



# Automatic Pronunciation Scoring using Learning to Rank and DP-based Score Segmentation

Liang-Yu Chen<sup>1</sup>, Jyh-Shing Roger Jang<sup>2</sup>

<sup>1</sup> Institute of Information Systems and Applications, National Tsing Hua University, Taiwan

<sup>2</sup> Department of Computer Science, National Tsing Hua University, Taiwan

davidson833@mirlab.org, jang@cs.nthu.edu.tw

## Abstract

This paper proposes a novel automatic pronunciation scoring framework using learning to rank. Human scores of the utterances are treated as ranks and are used as the ranking ground truths. Scores generated from various existing scoring methods are used as the features to train the learning to rank function. The output of the function is then segmented by the proposed DP-based method and hence boundaries between clusters can be used to determine the discrete computer scores. Experimental results show that the proposed framework improves upon the existing scoring methods. A non-native corpus with human ranks is also released.

**Index Terms:** automatic pronunciation scoring, computer-assisted pronunciation training, learning to rank

## 1. Introduction

With the introduction of CALL (computer-assisted language learning) systems, L2 learners (second language learners) have a new type of medium to improve their language skills without the presence of human teachers. One of the important functions of such a system, other than offering courses, is to provide feedback to the learners on their performances. This is especially essential for a CAPT (computer-assisted pronunciation training) system so that the learners are able to obtain the assessments on their pronunciations and possibly notifications on the mispronunciations.

Various scoring methods have been proposed for different tasks in CAPT applications. Witt and Young [1] proposed the GoP algorithms for detecting mispronounced phonemes in an utterance. Kim et al. [2] proposed three phone-based scoring methods: log-likelihood, log-posterior probability, and phone segment duration scores. Neumeyer et al. [3] further extended the work in [2] to five sentence-based scoring methods.

A number of researches has focused on combining different scoring methods to produce a single score for each utterance in order to acquire the benefits and to compensate the weakness of each scoring method. Franco et al. [4] explored various score combination methods such as linear regression, neural networks, and regression trees. Cincarek et al. [5] proposed the use of linear feature combination and multiplicative polynomial transformation to combine various sentence-based and word-based features.

In this paper, we propose a novel score combination method using learning to rank. By giving each utterance a score in a 1-5 scale and treating each score as a rank, not only the human raters can easily rank each utterance (rather than giving a precise score ranging from, say, 0-100) but also various existing learning to rank algorithms and tools are ready to use. We also argue that scores such as 75 or 70 in a 0-100 scale make no difference to the learners. In the proposed system, scoring methods based on [2], [3], and [5] are used as

the features and the human ranks as the ground truths to train the learning to rank (LTR) function. Since the LTR function only outputs the scores in a continuous and unknown numerical range, a segmentation stage is required to transform the continuous scores to the discrete ranks. Two methods based on k-means [6] and DP-based (dynamic programming) are also proposed for the segmentation stage.

In order to evaluate the proposed pronunciation scoring system, we have also established a non-native speech database called MIR-SD (Multimedia Information Retrieval lab, Stress Detection), a dataset that was originally designed for stress detection of multi-syllable English words and was recorded by Taiwanese speakers. This dataset, as well as the human scores, is publicly available in [7].

The rest of the paper is organized as follows. Section 2 describes the native and non-native speech corpora and the human ranks of the non-native speech corpus. Section 3 explains the proposed system in details. Section 4 reports the experimental results of our system, and section 5 concludes the paper and lists some of the future directions.

## 2. Speech corpora and human scores

### 2.1. Native corpus

The WSJ corpus [8] is used to train the biphone HMMs (hidden Markov models) [6] for speech recognition. 12 dimensions of MFCCs (Mel-frequency cepstral coefficients) as well as the energy and their first and second derivatives are used as the features. The WSJ corpus is also used to obtain the likelihood and duration statistics for each biphone model. Section 3.1 explains how this information is used in details.

### 2.2. Non-native corpus

The non-native corpus MIR-SD [7] was recorded by 22 Taiwanese speakers, consisting of 12 females and 10 males. The competence levels of the speakers are intermediate. Each speaker was responsible for recording over 200 English words, where each utterance contains only one word. Each word is a multi-syllable word selected from an English spelling contest for university students in Taiwan and appears only once in the corpus. The recording resolution is 16 bits and the sampling frequency is 16 kHz. In this research, only 50 utterances from each speaker are used, making a total of 1100 utterances.

### 2.3. Human ranks

Human ranks of the non-native corpus are required for both training the proposed system and evaluating the performance of the system. Each of the 1100 utterances was scored by two human experts on a scale of 1 (unintelligible) to 5 (native-like).

In order to evaluate the consistency between human raters, the word-based and speaker-based inter-rater correlations and

human ground truth correlations are computed and are shown in Table 1, where HR1 and HR2 denote the human rater 1 and 2 and GT denotes ground truths. The ground truth is computed as the mean of the human scores. Word-based correlations are computed using the scores for each word, and speaker-based correlations are computed by first averaging out the utterance scores for each speaker before computing the correlations.

Table 1: Consistency evaluation between human raters.

Correlation	Inter-rater	HR1-GT	HR2-GT
Word-based	0.58	0.84	0.89
Speaker-based	0.78	0.96	0.93

The correlations shown in Table 1 are comparable with most of the existing researches [2], [3], and [5]. This shows that the human raters are consistent in scoring. On the other hand, speaker-based scoring is more consistent than the word-based scoring, and this coincides with the finding in [2].

Table 2 shows the distribution of the ground truth ranks across all 1100 utterances. Most of the utterances are scored as 3 or 4, indicating most speakers have an intermediate competence level in pronunciation skill.

Table 2: Distribution of ground truth scores.

Score	1	2	3	4	5
Frequency	110	198	259	409	124
Percentage	10%	18%	24%	37%	11%

### 3. System description

The proposed pronunciation scoring system is depicted in Figure 1. In the training phase, various existing scoring methods are applied on the non-native corpus to obtain scores for each sentence. These scores are used as the features and the human ranks are used as the ground truths for training the LTR function. Since the LTR function does not output discrete ranks (1 to 5) but rather the scores that try to preserve the relative ranking order, these scores, called the LTR scores, need to be transformed into discrete ranks by a segmentation stage. The LTR scores segmentation training serves for the purpose of finding such score boundaries for each rank. The testing phase is similar to the training phase. Each sentence is first scored by various existing scoring methods. The trained LTR function then takes these scores and generates the LTR scores. Finally the LTR scores are transformed into ranks based on the trained segmentation boundaries. Details of each block are described in the following subsections.

#### 3.1. Existing scoring methods

Five scoring methods are applied in this block. The first three methods are based on the methods proposed in [2] and [3]. We also propose another two methods: likelihood distribution scores and rank ratio scores. These five scoring methods are based on forced alignment and phoneme recognition of the input utterance with HMMs trained from a large native corpus. This generates the required likelihood and duration information for the scoring methods described in the following sub-subsections. Note that these scoring methods are phone-based scores; the average of all phone scores within a word is used as the word-level score.

##### 3.1.1. Three phone-based scores

The following three scoring methods were proposed in [2] and [3] so that they are only briefly explained as follows.

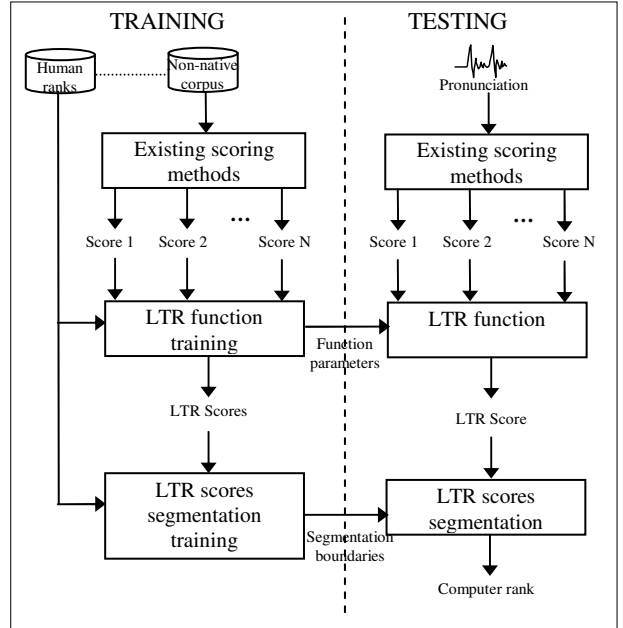


Figure 1: The automatic pronunciation scoring system

Duration distribution scores, denoted by *durDis*, refer to the likelihood of the phoneme model to have the given duration in the statistics of the native speech [2][3][5]. Firstly, the duration of a phoneme segment can be computed from forced alignment. This value is then normalized by the speech rate of the corresponding utterance, where the speech rate is defined as the number of phonemes occurring in a unit time. Lastly, the likelihood of the given duration can be computed using a log-normal distribution of the corresponding phoneme. The duration distribution of each phoneme is estimated from the WSJ corpus [8].

The HMM-based log-likelihood scores, denoted by *hmmLike*, are defined as the duration-normalized log-likelihood values of the phoneme segments [2].

The HMM-based log-posterior probability scores, denoted by *hmmPost*, are defined as the duration-normalized log-posterior probability of the phoneme segment [2], where the denominator term of the frame-based posterior probability represents the sum of prior-multiplied likelihoods of all competing models, including the correct model itself. In our experiment, we set the competing biphone models to be those models having the same right-dependent phoneme. For example, if *a*, *b*, *c*, *d*, *e* are five pseudo phonetic symbols, the biphone model */b+a/* would have the competing models */c+a/*, */d+a/*, */e+a/*, if they exist.

##### 3.1.2. Likelihood distribution scores

The likelihood distribution scores, denoted by *likeDis*, are very similar to the duration distribution scores, except that a Gaussian CDF of the log-likelihood values of the phoneme segments is used for each phoneme model instead of a Gaussian PDF. The likelihood distribution of each phoneme is estimated from the WSJ corpus [8].

##### 3.1.3. Rank ratio scores

Rank ratio scores, denoted by *rkRatio*, are based on the ranks of the correct phoneme models among all the competing phoneme models according to their likelihood values. 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:

$$\text{Rank Ratio} = \frac{\text{Rank} - 1}{p} \quad (1)$$

This rank ratio is then transformed to the 0-100 scale by using a bell function:

$$\text{rRatio} = \frac{100}{1 + \left(\frac{\text{rank ratio}}{a}\right)^b} \quad (2)$$

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

### 3.2. Learning to rank function

Learning to rank is originally a technique used in information retrieval. It is a supervised or semi-supervised machine learning algorithm for automatically constructing a ranking model, where the training data consists of lists of partially ordered items. Its objective is to generate scores for ranking and to minimize ranking errors.

Many learning to rank algorithms have been proposed and can be generally classified as the pointwise approach (e.g. Pranking [9]), the pairwise approach (e.g. RankSVM [10], RankBoost [11], and RankNet [12]), and the listwise approach (e.g. ListNet [13]). In our research, an implementation of the RankSVM algorithm developed in [10] is used.

### 3.3. LTR scores segmentation

Since the objective of the LTR function is to generate scores having a 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 the estimated ranks. We use this characteristic to transform the LTR scores to ranks by score segmentation, i.e. finding the boundary between each pair of adjacent clusters (ranks) from the training data. Two methods for score segmentation are proposed: the k-means method and the DP-based method.

The k-means [6] method iteratively finds  $k$  centers (5 in our case) in the training data and performs clustering for each sample point based on the Euclidean distance between the point and each center. Boundaries are then set as the middle of the outermost points of the adjacent clusters.

For the optimal segmentation of LTR scores, here we propose an efficient method based on DP that can guarantee to find optimal segmentation points to minimize the discrepancy between human's and computer's rankings. Let  $s = [s_1, s_2, \dots, s_n]$  be the LTR scores and  $r = [r_1, r_2, \dots, r_n]$  be the corresponding human's rankings with values between 1 and  $m$ . Without loss of generality, we shall assume the elements of vector  $s$  are sorted in an increasing order. Our goal is to find  $m-1$  boundaries  $\theta = [\theta_1, \theta_2, \dots, \theta_{m-1}]$  to map the original LTR scores to ranking. Specifically, the mapping function is defined as

$$s2r(s, \theta) = \begin{cases} 1, & \text{if } s \leq \theta_1 \\ k, & \text{if } \theta_{k-1} < s \leq \theta_k, \text{ where } 2 \leq k \leq m-1 \\ m, & \text{if } \theta_{m-1} < s \end{cases} \quad (3)$$

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

$$J(\theta) = \sum_{i=1}^n |r_i - s2r(s_i)| \quad (4)$$

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  $[s_1, s_2, \dots, s_i]$  ( $i \leq n$ ) to a rank range of  $[1, 2, \dots, j]$  ( $j \leq m$ ). We can then come up with the recurrent equation for  $D(i, j)$  as follows:

$$D(i, j) = |r_i - j| + \min\{D(i-1, j), D(i-1, j-1)\} \quad (5)$$

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

$$D(1, j) = |r_1 - j|, j \in [1, m] \quad (6)$$

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  $\theta$ .

## 4. Experimental results

To evaluate the effectiveness of the proposed system, we have conducted experiments on evaluating the existing scoring methods and the overall system performance using the best features and different segmentation methods. All experiments are carried out by speaker-wise 5-fold cross validation. The performance is measured by the correlation between computer scores and human ground truths (corr), the rank recognition rate (rRate), the rank recognition rate tolerating the confusion of  $\pm 1$  adjacent ranks (rRateT1), and the average absolute difference (AADiff). Larger values of the former three measures and smaller values of the last measure represent better performance.

### 4.1. Evaluation of the existing scoring methods

The five scoring methods described in section 3.1 are evaluated against the human ground truths. Since these scores are continuous values and the human ground truths are discrete ranks from 1 to 5, we have also conducted experiments on segmenting the continuous scores into discrete ranks by the DP-based method and the k-means method. Table 3 shows the performance of these scores.

A few points are worth noting. Firstly, by comparing the correlations, segmentation improves the effectiveness of hmmLike, hmmPost, and likeDis scores. The improvement in hmmPost is especially prominent. Secondly, we argue that correlation alone in some situations might not be a good performance measure. For example, in a 1-5 ranking scale, three utterances are ranked as 4, 5, 4 by humans and are ranked as 1, 2, 1 by the computer. This gives a correlation of 1 (which is perfect) but the computer actually ranks terribly. This phenomenon can also be observed by comparing the performance of the two outside tests of likeDis scores (as indicated by a darker gray background color). The k-means method generates a higher outside correlation than the DP-based method does, but DP-based method achieves much better rRate, rRateT1, and AADiff. Thirdly, by taking the rRate, rRate-1A, and AADiff into consideration, DP-based method clearly outperforms the k-means methods. Since the DP-based method is a supervised method while the k-means method is an unsupervised method, this observation is as expected.

Table 3: Performance evaluation of different scoring methods.

		Raw score	DP-based		k-means	
			inside	Outside	inside	outside
durDis	Corr	0.209	0.217	0.189	0.202	0.194
	rRate		0.342	0.309	0.281	0.276
	rRateT1		0.783	0.771	0.701	0.696
	AADiff		0.906	0.942	1.109	1.122
hmmLike	Corr	0.120	0.168	0.102	0.144	0.154
	rRate		0.325	0.306	0.258	0.255
	rRateT1		0.780	0.757	0.692	0.689
	AADiff		0.928	0.973	1.158	1.165
hmmPost	Corr	0.084	0.297	0.265	0.192	0.216
	rRate		0.344	0.330	0.170	0.162
	rRateT1		0.811	0.798	0.565	0.561
	AADiff		0.862	0.893	1.494	1.499
likeDis	Corr	0.141	0.160	0.125	0.141	0.143
	rRate		0.316	0.308	0.247	0.247
	rRate T1		0.789	0.774	0.665	0.671
	AADiff		0.924	0.948	1.207	1.203
rkRatio	Corr	0.240	0.232	0.198	0.229	0.236
	rRate		0.333	0.316	0.269	0.268
	rRateT1		0.789	0.779	0.699	0.698
	AADiff		0.898	0.929	1.120	1.124

#### 4.2. Performance evaluation of the system

The scores in the best form according to Table 3 (raw score and inside tests of the two segmentation methods) are chosen to be the features for the training of the LTR function (durDis and rkRatio in raw scores and DP-based segmented hmmLike, hmmPost, and likeDis scores). Linear kernel is used for the RankSVM algorithm and different SVM tolerance values are examined. The performance comparison is shown in Figure 2. The DP-based outside test results of hmmPost scores (indicated by a lighter gray background color) are chosen as the baseline in this comparison because they are the highest among all scores in Table 3.

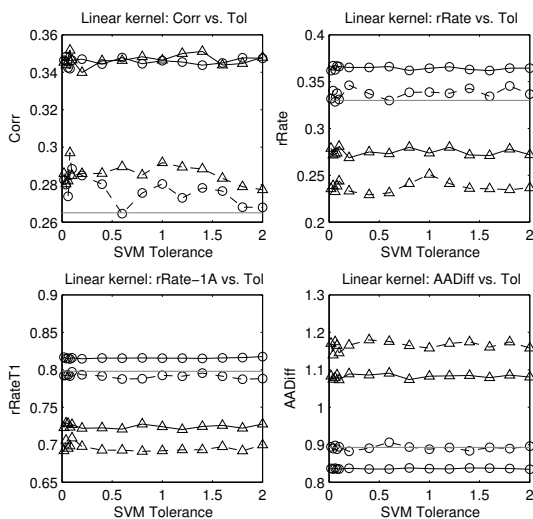


Figure 2: Overall performance comparison. Circles denote the DP-based method and triangles denote the k-means method for segmentation; Solid lines represent inside tests and dashed lines represent outside tests. The gray lines represent baselines.

It can be seen that DP-based method still outperforms the k-means method for segmentation. On the other hand, the system performance (based on outside tests of DP-based method) improves upon the baseline in corr and rRate, declines a bit in rRateT1, and almost stays the same in AADiff.

## 5. Conclusions and future work

This paper proposes a novel approach for automatic pronunciation scoring by using learning to rank as a score combination technique and a DP-based score segmentation method. Experimental results show that the DP-based method outperforms the k-means method for segmentation, and the learning to rank framework improves upon the baseline results. Future directions of research include using different learning to rank methods such as RankBoost [11] and using more scoring methods as the features for training the learning to rank function. In addition, we might change our current human ranking data to “comparison data”, because it is easier for human raters to decide whether an utterance is pronounced better than the other or not, and this is closer to the objectives of the pairwise learning to rank algorithms.

## 6. References

- [1] Witt, S. M. and Young, S. J., “Phone-level Pronunciation Scoring and Assessment for Interactive Language Learning”, *Speech Communication* 30, 95-108, 2000.
- [2] Kim, Y., Franco, H., and Neumeier, L., “Automatic Pronunciation Scoring of Specific phone Segments for Language Instruction”, in *Proceedings of the 4<sup>th</sup> European Conference on Speech Communication and Technology (Eurospeech '97)*, pp. 649-652, Rhodes, 1997.
- [3] Neumeier, L., Franco, H., Digalakis, V., and Weintraub, M., “Automatic Scoring of Pronunciation Quality”, *Speech Communication* 30, 83-93, 2000.
- [4] Franco, H., Neumeier, L., Digalakis, V., and Ronen, O., “Combination of Machine Scores for Automatic Grading of Pronunciation Quality”, *Speech Communication* 30, 121-130, 2000.
- [5] Cincared, T., Gruhn, R., Hacker, C., Nöth, E., and Nakamura, S., “Automatic Pronunciation Scoring of Words and Sentences Independent from the Non-Native’s First Language”, *Computer Speech and Language* 23, 65-88, 2009.
- [6] Rabiner, L. and Juang, B. H., “Fundamentals of Speech Recognition”, Englewood Cliffs, Prentice-Hall, NJ, 1993.
- [7] MIR-Stress Dataset, <http://mirlab.org/dataset/public/>, Multimedia Information Retrieval Lab, Department of Computer Science, National Tsing Hua University, Taiwan.
- [8] Charniak, E. et al., “BLLIP 1987-89 WSJ Corpus Release 1”, Linguistic Data Consortium, Philadelphia, <http://www ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2000T43>, assessed on 25 Apr 2010.
- [9] Crammer, K. and Singer, Y., “Pranking with Ranking”, in *proceedings of the conference on Neural Information Processing Systems (NIPS)*, 2001.
- [10] Joachims, T., “Optimizing Search Engines using Clickthrough Data”, in *proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*, ACM, 2002.
- [11] Freund, Y., Iyer, R., Schapire, R. E., and Singer, Y., “An Efficient Boosting Algorithm for Combining Preferences”, in *proceedings of ICML*, pp.170-178, 1998.
- [12] Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G., “Learning to Rank using Gradient Descent”, in *proceedings of ICML*, pp. 89-96, 2005.
- [13] Cao, Z., Qin, T., Liu, T. Y., Tsai, M. F., and Li, H., “Learning to Rank: From Pairwise Approach to Listwise Approach”, in *proceedings of the 24<sup>th</sup> International Conference on Machine Learning*, pp. 129-136, Corvallis, OR, 2007.