A System Demonstration of a Framework for Computer Assisted Pronunciation Training

Renlong Ai, Feiyu Xu
German Research Center for Artificial Intelligence, Language Technology Lab
Alt-Moabit 91c, 10559 Berlin, Germany
{renlong.ai,feiyu}@dfki.de

Abstract

In this paper, we demonstrate a system implementation of a framework for computer assisted pronunciation training for second language learner (L2). This framework supports an iterative improvement of the automatic pronunciation error recognition and classification by allowing integration of annotated error data. The annotated error data is acquired via an annotation tool for linguists. This paper will give a detailed description of the annotation tool and explains the error types. Furthermore, it will present the automatic error recognition method and the methods for automatic visual and audio feedback. This system demonstrates a novel approach to interactive and individualized learning for pronunciation training.

1 Introduction

Second language (L2) acquisition is a much greater challenge for human cognition than first language (L1) acquisition during the critical period. Especially by older children and adults, L2 is usually not acquired solely through imitation in daily communication but by dedicated language education. The teaching of second and further languages normally combines transmission of knowledge with practicing of skills. One major hurdle is the interference of L1 proficiency with L2.

Modern language learning courses are no longer exclusively based on books or face-to-face lectures. A clear trend is web-based, interactive, multimedia and personalized learning. Learners want to be flexible as to times and places for learning: home, trains, vacation, etc. Both face-to-face teaching and 7/24 personal online language learning services are very expensive. There is a growing economic pressure to employ computer-assisted methods for improving language learning in quality, efficiency and scalability. The current philosophy of Computer Assisted Language Learning (CALL) puts a strong emphasis on learner-centered materials that enable learners to work on their own. CALL tries to support two important features in language learning: interactive learning and individualized learning.

Natural language processing (NLP) technologies play a growing important role for CALL (Hubbard, 2009). They can help to identify errors in student input and provide feedback so that the learners can be aware of them (Heift, 2013; Nagata, 1993). Furthermore, they can help to build models of the achieved proficiency of the learners and provide materials and tasks appropriate.

In this paper, we demonstrate an implementation of a framework of computer assisted pronunciation training for L2. The current version, which is a further development of the Sprinter system (Ai et al., 2014), automatically recognises and classifies the pronunciation errors of learners and provides visual and audio feedback with respect to the error types. The framework contains two subsystems: 1) the annotation tool which provides linguists an easy environment for annotating pronunciation errors in learners' speech data; 2) the speech verification tool which identifies learners' prosody and pronunciation errors and helps to correct them.

The remainder of the paper is as follows: Section 2 describes the system architecture; Section 3 and 4 explain how each subsystem works; Section 5 evaluates the speech verification tool; A brief conclusion and future plan is mentioned at last in Section 6.

2 System Description

Figure 1 depicts the framework and the interaction between the annotation tool and the speech verification tool. As presented below, the speech ver-

ification tool is initialised with a language model trained with audio data from 10 learners. In case that learners' audio data can be collected and annotated in the online system, the error database can be updated on the fly and the language model can be updated dynamically, hence speech verification will improve itself iteratively until enough pronunciation errors are gathered and no annotation will be needed. Comparing to existing methods on

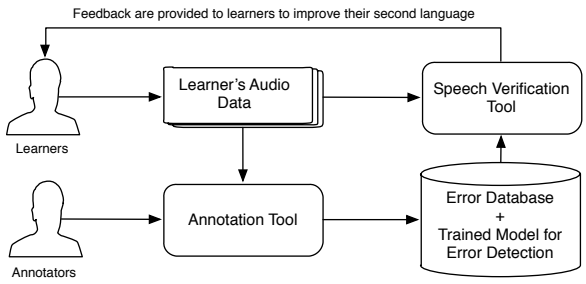

Figure 1: Overall Framework Architecture

pronunciation error detection, including building classifiers with Linear Discriminant Analysis or Decision Tree (Truong et al., 2004), or using Support Vector Machine (SVM) classifier based on applying transformation on Mel Frequency Cepstral coefficients (MFCC) of learners' audio data (Picard et al., 2010; Ananthakrishnan et al., 2011), our HMM classifier has the advantage that it is trained from finely annotated L2 speech error data, hence can recognise error types that are specifically defined in the annotations. Moreover, the results not only show the differences between gold standard and learner data, but also contain information of corrective feedback. Comparing to a general Goodness of Pronunciation (GOP) score or simply showing differences in waveform, our feedback pinpoints different types of error down to phoneme level and show learners how to pronounce them correctly via various means. The same annotating-training-verifying process can be adapted to all languages as far as our open source components support them, thus, with almost no extra efforts. Since people with the same L1 background tend to make the same pronunciation errors while learning a second language (Witt, 2012), we choose a specific L1-L2 pair in our experiment, namely, Germans who learn English.

3 Annotation Tool

To provide annotators an easy and efficient interface, we have further developed MAT (Ai and

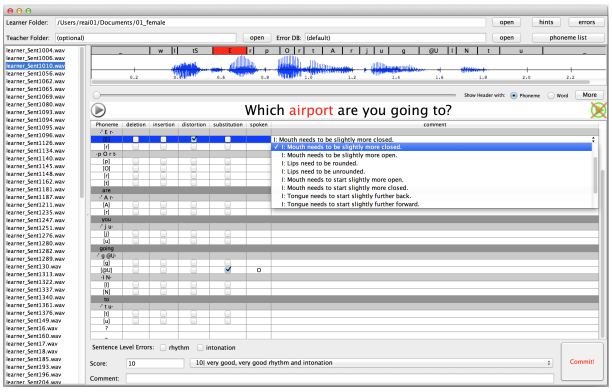

Figure 2: Screenshot of the Annotation Tool

Charfuelan, 2014), with which pronunciation errors can be annotated via simple mouse clicks and single phoneme inputs from keyboard. During the annotation, errors are automatically stored in an error database, which then updates the speech verification tool. We also add checking mechanisms to validate the annotations and ask annotators to deal with conflicts.

3.1 Target Pronunciation Errors

Four types of phoneme-level pronunciation errors can be annotated in the ways as shown in Figure 2. They are:

- Deletion: A phoneme is removed from learner's utterance, e.g. to omit /g/ in "England". Annotators only need to check the checkbox in column *deletion*.
- Insertion: A phoneme is inserted after another one, e.g. learner tries to pronounce a silent letter, like 'b' in "dumb". In this case the annotator should check *insertion* and type the inserted phoneme in column *spoken*.
- Substitution: A phoneme is replaced by another one, e.g. replacing /z/ with /s/ in "was". The annotator should check *substitution* and type the actually pronounced phoneme in *spoken*.
- Distortion: A phoneme is not pronounced fully correct, yet is not so wrong that it becomes another phoneme, e.g. pronouncing /u/ with tongue slightly backward. In this case the annotator should also select a hint from the drop down list in *hint* column to indicate how the error phoneme can be corrected.

3.2 Annotations

As depicted in Figure 2, annotations can be conducted easily and directly with this tool. If an annotator finds a pronunciation error, he/she can simply check the checkbox at line: phoneme and column: error type from the table, and provide extra information of the actually spoken phoneme and also hints of how to pronounce correctly. To make the annotation more convenient and efficient, several functions are built in the tool:

- Phoneme sequences are generated via MARY phonemizer. Phonemes are listed in the first column of the table and also in the header of the waveform panel. Clicking on a phoneme in the header will highlight the row of the corresponding phoneme in the table, and vice versa. The word, which the clicked phoneme belongs to, is also highlighted in the middle panel where the sentence is shown. Syllables in words are also retrieved from text analysis and displayed in the first column.
- Phoneme boundaries are recognized via forced alignment performed with HTK. Annotators can play any single phoneme, syllable or word by double-clicking them in the first column, or play any part of the sentence by choosing a range in the waveform panel with mouse and hit space on keyboard. In this way annotators can focus on error phonemes without bothering to listen to the whole sentence.
- There is already a list of hints on articulation provided. If a correction cannot be found in the list, annotators can click *hints* button and add the correction to the list.

After the annotation for a single audio file is done, the annotator clicks *Commit* to submit the annotations. The *Commit* function will check the annotation additionally:

- If *insertion* is marked, the inserted phoneme should be also written in *spoken*.
- If *substitution* is marked, the phoneme, which the original phoneme is substituted to, should also be written in *spoken*.
- If *distortion* is marked, a comment should be provided by annotators to indicate in which way this phoneme is distorted.
- Only one kind of error can be annotated per phoneme.

If the annotations are valid, the pronunciation errors and their diphone or triphone informations

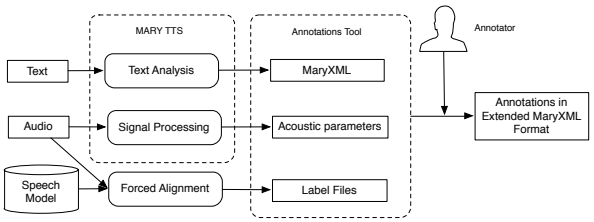

Figure 3: Components in annotation tool

are stored in an error database, which is later used in training the language model for error detection and also in feedback generation.

4 Speech Verification Tool

The speech verification tool identifies the pronunciation errors in learners' speech and provides feedback on correction, and also analyze the differences in pitch and duration between gold standard and learners' speech and display them in comprehensive ways. The goal is to help learners speak second language error-free and with rhythm and intonation like native speakers.

4.1 Pronunciation Error Detection

Pronunciation error detection is realized by performing phoneme recognition with HTK and comparing the correct and recognized phoneme sequence. Errors can be retrieved from the differences between the sequences. The sentence read by learner is processed by MARY phonemizer to generate the correct phoneme sequence. Possible pronunciation errors are then fetched from the error database based on the phonemes and their diphone and triphone information in the sequences. A grammar for phoneme recognition is then composed from the errors and the phoneme sequence.

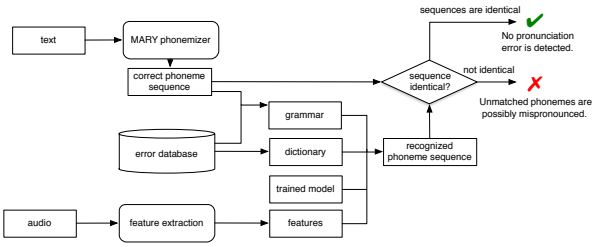

Figure 4: Workflow of Pronunciation Error Detection

Errors are handled differently as they appear in the grammar:

- Deletion means a phoneme can be optional in the grammar.

- Insertion means an extra phoneme, i.e. the inserted one, can be optional in the grammar.
- Substitution means multiple phonemes can appear at this position: the correct one and the substituted ones.
- Distortion also means multiple phonemes can appear at the same position: the correct and their distorted alias, which are represented with phoneme plus number. For example, the phoneme /ɑ/ can be distorted in two ways: either tongue is placed backward or tongue starts too forward. Since /ɑ/ is represented as /A/ in MARY, its two distorted versions are /A1/ and /A2/.

If a given sentence with its MARY phoneme sequence is:

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

A grammar with following content

(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)

will be generated, because the following pronunciation errors have been learned:

1. /i/ in *be* can be distorted.
2. /ʌ/ in *London* can be substituted with /ɑ/ or /ɔ/.
3. /d/ in *London* can be removed by learners while pronouncing.
4. /ɔː/ in *for* can be distorted.
5. /ð/ in *the* can be replaced with /z/.
6. /iː/ in *year* can be distorted.
7. /ə/ in *year* can be substituted with /ɑ/, and can also be deleted by learners.

If the predefined errors appear in learners' speech, they will be identified by comparing the different phonemes in the correct phoneme sequence generated with MARY phonemizer and the recognized sequence with HTK. Moreover, corrective feedback can also be created by fetching the hint, which annotators provide, from the error database. E.g. if /A1/ instead of /A/ is recognized for word *are*, we can provide the hint for this distorted /ɑ/: *Tongue needs to be slightly further forward*.

4.2 Prosody Differences

Prosody differences between gold standard and learners' speech are calculated and shown to learners, not only visually, but also with audio feedback. The prosody from gold standard is applied to learners' speech, so learners can hear their own voice with native prosody, in this way it might be easier for them to mimic the gold standard prosody (Flege, 1995).

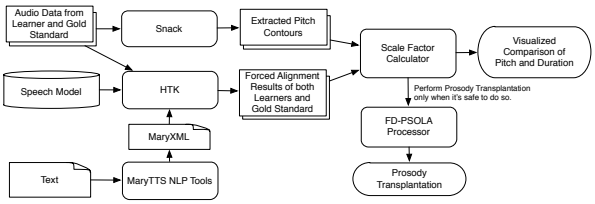

Figure 5: Workflow of Prosody Comparison and Transplantation

Duration differences can be calculated directly from the results of forced alignment. To gain more accurate alignment, the language model is trained with gold standard data and a selected set of learners' data. Before displaying the differences to learners as in Figure 6, the durations in learners' speech are normalized so that only words with large duration differences are shown.

Pitch differences are calculated with Snack Sound Toolkit[1]. Dynamic time warping is applied to the gold standard against speech data from learners, so that differences in pitch can be shown per phoneme synchronously. Due to the fact that each person has a distinct baseline in pitch, differences in pitch value are not considered as differences, we show only differences in pitch variation.

We use FD-PSOLA combined with DTW (Latsch and Netto, 2011) to perform prosody transplantation and generate audio feedback that learners can perceive. Despite the high performance of the method, there are still cases that prosody transplantation yields faulty results, e.g. the synthesized speech sounds very artificial or there are significant gaps between words or even phonemes. Therefore a scale factor calculator runs before the transplantation to determine if it makes sense to transplant the prosody, and will only do it and provide it as feedback when the scale factors do not exceed the threshold.

4.3 User Interface

Figure 6 shows a screenshot of the speech verification tool. Learners can choose sentences they want to practice from the list in the left, and

[1] http://www.speech.kth.se/snack/

click *Record* and speak to the microphone. After the audio is recorded, they can click *Check* to display the verification results. Pronunciation errors are marked in the first panel with colors other than green. Red, pink, yellow and purple are used for substitution, distortion, deletion and insertion. The second panel shows the differences in pitches, significant and medium differences are rendered with red and yellow. The panel below shows the differences in durations. Hints of correction or improvement are shown as text if learners click on the colored phonemes or words. If prosody transplantation is feasible, the button *Transplanted* is enabled and will play audio with learners' voice and transplanted gold standard prosody upon click.

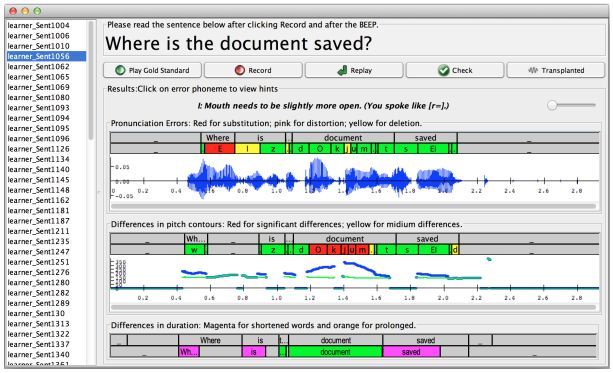

Figure 6: Screenshot of the Speech Verification Tool

5 Evaluation

Experiments are conducted both objectively and subjectively, in order to evaluate the performance of error detection and also how our feedback can help learners improve their second language skills

5.1 Objective Evaluation

To train the language model for pronunciation error detection, we let 1506 sentences read by native Britons. Given these sentences, 96 sentences are carefully selected, which cover almost all typical pronunciation errors made by Germans who learn English, and have been read by 10 Germans at different English skill levels. To evaluate our error detection system, we let 4 additional German people read these 96 sentences. The recordings were sent to both annotators and the error detection system. The results from error detection are transformed to MARYXML format and compared with the annotations.

	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.

The results show a very high precision in recognizing *deletion* and *insertion*. The recalls are also very good considering that there are new deletion phenomena in testers' speech that are not involved in old training data. Although a large amount of substitution errors appear in the test data, they have been detected accurately. This proves that training a language model considering specific L1 background is important for correct error recognition. For example in our case, most typical substitution errors made by Germans are well trained, like pronouncing /ð/ like /z/ in *the*, or /z/ like /s/ in *was*.

Detecting distortion errors seems a more difficult task for the system. Although a good recall is achieved, the precision is not satisfying. In CALL, this is perhaps not helpful because learners will be discouraged if they make correct pronunciation but are told wrong by the system. More speech data is required for training the model. Our annotators are experienced linguists but they may still holds different criterion on judging distortion. Having more linguists working on the annotation should also help to improve the accuracy of error detection.

5.2 Subjective Evaluation

To evaluate whether and how our feedback can help learners improve their second language, we designed a progressive experiment. 4 learners are chosen to read 30 sentences from the list. They can listen to gold standard as many times as they need, and record the speech when they are ready. If there are pronunciation errors or prosody differences, they can view hints or listen to gold standard, and then try again. If there are still prosody differences, they can then hear the transplanted speech, as many times as they need, and then try to record again. Insertions and deletions are easily corrected by learners once they have been pointed out. We examined the substitution errors that were not corrected, and found most of them are be-

	insertion	deletion	substitution	distortion	total
detected errors	13	4	467	220	704
corrected errors	13	4	429	116	562

Table 2: Amount of detected and corrected pronunciation errors from test speech data.

tween phoneme /æ/ and /e/, and also /əʊ/ and /ɔ/. The differences between the phoneme pairs were not easily perceived by learners, and they need to be taught systematically how to pronounce these phonemes. It was difficult for learners to correct distortions. Besides providing tutorial on how to pronounce error phonemes correctly, our system also needs to be modified so that it doesn't handle distortion so strictly.

	differences in count of phonemes (pitch) or words (duration)	correct after viewing hints	corrected after listening to prosody transplantation
pitch	474	343	438
duration	205	135	177

Table 3: Amount of detected and corrected prosody differences from test speech data.

The results on prosody differences show that most of these differences can be perceived and adjusted by learners, if they are given enough information. Almost all the remaining differences are from long sentences. Learners couldn't pay attention to all differences in one attempt. We believe that learners should be able to read all sentences in native intonation and rhythm if they try another several times.

6 Conclusion

In this paper we presents a framework for computer assisted pronunciation training. Its annotation tool substantially simplifies the acquisition of speech error data, while its speech verification tool automatically detects pronunciation errors and prosody differences in learners' speech and provide corrective feedback. Objective and subjective evaluations show that the system not only performs accurately in error detection but also helps learners realize and correct their errors.

In the future, we attempt to integrate more feedback content such as video tutorial or articulation animation for teaching pronunciation.

7 Acknowledgment

This research was partially supported by the German Federal Ministry of Education and Research (BMBF) through the projects Deependance (01IW11003) and ALL SIDES (01IW14002).

References

Renlong Ai and Marcela Charfuelan. 2014. 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.

Renlong Ai, Marcela Charfuelan, Walter Kasper, Tina Klwer, Hans Uszkoreit, Feiyu Xu, Sandra Gasber, and Philip Gienandt. 2014. Sprinter: Language technologies for interactive and multimedia language learning. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC-2014)*. European Language Resources Association.

Gopal Ananthakrishnan, Preben Wik, Olov Engwall, and Sherif Abdou. 2011. Using an ensemble of classifiers for mispronunciation feedback. In *SLaTE*, pages 49–52.

James E Flege. 1995. Second language speech learning: Theory, findings, and problems. *Speech Perception and Linguistic Experience: Theoretical and Methodological Issues*, pages 233–273.

Trude Heift. 2013. Learner control and error correction in icall: Browsers, peekers, and adamants. *Calico Journal*, 19(2):295–313.

Philip Hubbard. 2009. *Computer Assisted Language Learning: Critical Concepts in Linguistics*, volume I-IV. London & New York: Routledge.

V. L. Latsch and S. L. Netto. 2011. Pitch-synchronous time alignment of speech signals for prosody transplantation. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, Rio de Janeiro, Brazil.

Noriko Nagata. 1993. Intelligent computer feedback for second language instruction. *The Modern Language Journal*, 77(3):330–339.

Sébastien Picard, Gopal Ananthakrishnan, Preben Wik, Olov Engwall, and Sherif Abdou. 2010. Detection of specific mispronunciations using audiovisual features. In *AVSP*, pages 7–2.

Khiet Truong, Ambra Neri, Catia Cucchiarini, and Helmer Strik. 2004. Automatic pronunciation error detection: an acoustic-phonetic approach. In *InSTIL/ICALL Symposium 2004*.

S. M. Witt. 2012. Automatic error detection in pronunciation training: where we are and where we need to go. In *International Symposium on Automatic Detection of Errors in Pronunciation Training*, Stockholm, Sweden.