INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Pronunciation Error Detection for New Language Learners

Sean Robertson, Cosmin Munteanu, Gerald Penn

University of Toronto, Toronto, Canada
sdrobert@cs.toronto.edu, cosmin.munteanu@utoronto.ca, gpenn@cs.toronto.edu

Abstract

Existing pronunciation error detection research assumes that second language learners' speech is advanced enough that its segments are generally well articulated. However, learners just beginning their studies, especially when those studies are organized according to western, dialogue-driven pedagogies, are unlikely to abide by those assumptions. This paper presents an evaluation of pronunciation error detectors on the utterances of second language learners just beginning their studies. A corpus of nonnative speech data is collected through an experimental application teaching beginner French. Word-level binary labels are acquired through successive pairwise comparisons made by language experts with years of experience teaching. Six error detectors are trained to classify these data: a classifier inspired by phonetic distance algorithms; the Goodness of Pronunciation classifier [1]; and four GMM-based discriminative classifiers modelled after [2]. Three partitioning strategies for 4-fold cross-validation are tested: one based on corpus distribution, another leaving speakers out, and another leaving annotators out. The best error detector, a log-likelihood ratio of native versus nonnative GMMs, achieved detector-annotator agreement of up to $\kappa = .41$, near the expected between-annotator agreement.

Index Terms: pronunciation error detection, Computer Assisted Pronunciation Training (CAPT), Goodness of Pronunciation (GOP), Gaussian Mixture Models

1. Introduction

Computer-Assisted Pronunciation Training (CAPT) is becoming increasingly relevant to second language learning since it has the potential to complement materials that are used in classrooms or individual instruction. Pronunciation error detection, also known as mispronunciation detection, is a subfield of CAPT which focuses on finding nonnative pronunciations of segments of speech. It is distinguished from pronunciation scoring [3] in that it focuses on finding pronunciation errors rather than assessing their severity. Pronunciation error detection can be considered the groundwork that allows other modules of an application to provide formative feedback to language learners.

Existing pronunciation error detection research focused on providing detailed feedback to learners who have either had significant exposure to the target language or for tasks where learners are focused on articulation, such as reading. The famous Goodness of Pronunciation score detected phone-level errors on read phrases [1]. Later adaptations and improvements to this algorithm, with techniques such as Linear Discriminant Analysis [4] and Deep Neural Networks [5], were similarly analyzed on read phrases. When spontaneous speech has been evaluated, it has been with learners competent enough to generate responses to questions in standardized tests [6, 7]. Some work in nonnative prosody assessment has allowed for less skilled learners to practice in a semi-spontaneous environment [8], but evaluation has not been at the segmental level. Many other examples of pronunciation error detectors can be found in [9] and in the proceedings of modern language technology workshops such as SLaTE. This sort of research is valuable, but it assumes learners have some aptitude for the target language and have alloted time in their studies specifically for pronunciation.

The application of pronunciation error detectors to new learners of a second language has yet to be explored. There are a number of factors distinct to beginner learners: they must be assessed more leniently, since their utterances are likely always going to be mispronounced in some way; their segmental errors often span entire syllables or words, meaning accurately transcribing phone-level boundaries (cf. [10]) would be very difficult and isolating one problematic phone impossible (plus it assumes the learner is capable of understanding this granularity of feedback); and beginning learners are arguably more focused on other goals, such as word acquisition. Furthermore, popular communicative pedagogies [11] expect learners to begin dialogue practice right away, deferring reading comprehension to a later date. In short, beginning second language learners in realistic learning scenarios can be expected to produce lower quality utterances than are accounted for in current literature. For the same reason, beginning learners can receive the most benefits from pronunciation error detectors.

2. Contribution

To our knowledge, the research presented here is the first to adapt existing approaches of pronunciation error detection to beginning learner speech data. This is an emerging field offered by increasingly powerful mobile devices which can be used as complementary tools in educational settings. No research as of yet has been conducted to investigate how state-of-the-art pronunication error detectors perform on speech data collected from adult beginner learners of a second language in a realistic educational task where learners interact with each other in pairs, facilitated by a mobile-based tutoring app. Our research addresses this gap and presents the results of an evaluation of several error detectors, as described in section 3. Data collection is inspired by a communicative language learning curriculum wherein participants must recall appropriate phrases in a dialogue and sometimes generate new phrases, albeit highly restricted ones. Binary labels of more and less native word segments (herein referred to as native and nonnative for the sake of simplicity) are teased from expert French teachers using a series of pairwise comparisons. Word-level labels, if reliably predicted, could be incorporated into an implicit feedback mechanism appropriate to the pedagogy. Experimental setup, including the corpus description, can be found in section 4.1. The results of the experiment are discussed in section 5, including a discussion of potential pitfalls.

http://dx.doi.org/10.21437/Interspeech.2016-539

3. Methods

Six pronunciation error detectors, described below, are evaluated in this experiment. It is commonplace in pronunciation research to perform some form of threshold tuning per segment on the test set [12, 13, 1, 2, 14]. However, it has been shown that this practice can drastically improve the performance of some error detectors over untuned or minimally tuned varieties [15]. In addition, since these results may not align with real-world performance, "untuned" methods of choosing thresholds are explored. Both tuned thresholds (tuned to the Equal Error Rate (EER), matching the number of false and true positives) and untuned varieties are explored in this experiment.

Some of these models require Automatic Speech Recognition (ASR). In these cases, French acoustic models and phonetic dictionaries are from l'Université du Maine [16] and decoding is performed with Pocketsphinx [17]. A repository build of Pocketsphinx will normalize per-frame scores according to that frame's most likely phone to prevent underflow. For this experiment, Pocketsphinx has been patched to remove this safeguard.

3.1. Phonetic distance classifier

Often pronunciation error detectors build a neighbourhood of incorrect phonetic dictionary entries that are "close" to a canonical pronunciation of some word. If an alternate pronunciation is recognized via ASR, that word is considered mispronounced. An edit distance algorithm can be run to determine which specific phones or phonemes were mispronounced. Whether error detectors use rules, confusion matrices, or some other tool to develop alternate pronunciations [18, 13], they all serve to cluster together similar but incorrect pronunciations of words.

A Phonetic Distance Classifier (PDC), inspired by such error detectors, is tested in this experiment. It builds a ranked list of alternate pronunciations for every canonical one. This list has some fixed size N. To populate the list, an iterative algorithm generates all n size chains of phones from a list of m $(n-1)$-phone prefixes and sorts them in descending similarity from any canonical pronunciation of the word. The first m entries from the newly populated n-size list are then used to build chains of size $n+1$. This continues until chains reach the size of 2 phones past the maximum size of alternate canonical pronunciations. Then the top N most similar candidate are greedily selected from all prefix lists. Pruning prefix lists to size m ensures at each iteration this algorithm runs in $O(N)$. To measure the similarity between canonical and generated pronunciations, ALINE [19], an edit distance algorithm developed to find cognates, is employed.

After decoding speech with ASR, labelling words mispronounced is as simple as determining whether the recognized pronunciation is ranked above or below a threshold in the target word neighbourhood. For the untuned threshold, each word neighbourhood is bisected into equal halves.

3.2. Goodness of pronunciation

The GOP algorithm has been discussed extensively in pronunciation error detection literature [1, 12, 15]. Though Witt and Young [1] proposed the GOP score as a ratio between confidence scores of phones, their experiment invloved frame-based scores to avoid segment size mismatches. Given a phone inventory $\mathcal{P}$, a target phone $p^* \in \mathcal{P}$ with corresponding feature frames $\boldsymbol{o} \in \boldsymbol{O}$, with $|\boldsymbol{o}| = N$ and log-"likelihoods" $\ell(\Theta|o,p)$ derived from acoustic scores from forced alignment using an ASR system with parameters Θ, the GOP score is defined as

$$GOP(p^*, \boldsymbol{o}, \Theta) = \max_{p \in \mathcal{P}} \frac{1}{N} \left| \sum_{i=1}^{N} \left(\ell(\Theta|o_i, p*) - \ell(\Theta|o_i, p) \right) \right| \quad (1)$$

Adapting eq. (1) to the word level for this experiment is straightforward. Given a set of canonical pronunciations for a given word $\mathcal{W}$ with the i^{th}-indexed element of an vector $\boldsymbol{w} \in \mathcal{W}$ representing the i^{th} frame's aligned phone ($\boldsymbol{w}$ should traverse the canonical pronunciation), $\boldsymbol{p}$ a similar vector for free phone recognition, and with $\boldsymbol{o}$ spanning all the frames of the word, word-level GOP can be defined as

$$GOP(\mathcal{W}, \boldsymbol{o}, \Theta) = \frac{1}{N} \left| \max_{\boldsymbol{w} \in \mathcal{W}} \sum_{i=1}^{N} \left(\ell(\Theta|o_i, w_i) - \ell(\Theta|o_i, p_i) \right) \right| \quad (2)$$

Witt and Young suggested per-phone thresholds based on the number of times each phone was labelled mispronounced over the total number of mispronunciations. Letting $\mathcal{S}$ be the set of all experimental sessions in the training set and $c(\mathcal{W}', s)$ be the count of nonnative labels of a specific word in a specific session, a word-level adaptation can be defined as

$$T_{\mathcal{W}} = \alpha \left| \log \left(\frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \frac{c(\mathcal{W}, s)}{\sum_{\mathcal{W}'} c(\mathcal{W}', s)} \right) \right| \quad (3)$$

α is a hyperparameter not in the original phone-level threshold that is independent of $\mathcal{W}$ and allows thresholds to be scaled into the range of GOP scores.

In order to avoid wild inaccuracies due to improper segment boundaries from forced alignment, word-level scoring is performed on only the manually segmented frames (see section 4.1), plus a small amount of frames to the right and left of the segment.

3.3. GMM discriminative classifiers

Franco et al. [2] built a series of four discriminative classifiers which are implemented for this experiment. The error detectors, originally inspired by the speaker verification systems of Reynolds et al. [20], are independent of the size of the segments they classify and can therefore be reproduced almost verbatim. For this reason, technical details are omitted here.

The first system, dubbed GMM1, trains two GMMs per word: one GMM on nonnative word instances, the other on native instances. New instances of a word are classified by the Log-Likelihood Ratio (LLR) of matching the native GMM over the nonnative GMM. The untuned per-word threshold is zero: values over zero imply the native model is more likely and under zero the opposite.

The second system, GMM2, has an identical decision-making process as GMM1. However, the native and nonnative GMMs are Maximum A Posteriori (MAP) adaptations of a Universal Background Model GMM (GMM-UBM). The GMM-UBM is trained on all word instances, regardless of label. The individual GMMs are then MAP adapted with only label-specific data. MAP's hyperparameter, τ, controls the degree to which training with the new data impacts the old model. Again, zero is the untuned threshold.

The third system, GMM3, constructs per-word GMM-UBMs, but then MAP-adapts individual word instances in isolation to train another classifier. The weight and variance vectors of a word instance are concatenated into a "supervector." The supervectors are used to train a per-word linear SVM that

distinguishes between native and non-native instances. Supervectors are normalized so that the global mean of training data is 0 and its variance 1. The natural per-word untuned threshold for GMM3 is the decision boundary of its associated SVM. However, EER tuning can be performed on the signed distance to the hyperplane (ibid.).

The fourth system, GMM4, is simply a threshold on the weighted linear combination of scores of GMM2 and GMM3. In the previous work, weights were chosen by observation; GMM4 weighs scores of GMM2 and GMM3 according to the range of scores observed per word between the lowest score of the 2nd quartile and highest of the 3rd quartile.

For this experiment, GMMs are trained and adapted with the Microsoft Speaker Recognition Toolkit [21]. scikit-learn [22], which implements LIBLINEAR [23], is used to train and test SVMs.

4. Experiment

4.1. Corpus

Audio data were collected as part of a preliminary experiment at the University of Toronto. Beginner learners of French were recruited to participate in an hour-and-a-half long experiment wherein pairs took turns reciting phrases from dialogues. The dialogues, modelled to participants by recorded videos of actors, had no subtitles. Furthermore, participants were expected to make small changes to what they said according to context. They could review the dialogues at any time, but they could not shadow the actors. Pairs were expected to negotiate the meaning of phrases. This mode of interaction, as well as the dialogue content, are adaptations of a communicative pedagogical curriculum [11], designed by an expert in the field, to the complexities of CAPT.

Audio data from 29 sessions (58 participants) were transcribed and segmented by hand, corresponding to approximately 4.1 hours of recorded audio. Data were recorded on an iPad Mini 2 at 16 kHz mono PCM16. Efforts were made to reduce the impact of noise: the experiment took place in a relatively quiet room and participants were instructed to avoid extraneous noise. Since any deployed mobile speech application must be resilient to some noise, only severely distorted segments were removed from analysis. Additionally, beginners would often construct words piecemeal with either silence or filled pauses between syllables. Such segments would be due course for a CAPT application, and were thus retained. In total, 5286 segments are considered, which excludes contracted determiners (*l'*, *m'*, etc.), foreign and non-words, and any word with fewer than 30 instances (initial testing found words with few instances only served to inflate tuned accuracy).

Of those participants who filled out demographic information: 30 were female and 28 male; 26 spoke English as a first language, 11 Portuguese, and 7 either Cantonese or Mandarin; 23 were bilingual, 19 monolingual, and 10 trilingual; and the median age was 23. Though the vast majority of these participants ranked themselves a 1 on a 1-5 ascending scale of fluency in French (48), 15 participants in this set reported having some structured lessons in French. This is primarily due to the grades 4-8 core French program in Ontario, Canada. We included these participants in the analysis to explore a greater range of pronunciations.

Words were labelled as native or nonnative through a series of pairwise comparisons. Four annotators with French teaching experience were enlisted for the task. The definition of "nativeness" was intentionally left vague to better align with their individual criteria. Since this is not a well-defined task and most words were observed to be mispronounced in some way, direct labelling of word segments was replaced with a series of relative assessments that could be manipulated into a partial ranking of words by nativeness. Each annotator made a series of binary decisions based on which of two instances of a given word was more native. All pairs of 10 randomly drawn instances of each word were compared by each annotator. Comparing each instance with every other instance and counting the number of "wins" provided a full ranking of those 10 by nativeness. As this method requires a quadratic number of comparisons with respect to the number of word instances (every instance must be compared once to every other instance), the 4th and 6th ranked points were extracted from each 10 instance subset to act as boundaries for three discrete bins. Comparing new instances to these boundary points determines whether it should be labelled native, nonnative, or unsure (middling).

Table 1: *Annotator agreement (described below)*

	κ_1	κ_2	%N	%NN	%Con	%DB
A	.16	.48	23	36	12	55
B	.20	.52	29	41	9	10
C	.17	.50	29	37	13	10
D	.16	.51	31	23	14	25

Table 1 shows inter- and intra-annotator agreement over a 418 point overlap set. *%N* (*%NN*) is the percentage of instances labelled by that annotator as (non)native. The remainder were either unsure or simultaneously above and below the boundaries ("contradictory", or *%Con*). κ_1 measures one-versus-rest inter-annotator agreement using Cohen's κ over all points in the overlap set. κ_2 is Cohen's κ on only the points labelled native or nonnative by both the one annotator and the "rest" annotator. For reference, Fleiss' κ, a generalization of Cohen's κ for greater than two annotators [24], reported $\kappa_1 = .18$ and $\kappa_2 = .59$ with the same interpretation of subscripts (albeit for κ_2 the number of applicable points dropped to 80). The disparity between κ_1 and κ_2 suggests that nativeness was perceived along a continuum that was difficult to distinguish near its middle but easily distinguished at its poles. It is then expected that a good pronunciation error detector will lie between κ_1 and κ_2, since it needs only to decide between two labels. Finally, *%DB* lists the proportion of points labelled per annotator over all 5286 segments.

4.2. Preprocessing

Each error detector requires features to be extracted from the audio signal prior to classification. Feature frames are calculated by convolving the audio signal with a 25 ms Hamming Window, shifted 10 ms per frame. 12 Mel-frequency Cepstral Coefficients (MFCCs) plus 1 energy coefficient, their deltas, and their double deltas total a 39 coefficient feature vector per frame. Cepstral Mean-Variance Normalization (CMVN) and Automatic Gain Control (AGC) are performed on each utterance.

4.3. Partitioning

The error detectors outlined in section 3 are evaluated using 4-fold cross-validation on each of three different partitioning strategies.

4-Fold Stratified (4FS): The data set is split into four roughly equal quadrants, each containing a database-

proportional repesentation of each word but with the specific instances of those words randomly assigned.

4-Fold Participant (4FP): The data set is split into four roughly equal quadrants, each receiving all points from a unique set of randomly chosen pairs of participants, with the remaining pairs distributed to the smallest quadrants.

4-Fold Annotator (4FA): The data set is partitioned according to who annotated those points. The overlap set (which all annotators annotated) is distributed to the two smallest folds. Since some pairs of participants were only annotated by one annotator, 4FA cannot be perfectly discriminated from 4FP.

Points in the overlap set are labelled according to majority vote for 4FS and 4FP partitioning strategies. 4FA uses the labels of whatever annotator represents the fold the data are placed in.

Both the tuned and untuned thresholds described in section 3 are evaluated. To test how well the tuned thresholds generalize across folds, the unweighted average of per-word optimal thresholds is evaluated.

Each fold in each partitioning strategy is tested once for each value of a small set of adjustable hyperparameters. For PDC, neighbourhood sizes of 10, 50, and 100 generated words are tested. α of 0, 0.0005, 0.001, and 0.002 are tested for GOP. $\tau \in (0, 1, 4, 6)$ are checked for GMM2 and GMM3.

5. Results and discussion

	4FA		4FP		4FS	
	%Acc	κ	%Acc	κ	%Acc	κ
PDC	51	.09	51	.08	51	.08
GOP	60	.12	58	.11	58	.11
GMM1	57	.12	61	.12	65	.19
GMM2	**65**	**.27**	**62**	**.19**	**68**	**.31**
GMM3	61	.19	58	.11	64	.23
GMM4	61	.18	61	.16	67	.29

Table 2: *Results from untuned thresholds*

	4FA		4FP		4FS	
	%Acc	κ	%Acc	κ	%Acc	κ
PDC	53	.06	55	.09	55	.08
GOP	62	.23	63	.25	63	.24
GMM1	63	.23	63	.25	66	.31
GMM2	**68**	**.34**	**66**	**.31**	**71**	**.41**
GMM3	65	.27	62	.23	67	.33
GMM4	64	.27	65	.29	70	.38

Table 3: *Results from tuned thresholds*

	4FA		4FP		4FS	
	%Acc	κ	%Acc	κ	%Acc	κ
PDC	54	.06	57	.07	57	.06
GOP	58	.17	58	.15	58	.16
GMM1	50	.11	56	.15	61	.21
GMM2	**63**	**.27**	**60**	**.20**	**67**	**.32**
GMM3	59	.19	57	.13	63	.23
GMM4	60	.19	60	.19	66	.30

Table 4: *Results from average tuned thresholds*

Tables 2 to 4 show the unweighted average accuracy and detector-annotator Cohen's κ across folds for each partitioning strategy and each error detector.

GMM2 outperformed all other error detectors in all criteria. GMM1 and GMM2 both incorporate more data than GOP and PDC. The best setting of τ for GMM2 when using tuned thresholds is 0: the iterative MAP adaptation places no weight on the GMM-UBM when iterating. This means that GMM2 beats GMM1 by virtue of having smarter initial parameters (those of the GMM-UBM). GMM3 underperformed because there were too few word instance points to choose a hyperplane robust to noise. A lack of optimal setting for τ supports this hypothesis. Given the performance of GMM3, it is no surprise that it added minimal additional information to GMM4, the latter having split the difference between GMM2 and GMM3 in terms of performance.

PDC underperformed in all situations. This highlights that more than just phonotactic knowledge is necessary to build a phonetic distance classifier: knowledge about pronunciation errors specific to an L1-L2 pair is necessary.

The choice of partitioning strategy had a visible effect on error detector performance. In general, error detectors performed better when data from each annotator and each participant were distributed across folds (4FS). It is clear that the error detectors have difficulty generalizing across participants. However, because 4FP does not always share participants across folds, it is unclear how well the error detectors generalize across annotators.

All error detectors received impressive gains in perfomance when word-level thresholds were tuned on the test set. Slight gains are visible when average word-level optimal thresholds are employed versus untuned varieties, but to a far smaller effect than per-fold tuning. This shows that tuned thresholds do not necessarily generalize to new data. This is especially relevant to error detectors such as GOP and PDC: they are usually credited for needing little nonnative training data. To its credit, GOP benefited most from taking the average thresholds. Tuning thresholds on artificially permuted segments (cf. [4]) may be worthwhile.

Regardless of the above qualifications, GMM2 performed well within the expected range of gold-standard between-annotator agreement, making it well suited to the task of finding pronunciation errors at the word level.

6. Conclusions and future work

This research represents the first application of pronunciation error detectors for beginning second language learners' speech, especially as facilitated by interactions with a mobile tutoring app – an emerging field prompted by the increasing availability and computing capabilities of consumer-grade mobile devices. Results from this research, compared to those of previous word-level findings [25], suggest that existing pronunciation error detectors can be applied to more holistic language learning applications, assuming one carefully defines and restricts the task.

Towards this end, there are many possible future avenues of research. First, more complicated models, such as those based on deep neural networks, should be tested. This is a promising direction, considering the simplicity of this experiment's optimal model. Second, alternate labelling strategies that improve inter-annotator agreement should be explored. Performing comparisons with more extreme boundary points or training/testing on points two or more annotators agree on would be viable strategies. For new beginners, the distinction between intelligible and unintelligible [26] will likely be more pronounced. Finally, corpus-based results need to be extrinsically evaluated to determine just how useful word-level error detection is to beginner learners (especially in relation to feedback), a subject of our future research.

7. References

[1] S. M. Witt and S. J. Young, "Phone-level pronunciation scoring and assessment for interactive language learning," *Speech Communication*, vol. 30, no. 2, pp. 95–108, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167639399000448

[2] H. Franco, L. Ferrer, and H. Bratt, "Adaptive and discriminative modeling for improved mispronunciation detection," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 7709–7713.

[3] H. Kibishi and S. Nakagawa, "New feature parameters for pronunciation evaluation in english presentations at international conferences," in *12th Annual Conference of the International Speech Communication Association 2011 (INTERSPEECH 2011)*, 2011, pp. 1156–1159.

[4] H. Strik, K. Truong, F. de Wet, and C. Cucchiarini, "Comparing different approaches for automatic pronunciation error detection," *Speech Communication*, vol. 51, no. 10, pp. 845–852, 2009. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0167639309000715

[5] W. Hu, Y. Qian, F. K. Soong, and Y. Wang, "Improved mispronunciation detection with deep neural network trained acoustic models and transfer learning based logistic regression classifiers," *Speech Communication*, vol. 67, pp. 154–166, 2015.

[6] C. Cucchiarini, H. Strik, D. Binnenpoorte, and L. Boves, "Pronunciation evaluation in read and spontaneous speech: A comparison between human ratings and automatic scores," *Proceedings of the New Sounds*, 2000. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.18.5006&rep=rep1&type=pdf

[7] S.-Y. Yoon, L. Pierce, A. Huensch, E. Juul, S. Perkins, R. Sproat, and M. Hasegawa-Johnson, "Construction of a rated speech corpus of l2 learners' spontaneous speech," *CALICO Journal*, vol. 26, no. 3, 2013. [Online]. Available: https://journals.equinoxpub.com/index.php/CALICO/article/view/23066

[8] F. Hnig, A. Batliner, and E. Nth, "Automatic assessment of non-native prosody annotation, modelling and evaluation," in *Proc. of the International Symposium on Automatic Detection of Errors in Pronunciation Training (ISADEPT)*, 2012, pp. 6–8.

[9] S. M. Witt, "Automatic error detection in pronunciation training: Where we are and where we need to go," in *Proc. of the International Symposium on Automatic Detection of Errors in Pronunciation Training (ISADEPT)*, 2012, pp. 1–8.

[10] O. Mella, D. Fohr, and A. Bonneau, "Inter-annotator agreement for a speech corpus pronounced by french and german language learners," in *Workshop on Speech and Language Technology in Education (SLaTE)*, 2015, pp. 143–147.

[11] S. J. Savignon, "Communicative language teaching," *Theory into Practice*, vol. 26, no. 4, pp. pp. 235–242, 1987. [Online]. Available: http://www.jstor.org/stable/1476834

[12] S.-Y. Yoon, M. Hasegawa-Johnson, and R. Sproat, "Landmark-based automated pronunciation error detection," in *11th Annual Conference of the International Speech Communication Association 2010 (INTERSPEECH 2010)*, 2010, pp. 614–617.

[13] X. Qian, H. Meng, and F. Soong, "On mispronunciation lexicon generation using joint-sequence multigrams in computer-aided pronunciation training (capt)," in *12th Annual Conference of the International Speech Communication-Association 2011 (INTERSPEECH 2011)*, 2011, pp. 872–875.

[14] W. Hu, Y. Qian, and F. K. Soong, "An improved dnn-based approach to mispronunciation detection and diagnosis of l2 learners' speech," in *Workshop on Speech and Language Technology in Education (SLaTE)*, 2015, pp. 71–76.

[15] R. Kanters, C. Cucchiarini, and H. Strik, "The goodness of pronunciation algorithm : a detailed performance study," in *In SLaTE 2009 - 2009 ISCA Workshop on Speech and Language Technology in Education*, 2009, pp. 2–5.

[16] P. Delglise, Y. Estve, S. Meignier, and T. Merlin, "The lium speech transcription system: a cmu sphinx iii-based system for french broadcast news." in *6th Annual Conference of the International Speech Communication Association 2005 (INTERSPEECH 2005)*, 2005, pp. 1653–1656.

[17] D. Huggins-Daines, M. Kumar, A. Chan, A. Black, M. Ravishankar, and A. Rudnicky, "Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 1, 2006, pp. I–I.

[18] A. M. Harrison, W.-K. Lo, X.-j. Qian, and H. Meng, "Implementation of an extended recognition network for mispronunciation detection and diagnosis in computer-assisted pronunciation training," *Proc. of SLaTE2009*, 2009.

[19] G. Kondrak, "Alignment of phonetic sequences," Department of Computer Science, University of Toronto, Tech. Rep. CSRG-402, 1999.

[20] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19 – 41, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1051200499903615

[21] S. O. Sadjadi, M. Slaney, and L. Heck, "Msr identity toolbox v1.0: A matlab toolbox for speaker-recognition research," *Speech and Language Processing Technical Committee Newsletter*, 2013. [Online]. Available: http://research.microsoft.com/apps/pubs/default.aspx?id=205119

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[23] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, 2008. [Online]. Available: http://dl.acm.org/citation.cfm?id=1390681.1442794

[24] J. L. Fleiss, "Measuring nominal scale agreement among many raters," *Psychological bulletin*, vol. 76, no. 5, pp. 378–382, 1971. [Online]. Available: http://search.proquest.com/docview/614289059?accountid=14771

[25] T. Cincarek, R. Gruhn, C. Hacker, E. Nth, and S. Nakamura, "Automatic pronunciation scoring of words and sentences independent from the non-natives first language," *Computer Speech & Language*, vol. 23, no. 1, pp. 65–88, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0885230808000193

[26] T. Pongkittiphan, N. Minematsu, T. Makino, D. Saito, and K. Hirose, "Automatic prediction of intelligibility of english words spoken with japanese accents - comparative study of features and models used for prediction," in *Workshop on Speech and Language Technology in Education (SLaTE)*, 2015, pp. 19–22.