Automatic Pronunciation Error Detection and Feedback Generation for CALL Applications

Renlong Ai

DFKI GmbH, Language Technology Lab
Alt-Moabit 91c, 10559, Berlin, Germany
renlong.ai@dfki.de

Abstract. This paper describes a new method of automatic error detection in Computer Assisted Language Learning (CAPT) system. The method combines linguistic knowledge and modern speech technology. Our HMM classifier trained from annotations of linguists is not only capable of classifying correct and wrong phonemes, but also can tell how wrong an error phoneme is pronounced. Phone errors in L2's speech, like phoneme substitution or distortion are detected with high accuracy, and at the same time, corrective feedback with multimedia support, which demonstrates how exactly error phonemes should be pronounced, is also generated.

Index Terms: L2 pronunciation errors, automatic error detection, feedback.

1 Introduction

In recent years, second language (L2) learning has become more and more popular to meet the need of communicating and integrating with a foreign community or society. However, learning a second language takes time and dedication, not only from learners, but also from teachers, hence both face-to-face and 7/24 personal online language learning are very expensive. A large and still growing number of computer assisted language learning (CALL) in the market has shown a clear trend: language learning is going to be web-based, interactive, multimedia and personalized, so that learners are flexible as to times and places for learning.

Modern technologies allow computer to beat human teacher in many aspects of language teaching like building up vocabulary and checking grammar, but not in training pronunciation, although many attempts have been made. Some industrial CALL applications are applying automatic speech recognition (ASR) on learners' speech and trying to infer existence of errors from the confidence value in recognition result. This yields results with low accuracy because no specific model is trained to deal with all possible errors, hence is far less effective than traditional classroom teaching. Researches have been made to investigate or enhance how pronunciation errors can be automatically detected, including building classifiers with Linear Discriminant Analysis or Decision Tree[1], or using Support Vector Machine(SVM) classifier based on applying transformation

on Mel Frequency Cepstral coefficients (MFCC) of learners' audio data[2][3]. These methods either involve complex training process or have conditions in usage, such as targeting at a special second language, hence haven't been used in current CAPT systems yet.

We develop our method by studying the most common use case in CAPT: A learner firstly listens to the gold standard version of a sentence read by a native speaker, then tries to imitate what he/she has heard, and at last is reported how good he/she has spoken, in a comprehensive way. This means the sentence and also the correct phoneme sequence are known to the system. The system should also know all possible errors that could happen in this sentence, if such information is previously given to or continuously learned by the system. In our approach, we firstly gather learners' data and have them annotated by linguists (chapter 2). After analyzing annotated data, we set up classifiers to distinguish not only correct and wrong phonemes, but also in which way a phoneme is false pronounced. Thus, by applying a model trained with gold standard plus learners' data, our HMM network produces fine classified results, which contain information for generating corrective feedback (chapter 3). In our experiment, we are able to detect pronunciation error at phoneme level with 98.4% recall and 94.6% precision (chapter 4). Since our method targets at the use case in CAPT, integrating it into existing CALL applications is discussed at the end.

2 Corpus and tools

2.1 Corpus

L1 background of learners can affect the pronunciation errors they make in second language learning [4]. In order to locate the errors precisely, separate models for different L1-L2 pairs should be trained. To test our method, we target on German learning British English.

1506 sentences are chosen from LinguaTV [1]'s database, read by both native british female and male. Among these, 96 sentences, which cover most of the common pronunciation errors, like pronouncing /z/ as /s/, are then read by 14 female German learners at different English levels. 10 sets are used for training the error detection model and 4 sets are used for testing.

2.2 Annotations tool

Pronunciation errors in speech data from learners are annotated. We extend MAT[5] as shown in Figure 1 and focus only on phoneme errors, which are:

- Deletion: a phoneme in a word is removed while pronouncing.
- Insertion: a phoneme is inserted before or after another phoneme.
- Distortion: a phoneme is pronounced is a distorted way.
- Substitution: a phoneme is replaced with another one by the learner.

[1] www.linguatv.com

In case of insertion and substitution, the phoneme, that the learner inserted or substituted with, is also annotated. Token '-' or '+' used to indicate if the phoneme written in 'spoken' column is inserted before or after the original one. By distortion, annotators are asked to mark how a phoneme is distorted. Following are summarized ways of distortion that annotators use:

- Tongue needs to be slightly further forward.
- Tongue needs to be slightly further back.
- Mouth needs to be slightly more closed.
- Mouth needs to be slightly more open.
- Lips need to be rounded.
- Lips need to be unrounded.
- Mouth needs to start slightly more open.
- Mouth needs to start slightly more closed.
- Tongue needs to start slightly further back.
- Tongue needs to start slightly further forward.
- Lips need to be rounded at the end.
- Vowel needs to be longer.
- Vowel needs to be longer and tongue needs to be slightly further back.

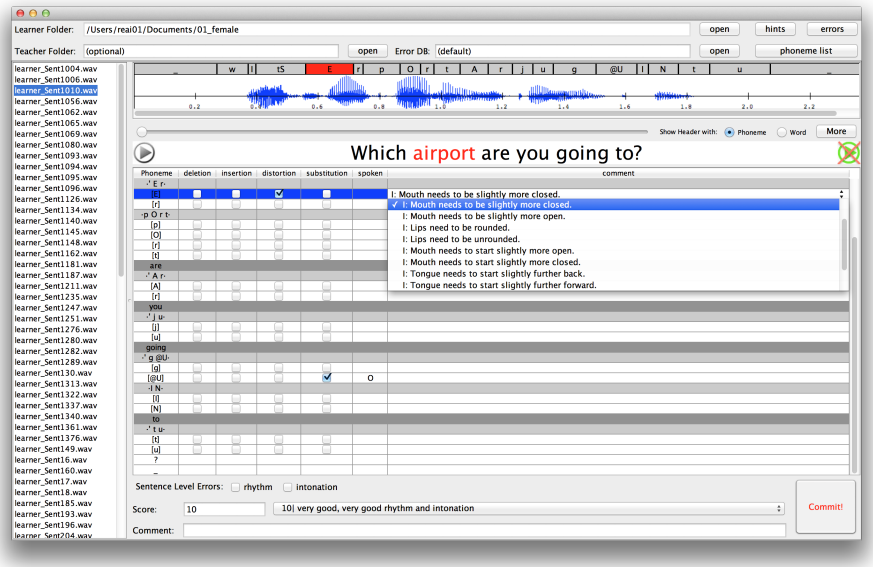

Fig. 1. With extended MAT, annotators can easily mark in which way an error phoneme is distorted.

3 Pronunciation Error Detection

The core of our method is to train a language model using HTK [2] for phoneme recognition. As a preparation of the training, errors found by annotators are classified. Then a model can be trained from correct and error phonemes. Before recognition, a grammar, which takes consideration of all possible errors that can appear in the given sentence, is generated. By passing the grammar and model, and also learner's audio to the recognizer, we can identify possible errors in learner's audio and also retrieve information for feedback from the recognizer's output.

3.1 Error Classification

After annotation, distorted phonemes are categorized by their ways of distortion and represented by new phonemes. For example, phoneme /ɑː/ in word 'are' can be distorted in two ways: either "Tongue needs to be slightly further forward." or "Tongue needs to start slightly further back.", so two new phonemes, A1 and A2, are created to represent wrongly pronounced /ɑː/. We use a database to keep track of all errors and integrate the database into MAT, so every newly annotated error is automatically classified and stored.

3.2 Language Model Training

The standard training for a phoneme recognition model using HTK is adapted to training a pronunciation error detection model, as shown in Figure 2. The audio data contains both gold standard data and learners' data. Gold standard data are handled in the same way as a normal training for phoneme recognition. As for learner's data, in order to keep the diphone and triphone information of error phonemes, we adjust the labels to make them represent the actually pronounced phoneme sequences. The output of MARY phonemizer is modified according to what type of error the corresponding audio file contains, which can be retrieved from the annotation.

- for deletion, the removed phoneme in learner's speech is also removed from the output of the phonemizer;
- for insertion, the inserted phoneme in speech is also inserted before or after the target phoneme, based on the annotation.
- for substitution, the annotated phoneme, which is actually spoken by the learner, replaces the original one.
- for distortion, the newly created distorted phoneme replaces the original one.

For example, the sentence "I'll be in London for the whole year." should have the right labels as (in MARY phoneme representations)

[2] http://htk.eng.cam.ac.uk/

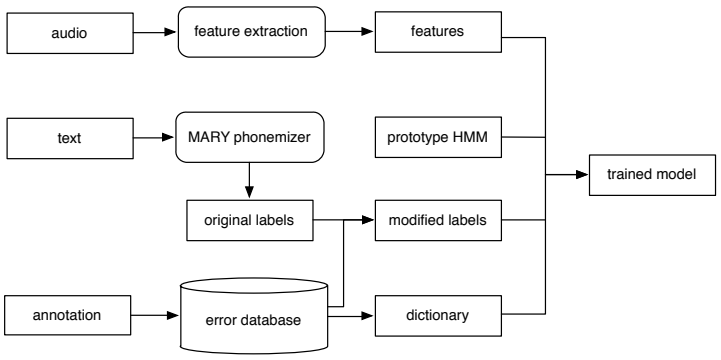

Fig. 2. Process to train a language model that detects pronunciation errors.

I'll	be	in	London	for	the	whole	year.
A l	b i	I n	l V n d	@ n	f O r	D @	h @U l	j I r

If a learner swallows /d/ in 'London', pronounces /ɔː/ in 'for' with backward tongue and replaces /ð/ with /z/ in 'the', the following labels are generated and used for training:

I'll	be	in	London	for	the	whole	year.
A l	b i	I n	l V n	@ n	f O2 r	z @	h @U l	j I r

During training, distorted phonemes are treated the same as normal ones and are also added to phone dictionary. Both gold standard and learners' data are send to iterations together so the trained model has information of inserted and removed phonemes, and is also able to deal with the differences between right phonemes and distorted ones.

3.3 Grammar Generation

To run phoneme recognition, HTK needs a grammar which defines the possible phoneme sequence of an input audio file. We generate grammars from the distribution of errors stored in database and texts that learners read. Taking the sentence "I'll be in London for the whole year" as example, firstly, the correct phoneme sequence is retrieved from MARY phonemizer and surrounds with 'sil', which represents the silence at the beginning and the end of the sentence. The grammar looks like

(sil A l b i I n l V n d @ n f O r D @ h @U l j I r sil)

Next, all possible errors made by learners in the same sentence are applied to the grammar, in this case, there could be errors in words 'London', 'for', 'the' and 'year', after this step the grammar is:

(sil A l b i I n l (V |A |O) n [d] @ n f (O |O2) r (D |z) @ h @U l j (I |I1) [(r |A)] sil)

At last, we observe errors in diphones and triphones and add them to the grammar too. These include errors in the same word in other sentences, and also errors with phonemes from other sentences that have the same pre and post phonemes as appeared in the target phoneme sequence. In this case the only other error found is in word 'be', so the final grammar is adapted to:

(sil A l b (i |i1) I n l (V |A |O) n d @ n f (O |O2) r (D |z) @ h @U l j (I |I1) [(r |A)] sil)

Unlike training language model, grammar is generated based on the incoming text in runtime of error detection, and compiled to a word network before HTK can use it in recognition.

3.4 Error Detection

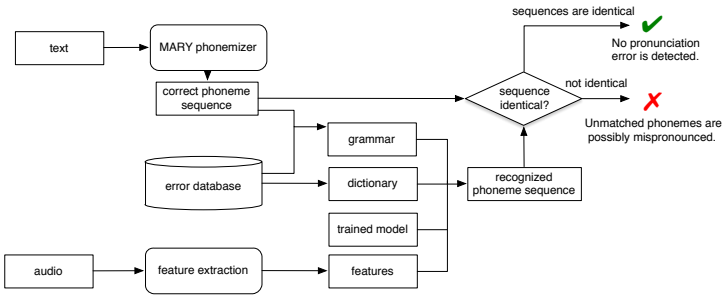

Fig. 3. Workflow of automatic error detection.

The process of automatic pronunciation error detection is illustrated in Figure 3. Phoneme recognition is performed using HTK with the trained model, adapted dictionary, generated grammar and extracted features. The recognition result is a phoneme sequence, which is then compared to the correct phoneme sequence generated from MARY phonemizer. If they are identical, no error is made in learner's pronunciation; if not, possible pronunciation errors can be traced from the difference between the two sequences in a simple way:

– if a distorted phoneme, e.g. I1, appears in the result, the original phoneme is distorted by the learner.
– if a phoneme from the correct sequence is missing, inserted or replaced in the result sequence, a deletion, insertion or substitution error can be inferred.

3.5 Feedback Generation

Finding out the errors is not the final destination. Intuitive feedback is needed so that learners know not only where the pronunciation errors are but also how to correct them. The advantage of our method is that these corrective information are retrieved at the same time as errors are detected. For example, if 'O2' is found in word 'for' in learner's pronunciation, we can show the annotation, from which this distorted phoneme is categorized, directly to the learner, and in this case it's "Tongue needs to be slightly further forward.". Or, if 'London' is recognized as 'l **O** n d @ n' instead of the correct 'l **V** n d @ n', we can tell the learner that he pronounces the first 'o' like /ɔː/ in 'often', but it should be like the /ʌ/ in 'cut'.

Simply displaying texts as instruction to learners is insufficient. Example of how exactly the error phoneme is pronounced, is needed. However, playing the gold standard version of the error word or sentence to learners is not enough either, because they may not be able to perceive the difference between the error phoneme and the correct one due to their L1 background [6]. In our evaluation system, we use a new way of feedback: the learner's own voice.

For each phoneme, we find out two words that are pronounced correctly from the voice data of a given learner. E.g. for /ʌ/ we have 'coming' and 'utter'. The words are chosen in the way that they have the target phoneme in different location and with different combination with other phonemes, and better represented by different letters. For /ʌ/, 'but' + 'cut' is not a good choice, neither is 'but' + 'utter'. Next, audio clips for each phoneme and its two example words are extracted. We also record some clips from native speaker. They are used for generating the final feedback. For example, if 'l **O** n @ n' is in the recognition result instead of 'l **V** n d @ n', the learner is presented with the a window as in figure 4. If she clicks on 'London' on the first row, the gold standard version of 'London' is played. If she clicks on the /ʌ/ on the second row, the following concatenated audio is played, where **/ʌ/** and **London** are extracted from gold standard voice, other underlined text are clips from the learner and the rest are pre-recorded audio prompts. We extract audio clips of phonemes and words by using the forced alignment information from trained model (for gold standard voice) and phoneme recognition result (for learners' voice). And the text is also displayed on screen.

"You pronounced **/ʌ/** in **London** like */ɔː/* in *'all'* and *'door'*. It should sound like */ʌ/* in *'coming'* and *'utter'*. Please try again."

Similarly, if /d/ and /ɔː/ are clicked, the following texts are displayed and corresponding audios are played:

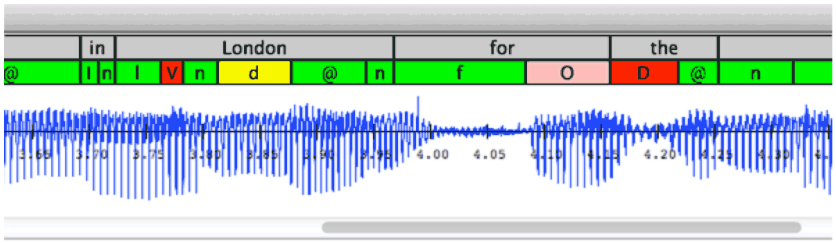

Fig. 4. A window showing learner's pronunciation error in our evaluation system. The background color of the phoneme shows what type of error the learner has made: green: no error, yellow: deletion, red: substitution, pink: distortion and purple: insertion (not presented in this example).

"You missed **/d/** in **'London'**, it should sound like */d/* in *'deny'* and *'good'*. Please try again."

"There is a little problem with the **/ɔː/** in **'for'**, it should sound like */ɔː/* in *'all'* and *'door'*. Tongue needs to be slightly further forward. Please try again."

In this way, learners are explained how to pronounce a phoneme correctly, in a way they are surely able to: by recalling how they used to sound it right in other words. Learners can perceive the difference between correct and wrong phonemes better, if they compare their own voices rather than comparing their voice with the gold standard [6].

4 Evaluation

We evaluate two contributions of our method: the performance of error detection and the effect of feedback. Precision and recall of our error detection method are evaluated objectively. We also apply progress evaluation to test if and to what level the automatic feedback can help language learners.

4.1 Precision and Recall

We run automatic error detection using the trained model on 4 sets of sentence, which have the same texts as the sentences used for training but read by 4 new learners. The results are then converted to extended MARY ALLOPHONES XML data with the same format as the annotations, so that they could be opened with the annotations tool for double-checking. Following are the results of comparing the generated data and the annotations, i.e. comparing errors detected by the system and errors found by annotators.

The result shows very high precision and recall for error types as deletion, insertion and substitution. In fact, the four deletion errors, which the system

	true positive	false positive	false negative	total	recall	precision
deletion	46	0	4	50	92%	100%
insertion	17	0	1	18	94.4%	100%
substitution	1264	14	2	1266	99.8%	98.9%
distortion	745	102	26	771	96.6%	88.0%
total	2072	116	33	2105	98.4%	94.6%

Table 1: A statistic of the error detection result. True positive: actually detected errors; false positive: correct pronounced phonemes detected as errors; false negative: errors not detected.

fails to detect, never appear in the training data, e.g. for the word 'central', the phoneme /r/ is removed by one of the testers. Substitutions are also detect very accurately. German tends to make the same substitution errors when speaking English, like replacing /ð/ in 'the' with /z/, and /z/ in 'was' with /s/. There are no new substitution errors in test data. Detecting distortions is not an easy task. In the 745 found errors, 114 of them are false categorized although they are successfully detected as distortion, e.g. the system returns "Tongue needs to be slightly further back." but the annotator thinks "Tongue needs to start slightly further back."

Despite a relative low accuracy at detecting distortion, we think the method is feasible for industrial CAPT applications, and we believe that the accuracy will raise if more training data is provided.

4.2 Feedback Evaluation

To use learners' own voice data as feedback, we are facing a dilemma: before a learner can pronounce a phoneme correctly, his/her correct voice data for this phoneme is not available. This problem becomes especially crucial when dealing with distortion because for some phoneme, beginners couldn't even pronounce them correctly only once, e.g. /ə/ at the end of 'number' or 'year'. In this case, we only display the annotator's hint as text, e.g. "Mouth needs to be slightly more open", to check if the learner manages to correct the pronunciation.

In our experiment, testers follow the scenario described in these steps:

1. Learner chooses a file with error and is presented with the window as in Figure 4. But at this time, clicking on the error phoneme only displays feedback as text.
2. Learner could click on the gray words on the first row to play the gold standard as many times as she wants. When she thinks she gets the information in the feedback, she press Record and speaks the whole sentence to the microphone again. Automatic error detection process runs again and presents the learner with a new window. In this window, clicking on error phonemes not only displays text but also play audio, as described in chapter 3.5.

3. If there are still errors shown in the new window, the learner can play the audio and check the text until she thinks she's able to correct the left errors, and then record again.
4. Another window should then show if the learner is able to correct all her errors.

	total	corrected after viewing text	corrected after listening to audio
deletion	20	19	20
insertion	6	6	6
substitution	641	430	608
distortion	338	104	125

Table 2: Statistics showing how feedback help learners correct their pronunciation errors.

Two of the four test learners took part in the experiment and the result is shown in table 2. By deletion and insertion, it's helpful enough to display the text information to make the learners realize what they missed or inserted. The only case that require a second time was a mistake: the learner did pronounce /s/ in 'months', but in the first time correction she focused on the /s/ and didn't pronounce the /θ/ before it clearly enough.

The case with substitution is interesting. We think there are three types of substitution. The first is like replacing /z/ with /s/ in 'Please' or /v/ with /f/ in 'of', the cause of which might be that learners forget the spelling rules. If prompt texts such as "like /z/ in 'zero' " or "like /v/ in 'very' " are given to learners, they understand instantly what the right pronunciations are. In learners' first attempt, most of this kind of substitution and those that were made by mistake were corrected. Example words play here an important role. Both learners have error with replacing /əʊ/ with /ɔ/ in 'most'. The learner with example word 'blow' and 'over' succeeded in correcting the error by only reading the textual feedback, while the other learner with 'hotel' and 'go' had to hear her own pronunciation of these two words to make successful correction. The second type is similar with the first, only that the original phoneme does not exist in learners' mother tongue, and is replaced with an existing one, e.g. /θ/ with /d/ in 'This'. The difficulty here is that a learner may not know how to pronounce it and makes no correct pronunciation on this phoneme, and hence no correct audio template can be generated. If this happens, our feedback won't work. The learner has to be taught systematically how to pronounce it. The third type is more in the way of a distortion, the error phonemes are distorted too much that they become another phoneme, e.g. replacing /æ/ with /e/ in 'exactly' or /ʌ/ with /a/ in 'number'. These errors are hard for learners to correct but after hearing their correct version of the same phoneme in other words, a large amount of them can be fixed.

The result shows that our feedback is not so good at helping to correct distortion errors as with other error types. Learners were able to correct around a third of the errors by changing their mouth, tongue or lips according the textual instruction. Playing audio wasn't helping much. We also notice that learners could distort a phoneme in her second attempt, although the same phoneme was correct in her first try. Our conclusion with distortion is that it's caused by learners' habit or accent, and might be hard to correct at once. In fact, distortion is still acceptable as long as the error phoneme is not distorted into a new phoneme, because learners may not even be able to perceive the difference between the correct phoneme and their distorted version, and will feel confused or discouraged if they are told that they pronounce wrongly every time they try to correct.

5 Conclusion and Discussion

This paper presents a method that automatically detects pronunciation error in learners' speech and generates corrective feedback. The methods targets at a very common use case in CAPT: Learners try to imitate a sentence after they listen to the gold standard, and wait for the system to tell them if they pronounce good enough. After training with annotated data, our system is able to detect phoneme errors like deletion, insertion, substitution and distortion with high accuracy, and provides feedback that could significantly help learners to correct their errors.

The model, which we trained with only voice data from 10 learners, already has good performance. In industrial usage, if learners allow their voice data to be collected, a more capable model can be expected.

Several aspects about feedback can be adjusted or improved in industrial systems:

- For learners that just start to use the system, there is no information about which phonemes they could pronounce error-free. In this case, words from learner's mother tongue could also be used as example words, if they contain the target phonemes. This could be an option for advanced learners too because they know how to pronounce their native words better.
- Extra video tutorial can be prepared for particular difficult phonemes such like how to pronounce /æ/ and /e/, /əʊ/ and /ɔ/, etc. When errors with these phonemes are detected, leaners can choose to watch corresponding video to learn the pronunciation systematically.
- It might make sense to distinguish beginners and advanced learners. Distortion errors are only displayed for advanced learners. Beginners should focus on those errors they could easily recognize and fix, like deletion or substitution. If they can't perceive the difference between the right phoneme and their distorted version, they won't be able to correct them and will be discouraged at last.
- Annotators should also provide hint of articulation to some substitution errors happening between similar phonemes such as replacing /æ/ with /e/.

In this case, the hint should be "Mouth needs to be slightly more open". Although the hint will not be used for categorizing distortion because no new phoneme is created, this information is helpful to the learners to correct such type of error.

Future work will seek to raise the precision of detecting distortion by studying the confidence value in HTK phoneme recognition result. The work of integrating this method into existing CALL application has already started.

6 Acknowledgement

This research was partially supported by the German Federal Ministry of Education and Research (BMBF) through the project Sprinter (contract 01IS12006A), Deependance (contract 01IW11003) and All Sides (contract 01IW14002).

References

1. K. Truong, A. Neri, C. Cucchiarini, and H. Strik, "Automatic pronunciation error detection: an acoustic-phonetic approach," in *InSTIL/ICALL Symposium 2004*, 2004.
2. S. Picard, G. Ananthakrishnan, P. Wik, O. Engwall, and S. Abdou, "Detection of specific mispronunciations using audiovisual features." in *AVSP*, 2010, pp. 7–2.
3. G. Ananthakrishnan, P. Wik, O. Engwall, and S. Abdou, "Using an ensemble of classifiers for mispronunciation feedback." in *SLaTE*, 2011, pp. 49–52.
4. J. Jenkins, "Global lntelligibility and local diversity: Possibility or poroolox?" *English in the world: Global rules, global roles*, p. 32, 2006.
5. R. Ai and M. Charfuelan, "Mat: a tool for l2 pronunciation errors annotation," in *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC-2014)*. European Language Resources Association, 2014.
6. J. E. Flege, "Second language speech learning: Theory, findings, and problems," *Speech Perception and Linguistic Experience: Theoretical and Methodological Issues*, pp. 233–273, 1995.