathcal{T} = \{t_1, t_2, \cdots, t_{|\mathcal{T}|}\}$, the phone set $\mathcal{R} = \{r_1, r_2, \cdots, r_{|\mathcal{R}|}\}$ and the tone set $\mathcal{V} = \{v_1, v_2, \cdots, v_{|\mathcal{V}|}\}$. As mentioned in Section 1, each syllable t has three parts: the initial $t^{\mathtt{i}} \in \mathcal{R}$, the final $t^{\mathtt{f}} \in \mathcal{R}$ and the tone $t^{\mathtt{v}} \in \mathcal{V}$. The speaker set is denoted as $\mathcal{S} = \{s_1, s_2, \cdots, s_{|\mathcal{S}|}\}$, and the isolated syllable observation set is denoted as $\mathcal{O} = \{o_1, o_2, \cdots, o_{|\mathcal{O}|}\}$. For any $o \in \mathcal{O}$, $s^o \in \mathcal{S}$ stands for the speaker who pronounced o, and $t^o \in \mathcal{T}$ stands for the reference syllable of observation o. Finally, $\mathcal{O}_s \subseteq \mathcal{O}$ is the set of observations spoken by speaker $s \in \mathcal{S}$, and $\mathcal{O}_t \subseteq \mathcal{O}$ is

the set of observations whose reference syllables are $t \in \mathcal{T}$.

2.2. Scaled Log-posterior Probability and Selective Maximum Likelihood Linear Regression

To access the goodness of pronunciation (GOP) score, scaled log-posterior probability (SLPP) [3] has been reported as a good parameter. In a HMM based speech recognizer, given observation o and phone r, the SLPP is

$$P(r|o) = \log \frac{\sum_{l \in \mathcal{L}_r} p(o|l)^\alpha p(r)}{\sum_{l \in \mathcal{L}} p(o|l)^\alpha p(r)} \quad (1)$$

where $\mathcal{L} = \{l_1, l_2, \cdots, l_{|\mathcal{L}|}\}$ is all the paths in the lattice from Viterbi decoding and $\mathcal{L}_r \subseteq \mathcal{L}$ stands for the paths that include phone r. The α is a scaling factor that scales the output score more meaningfully spreading between 0 and 1. This SLPP score is the raw score (feature) used in our method.

In the real scenario, it is possible that there are some mismatches between recognition model and the test data. To handle this problem, selective maximum likelihood linear regression (SMLLR) [3] can be used. The basic idea is to select high scored syllable for MLLR adaptation, which adapts good pronounced syllables and at same time prevents the model from adapting mispronunciations.

2.3. Normalization

As shown in [4], normalization is very effective in mispronunciation detection. In this paper, besides using speaker normalization, we also consider a new normalization technique: the syllable (phone and tone) normalization. Since each syllable has a initial, a final and a tone, and each normalization is an average on them, for the simplicity of describing our idea, here we write $t^{\mathtt{i}}$, $t^{\mathtt{f}}$ and $t^{\mathtt{v}}$, which have been defined in Section 2.1, together as syllable t, and more detailed form of normalization that was used in our experiment will be shown in Section 3.1. The speaker normalization Eq.(2) and syllable normalization Eq.(3) is shown below

$$P(s) = \frac{1}{|\mathcal{O}_s|} \sum\nolimits_{o \in \mathcal{O}_s} P(t^o|o) \quad (2)$$

$$P(t) = \frac{1}{|\mathcal{O}_t|} \sum\nolimits_{o \in \mathcal{O}_t} P(t^o|o) \quad (3)$$

As shown in Eq.(2), the speaker normalization is the average value of a speaker's all pronunciations. This can be taken as an indication of this speaker's speaking proficiency level. The syllable normalization Eq.(3) can be understood as a prior of the hardness of pronouncing a syllable. For example, if the normalization value of a specific syllable is low, we may assert this syllable is difficult to pronounce correctly and that other people would have a high probability to pronounce it incorrectly as well.

In contrast to previous work [4] that uses these normalization as weighting factors to normalize $P(t|o)$, in this paper we investigate the normalization in a classification view. In the classification view, raw score and normalization are considered as features. If without normalization, there is only one feature $P(t|o)$, which may not reflects the true level of pronunciation o because of model mismatch or speaker variation, and therefore sometimes it is impossible to separate correct and incorrect pronunciations by only looking at $P(t|o)$. But if we add the normalization information such as speaker normalization or syllable normalization, we take the speaker proficiency level or the hardness of pronouncing the syllable for consideration, which is very useful in making decision.

Take speaker normalization for example. If two observations o_1, o_2, where o_1 is correctly pronounced and o_2 is mispronunciation, have the same SLPP score: $P(t^{o_1}|o_1) = P(t^{o_2}|o_2)$, it is impossible to separate them. But if we incorporate the speaker information, such as $P(s^{o_1}) > P(s^{o_2})$ which means speaker s^{o_1} pronounces better than s^{o_2} on average, then we could separate them properly that o_1 is correctly pronounced and o_2 is mispronunciation. These more features make observations more separable from each other and thus easier for classification.

2.4. Algorithm Overview

The algorithm is as follows. First we construct a data set $\mathcal{D} = \mathcal{X} \times \mathcal{C}$ from $\mathcal{O}$, where $\mathcal{X} = \{x_1, \cdots, x_{|\mathcal{X}|}\}$. For each observation o, there is a corresponding instance $x_o \in \mathcal{X}$, which is a vector that take the raw score $P(t^o|o)$ and the normalization $P(t^o)$, $P(s^o)$ as features. With this data set $\mathcal{D}$, we apply machine learning method on them for training and testing. The flowchart of our algorithm is given in Fig.1.

3. EXPERIMENT

3.1. Detailed Form of Raw Score and Normalization

Here we introduce the more detailed form of $P(t^o|o)$ that was used in our experiment. As introduced in Section 2.2, for an pronunciation observation o, we scored on phones (the initial and final of a syllable), therefore we had two scores $P((t^o)^{\mathtt{i}}|o)$ and $P((t^o)^{\mathtt{f}}|o)$. Furthermore, in addition to the reference phone, the decoded phone (which means the most probable phone decoded by our model) was also calculated. We denote the decoded initial and final phone as $(t^o)^{\mathtt{di}}$ and $(t^o)^{\mathtt{df}}$. As a result the detailed form of $P(t^o|o)$ is four dimensional, as shown in Table 1.

With $P(t^o|o)$ changing to these more detailed forms, all the normalization should also be changed to corresponding more detailed forms. Speaker normalization and syllable normalization are shown in Table 2. Finally, we added the tone normalization $P((t^o)^{\mathtt{v}})$, which was calculated similar to other normalization. These 13 scores were our 13-dimensional features that were used for classification.

	initial phone	final phone
reference score	$P((t^o)^{\mathtt{i}}\|o)$	$P((t^o)^{\mathtt{f}}\|o)$
decoded score	$P((t^o)^{\mathtt{di}}\|o)$	$P((t^o)^{\mathtt{df}}\|o)$

Table 1. four raw scores

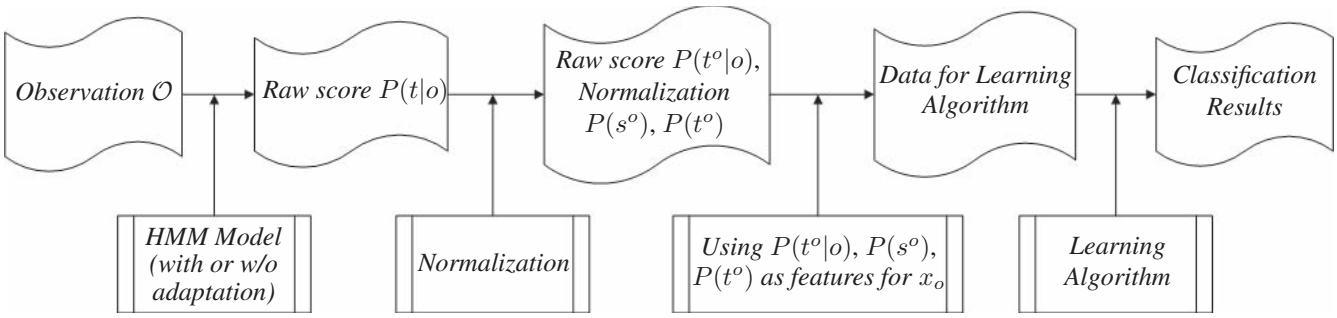


Fig. 1. Algorithm Overview

speaker normalization		
	initial phone	final phone
reference score	$P((s^o)^{\mathtt{i}})$	$P((s^o)^{\mathtt{f}})$
decoded score	$P((s^o)^{\mathtt{di}})$	$P((s^o)^{\mathtt{df}})$
syllable normalization		
	initial phone	final phone
reference score	$P((t^o)^{\mathtt{i}})$	$P((t^o)^{\mathtt{f}})$
decoded score	$P((t^o)^{\mathtt{di}})$	$P((t^o)^{\mathtt{df}})$

Table 2. Speaker and syllable normalization

From our experiment, we found that the normalization calculated on SLPP was not so reliable: the average FAR (defined in Section 3.5) was reduced by 26.7% on the data generated by model without adaptation and reduced by 3.2% in the data generated by model with adaptation, which is to say the normalization using SLPP score have great improvement on one data set and have small improvement on the other. Thus in this experiment, we used more reliable expert score in the train set instead. The general form (using syllable level denotation) of speaker normalization Eq.(4) and syllable normalization Eq.(5) using expert scoring is denoted using P_e, as shown in the following

$$P_e(s) = \frac{1}{|\mathcal{O}_{s,\mathtt{train}}|} \sum\nolimits_{o \in \mathcal{O}_{s,\mathtt{train}}} \mathtt{Expert}(o) \qquad (4)$$

$$P_e(t) = \frac{1}{|\mathcal{O}_{t,\mathtt{train}}|} \sum\nolimits_{o \in \mathcal{O}_{t,\mathtt{train}}} \mathtt{Expert}(o) \qquad (5)$$

where $\mathcal{O}_{s,\mathtt{train}}$ is the set of observations in training set spoken by s and $\mathcal{O}_{t,\mathtt{train}}$ is the set of observations whose reference syllable labels are t.

The detailed form of P_e is similar to P shown in Table 2, and the only difference is to change the normalization from using SLPP to using expert scoring in the training set.

3.2. Data Set

In our observation set $\mathcal{O}$, 100 native speakers had been tested and 9898 isolated syllables were pronounced in total. Proficiency level, such as correctly pronounced or mispronunciation, was given by two expert raters with national certificates. Among all the spoken syllables, 6285 were pronounced correctly and 3613 were mispronunciations.

Using our observation set $\mathcal{O}$ and their corresponding target class, we generated the data sets for our experiment as shown in Fig.1. To get access to the raw score $P(t^o|o)$, we used the Multi-space distribution Hidden Markov Model (MSD-HMM) [6] model which was trained on another 8000 syllables. Since there were mismatches between those 8000 syllables and our observation set $\mathcal{O}$, we also tried the MSD-HMM model with SMLLR adaptation. Using the MSD-HMM model and adapted MSD-HMM model, we generated two set of raw scores on $\mathcal{O}$ and then generate two data sets.

3.3. Evaluation

As mentioned in the Section 2, machine learning techniques are applied in learning the decision function. In our experiment, we used SVM classifier with RBF Kernel for our task[1].

The data set was randomly divided into 8 folders, and for each of the 8 folders, we took this folder for training and used the rest 7 folds for testing. The final result was the average value of the 8 test results.

We define the following two measures, false rejection rate (FRR) and false acceptance rate (FAR), as evaluation criteria.

$$\text{FRR} = \frac{\text{all mispronunciations that dected as correct ones}}{\text{all the mispronunciations}}$$

$$\text{FAR} = \frac{\text{all correct ones that dected as mispronunciations}}{\text{all the detect mispronunciations}}$$

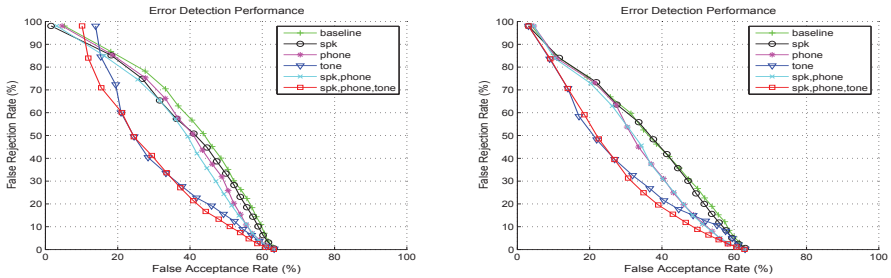
To fully reflect the changing performance FAR/FRR with different thresholds (in the SVM case, it is parallel hyperplanes), Detection-Error Tradeoff (DET) curve was used.

3.4. Experimental Result

We experimented our algorithm on two data sets and on different combinations of features:

- raw score with speaker normalization
- raw score with phone normalization
- raw score with tone normalization
- raw score with speaker, phone normalization
- raw score with speaker, phone and tone normalization

[1] We used the SVM Light [7] implementation



(a) Results on data set generated by non adapted model (b) Results on data set generated by adapted model

Fig. 2. DET curve of AMD on two data sets.

our baseline, using the method in [3], was also experimented.

The experimental result is shown in Fig. 2. Comparing the line "speaker, phone, tone" and the line "baseline", we observe a significant improvement when using all the normalization on both data sets. When taking the individual normalization and baseline for comparison, it can be concluded that each normalization would give rise to some extends of improvement and the most prominent normalization is the tone normalization. This is consistent with the statistics from the data set that tones have different mispronunciation rates, as shown in Table 3 (there are four Mandarin tones in all). It should be noticed that the tone 3 have the most high percentage of mispronunciation, which is because of the hardness of pronounce tone 3 correctly in Mandarin. This information can be taken as a sort of prior and experimental results support the idea of using this information.

tone	1	2	3	4
mispronunciation percentage	27%	29%	74%	29%

Table 3. Statistics on mispronunciation rate on each tone

Finally, when comparing the line with all normalization and the line with all except tone normalization, we also observe an improvement. This shows that the tone normalization could be combined with other nomalizations to obtain a better performance.

3.5. Result Summary

When given a FRR, we can find the corresponding FAR. Since the task of AMD in language learning requires to show a small number of possible mispronunciations, we focuses on the FAR when FRR is high. Therefor we uses the average FAR, which is the average number of FAR when FRR taking the value of 50%, 60%, 70%, 80% and 90%, to summary the result. The result in average FAR is shown in Table 4. We can see from the table that when using all the normalization, the performance is greatly enhanced, reducing the average FAR by 42.5% in the data set without adaptation and reducing the average FAR by 32.5% in the data set with adaptation.

4. CONCLUSION

In this paper, we propose to use normalization from syllable aspects to improve automatic mispronunciation detection. Machine learning method is utilized to make the final decision, not only avoids heuristics but also gets an extensible method for more normalization or features in the future. Experimental results support the effectiveness of the algorithm by reducing the average FAR by 42.5% in the data set without adaptation and reducing the average FAR by 32.5% in the data set with adaptation.

used normalization	w/o adaptation	with adaptation
baseline	30.7	22.7
speaker	28.1	22.4
phone	28.7	21.1
tone	19.9	16.5
speaker,phone	26.2	20.2
speaker,phone,tone	**17.6**	**15.3**

Table 4. Experimental result summary in average FAR(%)

5. ACKNOWLEDGEMENT

Thank iFlytek Speech Lab of University of Science and Technology of China for providing us the data used in this paper.

6. REFERENCES

[1] Silke M. Witt, *Use of Speech Recognition in Computer-Assisted Language Learning*, Ph.D. thesis, 1999.

[2] H. Franco, L. Neumeyer, and H. Bratt, "Automatic detection of phone-level mispronunciation for language," in *Learning, Proc. of Eurospeech*, 1999, pp. 851–854.

[3] F. Zhang, C. Huang, F.K. Soong, M. Chu, and R.H. Wang, "Automatic mispronunciation detection for mandarin," *Proc. of ICASSP*, pp. 5077–5080, 2008.

[4] C. Huang, F. Zhang, F.K. Soong, and M. Chu, "Mispronunciation detection for mandarin chinese," in *INTERSPEECH*, 2008.

[5] C. Cortes and V. Vapnik, "Support vector networks," in *Machine Learning*, 1995, pp. 273–297.

[6] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi, "Multi-space probability distribution hmm," *IEICE Trans. on Information and Systems*, 2002.

[7] T. Joachims, "Making large-scale support vector machine learning practical," pp. 169–184, 1999.