Automatic Pronunciation Scoring with Score Combination by Learning to Rank and Class-Normalized DP-Based Quantization

Liang-Yu Chen, *Student Member, IEEE*, and Jyh-Shing Roger Jang, *Member, IEEE*

***Abstract*—This paper proposes an automatic pronunciation scoring framework using learning to rank and class-normalized, dynamic-programming-based quantization. The goal is to train a model that is able to grade the pronunciation of a second language learner, such that the predicted score is as close as possible to the one given by a human teacher. Under this framework, each utterance is given a score of 1 to 5 by human raters, which is treated as a ground truth rank for the training algorithm. The corpus was rated by qualified English teachers in Taiwan (nonnative speakers). Nine phone-level scores are computed and converted into word-level scores through four conversion methods. We select the 16 best performing scores as the input features to train the learning-to-rank function. The output of the function is then quantized to a discrete rank on a 1-5 scale. The quantization is done with class normalization to alleviate the problem of data imbalance over different classes. Experimental results show that the proposed framework achieves a higher correlation to the human scores than other methods, along with higher accuracy in detecting instances of mispronunciation. We also release a new version of our nonnative corpus with human rankings.**

***Index Terms*—Automatic pronunciation scoring, computer assisted language learning (CALL), computer assisted pronunciation training (CAPT), learning to rank.**

I. Introduction

THE introduction of computer-assisted language learning systems provides L2 learners (second language learners) a new means by which to improve their language skills without the presence of human teachers. Such systems have been shown to be useful in improving students' language skills [1]–[3] and typically address one or more of the four basic aspects of language learning: listening, speaking, reading, and writing. Aside from offering static course material, such systems are also expected to provide performance feedback, particularly in computer-assisted pronunciation training (CAPT) systems, offering learners instant assessments on their pronunciation.

Various methods have been proposed for different tasks in CAPT applications. Some focus on detecting mispronunciation in L2 utterances [4]–[6], while others assess stress placement in a word or a sentence [7]–[9]. Black *et al.* [10] used four methods to verify (i.e., to accept or reject) children's pronunciation of English letter-names and letter-sounds, while other researchers tried to grade L2 learner pronunciation quality. Neumeyer *et al.* [11] proposed four sentence-level scoring methods based on log-likelihood and duration information obtained from forced alignment using hidden Markov models (HMMs) [12]. Franco *et al.* [13] expanded Neumeyer's work to propose another scoring method based on posterior probability by considering both log-likelihood and prior probability for a phone segment. These scoring methods were also used to grade 10 specific French phones in [14]. Our previous work proposed three word-level scoring methods [15][16]. In this study, we refer to the scoring methods based on a single aspect of pronunciation quality as basic pronunciation scoring, hence these scores are called *basic scores*.

On the other hand, a number of researchers have focused on combining different scores to produce a single score for an utterance in order to compensate for the various weaknesses of each scoring method. Franco *et al.* [17] explored various linear and nonlinear score combination methods to aggregate the three basic scores proposed in [11] and [13]. Results show that neural networks perform better than the other three score combination methods (linear regression, probability distribution estimation, and regression trees). However, Franco also mentioned that neural networks have a disadvantage in their relatively high computing costs for training, where different network architectures must be tested and their training parameters must be manually tuned to obtain optimal performance.

Cincared *et al.* [18] also proposed two score combination methods to aggregate various sentence-level and word-level basic scores. The first method treats each score (1-5) as a class and models each class as a Gaussian distribution. The score of the input utterance is then computed from the likelihood values of the Gaussian models and the class prior probabilities. The second method combines basic scores linearly, using a multiplicative polynomial to avoid producing skewed results from the linear combination step.

As suggested by Yang and Chen [19], it is easier for a human rater to make a relative judgment than to assign an exact score,

Manuscript received August 24, 2013; revised December 24, 2013; accepted June 08, 2015. Date of publication June 23, 2015; date of current version June 30, 2015. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Bowen Zhou.

L. Y. Chen is with the Institute of Information Systems and Applications, National Tsing Hua University, Hsinchu 30013, Taiwan (e-mail: davidson833@mirlab.org).

J. S. R. Jang is with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei 10617, Taiwan (e-mail: jang@csie.ntu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASLP.2015.2449089

i.e., people tend to judge performance based on a relative sense of other instances. To capture this relative information, our earlier work proposed a score combination method which integrated learning to rank and a DP-based (dynamic-programming-based) quantization algorithm to transform the output of the learning to rank function to pronunciation scores [15]. Under this framework, each utterance of the training data is given a human-labeled score from 1-5. By treating each score as a rank and treating basic pronunciation scores as input features, we can apply any existing learning to rank algorithms and thus grade the pronunciation quality. Results show that this learning to rank framework (using RankSVM) can yield higher scoring correlation to human raters than using k-means and all other basic scoring methods. In [16], we examined the use of three types of phone-level to word-level conversion methods (average-based, vowel-based, and consonant-based) to convert six basic scores, and compared the performance of two learning to rank algorithms, RankSVM and ListNet, to three existing score combination methods proposed in [17] and [18]. It was found that using all three types of phone-level to word-level conversion methods (i.e., all 18 word-level basic scores) yields better performance than using average-based method alone, and ListNet achieves the highest correlation among all five score combination methods.

Two issues were not considered in our previous work. First, our previous system simply adopts all types of basic scores as inputs for the learning to rank function. The effect of each individual basic score on overall performance was not examined. Second, the dataset used in the experiment is imbalanced over different classes. While this shows how data would be distributed in the practical situation, the imbalanced data issue was not considered in the algorithm.

In this paper, we aim to improve our previously proposed learning to rank framework as follows:

1) Nine basic scores are examined for their effectiveness under four different phone-level to word-level conversion methods. In addition to the three abovementioned conversion methods, the duration-weighted method is also examined.
2) A class-normalized quantization is proposed to alleviate the problem of imbalanced data over different classes.
3) We examine how basic scores in different forms affect the performance of score combination. For instance, for some basic scores, using the proposed quantization method to convert the raw basic scores to quantized ones can alleviate the skew distribution problem.
4) By adopting the above improvement, two learning to rank algorithms are evaluated against 5 other methods, including the three score combination methods proposed by [17] and [18] and two ordinal regression methods.

To evaluate the proposed method, we also established a nonnative speech database called MIR-SD (Multimedia Information Retrieval lab, Stress Detection) [20], a dataset that was originally designed for stress detection of multi-syllabic English words and was recorded by Taiwanese speakers. MIR-SD contains over 10,000 utterances recorded by 51 speakers, and currently only 2000 utterances from 25 speakers were rated by four qualified English teachers in Taiwan (nonnative speakers). This dataset, as well as the human scores, is freely available online.[1]

The rest of this paper is organized as follows. Section II describes the native and nonnative speech corpora and the human ranks of the nonnative speech corpus. Section III introduces nine basic pronunciation scoring methods. Section IV explains how to incorporate these basic scores into the learning to rank framework for score combination. Section V reports the experimental results of various system components as well as the overall performance comparison, and Section VI presents conclusions and some future research directions.

II. Speech Corpora

A. Native Corpus

To assess the pronunciation quality of an utterance, we need to first establish a speech recognition engine. In this study, the WSJ corpus [21] (wsj0, 64442 utterances) is used to train a set of biphone HMMs for speech recognition. Some previous work [22][23] shows that, for the pronunciation evaluation of nonnative English utterances, monophone (or biphone) models perform better than triphone models, if the models are trained from native speech. This is because monophopne and biphone models contain less contextual information than triphone models and are more tolerant of mismatches between native and nonnative pronunciation. The work in [22] suggests that biphone models perform better than triphone models in terms of having more tolerance for pronunciation mismatches while retaining a certain degree of context-dependency information. The work in [23], on the other hand, suggests the use of monophone models rather than triphone models. Based on our preliminary experiment, biphone models perform slightly better in basic pronunciation scoring than monophone models in terms of human-machine correlation. Biphone HMMs are therefore adopted in this study. Features used in this study include 12-dimensional Mel-frequency cepstral coefficients (MFCCs) and one dimensional energy, along with their first and second derivatives.

To evaluate the quality of the trained acoustic models, we perform forced alignment on the test set of the TIMIT dataset [24]. The performance is reported in terms of the percentage of the automatically-aligned boundaries being within a given time threshold of the corresponding manually-aligned boundaries. An HMM/ANN hybrid model trained using the TIMIT training set [25] and five acoustic-phonetic features and PLP features achieved 92.57% agreement using 20 ms threshold, where the model. Our acoustic models (HMM, trained using WSJ and tested on TIMIT) achieve 67.4% agreement using a 20 ms threshold and 86.3% using a 40 ms threshold. Despite our alignment agreement being lower due to mismatch of training and test data and less complex models, the majority of phonemes (86.3%) are still within a reasonable range of 40 ms.

A subset of the corpus (35487 utterances) is also used to compute various statistics of each biphone model for basic pronunciation scoring. Section III provides a detailed explanation for how this information is used.

[1]http://mirlab.org/dataset/public

TABLE I
GUIDELINES FOR HUMAN SCORING

Score	Explanation
1	Unintelligible pronunciation, difficult to determine which word was pronounced
2	Obvious mispronunciation but still intelligible, probably with more than 2 mispronounced syllables
3	Minor mispronunciation, probably with a misplaced stress or a mispronounced syllable
4	Intelligible pronunciation but with a minor nonnative accent
5	Intelligible pronunciation without an apparent nonnative accent

TABLE II
CONSISTENCY EVALUATION BETWEEN HUMAN RATERS

	Inter-rater	HR1-GT	HR2-GT	HR3-GT	HR4-GT
Correlation	0.66	0.82	0.72	0.88	0.81

TABLE III
DISTRIBUTION OF GROUND TRUTH SCORES

Score	1	2	3	4	5
Count	92	71	260	463	1114
%	4.6	3.5	13.0	23.1	55.7

B. Nonnative Corpus

Raab *et al.* [26] reviewed a number of nonnative speech databases. However, only a few of these satisfy the requirements of our research, and these databases are no longer publicly available. We therefore decided to construct our own nonnative speech database, MIR-SD [20], along with human ratings on pronunciation quality. It was recorded by 51 Taiwanese speakers as a take-home task for students in a postgraduate computer science course. They were instructed to record the prompted English words using their own unidirectional microphones. Similar to the corpus collection process used in [14], recordings with poor recording quality (either due to background noise or insufficiently loud recording volume) or with serious disfluency or hesitation were omitted, thus reducing the corpus size to 25 speakers (6 females and 19 males). The speakers' mother tongue is Mandarin and their English competence levels are mostly intermediate, meaning they are capable of pronouncing most of the words correctly. Each speaker was asked to record over 200 English words, where each utterance contains only one multi-syllabic English word selected from various sources including medical articles, vocabulary lists for university entrance exams in Taiwan, and an English spelling contest for university students in Taiwan. The recording resolution is 16 bits and the sampling frequency is 16 kHz. In this study, only 80 utterances from each speaker are used, for a total of 2000 utterances.

C. Human Rankings

Human rankings of the nonnative corpus are required for both training and evaluating the proposed system. Each of the 2000 utterances was scored by four human raters on a scale of 1 (unintelligible) to 5 (intelligible). The raters are qualified English teachers with master degrees from the foreign language departments of various highly ranked universities in Taiwan. To ensure consistency, the raters were given a scoring guideline, as shown in Table I, before commencing the scoring task. Compared to other studies, this scoring guideline was rather lenient in terms of consistency with native pronunciation. To be useful to a wide range of L2 students, our system aims to help them pronounce words intelligibly rather than train them to emulate native-speaker pronunciation. Although the human raters were not native English speakers, their English language skills were sufficient to determine the intelligibility of pronunciation.

Table II shows the word-based inter-rater correlation coefficient and human-to-ground-truth correlations, indicating the consistency between human raters, where HR1 to HR4 denote the four human raters, and GT denotes ground truth. The ground truth is computed by voting among all four raters; an average of the four raters is used in the event of a tie. Inter-rater correlations are computed as the average of correlations between all pairs of human raters. Word-based correlations are computed using the scores for each word. The correlations shown in Table II are comparable to most of the existing research [13], [14], [18], and this is the theoretical upper bound of the performance of score computation.

Table III shows the distribution of the ground truth scores across all 2000 utterances. In general, most of the utterances are scored 4 or 5, indicating most speakers have an intermediate competence level in pronunciation skill.

III. BASIC PRONUNCIATION SCORING

This section briefly describes the nine basic scoring methods used in this study. The required phonetic time alignment and likelihood for these basic scoring methods are computed by forced alignment with HMMs which are trained from a large native corpus (WSJ in this study). Some other sentence-based scoring methods [23], such as standard deviations of pitch and power, rate of speech, etc., were also examined. But in preliminary experimental results, these sentence-based scoring methods exhibit little correlation to human rankings, probably because the utterances used in this study only contain a single word, thus the change in the pitch or power is not as significant as it would be in a complete sentence. These sentence-based scoring methods are therefore not considered in this paper.

A. HMM-based Log-Likelihood Score

The HMM-based log-likelihood score $\hat{l}$, denoted by *hmmLike*, is defined as the duration-normalized log-likelihood of a phone segment [11]:

$$\hat{l} = \frac{1}{d} \sum_{t=t_0}^{t_0+d-1} \log p\left(y_t|q_i\right), \tag{1}$$

where $p(y_t|q_i)$ is the likelihood of the frame t with observation vector y_t, d is the duration of the phone segment in frames, t_0 is the index of the starting frame, and q_i represents the ith phone model.

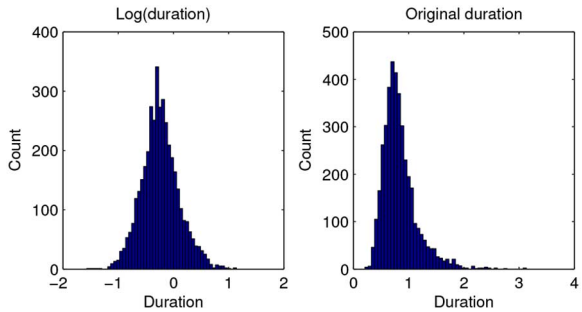

Fig. 1. Histograms showing the distributions of original duration and log(duration) of 4003 samples of /ih + f/ segment.

B. HMM-based Posterior and Log-Posterior Probability Score

The HMM-based log-posterior probability score $\hat{p}$, denoted by *hmmLPst*, is defined as the duration-normalized log-posterior probability of a phone segment [13]:

$$\hat{\rho} = \frac{1}{d} \sum_{t=t_0}^{t_0+d-1} \log P\left(q_i|y_t\right), \tag{2}$$

where the frame-based posterior probability $P(q_i|y_t)$ is defined as:

$$P\left(q_i|y_t\right) = \frac{p\left(y_t|q_i\right) P\left(q_i\right)}{\sum_{j=1}^{M} p\left(y_t|q_j\right) P\left(q_j\right)}, \tag{3}$$

where $P(q_i)$ is the prior probability of the phone model q_i, M is the number of competing models of q_i plus 1 and the denominator term denotes the sum of prior-multiplied likelihoods of all competing models, including the correct model itself. The competing biphone models here are defined as those models with the same right-context dependent phoneme. For example, if a, b, c, d, and e are five pseudo phonetic symbols, the biphone model /b + a/ would have competing models /c + a/, /d + a/, and /e + a/, if they exist in the training corpus. In this study, we also consider the posterior score, denoted as *hmmPost*, without taking log in Eq. (2). *hmmPost* ranges between 0 and 1 and might fit better to human scoring (as opposed to the log version where the score range is between $-\infty$ to 0).

C. Duration Distribution Score

Duration distribution score, denoted by *durDist*, refers to the likelihood of the phoneme model having the given duration based on statistics from the native speech corpus [11], [14], [18]. The duration (in frames) of a phone segment is computed via forced alignment. This value is then normalized by the speech rate of the corresponding utterance, where the speech rate is defined as the average number of phones per unit of time. The likelihood of the normalized duration can then be computed using a probability density function (PDF) of the corresponding biphone model. Based on [18] and our observations (as shown in Fig. 1), a PDF of a log-normal distribution is used to model the distribution of the normalized duration of each biphone. The duration distributions are estimated from the WSJ corpus.

D. Segment Classification Score

The segment classification score, denoted as *segClass*, is defined as the phone classification accuracy of a word, and thus it is a word-based score [11]. The idea is that, since the phone classifier is trained using a native speech corpus, the closer to native pronunciation the test speaker speaks, the closer the test speech is to the distribution in the training speech and thus the higher classification accuracy should be. This type of score is similar to the rank ratio score (to be described later) in that it disregards the relative magnitude of the likelihood values but only asks if the correct model has the highest likelihood among all competing models.

E. Likelihood Distribution Score

The likelihood distribution score, denoted by *likeDist*, refers to the probability of the phoneme model having the given likelihood value. Different from *durDist*, *likeDist* is based on a Gaussian CDF (cumulative density function) of the log-likelihood value of the phone segment instead of a Gaussian PDF [15]. This implies that a higher likelihood value generates a higher *likeDist* score. The likelihood distribution of each biphone is estimated from the WSJ corpus. The idea behind this scoring method is that vowel models tend to have higher likelihood values than consonant models do. They can be shifted to a common ground by transforming the original likelihood values using their distributions.

F. Posterior and Log-Posterior Distribution Score

The posterior and log-posterior distribution scores, respectively denoted by *postDist* and *lpstDist*, adopt the same formulation as the *likeDist* score [16]. Since the posterior probability is computed based on the likelihood and prior probability, it shares a common characteristic of having larger values for vowel models and smaller values for consonant models. Log-posterior distribution score is devised to avoid this problem. We also examine the use of *postDist* (without taking logarithm).

G. Rank Ratio Score

The rank ratio score, denoted by *rkRatio*, is based on the rank of the correct biphone model among all competing models according to their likelihood values [27]. This type of score also aims to eliminate the natural difference between the likelihood values of vowel and consonant models. The model with the highest likelihood value is ranked as 1 and the lowest as $p + 1$ where p is the number of competing models (excluding the correct model). The rank ratio is defined as

$$\text{Rank Ratio} = \frac{Rank - 1}{p}. \tag{4}$$

This rank ratio is then transformed to a 0-100 score using a bell-shaped function:

$$rkRatio = \frac{100}{1 + \left(\frac{\text{Rank Ratio}}{a}\right)^b}, \tag{5}$$

where a and b are set differently for each biphone model to maximize the scoring performance of each model.

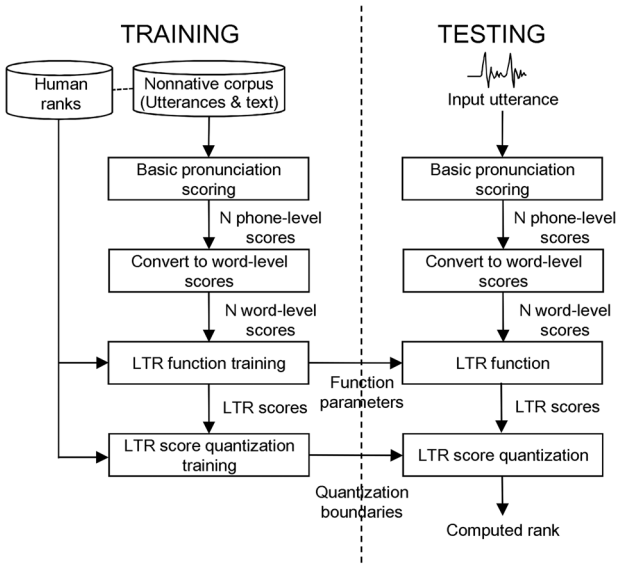

Fig. 2. Schematic diagram of the proposed score combination method for automatic pronunciation scoring.

IV. Score Combination Using Learning to Rank

In this study we propose an improved version of the score combination method based on a learning to rank framework. This method is depicted in Fig. 2. In the training phase, various phone-level basic pronunciation scores are computed from the nonnative corpus. These phone-level scores are then converted to word-level scores and are used as the features, along with the human rankings as the ground truth, to train the LTR function. The last step is to train the required quantization parameters to transform the continuous output of the LTR function, called LTR scores, to discrete ranks between 1 and 5. On the other hand, the testing phase undergoes similar steps as the training phase. The same set of phone-level basic scores are first computed and converted to word-level scores for the input utterance. The LTR function then generates an LTR score from these word-level scores based on the trained function parameters. The LTR score is then transformed into a discrete rank by the quantization step. Details of each step are described in the following subsections.

A. Basic Pronunciation Scoring

In this stage basic pronunciation scores are extracted from the input utterances. The required parameters for these basic scoring functions are estimated from the native corpus, including the duration, likelihood, and posterior probability distributions, prior probabilities, and parameters a and b for converting a rank ratio into a rank ratio score.

B. Conversion From Phone-level to Word-level Scores

This step converts the phone-level scores computed from the previous step into word-level scores. In our previous work [15], we used average-based word-level scores by averaging out all phone-based scores within a word instead of using a different weight for each phoneme based on its duration relative to the entire word [28]. This is because consonants tend to have relatively shorter durations, but they are as equally important as vowels to human listening perception. However, we still keep the original duration-weighted method in this work because forced alignment tends to be less accurate on nonnative speech by using HMMs trained from native data. Thus the forced alignment performance for some phones (such as fricatives and plosives) that tend to have a shorter duration might be unreliable. We therefore want the phones that have longer durations, which are assumed to be more stable, to contribute more towards the word-level score. In this paper, we also use another two methods for the conversion to word-level scores (vowel-based and consonant-based word-level scores) to further emphasize the effect of vowels and consonants on determining the pronunciation quality [16]. Vowel-based word-level scores are computed by averaging only phone-level scores of all vowel segments within a word; consonant-based word-level scores are computed by averaging only phone-level scores of all consonant segments within a word.

The four conversion methods (average-based, vowel-based, consonant-based, and duration-weighted) are applied to all 9 basic scores to render 36 word-level scores. The performance of these 36 scores is examined in the experimental section.

These basic scores in their original forms, called raw basic scores, can be used directly as the input to the LTR function. However, we found that some of the basic scores have wider ranges than other basic scores, and preliminary experimental results show that the performance of some score combination methods can be improved by re-scaling these basic scores. We will also discuss this issue in the experimental section.

C. Learning to Rank Function

Learning to rank [29] is originally a technique used in information retrieval (IR). It is a supervised or semi-supervised machine learning algorithm for automatically building a ranking model to rank items (or often referred to as a document in IR) under a certain query based on the given features in a ranking problem. Many algorithms have been proposed in the past decades and can be generally classified as the pointwise approach (e.g. PRank [30]), the pairwise approach (e.g. RankSVM [31], RankBoost [32], and RankNet [33]), and the listwise approach (e.g. ListNet [34]). From a discriminative learning viewpoint, these approaches define different input and output spaces, employ different hypotheses, and adopt different loss functions [29].

In the pointwise approach, a single item is used as a training instance and the output is a score or a relevance degree of that single item to the corresponding query. The hypothesis space contains functions that take the feature vector of an item as the input, and output the predicted score or relevance degree. This function is generally regarded as a scoring function. The loss function measures the difference of the predicted score to the ground truth score for each item. The final ranked list can be generated by sorting the output scores of all items.

In the pairwise approach, a pair of items is used as a training instance and the output is a binary-valued pairwise preference, indicating whether or not the first item is preferable to the second item for the corresponding query. In addition to the binary-valued pairwise preference, Zhou *et al.* [35] introduced an extra tie preference relation to make use of the tie pairs which are skipped in most of the pairwise algorithms. The hypothesis

space contains functions that operate on the feature vector of a pair of items and output the predicted preference of that item pair. The loss function measures inconsistency between the predicted preference and the ground truth preference. The final ranking list can be reconstructed either by using a greedy algorithm that aims to satisfy as many ordered pairs as possible [36] or simply by sorting the scores which are generated by the hypothesis function for computing the pairwise preference.

In the listwise approach, a list of items is used as a training instance. The output may be either a ranked list itself (no additional computation is required to generate the final ranking list) or relevance degrees (sorting is required to generate a ranking list). The hypothesis function is a multivariate function, often implemented as a scoring function that operates on a group of items and outputs their permutation or predicted relevance degrees. The loss function is defined on the approximation or bound of some IR evaluation measures (if the output is relevance degrees) or measures the difference between the predicted ranked list and the ground truth ranked list (if the output is a ranked list).

Note that, although the ground truth data are different in all three approaches (relevance degrees, pairwise preferences, or ranked lists), they can be converted easily from one type to another. For instance, pairwise preference data can be obtained from the relevance degree data by pairing two samples in different permutations.

Since we only focus on validating the effectiveness of the learning to rank framework as a score combination method, we do not use any human-labeled pairwise or listwise scoring data. The required pairwise and listwise labels for the training data are generated from the human ranking data (as described in Section II-C).

Similar to the pointwise approach in learning to rank, most of the existing score combination methods for pronunciation scoring adopt loss functions that measure the difference between the predicted score and the ground truth score of each single utterance without considering the relative order among different utterances. We therefore propose the use of pairwise and listwise learning to rank algorithms to combine various basic pronunciation scores. The advantage of these algorithms is that their loss functions naturally take the relative order among utterances into account: the pairwise approach considers the relative order between two items and the listwise approach considers the relative order among a list of items. Two learning to rank algorithms, RankSVM and ListNet are examined in this study.

Before explaining what RankSVM and ListNet do, we first state the ranking problem as follows. Suppose there are m queries $\{q^{(1)}, \cdots, q^{(m)}\}$ in the training set, and there are $n^{(i)}$ items $\{d_1^{(i)}, \cdots, d_{n^{(i)}}^{(i)}\}$ associated with the ith query $q^{(i)}$. A feature vector $x_j^{(i)}$ can be extracted from the query-item pair $(q^{(i)}, d_j^{(i)})$, and this item $d_j^{(i)}$ has a ground truth judgment or score $y_j^{(i)}$ with respect to the query $q^{(i)}$. The goal of the ranking problem is to find a ranking function f such that the predicted ranking order, based on the computed score $z_j^{(i)} = f(x_j^{(i)})$ of each item $d_j^{(i)}$, is as close to the ground truth ranking order (based on $y_j^{(i)}$ of each item) as possible.

In our pronunciation scoring problem, we assume that all utterances are associated with a single query ($m = 1$) so that the ground truth scores for all utterances can be compared with one another. The formulation in the rest of this subsection is presented in the context of pronunciation scoring, i.e. there are n utterances associated with a single query. Each utterance d_j has a feature vector x_j and ground truth scores y_j.

RankSVM [31] is a pairwise learning to rank algorithm that uses SVM (support vector machine) [37] to make a binary decision on the preference of a pair of utterances. RankSVM has exactly the same objective function as SVM but is subjective to different constraints. RankSVM's constraints are constructed from utterance pairs. The loss function for RankSVM is a hinge loss that is also defined on utterance pairs. We choose RankSVM for the pairwise approach because of its various advantages inherited from SVM, such as good generalization with margin maximization and kernelized methods for handing nonlinear problems.

ListNet [34] is a listwise learning to rank algorithm. A simple linear neural network shown in Eq. (6) is used as the scoring function.

$$z_j = f(x_j) = w^T x_j. \tag{6}$$

The probability of having a certain permutation π of given scores s of n utterances can then be computed. However, the number of permutations is of order $O(n!)$, making it impractical to compute the probabilities for all possible permutations. The number of permutations can be greatly reduced by computing the top-k probability of k out of n utterances in the order of $\{d_{j_1}, d_{j_2}, \cdots, d_{j_k}\}$ (j_k is the index of the item that is ranked at position k), i.e. the probability of these k utterances being ranked on the top k positions in the given order. By assuming $k = 1$ as in [34], the top-1 probability of n utterances simplifies to the following form:

$$P(s_{j_1}) = \frac{\exp(s_{j_1})}{\sum_{t=1}^{n} \exp(s_{j_t})}, \tag{7}$$

where s_{j_1} is the score of the utterance that is ranked on the first position. By computing $P(s_j)$ for all utterances, we can obtain a probability distribution of the scores s. Thus, for our pronunciation scoring problem, we can have a probability distribution of ground truth judgments $P(y_j)$ and a probability distribution of computed scores $P(z_j)$ for $j = 1, 2, \ldots, n$. Cross entropy is then adopted as the listwise loss function to measure the difference between the two distributions:

$$L(y, z) = -\sum_{j=1}^{n} P(y_j) \log P(z_j). \tag{8}$$

Next, gradient descent is used as the optimization algorithm to minimize this loss function. After differentiating Eq. (8), we can obtain the gradient of $L(y, z)$ with respect to parameter w as:

$$\Delta w = \sum_{j=1}^{n} \left(P(z_j) - P(y_j)\right) x_j. \tag{9}$$

Thus, the parameter w can be updated iteratively by $\hat{w} = w - \eta \Delta w$, where η is the learning rate of gradient descent. The

learning process continues through several iterations until a certain condition is met, say the change in loss is below a threshold or a given number of iterations are reached.

For a given new utterance d', the score z', called the LTR score in Fig. 2, can be computed using Eq. (6) with the trained parameter w and the utterance's feature vectors x'. Detailed derivation and explanation of the ListNet algorithm can be found in the literature [19]–[38].

D. LTR Score Quantization

There are two reasons for adopting this score quantization technique. First, since the objective of the LTR function is to generate scores that maintain the ranking order as close to the correct order as possible, the LTR scores are not guaranteed to fall into a specific numerical range. What is guaranteed is that the input samples with the same estimated rank have similar LTR scores, i.e. the input samples are orderly clustered based on their estimated ranks. We use this characteristic to transform the LTR scores to discrete ranks by applying score quantization, i.e. finding the boundary between each pair of adjacent clusters (ranks) from the training data. Second, in our experiment, we found that some basic scoring methods, e.g. *hmmLPst*, generate scores that mostly stay in a certain range but occasionally deviate significantly from this range. These few super low (or super high) scores degrade the performance in terms of the correlation coefficient. When we apply our score quantization technique to these basic scores, the super low scores become 1 while the super high scores become 5.

Chen and Jang [15] proposed a DP-based method to find the optimal quantization boundaries to minimize the discrepancy between human and computer rankings. Let $s = [s_1, s_2, \ldots, s_n]$ be the LTR scores and $r = [r_1, r_2, \ldots, r_n]$ be the corresponding human rankings with values between 1 and m. Without loss of generality, we assume the elements of vector s are sorted in an increasing order. Our goal is to find $m-1$ boundaries $\theta = [\theta_1, \theta_2, \cdots, \theta_{m-1}]$ to map the original LTR scores to ranking. Specifically, the mapping function is defined as

$$\mathrm{s2r}\left(s,\theta\right) = \begin{cases} 1, if s \le \theta_1 \\ k,\ if\ \theta_{k-1} < s \le \theta_k, where 2 \le k \le m-1 \\ m, if \theta_{m-1} < s \end{cases}. \tag{10}$$

We need to find θ such that, after mapping, the rankings can be as close as possible to those labeled by humans. We can then define the objective function as follows:

$$J\left(\theta\right) = \sum_{i=1}^{n} |r_i - s2r\left(s_i\right)|, \tag{11}$$

where r_i is the desired rank while $s2r(s_i)$ is the computed rank. By minimizing $J(\theta)$, the computed rank can be made as close as possible to the human rank. To deal with such a problem in a DP framework, we first need to define the optimum-valued function $D(i,j)$ representing the minimum cost of mapping $[s_1, s_2, \cdots, s_i]$ $(i \le n)$ to a rank range of $[1, 2, \cdots, j]$ $(j \le m)$. We can then come up with the recurrent equation for $D(i,j)$ as follows:

$$D(i,j) = |r_i - j| + \min\left\{D(i-1,j), D(i-1,j-1)\right\}, \tag{12}$$

where $i \in [1,n]$ and $j \in [1,m]$. The initial conditions are

$$D(1,j) = |r_1 - j|, j \in [1,m]. \tag{13}$$

The optimum cost is equal to $D(n,m)$. As a common practice in DP, after $D(n,m)$ is found, we can backtrack to find the optimum path together with the optimum values of θ.

The above algorithm works well in most of the cases. However, in this study where the data distribution among all classes is highly imbalanced–over 55% of utterances are scored as 5 while only less than 10% are scored as either 1 or 2–preliminary experimental results show that the current algorithm strongly favors the larger class, i.e. the class with more training data. As a result, boundaries around the smaller classes are poorly estimated while the boundaries around the larger class are excessively widened to cover almost all data. This phenomenon yields a good recognition rate, since most of the data belongs to class 4 or 5, and yet a poor correlation coefficient is obtained.

Here we propose a class-normalized DP-based quantization algorithm to alleviate this problem. A normalizing term is added to Eq. (11) as follows.

$$J\left(\theta\right) = \sum_{i=1}^{n} \frac{1}{N_{r_i}^{\alpha}} |r_i - s2r\left(s_i\right)|, \tag{14}$$

where N_{r_i} represents the number of data points in class r_i, and $\alpha \in [0,1]$ is the parameter controlling the degree of normalization. The corresponding recurrent equation for DP can be reformulated accordingly. If $\alpha = 0$, this normalizing term becomes 1 and the algorithm is exactly the same as the original one where each data point contributes equally and the larger class dominates the learning process. On the other hand, if $\alpha = 1$, the dominating effect of the larger class is suppressed and each class contributes equally to the learning process. In the experimental section we examine the effect of different settings for α on quantization performance.

V. Experimental Results

In this section, we start from evaluating the performance of each basic pronunciation score using four phone-level to word-level score conversion methods. The best basic scoring method serves as the baseline for the score combination methods. We also test the effectiveness of class-normalized DP-based score quantization on the best performing basic score under different parameter settings. The best performing score quantization setting is then applied to all basic scores. Lastly, the performance of the proposed method is compared with that of the baseline as well as three existing score combination methods proposed in [17] and [18]. Only the better performing basic scores (either quantized or in their raw forms) are used for score combination.

In the experiment, utterances from 20 speakers are used as the training set while utterances from the remaining five speakers are used as the test set. The training set is used for basic score selection and parameter training/tuning for both the proposed

score quantization method and all score combination methods. The test set is only used for the final performance comparison of all score combination methods. The WSJ corpus is used to train 1492 biphone HMMs for speech recognition. For computing the basic pronunciation scores involving distribution estimation (duration distribution score, likelihood distribution score, and posterior distribution score), we estimate the distributions of all biphone models as well as their main monophones. Here we do not train another set of monophone models exclusively; we instead collect all samples from their biphone derivatives and estimate the distribution (e.g. for estimating the distribution for monophone /a/, we collect all samples from its biphone derivatives $/\mathrm{a}+\mathrm{b}/$, $/\mathrm{a}+\mathrm{c}/$, $/\mathrm{a}+\mathrm{d}/$, … and so on). If an unseen biphone model is found (i.e. a biphone model that exists in the corpus that is used to train HMMs but does not exist in the part of the corpus that is used to estimate the distributions) we roll back the distribution of the biphone model to its monophone version. The monophone distribution is also used instead of the biphone one when the variance of the distribution is zero (i.e. a biphone model that only has one sample or has multiple samples with the same value). This roll back strategy avoids unstable statistics for the rarely seen biphone models.

In terms of measuring the closeness of the computed scores to the human scores, we use the correlation coefficient as the performance measure. This approach has been adopted in most of previous research on automatic pronunciation scoring [14]–[17]. The reason is that users of a CAPT system would accept frequent slightly inaccurate scoring results (higher correlation) rather than occasional extremely inaccurate scoring results (higher recognition rate). However, since the proposed class-normalized quantization is based on a criterion to minimize the difference in computed and human scores, we will also show the change in class recognition rate for the analysis purpose (Section V-B). In this study, the correlation computed using the scores (either basic or combined) in their raw forms is denoted as raw correlation (raw corr) and the correlation computed using the scores in the quantized form is denoted as rank correlation (rank corr).

A. Evaluation of Basic Scores using Different Conversion Methods

Fig. 3 shows the performance of the 9 basic pronunciation scores using different phone-level to word-level conversion methods described in section IV-B. This experiment is performed only on the training set. From this figure, we can observe the following issues.

First, the result shows that *rkRatio*, *hmmLPst*, *hmmPost*, *lpstDist*, and *segClass* are generally better basic scoring methods for our dataset, while *hmmLike* and *likeDist* show extremely unstable performance. This is similar to the result shown in [13], i.e. *hmmLike* is not as stable as other scoring methods.

Second, the average-based method performs better than the other three conversion methods. This result is reasonable since the influence of a phoneme of a word towards the overall pronunciation quality is not always directly proportional to its duration. Therefore, a duration-weighted method might underestimate some shorter phonemes such as fricatives and plosives. It is also clear that vowel-based or consonant-based basic scores alone yield poorer performance when they are compared with

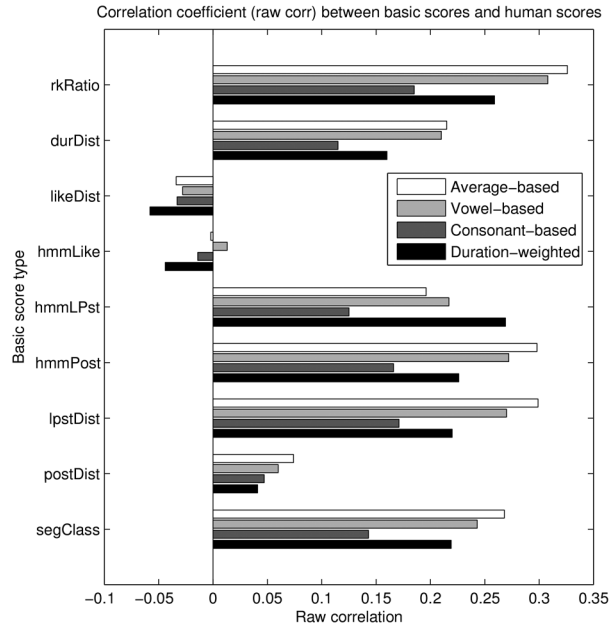

Fig. 3. Correlation coefficient (raw corr) between basic scores and human scores with different phone-level to word-level score conversion methods.

TABLE IV
SCORE DISTRIBUTION OF THE BASIC SCORE *HMMLPST* (AVERAGE-BASED)

Data range	Frequency	Percentage
-56 < *hmmLPst* ≤ -10	11	0.7%
-10 < *hmmLPst* ≤ -8	11	0.7%
-8 < *hmmLPst* ≤ -6	20	1.3%
-6 < *hmmLPst* ≤ -4	85	5.3%
-4 < *hmmLPst* ≤ -2	398	24.9%
-2 < *hmmLPst* ≤ 0	1075	67.2%

the corresponding average-based basic scores. This is also expected since the pronunciation quality cannot be judged solely based on either vowels or consonants, as shown in the tight/sight example. However, [16] shows that incorporating both vowel-based and consonant-based basic scores can enhance the performance of score combination, thus we still keep these scores for score combination.

Third, *hmmLPst* (the one with log) performs worse than *hmmPost* (the one without log). This is probably because *hmmPost* does not have a skewed distribution as *hmmLPst* does. This skewed distribution of *hmmLPst* amplifies the difference between the computed scores and the ground truth scores for computing the correlation coefficient, especially for those scores that seriously deviate from the rest. Table IV shows the score distribution of averaged-based *hmmLPst* on the training set. Over 92% of the scores are concentrated in the range between -4 and 0 while less than 3% are in the range between -56 and -6. On the other hand, *hmmPost* is always in the range between 0 and 1 and does not have this problem. However, when it comes to the distribution score, the score with log (*lpstDist*) performs significantly better than the score without log (*postDist*). We assume that the scores with log are still a better representation of human judgment than the scores without log, but that the skewed distribution of *hmmLPst* is a disadvantage in computing the correlation coefficient. We will justify this assumption in the next experiment.

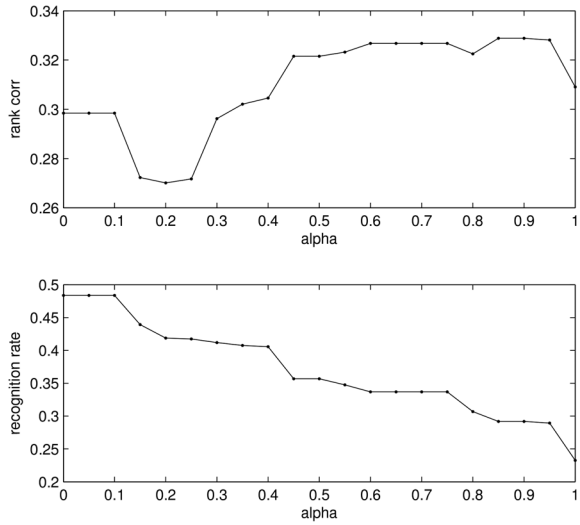

Fig. 4. Performance comparison on quantizing average-based *rkRatio* with different α values (raw correlation is 0.326).

Finally, *durDist* is shown to be somewhat effective as demonstrated in [11] except for the consonant-based *durDist*, possibly due to the unreliable performance of the forced alignment of short consonant segments (particularly obstruents, such as /t/ and /s/).

B. Evaluation of Class-Normalized Score Quantization

In this experiment we examine the effectiveness of applying the proposed score quantization technique to basic scores. For simplicity, we apply score quantization, under different settings of α, only to the best performing basic score (i.e. averaged-based *rkRatio*) on the training data to find the best α value. This α value is then applied to quantize all other basic scores.

Fig. 4 shows the result of finding the best α. The top figure shows the change in rank correlation with different values of α, while the bottom figure shows the change in score recognition rate (by treating scores 1 to 5 as five different classes). It can be seen that the rank correlation generally increases when α increases from 0.2, but not very stably. This is probably because the optimization on score quantization is based on minimizing the difference between computed scores and ground truth scores and is not directly based on correlation (efficient optimization directly based on correlation is difficult, since the sample mean is unknown during the DP computation). However, by looking at the change in the score recognition rate, the influence of class-normalized score quantization is obvious. When α increases, the smaller classes contribute more towards the learning process while larger classes receive less attention, hence the decrease in recognition rate. Although more elements are misclassified after class normalization, the misclassified elements are closer to their ground truth values and thus the correlation is higher.

As shown in Fig. 4, the correlation reaches its highest point when α is 0.85 and 0.9. We use $\alpha = 0.85$ to quantize all other basic scores and also for the LTR score quantization stage.

Table V shows the change in correlation after quantizing all basic scores with $\alpha = 0.85$. The top row shows the raw correlation (basic score before quantization) while the bottom row shows the rank correlation (basic score after quantization). Most of the basic scores improve after quantization, especially for *hmmLPst*. The improvement confirms that quantization can alleviate the skewed distribution problem. It is also clear that, after quantization, the scores related to log-posterior probability (*hmmLPst* and *lpstDist*) perform better than their counterpart scores without taking logarithm (*hmmPost* and *postDist*).

TABLE V
CORRELATION COEFFICIENTS BETWEEN BASIC SCORES AND HUMAN SCORES. THE TOP ROW SHOWS RAW CORRELATION (BASIC SCORE BEFORE QUANTIZATION) WHILE THE BOTTOM ROW SHOWS RANK CORRELATION (BASIC SCORE AFTER QUANTIZATION). GRAY REGIONS SHOW THE BASIC SCORES SELECTED FOR SCORE COMBINATION

	Average Based	Vowel based	Consonant based	Duration Weighted
rkRatio	0.326	0.308	0.185	0.259
	0.329	0.299	0.205	0.277
durDist	0.215	0.210	0.115	0.160
	0.224	0.202	0.139	0.173
likeDist	-0.034	-0.028	-0.033	-0.058
	0.012	0.015	0.031	0.027
hmmLike	-0.002	0.013	-0.014	-0.044
	0.069	0.050	0.069	0.068
hmmLPst	0.196	0.217	0.125	0.269
	0.340	0.310	0.228	0.317
hmmPost	0.298	0.272	0.166	0.226
	0.304	0.266	0.181	0.225
lpstDist	0.299	0.270	0.171	0.220
	0.295	0.270	0.175	0.245
postDist	0.074	0.060	0.047	0.041
	0.080	0.119	0.067	0.054
segClass	0.268	0.243	0.143	0.219
	0.243	0.230	0.149	0.229

C. Performance Comparison of Various Score Combination Methods

Based on the performance shown in Table V, we select only the 16 basic scores for which raw or rank correlations are greater than 0.2 for further score combination. If the scores both before and after quantization show a correlation greater than 0.2, the higher one is selected. Also, the log-posterior related scores (*hmmLPst* and *lpstDist*) perform better than the corresponding scores without taking logarithm (*hmmPost* and *postDist*) so that such scores are not selected here. The selected basic scores are shown as the gray regions in Table V.

To evaluate the effectiveness of the proposed method, we compare it with the best performing basic score, average-based *rkRatio*, as well as three other existing score combination methods described in section I: *GCE* (Gaussian classifiers with expectation, the first method proposed in [18]), *LCA* (linear classifiers with score adjustment, the second method proposed in [18]), and *NN* (neural network, the best performing method proposed in [17]). The basic score *rkRatio* serves as the baseline for all score combination methods. *GCE* estimates the Gaussian distributions from the training data and the expectation is computed based on the likelihood of five Gaussian distributions. As for the *LCA* approach, different degrees ($2 \sim 6$) of multiplicative polynomial transformation are tested and it turns out that degree of 6 performs the best. As for the *NN* approach, we adopt the same architecture as suggested in [17]: a two-layer perceptron with a single linear output layer and $4 \sim 32$ sigmoidal units for the hidden layer. The training algorithm is backpropagation with minimum mean square error criterion. The best performing number of sigmoidal units (4 in

this study) is chosen for comparison. Since these three score combination methods learn from the training data to produce combined scores that are as close to their ground truth scores as possible (unlike the learning to rank methods that learn to produce combined scores for preserving the relative order in magnitude), we simply round off the raw basic scores to obtain quantized scores. For comparison purposes, we also try quantizing the raw predicted scores of these three methods using the proposed class-normalized score quantization method (denoted as $GCE + cnsq$, $LCA + cnsq$, and $NN + cnsq$).

Since we intend to generate the scores from a ranking perspective, we also compare the proposed method with two ordinal regression models: the *ordered logit* model and the *ordered probit* model [39]. Both methods, similar to the pointwise approach in learning to rank, estimate the predicted probability of each category (score 1-5) for an input utterance, and the category with the highest predicted probability is chosen as the computed rank. Since their output is a discrete rank, raw correlation is not computed for these two methods.

For the proposed method, we use two learning to rank algorithms: *RankSVM* and *ListNet*. For *RankSVM*, we used the implementation provided by Joachims [31]. Different values for the regularization parameter (i.e. the "C" multiplier that controls the trade-off between training error and margin) with a linear kernel are tested and the best performing parameter ($\mathrm{C} = 0.5743$) is chosen for comparison. For *ListNet*, different settings of the loss change threshold and learning rate are tested by grid search and the best performing setting (1.5 learning rate, 10^{-9} loss change and 475 max iteration) is chosen for comparison.

One issue we found during the preliminary experiment is that *ListNet* is very susceptible to the numerical range of each dimension of the feature. Its performance is significantly lowered when the selected 16 basic scores are used directly. Among these 16 basic scores, the six raw basic scores (shown as the shaded top-row scores in Table IV) range between 0 and 1, while the other 10 quantized basic scores (shown as the shaded bottom-row scores in Table IV) are between 1 and 5. We simply divide the quantized basic scores by 5 to make all 16 basic scores in the same range (i.e. 0 to 1). *ListNet* performs the best when all basic scores are in the same range. On the other hand, the other four score combination methods perform exactly the same or with only very subtle differences regardless of whether the ranges of all basic scores are the same. The following experiment presents results for all basic scores in the same range.

Given that the dataset is imbalanced (i.e., most utterances are rated as 4 and 5) and correlation alone might not completely reflect performance, we further analyze performance in terms of mispronunciation detection, as this better fits the goal of guiding L2 learners to speak intelligibly. Based on the scoring guideline shown in Table I, we label scores of 4 and 5 as "intelligible" and score of $1 \sim 3$ as "mispronounced" for both human scores and computed quantized scores. Performance is measured in terms of pFA (probability of false alarm, i.e., the proportion of actual intelligible pronunciation which is incorrectly identified as mispronunciation) and pMiss (probability of miss or miss rate, i.e., the proportion of actual mispronounced utterances which is incorrectly identified as intelligible pronunciation). Given a learning scenario in which an L2 student is asked to practice the pronunciation of a word until he or she succeeds in producing a correct pronunciation, it is more important to correctly detect mispronounced utterances, i.e. pMiss is more important than pFA.

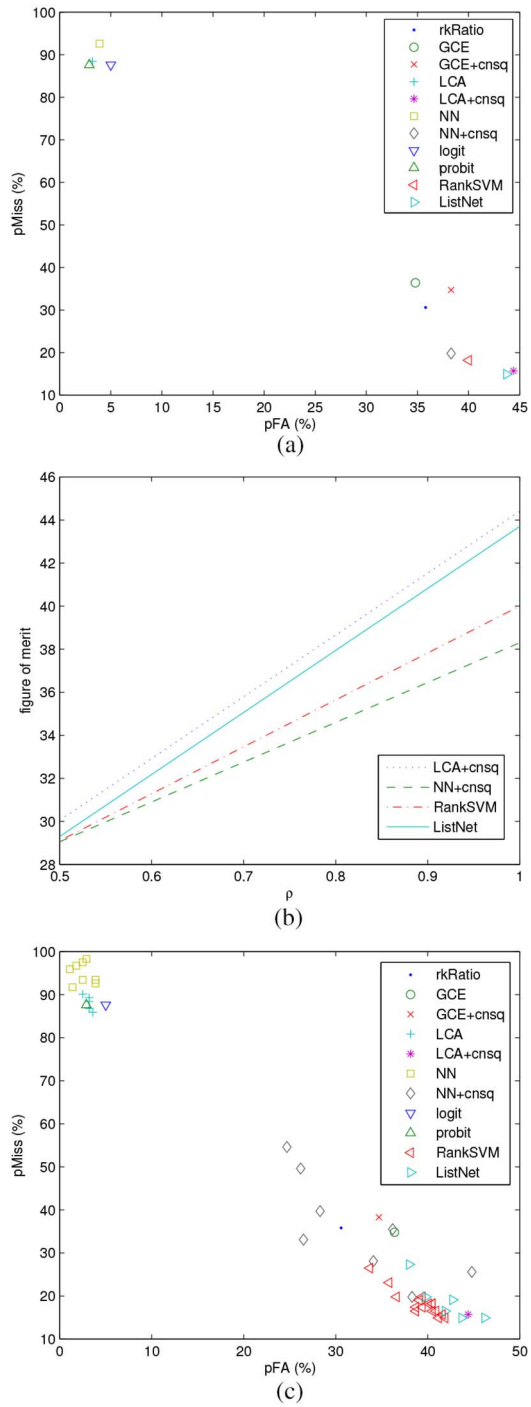

Fig. 5. Performance comparison for mispronunciation detection in terms of pFA and pMiss. (a) plot of all methods with trained parameter set, (b) Figure of merit for the four best performing methods, (c) DET plot of all methods with different parameter settings.

The final performance comparison is shown in Table VI and Fig. 5. Fig. 5(a) shows the DET plot of all methods with parameters trained on the training data. Since it is not obvious to identify which method performs the best among the four best

TABLE VI
OVERALL PERFORMANCE COMPARISON. GRAY REGIONS INDICATE THE BEST VALUE OF EACH PERFORMANCE MEASURE

Method	raw corr.	rank corr.	pFA	pMiss
Averaged-based rkRatio	0.329	0.327	35.8%	30.6%
GCE	0.273	0.264	34.8%	36.4%
GCE+cnsq		0.286	38.3%	34.7%
LCA	0.391	0.335	3.2%	88.4%
LCA+cnsq		0.355	44.4%	15.7%
NN	0.378	0.295	3.9%	92.6%
NN+cnsq		0.349	38.3%	19.8%
Ordered Logit	n/a	0.216	5.0%	87.6%
Ordered Probit	n/a	0.247	2.9%	87.6%
RankSVM (proposed)	0.397	0.369	40.0%	18.2%
ListNet (proposed)	0.397	0.402	43.7%	14.9%

performing methods ($LCA + cnsq$, $NN + cnsq$, *RankSVM*, and *ListNet*) in Fig. 5(a), we further compare their performance, as shown in Fig. 5(b), by using figure of merit (FoM) which is defined as

$$FoM = \rho \cdot (1 - pMiss) + (1 - \rho) \cdot (1 - pFA), \qquad (15)$$

where ρ is a value between 0 and 1, representing the importance of pMiss over pFA. Since pMiss is more important than pFA, we only show the section where $\rho > 0.5$. In addition, we only show the four methods that clearly outperform the other seven methods in Fig. 5(b) for the sake of simplicity.

In order to show the performance of all methods across their full operating range, we also include the DET plot of all methods with various parameter settings in Fig. 5(c). Note that these scoring methods are designed for predicting scores of 1 to 5 so that, unlike any binary classification problem, it does not have a discrimination threshold. Hence we only show the discrete points representing the pFA and pMiss of all methods with various parameter settings. Also note that average-based *rkRatio*, *GCE*, $GCE + cnsq$, *ordered logit*, and *ordered probit* do not have any tunable parameters so that each of the five methods only has one point in Fig. 5(c). The LCA + cnsq method using different degrees of multiplicative polynomial transformation yields the same pFA and pMiss so that it also has only one point in Fig. 5(c). All the parameter training and tuning described above is performed on the training data, and the performance shown in Table VI and Figure 5 uses the test data. This explains why the performance of $NN+cnsq$ and *RankSVM* in Fig. 5(a) is not the best one in Fig. 5(c). Experimental observations are as follows.

1) The *GCE* approaches (*GCE* and $GCE + cnsq$) do not outperform the basic score average-based *rkRatio* in terms of both raw correlation and rank correlation, as shown in Table VI. This might be due to the distribution of our data being not Gaussian-like, thus producing poor distribution estimates. Another potential reason is that the GCE approaches are less flexible as they only have a few parameters to define the prediction process (i.e., only one mean and one variance for each Gaussian distribution).
2) Without using the proposed quantization method, *LCA* and *NN* suffer from the problem of imbalanced data–most of the recognized utterances produce scores of 4 or 5, thus yielding a low pFA and high pMiss. The proposed score quantization method alleviates this problem and the pMiss values of both methods ($LCA + cnsq$ and $NN + cnsq$) drop below 20%, as shown in Table VI. The improvement is also shown in terms of correlation.
3) The two ordinal regression methods, *ordered logit* and *ordered probit*, perform the worst among all methods, as shown in Table VI, since these pointwise methods do not consider the relative order between utterances in the learning process [29]. On the other hand, low pFA and high pMiss shows that these methods are also affected by the imbalanced distribution of the dataset.
4) The proposed methods using *RankSVM* and *ListNet* perform the best among all methods in terms of both raw and rank correlations, as shown in Table VI. Despite the effectiveness of the learning-to-rank algorithm itself (as demonstrated by its high raw correlations), class-normalized score quantization minimizes the error in each class so that the larger classes do not dominate the learning process and thus a better rank correlation is achieved. As expected, *ListNet* performs better than *RankSVM*, since *ListNet* considers the relative order of the entire training data rather than just a single data pair at a time as in *RankSVM*. A Wilcoxon signed rank test performed on the rank output of the $LCA + cnsq$ method, the best performing previous method, and the proposed ListNet method shows that the p-value is less than the 0.05 level ($p = 0.0137$), which indicates the improvement over the previous method is statistically significant. ListNet also achieves the lowest pMiss and the best figure of merit in Fig. 5(b) for $\rho > 0.52$, showing that its performance in terms of mispronunciation detection is also satisfactory.

Also note that the required computation time for each score combination method is different. On the same computer, *GCE* and *LCA*, including testing for five different degrees multiplicative polynomial transformation, take less than one minute; while *NN* takes around 20 minutes, including experimenting with 15 different numbers of sigmoidal units; *RankSVM* takes about $2 \sim 3$ hours, including experimenting with 16 different values for the regularization parameter; both ordinal regression methods take a few seconds; and *ListNet* also takes about a few seconds for each trial of the different parameter sets, and a total of around $15 \sim 20$ trials were performed to find the best parameter set by grid search. All experiments were conducted on a laptop with an Intel Core i5 CPU (2.27 GHz) and 8 GB of RAM.

VI. CONCLUSION

This paper proposes an automatic pronunciation scoring framework with score combination based on learning to rank and class-normalized DP-based quantization. The key findings are as follows.

1) We examine nine basic pronunciation scoring methods with four methods to convert phone-level scores to word-level scores. Experimental results show that average-based scores outperform the other three conversion methods.

2) Applying the proposed class normalization technique to score quantization alleviates the problem of imbalanced data over different classes.
3) Applying this class-normalized score quantization to some of the basic scores can improve their correlation to human rankings.
4) The proposed learning-to-rank framework, particularly the ListNet method, achieves a better correlation to human rankings and a better accuracy in detecting mispronunciations than other score combination methods, proving the effectiveness of the proposed framework.

A number of future research directions are identified as follows. Firstly, the basic scores used in this research are based on the likelihood and duration of the phoneme segment. Acoustic features such as stress information (for an English word) and intonation (for a sentence) are not yet considered in this study. However, our dataset consists of the pronunciation of a single English word in each utterance. Preliminary error analysis shows that some utterances have good pronunciation but with incorrect stress placement. This is currently not captured by our system but shall be considered in the grading process. Secondly, different learning to rank approaches can be tested instead of RankSVM. According to [29], RankSVM has the best performance of all pairwise algorithms, but at a price of increased computation complexity. Other pairwise algorithms such as RankBoost or RankNet can be tested to obtain a balance between the computation cost and performance. On the other hand, we can also adopt the tie data to best make use of the limited training data. Thirdly, we currently use the human ranking data (1-5 scale) to simulate the pairwise data by making combinations of each possible pair of the available data. In real situations, however, scoring directly in a pairwise manner poses a data sparsity problem since the number of data pairs is now quadratic to the data size. It would be impractical to label all pairs and we have to work with sparse data. A possible solution is to use a greedy algorithm to approximate the global order from partial pairwise comparisons [19][36].

REFERENCES

[1] A. Neri, C. Cucchiarini, and H. Strik, "Segmental errors in dutch as a second language: How to establish priorities for CAPT," in *Proc. InSTIL/ICALL Symp.*, Venice, Italy, Jun. 2004, pp. 13–16.

[2] A. Neri, C. Cucchiarini, and H. Strik, "ASR-based corrective feedback on pronunciation: Does it really work?," in *Proc. Interspeech*, Pittsburgh, PA, Sep. 2006, pp. 1982–1985.

[3] H. Wang, C. J. Waple, and T. Kawahara, "Computer assisted language learning system based on dynamic question generation and error prediction for automatic speech recognition," *Speech Commun.*, vol. 51, pp. 995–1005, Oct. 2009.

[4] O. Ronen, L. Neumeyer, and H. Franco, "Automatic detection of mispronunciation for language instruction," in *Proc. 5th Eur. Conf. Speech Commun. Technol. (Eurospeech '97)*, Rhodes, Greece, Sep. 1997, pp. 645–648.

[5] S. M. Witt and S. J. Young, "Phone-level pronunciation scoring and assessment for interactive language learning," *Speech Commun.*, vol. 30, no. 2–3, pp. 95–108, Feb. 2000.

[6] A. M. Harrison, W. Lo, X. Qian, and H. Meng, "Implementation of an extended recognition network for mispronunciation detection and diagnosis in computer-assisted pronunciation training," in *Proc. 2nd ISCA Workshop Speech Lang. Technol. Educ. (SLaTE)*, Warrickshire, U.K., Sep. 2009, pp. 45–48.

[7] Y. Tsubota, T. Kawahara, and M. Dantsuji, "Practical use of English pronunciation system for Japanese students in the CALL classroom," in *Proc. Interspeech*, Jeju Island, Korea, Oct. 2004, pp. 1689–1692.

[8] J. Tepperman and S. Narayanan, "Automatic syllable stress detection using prosodic features for pronunciation evaluation of language learners," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Philadelphia, PA, Mar. 2005, pp. 937–940.

[9] J. P. Arias, N. B. Yoma, and H. Vivanco, "Word stress assessment for computer aided language learning," in *Proc. Interspeech*, Brighton, U.K., Sep. 2009, pp. 1135–1138.

[10] M. P. Black, A. Kazemzadeh, J. Tepperman, and S. S. Narayanan, "Automatically assessing the ABCs: Verification of children's spoken letter-names and letter-sounds," *ACM Trans. Speech Lang. Process.*, vol. 7, no. 4, Aug. 2011, article 15.

[11] L. Neumeyer, H. Franco, M. Weintraub, and P. Price, "Automatic text-independent pronunciation scoring of foreign language student speech," in *Proc. Int. Conf. Spoken Lang. Process.*, Philadelphia, PA, Oct. 1996, pp. 1457–1460.

[12] L. R. Rabiner, "A tutorial on hidden Markov models and selected application in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[13] H. Franco, L. Neumeyer, Y. Kim, and O. Ronen, "Automatic pronunciation scoring for language instruction," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Munich, Germany, Apr. 1997, pp. 1471–1474.

[14] Y. Kim, H. Franco, and L. Neumeyer, "Automatic pronunciation scoring of specific phone segments for language instruction," in *Proc. 5th Eur. Conf. Speech Commun. Technol. (Eurospeech '97)*, Rhodes, Greece, Sep. 1997, pp. 649–652.

[15] L. Y. Chen and J. S. R. Jang, "Automatic pronunciation scoring using learning to rank and DP-based score segmentation," in *Proc. Interspeech*, Makuhari, Japan, Sep. 2010, pp. 761–764.

[16] L. Y. Chen and J. S. R. Jang, "Improvement in automatic pronunciation scoring using additional basic scores and learning to rank," in *Proc. Interspeech*, Portland, OR, USA, Sep. 2012.

[17] H. Franco, L. Neumeyer, V. Digalakis, and O. Ronen, "Combination of machine scores for automatic grading of pronunciation quality," *Speech Commun.*, vol. 30, no. 2–3, pp. 121–130, Feb. 2000.

[18] T. Cincared, R. Gruhn, C. Hacker, E. Nöth, and S. Nakamura, "Automatic pronunciation scoring of words and sentences independent from the non-native's first language," *Comput. Speech Lang.*, vol. 23, no. 1, pp. 65–88, Jan. 2009.

[19] Y. H. Yang and H. H. Chen, "Ranking-based emotion recognition for music organization and retrieval," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 762–774, May 2011.

[20] "MIR-Stress Dataset," Multimedia Inf. Retrieval Lab, Dept. of Comput. Sci., National Tsing Hua Univ.. Hsinchu, Taiwan [Online]. Available: http://mirlab.org/dataset/public/

[21] E. Charniak, D. Blaheta, N. Ge, K. Hall, J. Hale, and M. Johnson, *BLLIP 1987-89 WSJ Corpus Release 1*. Philadelphia, PA, USA: Linguistic Data Consortium, 2000 [Online]. Available: http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2000T43

[22] X. He and Y. Zhao, "Model complexity optimization for nonnative english speakers," in *Proc. 7th Eur. Conf. Speech Commun. Technol. (Eurospeech'01)*, Aalborg, Denmark, Sep. 2001, pp. 1461–1464.

[23] K. Hirabayashi and S. Nakagawa, "Automatic evaluation of english pronunciation by Japanese speakers using various acoustic features and pattern recognition techniques," in *Proc. Interspeech*, Makuhari, Japan, Sep. 2010, pp. 598–601.

[24] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue, "TIMIT acoustic-phonetic continuous speech corpus," Linguistic Data Consortium, Philadelphia, PA, USA, 1993 [Online]. Available: http://catalog.ldc.upenn.edu/LDC93S1

[25] J.-P. Hosom, "Automatic phoneme alignment based on acoustic-phonetic modeling," in *Proc. Interspeech*, Denver, CO, USA, Sep. 2002.

[26] M. Raab, R. Gruhn, and E. Noeth, "Non-native speech databases," in *Proc. IEEE Workshop Autom. Speech Recogn. Understand. (ASRU)*, Kyoto, Japan, Dec. 2007, pp. 413–418.

[27] J. C. Chen, J. S. R. Jang, and T. L. Tsai, "Automatic pronunciation assessment for mandarin Chinese: Approaches and system overview," *Int. J. Comput. Linguist. Chinese Lang. Process.*, vol. 12, no. 4, pp. 443–458, Dec. 2007.

[28] J. C. Chen, J. L. Lo, and J. S. R. Jang, "Computer assisted spoken English learning for Chinese in Taiwan," in *Proc. Int. Symp. Chinese Spoken Lang. Process.*, Hong Kong, Oct. 2004, pp. 337–340.

[29] T. Y. Liu, "Learning to Rank for Information Retrieval," in *Foundations and Trends in Information Retrieval*. Boston, MA, USA: Now, Mar. 2009, ch. 1, no. 3.

[30] K. Crammer and Y. Singer, "Pranking with ranking," in *Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2001, pp. 641–647.
[31] T. Joachims, "Training linear SVMs in linear time," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Disc. Data Mining*, Philadelphia, PA, USA, Aug. 2006, pp. 217–226.
[32] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer, "An efficient boosting algorithm for combining preferences," *J. Mach. Learn. Res.*, vol. 4, pp. 933–969, Nov. 2003.
[33] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, "Learning to rank using gradient descent," in *Proc. 22nd Int. Conf. Mach. Learn.*, Bonn, Germany, Aug. 2005, pp. 89–96.
[34] Z. Cao, T. Qin, T. Y. Liu, M. F. Tsai, and H. Li, "Learning to rank: From pairwise approach to listwise approach," in *Proc. 24th Int. Conf. Mach. Learn.*, Corvallis, OR, USA, Jun. 2007, pp. 129–136.
[35] K. Zhou, G. R. Xue, H. Zha, and Y. Yu, "Learning to rank with ties," in *Proc. 31st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, Singapore, Jul. 2008, pp. 275–282.
[36] W. W. Cohen, R. E. Schapire, and Y. Singer, "Learning to order things," *J. Artif. Intell. Res.*, vol. 10, pp. 243–270, May 1999.
[37] C. Cortes and V. N. Vapnik., "Support-vector networks," *J. Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.
[38] Z. Cao, T. Qin, T. Y. Liu, M. F. Tsai, and H. Li, Learning to rank: From pairwise approach to listwise approach Microsoft Corporation, Tech. Rep. MSR-TR-2007-40, Apr. 2007.
[39] J. Hardin and J. Joseph, *Generalized Linear Models and Extensions*, 2nd Ed. ed. College Station, TX, USA: Stata Press, Feb. 2007, ch. 15.

Liang-Yu Chen (S'08) was born in Taipei, Taiwan, in 1982. He received the B.Sc.(Eng) degree in electrical and information engineering (information option) in 2004 and M.Eng. degree in information engineering in 2005, both from the University of the Witwatersrand, Johannesburg, South Africa. He is a Ph.D. candidate in information systems and applications at National Tsing Hua University, Taiwan. His research interests include computer-assisted pronunciation training, speech recognition, pattern recognition, and machine learning.

Jyh-Shing Roger Jang (M'93) received his Ph.D. from the EECS Department at the University of California at Berkeley. He studied artificial networks and fuzzy logic with Prof. Lotfi Zadeh, the father of fuzzy logic. As of 2013, his seminal paper on ANFIS (adaptive neuro-fuzzy inference systems) published in 1993 has received over 6000 citations.

After obtaining his Ph.D., he joined MathWorks to coauthor the Fuzzy Logic Toolbox used with MATLAB. Since then, he has cultivated a keen interest in implementing industry-strength software for pattern recognition and computational intelligence. He was with the CS Dept. of National Tsing Hua Univ., Taiwan, from 1995 to 2012. Since August 2012, he has been with the CSIE Dept. of National Taiwan Univ., Taiwan. He has published one book on "Neuro-Fuzzy and Soft Computing," two books on MATLAB programming, and one book on JavaScript Programming. He has also maintained toolboxes of machine learning and speech/audio signal processing, and on-line tutorials on "Data Clustering and Pattern Recognition" and "Audio Signal Processing and Recognition." His research interests include machine learning and pattern recognition, with applications for speech recognition/assessment/synthesis, music analysis/retrieval, and image identification/retrieval.