Automatic Pronunciation Evaluation And Mispronunciation Detection Using CMUSphinx

Ronanki Srikanth[1] *Li Bo*[2] *James Salsman*[3]
(1) International Institute of Information Technology, Hyderabad, India
(2) National University of Singapore, Singapore
(3) Talknicer, USA
srikanth.ronanki@research.iiit.ac.in, li-bo@outlook.com, jsalsman@talknicer.com

Abstract
Feedback on pronunciation is vital for spoken language teaching. Automatic pronunciation evaluation and feedback can help non-native speakers to identify their errors, learn sounds and vocabulary, and improve their pronunciation performance. These evaluations commonly rely on automatic speech recognition, which could be performed using Sphinx trained on a database of native exemplar pronunciation and non-native examples of frequent mistakes. Adaptation techniques using target users' enrollment data would yield much better recognition of non-native speech. Pronunciation scores can be calculated for each phoneme, word, and phrase by means of Hidden Markov Model alignment with the phonemes of the expected text. In addition to the basic acoustic alignment scores, we have also adopted the edit distance based criterion to compare the scores of the spoken phrase with those of models for various mispronunciations and alternative correct pronunciations. These scores may be augmented with factors such as expected duration and relative pitch to achieve more accurate agreement with expert phoneticians' average manual subjective pronunciation scores. Such a system is built and documented using the CMU Sphinx3 system and an Adobe Flash microphone recording, HTML/JavaScript, and rtmplite/Python user interface.

Keywords: Pronunciation Evaluation, Text-independent, forced-alignment, edit-distance neighbor phones decoding, CMUSphinx.

Proceedings of the Workshop on Speech and Language Processing Tools in Education, pages 61–68,
COLING 2012, Mumbai, December 2012.

1 Introduction

Pronunciation learning is one of the most important parts of second language acquisition. The aim of this work is to utilize automatic speech recognition technology to facilitate learning spoken language and reading skills. Computer Aided Language Learning (CALL) has received a considerable attention in recent years. Many research efforts have been done for improvement of such systems especially in the field of second language teaching. Two desirable features of speech enabled computer-based language learning applications are the ability to recognize accented or mispronounced speech produced by language learners, and the ability to provide meaningful feedback on pronunciation quality.

The paper is organized into the following sections : Section 2 discusses in detail some of the popular and best performing approaches proposed for pronunciation scoring and computer-aided language learning. We present in Section 3 our database preparation for evaluation of the proposed method along with description of TIMIT database used as reference statistics in Text-independent approach and is explained in section 5. Section 4 presents an algorithm to detect mispronunciations based on neighbor phones decoding. Section 5 presents scoring routines for both Text-dependent and Text-independent approaches and finally results are tabulated in section 6 followed by conclusions.

2 Related Work

The EduSpeak system (Franco H. Abrash and J, 2000) is a software development toolkit that enables developers to use speech recognition and pronunciation scoring technology. The paper presents some adaptation techniques to recognize both native and non-native speech in a speaker-independent manner. (L. Neumeyer and Price, 1996) developed automatic Text-independent pronunciation scoring of foreign language student speech by using expert judge scores.

(Seymore and R, 1996) created a system called Fluency (Eskenazi, 2009) to detect and correct foreign speakers pronunciation errors in English. She also used automatic speech recognition to detect pronunciation errors and to provide appropriate correct information.

(Peabody, 2011) focused on the problem of identifying mispronunciations made by non-native speakers using a CALL system. He also proposed a novel method for transforming mel-frequency cepstral coefficients (MFCCs) into a feature space that represents four key positions of English vowel production for robust pronunciation evaluation. (Moustroufas and Digalakis, 2007) presented various techniques to evaluate the pronunciation of students of a foreign language, again without using any knowledge of the uttered text. The authors used native speech corpora for training pronunciation evaluation.

(Sherif Mahdy Abdou and Nazih, 2006) described the implementation of a speech enabled computer-aided pronunciation learning system called HAFSS. The system was developed for teaching Arabic pronunciation to non-native speakers. It used a speech recognizer and a phoneme duration classification algorithm implemented to detect pronunciation errors. The authors also used maximum likelihood linear regression (MLLR) speaker adaptation algorithms.

(Chitralekha Bhat, 2010) designed a pronunciation scoring system using a phone recognizer using both the popular HTK and CMU Sphinx speech recognition toolkits. The system was evaluated on Indian English speech with models trained on the Timit Database. They used forced alignment decoding with both HTK and Sphinx3.

(S. Pakhomov and G.Sales, 2008) and (Eskenazi, 2002) described the measurement of different automatic speech recognition (ASR) technologies applied to the assessment of young children's basic English vocabulary. Former authors used the HTK version 3.4 toolkit for ASR. They calculated acoustic confidence scores using forced alignment and compared those to edit distance between the expected and actual ASR output. They trained three types of phoneme level language models: fixed phonemes, free phonemes and a biphone model.

3 The Data

3.1 Training: TIMIT Data

We used standard TIMIT corpus for training the Text-Independent pronunciation evaluation system and is explained in section 5.2. TIMIT contains broadband recordings of 630 speakers of eight major dialects of American English, each reading ten phonetically rich sentences. The TIMIT corpus includes time-aligned orthographic, phonetic and word transcriptions as well as a 16-bit, 16kHz speech waveform file for each utterance.

3.2 Testing: Data Preparation

We prepared a Non-native database in Indian accent to test the proposed pronunciation evaluation system. The corpus contains recordings of 8 non-native speakers of English from four different regions of India, each reading five sentences and five words. We asked the speakers to pronounce each word 10 times in one complete recording and then mispronounce 10 times either by spelling one of the phones incorrect or by skipping some of the phones in each word, each time. We also asked the speakers to pronounce each sentence 3 times in one complete recording and then mispronounce 3 times by spelling one or more than one word incorrectly. Later, we manually chopped the wav files into recordings of each word, sentence in separate individual files. Thus, we have 400 correct and incorrect recordings of 5 words and 120 correct and incorrect recordings of 5 sentences from 8 Non-Native speakers of English.

4 Edit-distance Neighbor phones decoding

We started our work as to identify the mispronunciations using the help of speech recognition tool Sphinx3. The decoding results shown that both word level and phrase level decoding using Java State Grammar Format (JSGF) are almost same. This method helps to detect the mispronunciations at phone level and to detect homographs as well if the percentage of error in decoding can be reduced.

4.1 Phoneset, Models and Sphinx Decoder

The decoder we used in this paper is Sphinx3_decode which requires either Language Model(LM) or Finite State Grammar(FSG) along with acoustic models trained on large vocabulary database. We used WSJ1 (Lee, 1989) acoustic models for wideband (16kHz) microphone speech, consisting 4000 senone and 32 Gaussian mixtures per stature as Hidden Markov Models (HMM) models to train the system.

Finite State Grammar(FSG) which can be derived from JSGF contains the transition probabilities from one state to another and is supplied as input to the decoder instead of Language model along with acoustic models.

Since we are using CMUSphinx decoder, the phoneset being used in this algorithm is CMU-Arctic phoneset which is also known as CMUbet. Worldbet, CGIbet,ARPAbet are few such other ASCII based phonetic alphabets. Neighboring phones are the list of phones which contains most similar other phonemes for each phoneme in CMUbet. We have chosen the neighbor phones for each phoneme in such a way that the mispronunciation can occur with similar sounding phonemes. For example, the neighbors for phoneme /N/ are (/N/|/M/|/NG/), and /TH/ are (/TH/|/S/|/DH/|/F/|/HH/) etc.,

Along with these. we used CMU dictionary which contains all words in English vocabulary with corresponding representation of phones in CMUbet. In languages like English it is very common to find that the same word can be pronounced in several different ways also known as homographs. The dictionary file in Sphinx is allowed to have several entries for the same word. However, for the system to work properly, the transcription file must state which pronunciation alternative is used for each word. Sphinx provides a way to do this automatically, which is called forced alignment.

4.2 Sphinx Forced-Alignment

The process of force-alignment takes an existing transcript, and finds out which, among the many pronunciations for the words occuring in the transcript, are the correct pronunciations. The output is written into a file with an option phsegdir in sphinx3_align and it contains each phone start and end positions in terms of frames on time scale along with large negative acoustic spectral match score.

SFrm	EFrm	SegAScr	Phone
0	9	-64725	SIL
10	21	-63864	W SIL IH b
22	30	-126819	IH W TH i
31	41	-21470	TH IH SIL e

Table 1: Format of phseg file for a sample word: “WITH”

4.3 Algorithm to detect Mispronunciations

Based on the forced-alignment output, we designed few decoders such as single-phone decoder, word decoder and phrase decoder. Initially, the wav file is chopped into individual phones in case of single-phone decoder, words in case of word decoder and complete phrase is taken in case of phrase decoder. JSGF file is specified in such a way that, each time, the phone is supplied along with its neighbor phones. In single-phone decoder, each phoneme chopped in a separate wav file is decoded along with its neighbor phones. In phrase decoder, all phones along with its neighbor phones is given as input to JSGF. In case of word-decoder, to identify mispronunciation at phone-level, each time only one phoneme is supplied with its neighbor phones keeping the rest constant. For example, word - “WITH” is presented as
public <phonelist> = ((W | L | Y) (IH) (TH));
public <phonelist> = ((W) (IH | IY | AX | EH) (TH));
public <phonelist> = ((W) (IH) (TH | S | DH | F | HH));

The accuracy of each decoder for SA1, SA2 in TIMIT and for some recorded external phrase

is reported in Table 2. Both Word decoder and Phrase decoder perform at equal level since the decoding of context-independent phones doesn't vary much across word boundaries. Since, the error-rate can't be negligible even if it is too low, we moved to threshold based scoring method which is explained in next section.

Type	Single-Phone	Word	Phrase
SA1	41.3%	86.1%	84.4%
SA2	42.5%	87%	85.2%
ext. phrase	29%	73.2%	72.1%

Table 2: Decoding Accuracy of each decoder

5 Scoring Routines

5.1 Text-dependent

In Text-dependent approach, we can do pronunciation scoring only for those words/phrases for which we have at least 10-50 native exemplar recordings. This method is completely based on exemplar recordings for each phrase. Initially, Sphinx forced alignment is applied on native exemplar recordings of each phrase in the training dataset. Later, mean acoustic score, mean duration along with standard deviations are calculated for each of the phones in the phrase from the forced-alignment output. Since the acoustic scores are in large negative values, logarithm is applied i.e., log(1-acs) is considered into account where acs is the acoustic score of each phone. Now, given the test recording, each phoneme in the phrase is then compared with exemplar statistics with respect to position of the phoneme in the phrase. The standard score of a raw score x is:

$$z = \frac{x - \mu_i}{\sigma_i} \tag{1}$$

z-scores are calculated from equation (1) for both acoustic score and duration and then normalized scores from 1-5 are calculated based on maximum and minimum of z-scores of each phoneme from native exemplar statistics. All phoneme scores are averaged over each word and then all word scores are aggregated with some weightage given with respect to parts of speech(POS) to get the complete phrase score.

POS	weight	POS	weight	POS	weight
Quantifier	1.0	Adverb	0.8	Possessive	0.6
Noun	0.9	Adjective	0.8	Conjunction	0.5
Verb	0.9	Pronoun	0.7	articles	0.4
Negative	0.8	Preposition	0.6		

Table 3: Weightage of a word based on parts of speech

5.2 Text-independent

The advantage of this Text-independent approach is that we can do pronunciation scoring given any random word or phrase without the requirement of native exemplar recordings for that particular word or phrase. This algorithm is based on pre-determined statistics built from some corpus. Here, in this paper, we used TIMIT corpus to build statistics.

There are 630 speakers in TIMIT each recording 10 sentences. All the wav files are forced-aligned with its transcription to get spectral acoustic match score and duration. Later, we derived statistics for each phone based on its position (begin/middle/end) in the word.

Now, given any random test file, each phone acoustic score, duration is compared with corresponding phone statistics based on its position. The scoring method is same as to that of Text-dependent system.

6 Results

Our main aim of the proposed algorithm is to detect mispronunciations and give reasonable feedback with a score of 1-10. We mainly concentrated on two factors: pronunciation match with correct phone and duration. Edit-distance neighboring phones decoding works well within limits of error-free decoding. Demo of the system is at http://talknicer.net/~ronanki/test/

Initially, we tested the Text-independent system with TIMIT, SA1 and SA2 sentences. The results in Table 4 shows that threshold greater than 7.5 is reasonably good for correct pronunciation. So, we made 7.5 as hard threshold boundary between correct and incorrect pronunciation for any phrase and evaluated the performance of system on our database mentioned in section 3.2. From table 5 and 6, it is observed that Text-independent system works well for phrases even with hard-bounded threshold value.

Sentence	Min.	Max.	Mean	Thres. > 7	Thres. > 7.5	Thres. > 8
SA1	7.14	9.01	8.58	630/630	627/630	612/630
SA2	7.38	8.93	8.50	630/630	627/630	611/630

Table 4: Performance of the TIMIT sentences using Text-independent system

Type	Mean	Thres. > 6	Thres. > 6.5	Thres. > 7
Correct	7.07	354/400	302/400	219/400
Type	Mean	Thres. < 6	Thres. < 6.5	Thres. < 7
Wrong	6.13	170/400	255/400	320/400

Table 5: Performance of words in both cases using Text-independent system

Type	Mean	Thres. > 7	Thres. > 7.5	Thres. > 8
Correct	7.79	108/120	96/120	88/120
Type	Mean	Thres. < 7	Thres. < 7.5	Thres. < 8
Wrong	6.73	86/120	98/120	118/120

Table 6: Performance of sentences in both cases using Text-independent system

Conclusions

Our future work is to concentrate on CART modelling to get better reference statistics based on contextual information of the phone. This tree based clustering model really helps the system to get more efficient scores. Future work will also include deployment on web and stand-alone servers using CMU Sphinx v3 in C, SQL, JavaScript, PHP, Dalvik Java and Objective C. The pronunciation evaluation system really helps second-language learners to improve their pronunciation by trying multiple times and it lets you correct your-self by giving necessary feedback at phone, word level.

References

Chitralekha Bhat, K.L. Srinivas, P. R. (2010). Pronunciation scoring for indian english learners using a phone recognition system. In *Proceedings of the First International Conference on Intelligent Interactive Technologies and Multimedia*, pages 135–139.

Eskenazi, M. (2009). An overview of spoken language technology for education. In *Proc. of Speech Communication, Elsevier, vol 51 issue 10*, pages 832–844.

Eskenazi, M., P. G. (2002). Pinpointing pronunciation errors in children's speech: examining the role of the speech recognizer. In *Proposed to the Pronunciation Modeling and Lexicon Adaptation for Spoken Language Technology Workshop, Sept 2002, Colorado.*

Franco H. Abrash, V. Precoda, K. B. H. R. and J, B. (2000). The sri eduspeak system: Recognition and pronunciation scoring for language learning. In *Proceedings of InSTIL, Scotland*, pages 123–128.

L. Neumeyer, H. Franco, M. W. and Price, P. (1996). Automatic text-independent pronunciation scoring of foreign language student speech. In *Proc. of ICSLP 96, Philadelphia, Pennsylvania*, pages 1457–1460.

Lee, K.-F. (1989). Automatic speech recognition: The development of the sphinx system. In *Kluwer Academic Publishers, Boston.*

Moustroufas, N. and Digalakis, V. (2007). Automatic pronunciation evaluation of foreign speakers using unknown text. In *Comput. Speech Language*, page 219–230.

Peabody, M. A. (2011). *Methods for Pronunciation Assessment in Computer Aided Language Learning.* PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA.

S. Pakhomov, J. Richardson, M. F.-D. and G.Sales (2008). Forced-alignment and edit-distance scoring for vocabulary tutoring applications. In *Lecture Notes in Computer Science, Volume 5246/2008*, pages 443–450.

Seymore, K., C. S. E. S. and R, R. (1996). Language and pronunciation modelling in the cmu 1996 hub-4 evaluation. In *Proc. of DARPA Speech Recognition workshop, chantilly, Virginia, Morgam kaufmann Publishers.*

Sherif Mahdy Abdou, Salah Eldeen Hamid, M. R. A. S. O. A.-H. M. S. and Nazih, W. (2006). Computer aided pronunciation learning system using speech recognition techniques. In *in Interspeech.*