

# Improving Pronunciation Modeling for Non-Native Speech Recognition

Tien-Ping Tan, Laurent Besacier

LIG/ GETALP Laboratory, UMR CNRS 5217

UJF - BP 53, 38041 Grenoble Cedex 9, France

tien-ping.tan@imag.fr, laurent.besacier@imag.fr

## Abstract

In this paper, three different approaches to pronunciation modeling are investigated. Two existing pronunciation modeling approaches, namely the pronunciation dictionary and n-best rescoring approach are modified to work with little amount of non-native speech. We also propose a speaker clustering approach, which capable of grouping the speakers based on their pronunciation habits. Given some speech, the approach can also be used for pronunciation adaptation. This approach is called latent pronunciation analysis. The results show that conventional pronunciation dictionary perform slightly better than n-best list rescoring, while the latent pronunciation analysis has shown to be beneficial for speaker clustering, and it can produce nearly the same improvement as the pronunciation dictionary approach, without the need to know the origin of the speaker.

**Index Terms:** non-native ASR, decision trees, n-best list rescoring, latent phonemic analysis

## 1. Introduction

Automatic speech recognition (ASR) systems are increasing popular to be embedded in different kind of systems and applications. However, speech recognition system performance for recognizing non-native speech is still low compared to native speech.

Non-native ASR systems use the accent knowledge about the speaker to select the best models which correspond to the speaker to decode the speech. However, the challenge in training non-native models is the lacking of non-native data. Therefore, works in acoustic modeling make use of the speaker's native language or a little non-native speech for adaptation [1][2]. However, the native language of the speaker alone seems to be less effective for modeling pronunciation variants in a pronunciation dictionary for non-native speakers [3]. The availability of some non-native speech is a prerequisite for modeling pronunciation variants. Works in pronunciation modeling can generally be divided based on the component where the pronunciation variants are modeled and used [4]. The four possible places are pronunciation dictionary, acoustic model, language model and rescoring module.

In this paper, we modify two of the existing pronunciation modeling approaches, so that they can work with limited amount of non-native speech. In *Section 2* and *Section 3*, the non-native pronunciation modeling using pronunciation dictionary and rescoring module are presented respectively. We also propose a speaker clustering approach called latent pronunciation analysis which groups the speakers based on their pronunciation habits. It is also able to adapt the pronunciation dictionary given some non-native speech. It will be presented in *Section 4*. In *Section 5*, the approaches

will be experimented and compared. Conclusions are drawn in *Section 6*.

## 2. Pronunciation Dictionary: Decision Trees

There are few methods to derive the pronunciation variants, and one of it is through the use of decision trees [5]. The procedure used here is a general one, except that we derive the pronunciation variants by going through two passes, since we only have a little amount of non-native speech and the phoneme recognizer used for estimating the pronunciation variants produces around 50% recognition errors. Thus, it is important to have the hypotheses as accurate as possible. In the first pass, the purpose is to remove the unlikely variants. In the second pass, the pronunciation variants observed will be generalized according to the features of the pronunciation context using decision trees to predict unobserved variants. Figure 1 shows the steps for deriving the pronunciation variants.

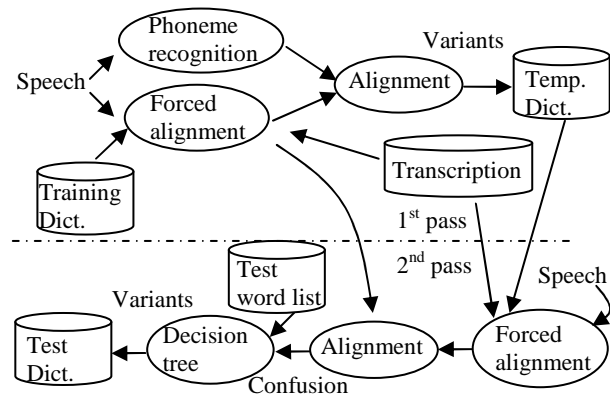


Figure 1. Generating pronunciation variants using decision tree

The objective of the first pass is to retain the more likely pronunciation variants and remove those less likely. The hypothesis phoneme strings from the phoneme recognizer are aligned against the reference phoneme time stamps using for example time alignment. A triphone confusion matrix is then created from the alignment. A low threshold is set to the triphone confusion matrix so that the pronunciation substitutions or variants which exist more than the threshold are selected. All possible pronunciation variants for each word will be generated and added into a temporary pronunciation dictionary. This will produce a pronunciation dictionary which is very much bigger than the original.

In the second pass, pronunciation variants will be generalized according to the context (left and right) features using decision trees. The first step is to re-estimate the hypotheses, this time by force aligning the speech using the new pronunciation dictionary created in the first pass. The

new hypothesis phoneme time stamps will then be aligned against the corresponding reference phoneme time stamps estimated earlier in the first pass.

The triphone confusions will be collected together, and a tree will be built for each base phoneme except silence. The left and right context phonemes need to be translated to a particular feature vector (Figure 2b). IPA based articulation features may be used for this purpose. The decision tree algorithm will then classify according to the features defined. Decision tree algorithms such as CART or C4.5 can be applied for grouping the confusion based on the defined features. A threshold is used to extract pronunciation variants from the decision trees, and the variants will subsequently be added into the pronunciation dictionary.

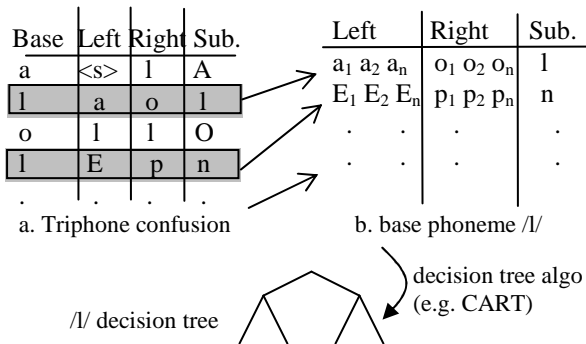


Figure 2. Sub-steps to create the decision trees in the decision tree process. a) Triphone confusions are obtained from the alignment of hypotheses and references. b) All triphones with the same base phoneme (e.g. /l/) are gathered and their left and right context phonemes are converted to feature vectors (e.g. IPA). c) Decision tree is built for each base phoneme

### 3. N-Best List Rescoring

Typically, pronunciation variants are added into the pronunciation dictionary, and the speech recognition system will select the best pronunciation variant during decoding. On the contrary, it is also possible to evaluate the pronunciation variants on the word lattice or n-best list using additional information from the phoneme recognizer [6]. Figure 3 shows the architecture of the n-best rescoring system. The approach applied here is similar to the one suggested in [6], except that we attempt to use a triphone model to represent the variants instead of a word model because of data sparsity.

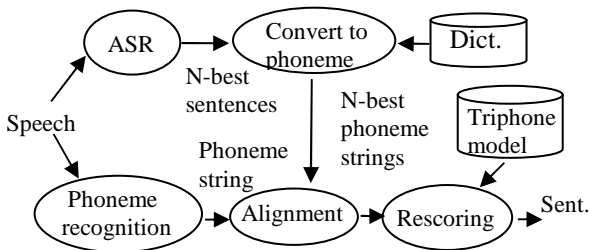


Figure 3. n-best rescoring architecture

Before pronunciation rescoring can be carried out, the triphone model has to be created first. The triphone model actually contains the triphone confusions. It is created with the same procedure as before in one pass. Some non-native speech is required for training the triphone model. The non-native speech is decoded by the phoneme recognizer, and the hypotheses produced are then aligned against the

corresponding reference phoneme strings. For smoothing the triphone confusion matrix, the triphone confusion values are interpolated with the corresponding monophone confusion probability. A floor value is used instead if both the confusion values are zero. Thus, contrary to the pronunciation dictionary approach, the pronunciation variants used here have a weight.

$$P'(sub|base, left, right) = w P(sub|base, left, right) + (1-w) P(sub|base), P(sub|base) > 0, 0 \leq w \leq 1 \quad (1)$$

$$P'(sub|base, left, right) = \text{floor probability}, P(sub|base) = 0 \quad (2)$$

During evaluation, the non-native speech is decoded by the speech recognition system and the phoneme recognizer. Note that the pronunciation dictionary used during decoding contains only the baseform representations or standard pronunciations of the words. The output of decoding of the speech recognition system contains the n-best sentences. These sentences will then be converted to the corresponding (reference) phonemes strings from the word strings using the pronunciation dictionary. The hypothesis phoneme string from the phoneme recognizer will then be aligned against each n-th "reference" phoneme string. The pronunciation score for each sentence from the n-best list will be calculated by considering the triphone confusions of the hypothesis phoneme string against the reference string using triphone model. The pronunciation score (log probability) for each sentence is then added to the total score which comprise of acoustic and language score. The sentence from the n-best list with the highest composite score will be selected. Figure 4 below shows an example of pronunciation rescoring using a 2-best list.

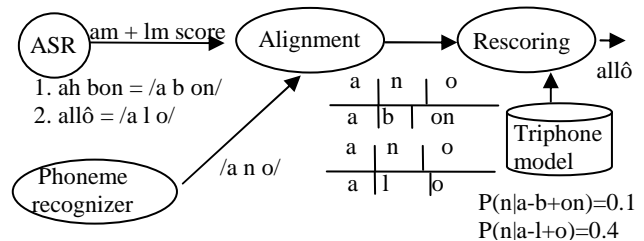


Figure 4. An example of a 2-best list rescoring. ASR gives 2 most probable sentences (ah bon and allô), which are converted to phoneme strings, and then aligned with the hypothesis (/a n o/) from phoneme recognizer. The pronunciation scores are calculated for each sentence and then added to the acoustic and language model score.

### 4. Latent Pronunciation Analysis

In this approach, we attempt to cluster non-native speakers into groups based on their pronunciation habits and subsequently use it for adaptation. An unsupervised speaker clustering method based on pronunciation habits is proposed here. The approach is inspired by eigenvoices approach [7] and also from idea given (but not experimented) in [8]. We call it 'latent pronunciation analysis' by analogy to 'latent semantic analysis' in natural language processing domain. The idea is to create a set of speaker dependent pronunciation confusion vectors, which will be used to derive the pronunciation confusion space. Then we can determine the position of the test speaker in the pronunciation space given some speech. Pronunciation variances can then be estimated for the test speaker.

For each speaker, a speaker dependent pronunciation confusion supervector will be created. The supervector is in fact the triphone confusion matrix which is laid out in a vector format. Table 1 below shows an example of K-supervectors created for K speakers (in actual case column vector is used instead of row vector). The supervectors can be derived from their corresponding speaker dependent pronunciation decision trees. The procedures to create the decision trees are the same as described in Section 2. From the test pronunciation dictionary, all pronunciation contexts are extracted. The possible substitutions for all the contexts are extracted from every speaker dependent decision tree by using a threshold and put in a pronunciation confusion vector. Since every speaker may have different set of substitution, a standard pronunciation confusion vector (supervector) must contain all the possible substitutions for every speaker and in the same order. Then for each context, the total probability of the substitution for each context will be normalized to 1.0.

Table 1. *K supervectors of pronunciation confusion. The context row shows triphones*

Contexts	b-a+n			d-ə+p		
	a	an	A	ə	DEL	
speaker 1	0.9	0.1	0.0	0.9	0.1	...
speaker 2	0.7	0.2	0.1	1.0	0.0	...
...						
speaker K	0.9	0.05	0.05	0.9	0.1	...

D

Principal component analysis (PCA) is then used to estimate the eigenvectors and eigenvalues. Eigenvectors  $E=e(1)...e(k)$  are derived from the covariance matrix of K supervectors V with dimension D, where k is less than K, and  $K \ll D$ . Notice that only the first k (principal component) of K eigenvectors are used. For clustering the speakers, the eigenvalues (weights) of the speaker is found using equation 3 and plotted to the k-space of the eigenspace. The speakers can then be separated to groups manually or automatically using some clustering approaches.

$$w = E^T \times V \quad (3)$$

For estimating pronunciation variants of an unknown speaker, some adaptation speech with transcription from the test speaker is required. This can be achieved by asking the speaker to read some sentences, which is common for configuring an ASR application. The speech will be forced aligned using the standard dictionary to get the reference phoneme string. To obtain the hypotheses, the same speech will also be forced aligned using another dictionary which contains the same variants or substitutions found in the supervectors. The corresponding phoneme strings will be aligned to create the confusion matrices. A supervector is constructed for each test speaker by finding its triphone and monophone confusion matrix, and subsequently interpolating them, and filling the supervector. The position/weights of the speaker on the eigenspace are estimated and then used to reconstruct the test speaker's supervector to obtain  $V'$  (eq. 4):

$$V' = E \times w \quad (4)$$

Finally, a threshold is used to extract the speaker specific pronunciation variants from supervector  $V'$ . The variants are then added into the test pronunciation dictionary.

## 5. Experiments

The experiments were carried out on non-native French and English speakers using CMU Sphinx ASR system. The non-native French corpus (NNF) [9] contains native Chinese and Vietnamese speakers. On the other hand, the non-native English speakers are of native German and Italian origin (ISLE) [10]. The baseline 16 Gaussians French and English context independent acoustic models were created using BREF120 corpus [11] and TIMIT [12] respectively. Table 2 describes how the corpora were used. Both non-native corpora are very challenging for an ASR task.

Table 2. *Summary of corpora used for training and testing*

Language	Task	Corpus	Speakers	Hours
French	Training	BREF120	120	100+
	Testing	NNF	15	1.5
English	Training	TIMIT	630	4
	Testing	ISLE	46	18

### 5.1. Pronunciation Dictionary: Decision Trees

This section presents the experiments carried out for testing pronunciation modeling using pronunciation dictionary. Table 3 below shows the test and modeling condition.

Table 3. *Number of speakers involved in the tests*

Corpus	Description	# Speaker
NNF	Test	10
	Modeling, Vietnamese	3
	Modeling, Chinese	2
ISLE	Test	34
	Modeling, German	6
	Modeling, Italian	6

Wagon utility with CART algorithm from the Festival speech synthesis system [13] was used to create the decision trees. The threshold for the confusion matrix was set to 0.15, which created on average about 10 variants per pronunciation in the first pass. IPA based articulation features were used to represent the phoneme for classification. The threshold for the decision trees was set to 0.3, producing about one extra variant per pronunciation in the second pass. Table 4 below shows the performance improvement.

Table 4. *WER of ASR by using pronunciation dictionary (decision tree) to model pronunciation variants*

Approach	Non-native French		Non-native English	
	Chinese	Vietnamese	German	Italian
Baseline	56.2	58.1	58.7	81.5
Decision Tree	54.9	56.3	56.1	76.9

### 5.2. N-Best List Rescoring

The same speakers used in section above (Table 3) were used for generating the triphone model and also for testing. To create the triphone model, the triphone confusion is interpolated with monophone confusion with the weight 0.8 and 0.2 respectively. The number of n-best sentences is set at 100. Table 5 below shows the improvement after rescoring. The results show that pronunciation dictionary approach is slightly better than n-best list rescoring approach for our experimental setup.

Table 5. WER of ASR by using n-best list rescoring to model pronunciation variants

Approach	Non-native French		Non-native English	
	Chinese	Vietnamese	German	Italian
Baseline	56.2	58.1	58.7	81.5
n-best resc.	55.5	56.8	56.9	79.6

### 5.3. Latent Pronunciation Analysis

In this test, only the non-native English were evaluated. Eighteen non-native English speakers, each with about 15 minutes of transcribed speech from native German and Italian were selected as the training set to create the pronunciation clusters. The remaining ten speakers were used as test set.

For creating the supervectors, speaker dependent decision trees were grown with Wagon utility. In total, 1464 triphone contexts were extracted from the test dictionary to create the barebones of the supervectors. The threshold for the decision trees was set at 0.5, and pronunciation variants for the triphone contexts were extracted from the speaker dependent decision trees to create supervectors with 4023 dimensions each. This means that for each context there were about 4 possible variants. A covariance matrix was calculated, and subsequently the eigenvectors were derived.

First, for evaluating the performance of the approach in clustering non-native speakers, we plotted the eigenvalues of the training speakers on eigenvalue 1 and eigenvalue 2 axes (see Figure 6). It shows that the two groups of non-native speakers can be separated by using the first eigenvector.

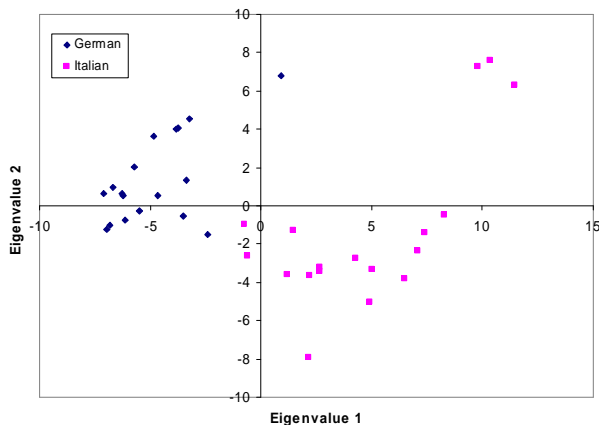


Figure 6. Plotting the speakers on the eigenspace

Next, for speaker dependent pronunciation adaptation, ten principal components were used. Two minutes of transcribed speech from each test speaker was used for creating the pronunciation confusion supervectors. The threshold was set at 0.3 to extract the pronunciation variants from the test supervector to be added into the pronunciation dictionary. Table 6 shows the results. Since only ten speakers were involved in the test, the baseline results in this experiment were different from the previous two tests. They show that the latent pronunciation analysis method is able to predict the pronunciation variants rather well with the reduction in WER, without knowing in advance the accent of the speaker. However, if the accent of the speaker is known in advance, it is better to use the accent specific dictionary, since it is slightly more effective.

We have randomly selected an Italian speaker and compared the average difference (omission and addition) in variants between that speaker and others. The average number of differences for that Italian speaker and other Italian

speakers is 209, while the average difference compared to German speakers is 411. This shows that the variants generated are speaker specific and more related to the accent.

Table 6. Comparing latent pronunciation analysis and decision tree approach for pronunciation modeling

Speaker	Baseline ( $\approx$ 1k words)	LPA		Decision Tree	
		200 variants	400 variants	200 variants	400 variants
Italian	75.5	72.6	72.2	73.0	71.1
German	59.0	57.6	57.2	56.3	56.0

## 6. Conclusions

We have presented three methods for modeling pronunciation variants. Adding variants created from decision trees into pronunciation dictionary reduce the WER more than the n-best list rescoring method. Furthermore, the n-best list rescoring requires more processing compared to the traditional dictionary approach. As for the latent pronunciation analysis, the method can be used in situation when we do not know in advance the accent of the speakers. But if the accent of the speaker is known, it is better to use the accent specific pronunciation dictionary. The latent pronunciation analysis is also potentially useful for automatic accent identification. We will investigate this in future works.

## 7. References

- [1] Uebler, U. and Boros M., "Recognition of Non-native German Speech with Multilingual Recognizers", Eurospeech'99, Budapest, 2: 911-913, 1999.
- [2] Tan T.-P and Besacier, L., "Modeling Context and Language Variation for Non-Native Speech Recognition", Interspeech'07, Antwerp, 1429-1432, 2007.
- [3] Goronzy, S., Kompe, R. and Rapp, S., "Generating Non-Native Pronunciation Variants for Lexicon Adaptation", ISCA'01, Sophia Antipolis, 143-146, 2001.
- [4] Strik, H. and Cucchiari, C., "Modeling Pronunciation Variation for ASR: A Survey of the Literature," Speech Communication, 29: 225-246, 1999.
- [5] Humpries, J.J. and Woodland, P., "Using Accent-Specific Pronunciation Modelling for Improved Large Vocabulary Continuous Speech Recognition", Eurospeech'97, Rhodes, 2367-2370, 1997.
- [6] Gruhn, R., Markov, K., and Nakamura, S., "A Statistical Lexicon for Non-Native Speech Recognition", ICSLP'04, 1497-1500, 2004.
- [7] Kuhn, R., Nguyen, P., Goldwasser, L., Niedzielski, N., Fincke, S. and Contolini, M., "Eigenvoices for Speaker Adaptation", ICSLP'98, Sydney, Australia, 1774-1777, 1998.
- [8] Goronzy, S., "Robust Adaptation to Non-Native Accents in Automatic Speech Recognition", Springer Verlag, Berlin, 2002.
- [9] Tan, T.-P and Besacier, L., "A French Non-Native Corpus for Automatic Speech Recognition", LREC'06, Genoa, 1610-1613, 2006.
- [10] Menzel, W., Atwell, E., Bonaventura, P., Herron, D., Howarth, P., Morton R. and Souter, C., "The ISLE Corpus of Non-Native Spoken English", LREC'00, Athens, 957-963, 2000.
- [11] Lamel, L.F., Gauvain, J.L. and Eskenazi, M., "BREF, a Large Vocabulary Spoken Corpus for French", Eurospeech'91, Genoa, 505-508, 1991.
- [12] Fisher, W.M., Doddington, G.R. and Goudie-Marshall, K.M., "The DARPA Speech Recognition Research Database: Specifications and Status", Proceedings of DARPA Workshop on Speech Recognition, 93-99, 1986.
- [13] Taylor, P.A., Black A. and Caley R., "The Architecture of the Festival Speech Synthesis System", The Third ESCA Workshop in Speech Synthesis, Jenolan Caves, Australia, 147-151, 1998.