A Recursive Dialogue Game for Personalized Computer-Aided Pronunciation Training

Pei-hao Su, Chuan-hsun Wu, and Lin-shan Lee, *Fellow, IEEE*

***Abstract*—Learning languages in addition to the native language is very important for all people in the globalized world today, and computer-aided pronunciation training (CAPT) is attractive since the software can be used anywhere at any time, and repeated as many times as desired. In this paper, we introduce the immersive interaction scenario offered by spoken dialogues to CAPT by proposing a recursive dialogue game to make CAPT personalized. A number of tree-structured sub-dialogues are linked sequentially and recursively as the script for the game. The system policy at each dialogue turn is to select in real-time along the dialogue the best training sentence for each specific individual learner within the dialogue script, considering the learner's learning status and the future possible dialogue paths in the script, such that the learner can have the scores for all pronunciation units considered reaching a predefined standard in a minimum number of turns. The purpose here is that those pronunciation units poorly produced by the specific learner can be offered with more practice opportunities in the future sentences along the dialogue, which enables the learner to improve the pronunciation without having to repeat the same training sentences many times. This makes the learning process for each learner completely personalized. The dialogue policy is modeled by Markov decision process (MDP) with high-dimensional continuous state space, and trained with fitted value iteration using a huge number of simulated learners. These simulated leaners have the behavior similar to real learners, and were generated from a corpus of real learner data. The experiments demonstrated very promising results and a real cloud-based system is also successfully implemented.**

***Index Terms*—Computer-aided pronunciation training (CAPT), computer-assisted language learning, dialogue game, Markov decision process, reinforcement learning.**

I. Introduction

In the world of globalization today, the geographical distance is no longer a barrier for the communication between people, but instead different languages and cultural backgrounds appear to be. This leads to a fast growth in the demand on second language acquisition in recent years. Traditional in-classroom lessons are useful, but one-to-one tutoring offers a much more effective language learning environment despite its high cost. Computer-assisted language learning (CALL) becomes very attractive as speech and language processing technologies advance. Although computers cannot serve as good as human tutors, softwares can be easily spread and repeatedly used as desired by learners anywhere at any time.

Correct pronunciation is usually the first and most important issue in language learning. Computer-aided pronunciation training (CAPT) is thus an important sub-area of CALL, including automatic pronunciation evaluation which offers numerical feedback, descriptive judgments or corrective suggestions. Goodness-of-Pronunciation (GOP) proposed by Witt and Young [1] is a good example of posterior probability based pronunciation evaluation score derived with Automatic Speech Recognition (ASR) technologies. ASR has been further utilized or extended in many following studies [2]. Harrison *et al.* developed an extended recognition network that included erroneous pronunciation patterns with an ASR framework in order to detect the mispronunciation within learners' utterances [3]. A series of work led by Minematsu and Hirose also introduced speaker-independent structural representation with a goal to eliminate the difference caused by different speakers and acoustic conditions for better pronunciation evaluation [4]–[7]. The above are several instances demonstrating the multifaceted research directions towards the goals of CAPT.

The objective of CALL can be realized in more comprehensive ways such as systems and games. Strik *et al.* developed a web-based role-playing environment for practicing conversation in Dutch in the DISCO and GOBL projects [8], [9]; Educational Testing Service Research (ETS Research) presented an online practice test, SpeechRater, for the Test of English as a Foreign Language internet-based test (TOEFL iBT) and analyzed the effectiveness of different information offered [10]–[12]; A series of games for language learning led by Seneff used speech and language processing techniques [13]–[17]. NTU Chinese developed at National Taiwan University (NTU) is an online software system which is able to provide pronunciation evaluation and corrective feedback for non-native Chinese learners [18]. Also, "Rosetta Stone" [19] and "English Town" [20] are popular language learning products nowadays being used by many language learners. These are typical examples of CALL systems focusing differently from pronunciation learning to vocabulary learning, from spoken dialogues to different learning scenarios.

On the other hand, spoken dialogues have been extensively investigated and widely used for long in speech processing com-

Manuscript received April 22, 2014; revised September 09, 2014; accepted November 09, 2014. Date of publication December 02, 2014; date of current version January 14, 2015. This work was supported by the Ministry of Science and Technology in Taiwan under contract MOST 103-2221-E-002-136-MY3. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Chung-Hsien Wu.

P.-h. Su is the Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, U.K. (e-mail: phs26@cam.ac.uk).

C.-h. Wu is the Department of Computer Science and Information Engineering, National Taiwan University, Taipei 10617, Taiwan (e-mail: r02922002@ntu.edu.tw).

L.-s. Lee is the Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan (e-mail: lslee@gate.sinica.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.2014.2375572

munity. Traditionally, most spoken dialogue systems have been developed to serve specific purposes [21], [22], for example slot-filling tasks such as airline ticket booking or city information querying [23], [24], in which the user-system interaction is very often modeled in a statistical framework. When spoken dialogues were used for language learning, a more immersive learning environment including language interaction experiences can be provided. Raux and Eskenazi proposed to use correction prompts in task-oriented spoken dialogues for language learning [25]. Johnson reported the Tactical Language Training System (TLTS) which offered interactive lessons and games for foreign language communication skill training [26]. These are just a few examples of using spoken dialogues in CALL.

Recently we tried to extend the functionalities of pronunciation evaluation and corrective feedback provided by NTU Chinese [18] for the purpose of offering furthermore the immersive learning environment by spoken dialogues. We proposed a new dialogue game framework on top of the NTU Chinese pronunciation evaluation software [27]. In this framework, sentences to be practiced were adaptively selected on-line along the dialogue for each learner, based on the scores the learner has received for each pronunciation unit so far. The sentence selection was in such a way that those pronunciation units with lower scores could be offered with more practice opportunities along the dialogue, while those with high scores were not cared. Thus these poorly-pronounced units could be practiced repeatedly along the dialogue, and the learner did not have to repeat the same training sentence many times. This design enabled the learning materials for each learner to be completely personalized. The dialogue manager was modeled as a Markov decision process (MDP) [28], [29] trained with reinforcement learning and fitted value iteration using simulated learners generated from real learner data. This framework was then improved to become a recursive dialogue game [30]. The dialogue script was made recursive and the dialogue paths in the script could be infinitively long. The dialogue management policy was optimized such that the learner's scores of the pronunciation units (or a selected subset) could be practiced and improved to achieve a pre-defined standard in a minimum number of turns. This dialogue management policy was again optimized by an MDP, but with high-dimensional continuous state space for precise representation of the learning status for all pronunciation units considered.

In this paper, we present a complete framework integrating the above dialogue game concepts for extension of NTU Chinese including additional recent experimental analysis over different learning scenarios. A dialogue game system is also successfully implemented and reported here as well. The rest of this paper is organized as follows. Section II introduces the automatic pronunciation evaluator used in this framework, the NTU Chinese, the dataset used in the experiments, and the real learner data from the NTU Chinese project. Section III then presents the complete framework of the dialogue game proposed in this paper, including the dialogue script, the dialogue manager modeled with MDP, and the simulated learners. Sections IV and V describe the experimental setup and results including analysis and discussions on the results. Section VI presents the real system implementation and its features. Concluding remarks are finally made in the last section.

II. Automatic Pronunciation Evaluator and Dataset From the NTU Chinese Project

The proposed framework requires an automatic pronunciation evaluator to offer and update the scores in real-time for all considered pronunciation units produced by a specific learner. The evaluator serves as a guide for selecting the future sentences for the learner along the dialogue. In the work presented in this paper, we use the NTU Chinese [18] as the automatic pronunciation evaluator for learning Chinese as an example, although the concept is equally applicable to all languages with different pronunciation evaluators.

A. Automatic Pronunciation Evaluator: NTU Chinese

As mentioned above, NTU Chinese is a successfully operating online Chinese pronunciation evaluation software, specifically designed for providing the learners quantitative assessments and multimedia corrective feedback for their pronunciation. Learners can practice their listening and speaking skills anywhere and at anytime. It was produced by a joint effort between National Taiwan University (NTU) and some industry partners. NTU Chinese is able to evaluate the utterances produced by an individual learner with numerical scores. Those scores are given to each pronunciation units in four different aspects: pronunciation, pitch, timing and emphasis; where the first is primarily phonetic evaluation and the other three are primarily prosodic evaluation. For those phonemes with scores below a threshold, a 3-dimensional video will appear on the screen to demonstrate the correct ways of the vocalization when producing the phoneme, including the relative positions among the lip, tongue, teeth and other articulators. Description judgments or corrective suggestions for improving the pronunciation will also appear on the screen when needed. Such feedback in CALL systems have been shown in earlier studies [31]–[33] to be able to help the learner improve their pronunciation skills. The scoring algorithm is not only based on signal analysis with acoustic and prosodic models such as those using posterior probabilities, but further improved by learning from the scores given by professional human Chinese teachers over a corpus produced by a group of real non-native Chinese learners. The above learning's corpus, scored by human Chinese teachers, all learning sentences and the course content currently used in this software were contributed by the International Chinese Language Program (ICLP) of National Taiwan University.

B. Real Learner Data from NTU Chinese

The real learner data used in this paper are a set of read speech corpus collected in 2008 and 2009 from real learners practicing their Mandarin Chinese with NTU Chinese. A total of 278 learners at different learning levels (beginners, intermediate and advanced) from 36 different countries, with balanced gender and a wide variety of native languages, participated in the recording task. Each learner was asked to read a set of 30 phonetically balanced and prosodically rich sentences,

covering almost all frequently used Mandarin syllables and tone patterns, each of which contained 6 to 24 Chinese characters. These 30 sentences were selected from the learning materials used in NTU Chinese.

III. Proposed Dialogue Game Framework

The complete framework proposed is presented in this section. It includes three major components: the dialogue script (subsection III-C), the Markov decision process (MDP) (Sections III-D, III-E, III-F), and the simulated learner generation (Section III-G). They will be presented after the background ideas (Section III-A) and a framework overview (Section III-B).

A. Background Ideas: Repeated Practice in Dialogue

The background ideas here come from the extension of NTU Chinese. NTU Chinese is currently used by many Chinese learners in many institutions. However, the contents or learning materials of the software include only many question/reply pairs but no further dialogue and interaction. When a learner receives scores, feedback or suggestions with regard to the pronunciation of the produced utterance, he or she needs to repeat the same sentences again and again, trying to improve the pronunciation. But repeated practice on the same sentences is in any case boring to the learner. Therefore in the framework proposed here, we try to introduce a dialogue game scenario based on the existing pronunciation evaluation, such that the learners can continue interacting with the computer in interesting dialogues rather than repeating the same sentences. At the same time, those pronunciation units with lower scores will appear much more frequently within the dialogue in the near future, so the learner can practice these units much more times but in many different sentences.

In addition to the feedback of evaluation scores, description judgments or corrective suggestions to the learners as the core mechanisms for helping the learners in CALL systems [31]–[33], the necessity and effectiveness of repeated pronunciation practice has also been clearly shown in related literatures [34]–[36]. Such repeated practice is even considered as the *repetition drill* in language learning approaches [37], which underlines the importance of intensive practice in language learning for memorization and automatism by the learners with immediately offered quantitative assessments and corrective feedbacks. However, repeated practice on the same sentence is boring, which leads to the dialogue game approach proposed here: learners can not only practice different learning sentences with instructions and feedbacks along the dialogue, but also practice those poorly-produced pronunciation units repeatedly within different sentences along the dialogue. Besides, as mentioned earlier, spoken dialogues can offer more immersive learning environments including language interaction experiences. Therefore, the role-playing in practical dialogue scenario can further familiarize the leaner with real life interactions and help build the learner's self confidence while conversing with the target language [38].

B. Framework Overview

The overall system block diagram of the proposed framework is shown in Fig. 1. Interaction between the Learner (at the lower middle part) and the system involves Utterances Produced and Selected Sentences. The Utterances Produced by the learner are the input to the system, and the Selected Sentences offered by the system are practiced by the learner. The Automatic Pronunciation Evaluator (NTU Chinese in this work, at the lower right corner) evaluates each pronunciation unit in the utterance. In addition to giving feedback to the learner immediately after each utterance is pronounced, the scores of all pronunciation units are updated and sent to the Pedagogical Dialogue Manager (on the left side of the Learner), driven by the Sentence Selection Policy (at the lower left corner), for recommending the next sentence for the learner out of the Dialogue Script (at the upper right corner). A set of Real Learner Data in the corpora (on the left side of the Dialogue Script) is used to construct the Learner Simulation Model, which generates the Simulated Learners (at the upper left corner) to train the Sentence Selection Policy based on the Dialogue Script using the Fitted Value Iteration algorithm.

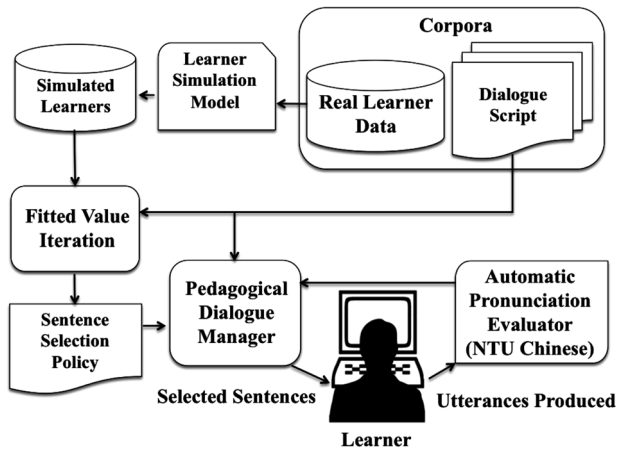

Fig. 1. System block diagram of the proposed dialogue game framework.

In this framework, both the computer and the learner need to have multiple sentence choices at each turn along the dialogue. A dialogue script properly designed for this purpose is therefore needed. Also, every sentence chosen by either the computer or the learner influences the future sentences along the dialogue significantly, resulting in very different distributions for the counts of different pronunciation units for the learners to practice in the future. The dialogue policy here is to select the most appropriate sentence for the learner to practice at each turn considering the learning status of the learner, such that more opportunities are given to practice those poorly-produced pronunciation units along the dialogue. Note that the sentences in the dialogue script have fixed question/reply sequential relationships. Thus selecting the next sentence containing the highest frequency counts for the poorly-produced units doesn't imply such high frequency counts for those units will continue to appear in the future sentences. So clearly the sentence selection policy needs to consider future sentences along the dialogue. In this way the learner can achieve the goal of having the scores of all (or a selected subset of) pronunciation units improved to achieve a pre-defined standard in a minimum number of turns.

The advantage of this framework is when participating in an interesting dialogue game, the learner can have diversified interactions with the system in an immersive environment; the learner can practice those poorly-produced pronunciation units

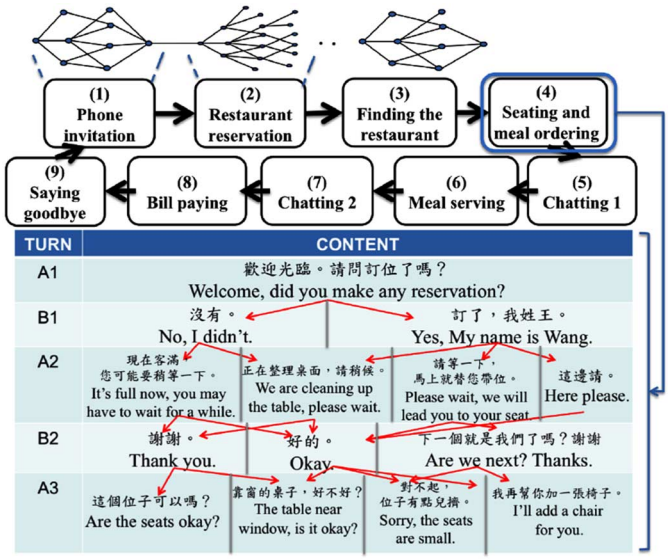

Fig. 2. The example script of the tree-structured dialogue game in the restaurant scenario used here: starting from (1) Phone invitation and (2) Restaurant reservation, all the way to (9) Saying goodbye and returning to (1) for next meal. A segment of the scenario (4) Seating and meal ordering is also shown.

many times in very different sentences along the dialogue, rather than producing the same set of boring sentences repeatedly. This also provides very high flexibility for learners to have completely personalized sentence practice opportunities.

C. Recursive Dialogue Script

The dialogue script is the backbone of the proposed framework, since the progress of the dialogue game is based on the script. Here the script for the proposed dialogue game consists of a series of tree-structured sub-dialogues which are cascaded into a loop, with the last sub-dialogue linked to the first. The initial script used in the preliminary experiments is the example dialogue in the restaurant scenario and includes a total of nine sub-dialogue, as shown in Fig. 2. It starts with the sub-dialogue (1) for phone invitation, followed by sub-dialogue (2) for restaurant reservation and so on, all the way to the last sub-dialogue (9) for saying goodbye, and returning to the sub-dialogue (1) for the next meal. In this design there can be almost unlimited numbers of dialogue paths within the script with any desired length. The nine sub-dialogues in the initial example script have a total of 176 turns. Each sub-dialogue contains conversations between roles A and B - one the computer and the other the learner. After an utterance is chosen and produced by one side, there are a number of choices for the other. This results in the script to be tree-structured. A segment of the sub-dialogue (4) "Seating and meal ordering" is also shown in Fig. 2, where A is the waiter and B the customer. The tree structure and multiple dialogue paths are also clearly shown. This dialogue script used below is designed by language teachers of National Taiwan University, such that the sentences included are phonetically balanced and prosodically rich, with good coverage of commonly used words on properly chosen learning level. The script can be utilized in many different ways in addition to looping the sub-dialogues recursively as shown in Fig. 2. For example, the learner can practice only one sub-dialogue but recursively, or arbitrarily cascading several sub-dialogues (e.g. sub-dialogue (1) followed by sub-dialogue (3)(4)), or with different ordering but recursively, etc., based on the learner's preference.

As mentioned above, since both the computer and the learner have multiple sentence choices at each dialogue turn, every choice influences the future path significantly. This results in very different distributions for the counts of different pronunciation units for the learner to practice in the future, and is where the sentence selection policy and Markov decision process can operate and offer personalized learning.

D. Markov Decision Process (MDP)

The dialogue manager is the core technique in the approach proposed in this paper. In order to select good training sentences personalized for the learner based on the learning status in real time during the interaction within the dialogue, we need a flexible system policy capable of dealing with all possible different learner situations. However, at each turn the learner's pronunciation performance for all pronunciation units (considered as "system state") may be changed and is unknown after producing the next utterance, which implies the "state transition" is uncertain or non-deterministic. On the other hand, the learning process for a specific learner has a given final goal where the scores for all pronunciation units reach a pre-defined standard (considered as "goal state"). So the desired system policy is to select a sequence of sentences along the dialogue script (considered as a sequence of "actions") such that the "goal state" can be reached in minimum number of turns. This implies to select a sequence of "actions" toward some goal over an uncertain "state space" considering some desired objective function, which is exactly a planning problem under uncertainty and is often handled with Markov decision process (MDP). In this approach the uncertain state problem are properly taken care of with the concept of expectation, and the desired best policy can be learned from data. This is a data-driven approach, in which the best policy is learned completely based on the training materials (here the expert-designed dialogue script and simulated/real learner data) without any hand-crafted effort. This data-driven nature also implies the approach is equally applicable to all different languages, all different learning scenarios (e.g. the script here is in restaurant scenario, but can be others such as shopping or traveling), and all different learning goals (e.g. for learning any desired set of pronunciation units), as long as given the needed training data for the propose. Thus the Chinese pronunciation learning in restaurant scenario is simply an example illustrating the feasibility of the approach. It can certainly be extended to all other languages with all different purposes. The above reasons explain why MDP is chosen for dialogue manager modeling.

Of course, policy can also be established by some heuristic methods or some other planning algorithms. Heuristic methods are often hand-crafted, based on some principles or criteria. Good examples include the arbitrary method (randomly choosing the next sentence for the learner to practice in the case here) and the greedy method (selecting the next sentence with the most count of the learner's poorly-pronounced units in the case here), but both of them can offer very little benefits since they are limited to the chosen principles or criteria. The greedy method is not very helpful because selecting the next sentence with the most count of the learner's poorly-pronounced units

never implies the same for future sentences in the dialogue script. The well-known planning algorithms developed for deterministic problems such as A^* search algorithm similarly can only yield very limited effect, because the uncertain state space considered here makes the case non-deterministic. Detailed experimental results comparing between different methods with the proposed policy using MDP will be shown in Section V. Moreover, since the learner's status may be misjudged by the system (the scores of the pronunciation units estimated by the Automatic Pronunciation Evaluator may be wrong), a more generalized version of MDP called partial observable Markov decision process (POMDP) can certainly be applied here, which regards the learner's status estimated by the Automatic Pronunciation Evaluator as one sample from the true learner's status and maintains a probability distribution over all possible learners' status. Here in the initial work of this paper we choose to investigate the use of MDP with preliminary experiments, while the application of POMDP will be considered in the future.

An MDP [39], [40] is a mathematical framework for modeling sequential decision making problems, formally represented by the 5-tuple $\{S, A, R, T, \gamma\}$, including the set of all states S, the set of possible actions A, the reward function R, the Markovian state transition function T, and the discount factor γ which determines the effect of future outcomes on the current state s. When an action a is taken at the state s, a reward r is received and the state is transited to a new state s'. Solving the MDP consists of determining the best action a to be taken at each state s called a *policy* π, which maximizes the expected total discounted reward starting from the state s, or value function : $V^\pi(s) = E[\sum_{k=0}^{\infty} \gamma^k r_k | s_0 = s, \pi]$, where s_0 is the starting state, r_k is the reward gained in the k-th state transition, and the policy $\pi : S \rightarrow A$ maps each state s to an action a. The above value function can be further analyzed by the state-action (Q) value function, which is defined as the value function of taking action a at state s based on the policy $\pi : Q^\pi(s, a) = E[\sum_{k=0}^{\infty} \gamma^k r_k | s_0 = s, a_0 = a, \pi]$, where a_0 is the action taken at state s_0. Thus the optimal policy π^* can be expressed as $\pi^*(s) = \arg\max_{a \in A} Q^\pi(s, a)$ by a greedy selection over the state-action pairs, where the maximization considers every $Q^\pi(s, a)$ for all possible policies π while focusing on the action a to be taken at the state s only. The goal of finding the optimal policy is therefore equivalent to maximizing these Q functions.

The above Q functions can be updated iteratively toward the optimal values in a process known as *fitted value iteration* based on the *Dynamic Programming* (or *Bellman*) Equation:

$$Q^\pi = B^\pi(Q^\pi), \tag{1}$$

where the Bellman mapping (also called the Bellman backup operator) B^π is defined as:

$$[B^\pi(Q^\pi)](s, a) = E_{s' \sim T(s,a)}[R(s, a, s') + \gamma Q^\pi(s', \pi(s'))], \tag{2}$$

where $E_{s' \sim T(s,a)}$ stands for the expectation evaluated for the next state s' following transition probability distribution T from state s with action a taken, and $R(s, a, s')$ is the reward gained when transiting from state s to s' by taking action a. So in every iteration each $Q^\pi(s, a)$ is updated as in (1) by the backup operator as in (2).

E. MDP for the Dialogue Game

Here we describe how the proposed dialogue game is modeled using the Markov decision process.

Continuous state Representation: The state represents the learning status of the learner. It consists of the average scores obtained so far for all the pronunciation units considered in the game given by the Automatic Pronunciation Evaluator in Fig. 1, each being a continuous value (ranging from 0 to 100 in the case of NTU Chinese in the work presented here), and the present value is directly observable by the system. This results in a high-dimensional continuous state space ($s \in [0, 100]^U$ for the work here, where U is the total number of pronunciation units considered.). Note that the state is changed after an additional sentence is produced by the learner and the state transition is uncertain. In addition, since the system needs to know which dialogue turn the learner is at, the dialogue turn index t is also included in the state space.

Action Set: In each state at dialogue turn t, the actions to be taken are the available sentence options to be selected for practice. Since the dialogue script used is in tree structure, the number of actions (next available sentences to be chosen) in every specific state may vary even at the same given turn.

Reward Function: A dialogue *episode* E contains a sequence of state transitions $\{s_0, a_0, s_1, a_1, \ldots, s_K\}$, where s_k is the k-th state and a_k is the action taken at s_k, and s_K represents the terminal state, in which the scores of all pronunciation units considered reach a pre-defined standard. The goal here is thus to train a policy that can offer the learner at each turn the best selected sentence to practice considering the learning status, such that the terminal state s_K can be reached within a minimum number of turns. Hence every state transition is rewarded -1 as the penalty for an extra turn ($r_k = -1, k \leq K - 1$), and r_K is the finishing reward gained when the terminal state s_K is reached. The final total reward R is then the sum of all rewards obtained: $R = \sum_{k=0}^{K} r_k$. So the goal here is to maximize R. In addition, a timeout count of state transitions L is used to limit the episode lengths.

F. Policy Training for the Dialogue Game

Here we wish to find the optimal policy which maximizes the final total reward R as defined above [40]. Since we have a high-dimensional continuous state space as mentioned above, we use the function approximation method [41]–[43] to approximate the exact value function $Q^\pi(s, a)$ with a set of m basis functions:

$$\hat{Q}_{\underline{\theta}}(s, a) = \sum_{i=1}^{m} \theta_i \phi_i(s, a) = \underline{\theta}^T \underline{\phi}(s, a), \tag{3}$$

where $\underline{\theta}$ is the parameter (weight) vector corresponding to the basis function vector $\underline{\phi}(s, a)$. The goal of finding the optimal policy can then be reduced to finding the appropriate parameters $\underline{\theta}$ for a good approximation $\hat{Q}_{\underline{\theta}}(s, a)$ for $Q^\pi(s, a)$. A *sampled* version of the Bellman backup operator $B^\pi(Q^\pi)$ in (2)

is introduced for the i-th sampled transition (s_i, a_i, r_i, s_i') as $\hat{B}(Q(s_i, a_i))$ in (4),

$$\hat{B}(Q(s_i, a_i)) = r_i + \gamma \max_{a \in A} Q(s_i{}', a_i). \tag{4}$$

With a batch of transition samples $\{s_j, a_j, r_j, s_j^{'} | j = 1, \ldots, J\}$ for the n-th iteration, *least-square linear regression* can be performed to find the new parameter vector $\underline{\theta}_n$ at the n-th iteration so that $\hat{Q}_{\underline{\theta}_n}(s, a)$ approaches $Q^{\pi}(s, a)$ as precisely as possible. The parameter vector is updated as

$$\underline{\theta}_n = \arg \min_{\underline{\theta} \in \mathbb{R}^m} \sum_{j=1}^{J} (\hat{Q}_{\underline{\theta}})(s_j, a_j) - \hat{B}(Q_{\underline{\theta}_{n-1}}(s_j, a_j))^2 + \frac{\lambda}{2} \|\underline{\theta}\|^2, \tag{5}$$

where the second term is the 2-norm regularized term weighted by λ to prevent over-fitting. Therefore the approximation $\hat{Q}_{\underline{\theta}_n}(s_j, a_j)$ approaches and converges to $\hat{B}(Q_{\underline{\theta}_{n-1}}(s_j, a_j))$ along with the training iterations.

The above linear regression has a closed form solution that can be easily performed at each iteration. If each transition sample (s_j, a_j) is viewed as a row vector $\underline{\mathbf{x}}_j$ and $\hat{B}(Q(s_j, a_j))$ as a scalar y_j, the vectors $\underline{\mathbf{x}}_j$ and values y_j for the training samples $\{s_j, a_j, r_j, s_j' | j = 1, \ldots, J\}$ can form a matrix $\mathbf{X}$ and a vector $\underline{\mathbf{y}}$, then the linear regression solution of the updated $\underline{\theta}_n$ at the n-th iteration is simply:

$$\underline{\theta}_n = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \underline{\mathbf{y}}, \tag{6}$$

where $\mathbf{I}$ is the identity matrix and $\lambda \mathbf{I}$ is the regularized term.

G. Simulated Learner Generation from Real Learner Data

The MDP presented above requires a huge quantity of data for training, and the training data should reflect the pronunciation behavior of real learners. Here we present the approaches to generate simulated learners from real learner data for this purpose. The Automatic Pronunciation Evaluator in Fig. 1 assigns scores to each pronunciation unit in every utterance of the real learner data. These scores for each utterance pronounced by a real learner are used to construct an utterance-wise score vector $\underline{z}$, whose dimensionality U is the number of pronunciation units considered. Every component of the score vector $\underline{z}$ is the average score of the corresponding unit in the utterance; those units unseen in the utterance are viewed as latent data and treated with the expectation-maximization (EM) algorithm [44], [45]. Hence every utterance produced by the real learner is used to generate a separate score vector $\underline{z}$. Such utterance-wise score vector $\underline{z}$ can reflect the score correlation across different pronunciation units with certain context relationships, as will be further explained below. In our dataset as mentioned in Section II-B, we have 278 learners, each with 30 utterances. Therefore we have around 8000 score vectors for GMM training. The score vector $\underline{z}$ constructed from all utterances produced by all real learners are then used to train a Gaussian mixture model (GMM) with the EM algorithm that handles latent data. Note that the trainable parameters within GMM are priors, mean vectors and covariance matrices. When calculating and maximizing the likelihood of a certain Gaussian mixture given a set of score vectors, each missing score in $\underline{z}$ (where the pronunciation unit in the corresponding utterance is not presented) is simply substituted by the value of the corresponding index within the mean vector of the considered Gaussian estimated up to the previous iteration. In this way all missing data in the score vectors $\underline{z}$ can be properly taken care of. This GMM is referred to as the Learner Simulation Model and is shown in the left half of Fig. 3.

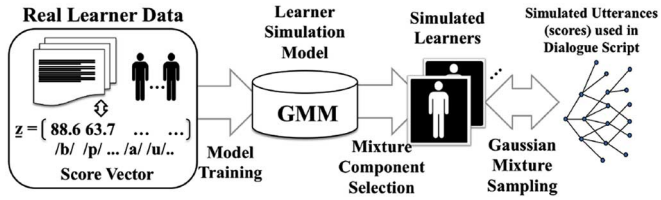

Fig. 3. Learner Simulation Model for simulated learner generation.

The GMM here not only aggregates the utterance-wise score distribution statistics of the real learners, but also reflects the utterance-wise correlation of scores across different pronunciation units within different contexts. For example, some Chinese learners have difficulties pronouncing all retroflexed phonemes (appearing frequently in Mandarin but not necessarily in other languages) within the context of certain specific vocal tract articulation. Such situation may be reflected in the above GMM trained with the utterance-wise correlated score vectors $\underline{z}$. Therefore each mixture of this GMM may represent the pronunciation error distribution patterns for a certain group of learners with similar native language backgrounds. On the other hand, different Gaussian mixtures in the GMM may also represent different learning levels (beginners, intermediate and advanced) across the real learners. With this approach we discover clusters of alike learning behavior without building a speaker-wise models, and we can directly train the GMM from raw data.

The adequate number of the mixtures in GMM is usually not easy to determined manually. Increasing this number (adding parameters) often yields higher model likelihood but may result in over-fitting and lack of generalization as well. Therefore Bayesian information criterion (BIC) [46], [47] is employed here on GMM to handle this problem by taking into account both the model likelihood and the parameter complexity penalty. Assume that I is the number of Gaussians, the total number M of continuously-valued free variables is:

$$M = I(1 + D + \frac{1}{2} D(D + 1)) - 1, \tag{7}$$

where each Gaussian needs one variable for prior probability, D variables for the mean, and $\frac{1}{2} D(D + 1)$ variables for the covariance matrix. With this total number of free variables, the value of the BIC is defined as the negative of the log likelihood function plus a term for the model complexity penalty weighted by the size N of the training data X,

$$BIC = -\log P(X|\Theta) + \frac{1}{2} M \ln N, \tag{8}$$

where Θ is the parameter set for all the Gaussians. The goal is then to find the number of Gaussians I and the complete parameter set Θ that minimizes (8). Through this criterion we can obtain the proper parameter setting given the training data.

With the GMM trained, for the MDP policy training, we first randomly select a Gaussian mixture component as a simulated learner group for a certain native language background

TABLE I
A LIST OF ALL THE 101 MANDARIN PRONUNCIATION UNITS CONSIDERED IN THE EXPERIMENTS, INCLUDING PHONETIC UNITS (22 INITIALS, 36 FINALS) AND PROSODIC UNITS (5 UNI-TONES, 19 INTRA-WORD BI-TONES AND 19 INTER-WORD BI-TONES). THE ID INDICES IN THE PARENTHESES ARE THOSE USED IN THE HORIZONTAL AXES IN FIG. 6 AND 11

Pronunciation Units										
Initials	(1) b	(2) p	(3) m	(4) f	(5) d	(6) t	(7) n	(8) l	(9) g	(10) k
	(11) h	(12) ji	(13) chi	(14) shi	(15) j	(16) ch	(17) sh	(18) r	(19) tz	(20) ts
	(21) s	(22) sic								
Finals	(23) a	(24) o	(25) ai	(26) ei	(27) au	(28) en	(29) an	(30) ang	(31) eng	(32) i
	(33) ie	(34) iau	(35) ian	(36) in	(37) ing	(38) u	(39) ou	(40) iou	(41) e	(42) uo
	(43) uei	(44) uan	(45) uen	(46) ueng	(47) iang	(48) iue	(49) iu	(50) ia	(51) iuan	(52) uai
	(53) uang	(54) iun	(55) iung	(56) empt1	(57) empt2	(58) er				
uni-tones	(59) 1	(60) 2	(61) 3	(62) 4	(63) 5					
intra-word bi-tones	(64) 11	(65) 12	(66) 13	(67) 14	(68) 21	(69) 22	(70) 23	(71) 24	(72) 25	(73) 31
	(74) 32	(75) 33	(76) 34	(77) 35	(78) 41	(79) 42	(80) 43	(81) 44	(82) 45	
inter-word bi-tones	(83) b11	(84) b12	(85) b13	(86) b14	(87) b21	(88) b22	(89) b23	(90) b24	(91) b25	(92) b31
	(93) b32	(94) b33	(95) b34	(96) b35	(97) b41	(98) b42	(99) b43	(100) b44	(101) b45	

[48]–[50]. The mean vector of the selected Gaussian mixture stands for the level of the simulated learners in this group on each pronunciation unit, while the covariance matrix represents the score variation within each unit and the score correlation between units. When a sentence is to be pronounced by a simulated learner, a score vector $\underline{z}$ randomly sampled from this mixture yields the scores for the units in this sentence, taken as the simulated utterance produced by this simulated learner within the group. This is then repeated many times both within the group and across many different groups, as shown in the right half of Fig. 3.

The goal here is to provide proper sentences for each individual learner to practice in the dialogue such that the learner's pronunciation for all units considered reaches a pre-defined standard in a minimum number of turns. However, the learner simulation model mentioned above only models the pronunciation performance given the real learner data but not the characteristics of pronunciation improvement. Hence we need to develop in addition an incremental improvement model for the scores of pronunciation units produced by the simulated learners. This can be achieved by assuming whenever the i-th pronunciation unit has been practiced $\mathcal{C}$ times by a simulated learner, the i-th component of the mean vector in the Gaussian mixture is increased by α (to a higher level) and the (i,i)-th element in the covariance matrix of the Gaussian mixture decreased by β (with more stable scores). Here $\mathcal{C}$, α, and β are all Gaussian random variables with means and variances assigned according to the overall pronunciation performance of the simulated learner. In other words, when more units in the mean vector are closer to the pre-defined standard, the mean and variance of the variable $\mathcal{C}$ become smaller, and the means of the variables α and β become larger and their variances smaller. So the pronunciation of the simulated learner not only improves incrementally along the dialogue path, but the improvement becomes faster and more stable when the overall pronunciation performance gets better. Detailed parameter setting for $\mathcal{C}, \alpha$, and β within our framework design will be mentioned below.

IV. EXPERIMENTAL SETUP

Experiments were performed on the complete recursive script of the nine sub-dialogues for learning Mandarin Chinese as described in Section III-B. The results reported below are all for the computer as role A and the learner as role B. Totally 101 Mandarin pronunciation units were considered, including 58 phonetic units (context-independent Initial/Finals of Mandarin syllables, where Initial is the initial consonant of a syllable, and Final is the vowel or diphthong part but including optional medial and nasal ending) and 43 prosodic units or tone patterns (uni-tones, intra-word bi-tones (tones for two consecutive syllables within a word) and inter-word bi-tones (tones for two consecutive syllables across a word boundary)). Detailed enumeration of all 101 units is listed in Table I. Several different learning scenarios were tested: learning tone patterns only, phonetic units only, both, and focusing on several selected subsets of the units. NTU Chinese [18] was used as the automatic pronunciation evaluator for unit scoring of the real learner data. In the MDP setting, the terminal state s_K was defined as the situation when 95% of all pronunciation units considered were produced with scores over 75 more than five times. The reward at the dialogue terminal state r_K was set to 0 and timeout count L was 500, that is, the dialogue game might end with reward -500 (reward -1 as penalty for each turn) if the system didn't arrive at the terminal state s_K in 500 turns. Multivariate Gaussian functions of 101 dimensions served as the basis functions $\phi(s,a)$ in (3) to approximate the value function. We set the number of basis functions m to 5 in this paper empirically, since higher parameter complexities resulted in over-fitting [30]. These Gaussian functions had fixed covariance matrices and we tried to have their means evenly spread on the state space. All these Gaussian basis functions had initial weights set to 1, then updated with (4), (5). The system's initial policy was always to choose the first sentence among the candidate sentences. Five-fold cross-validation was used: in each training iteration, four-fifths of the real learner data were used to construct the GMM to generate simulated learners for policy training, while the rest was used to train another GMM to generate simulated learners for testing. This was repeated five times and the average was taken. The number of mixtures within the GMM for generating the simulated learners was determined by BIC in (8). The parameters $\mathcal{C}, \alpha$, and β (all conforms to Gaussian) in the incremental improvement model for pronunciation scores mentioned in subsection 3.7 were set based on the overall pronunciation performance. The mean and variance of $\mathcal{C}$ were respectively set $6-b*2$ and $2-b$, where b is the percentage of well-produced unit with average score above 75. The mean and variance of α and β were both set $6+b*4$ and $2-b$. In the testing phase, the testing simulated learners

went through the nine sub-dialogues either in sequential order recursively or in arbitrary order until the terminal state s_K was reached. All MDP testing results reported were the average of 100 testing simulated learners.

The approach proposed here is a new framework for pronunciation training with spoken dialogues. The standard evaluation metrics for spoken dialogue systems, such as the task success rate assessed by either simulated or real users under different recognition word error rate (WER) [51], [52], cannot be applied because of the very different task goal and application scenario. There exists no any prior work with similar task goal which can be compared with either. We therefore designed the evaluation methodologies for the specific task based on the objective of personalized pronunciation training, in which the system is to provide personalized training sentences as effective as possible. We also try to compare the proposed approach with several existing methods for the objective here when possible. This leads to the following ways of demonstrating the experimental results:

1) Average number of dialogue turns needed for the test learners to reach the terminal state, or all pronunciation units considered having scores achieving a pre-defined standard, is taken as the primary indicator for system performance. This is easily understandable, since the goal of the system is to have this number as small as possible.
2) Learning curves for the average dialogue turns needed are used to show how the policy is improved during learning. Learning curves are usually used in machine learning applications to show how the system objective and model parameters are optimized along with the increase of training iterations. In the experiments below we use the learning curves for the average number of dialogue turns needed (system objective) to demonstrate the way the policy was learned with different setting of model parameters, or under the goal of focusing on different subsets of pronunciation units.
3) Number of repeated practice opportunities on each poorly-produced pronunciation unit for a typical example simulated learner is used to show the effectiveness of the system policy when comparing with other existing methods. As mentioned previously, repeated practice is a major approach for language learning and the goal here is to offer repeated practice to poorly-produced units. Such experimental results can illustrate how the proposed approach actually works.

V. Experimental Results

A. Parameter Tuning

We used the basis function vector $\underline{\phi}(s,a)$ as in (3) to approximate the value function, and casted all basis functions within this vector as multivariate Gaussians with fixed covariance matrices. Here a good estimate for these covariance matrices are needed, so we assumed these matrices to be diagonal with identical diagonal values v. In addition, we also need to estimate an appropriate value for the regularized term weight λ in (5). Fig. 4 demonstrates the learning curves for the number of dialogue turns averaged over 100 testing simulated learners needed to reach the terminal state, plotted as functions of the number of training iterations, under different parameter combinations:

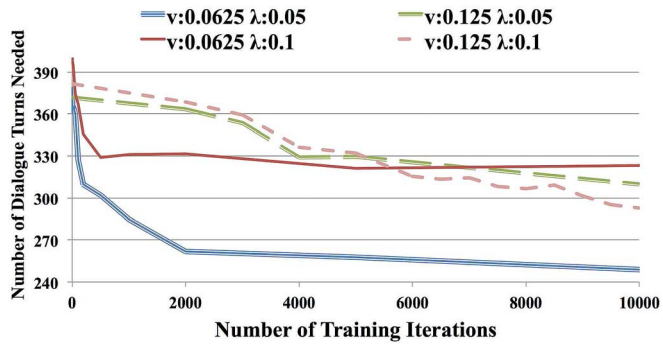

Fig. 4. Number of dialogue turns needed with respect to number of training iterations under different parameter settings when all 101 units were considered.

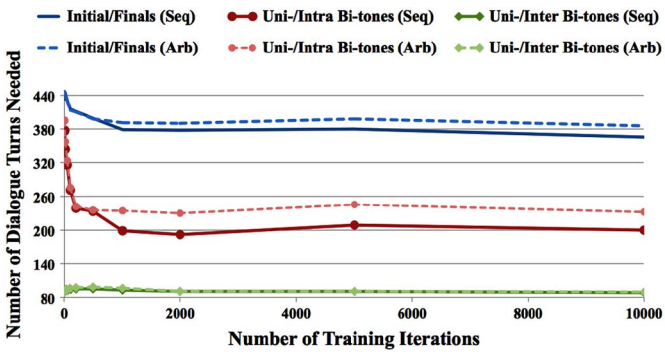

Fig. 5. Number of dialogue turns needed with respect to the number of training iterations for different sets of pronunciation units.

$v = 0.0625, 0.125$ and $\lambda = 0.05, 0.1$. In these cases all the 101 units mentioned above were considered and all the nine sub-dialogues trees were went through in sequential order recursively. As shown in the Fig., the system indeed learned to optimize its policy in order to offer better dialogue paths in the script to every individual testing simulated learner along with the training process. It is worth mentioning that with $v = 0.0625$ and $\lambda = 0.05$, the system performed the best which actually converged at an average of 250.66 turns, while other settings ended with relatively over-fitted results. Therefore, in all following experiments we used the parameter setting of $v = 0.0625$ and $\lambda = 0.05$. Note that the parameters obtained here are certainly not optimal since the too limited quantity of the available training data and better settings are definitely needed. However, we still tried to find adequate system policy with the very limited training data.

B. Number of Dialogue Turns Needed for Different Sets of Pronunciation Units

In Fig. 5, we plot the learning curves for number of turns needed to reach the terminal state averaged over 100 testing simulated learners with respect to the number of the training iterations (the training processes went through the nine sub-dialogues in sequential order), when different subsets of units were considered: 58 Initial/Finals only (blue), 5 uni-tones plus 19 intra-word bi-tones (red), and 5 uni-tones plus 19 inter-words bi-tone (green). The solid curves (labeled "Seq") are those when the testing simulated learners went through the nine sub-dialogues sequentially and recursively. The dashed curves (labeled "Arb") are those when the testing simulated learners went

through the sub-dialogues in arbitrary order, while the system policies were trained with training simulated learners going through the sub-dialogues in sequential order. For example, the testing learner could jump to the fourth sub-dialogue after finishing the second sub-dialogue (after restaurant reservation, the learner wishes to learn how to order meals first). Clearly the three solid curves (labeled "Seq") for different sets of target units considered yielded promising results. The number of needed turns for blue, red and green solid curves converged at 365.60, 199.36, 89.52 turns respectively. As the number of target units becomes smaller, the needed number of turns is also smaller. Since there was a total of 84 turns for role B in the nine consecutive sub-dialogues, the results in Fig. 5 indicated that going through all nine sub-dialogues and restarting from the first sub-dialogue was necessary for the testing simulated learners here, which was also true for Fig. 4. Moreover, the extra turns needed for the dashed curves compared to the solid curves show the cost paid when the user decided to practice the desired contents (sub-dialogues) that are different from the way the policy was trained.

It is worth noting that the needed turns of considering uni-tones plus intra-word bi-tones (red) were much higher than considering uni-tones plus inter-word bi-tones (green) (199.36 vs. 89.52 turns for solid curves). This is obviously because of the well known tone sandhi phenomena in Mandarin Chinese, that is, the tone pattern of a syllable is highly dependent on the tones of the left and right context syllables if these syllables are within the same word, but such context dependency of tone patterns becomes much less observable across word boundaries. This is very difficult for learners. By observing the scores for the real learners, it is easy to find the scores for inter-word bi-tones are usually better, while those for intra-word bi-tones are significantly worse in average, because the tone sandhi within words are much difficult to learn. The results in Fig. 5 show that such situation was clearly reflected in the score vectors of the simulated learners, and as a result was also illustrated in the policies and the number of turns needed.

When we compare the solid blue curves between Fig. 4 and 5, we can notice that considering all 101 pronunciation units converged at a much lower turn number (250.66. in Fig. 4) than taking only the 58 Initial/Finals into account (365.60 in Fig. 5). This seems to be difficult to understand. Analysis showed that the occurrence frequencies of some low-frequency Initial/Finals in the whole dialogue script were very low, despite that great effort has been made to try to make the script as phonetically balanced as possible. Therefore when a simulated learner had bad performance on these low-frequency units, it might take much more dialogue turns to reach the system goal (Fig. 5), i.e. 95% of all pronunciation units were produced with scores over 75 more than five times. When considering all 101 units (Fig. 4), all the tone patterns were considered, most of which had much higher occurrence frequencies, therefore the goal of 95% was much more easier to achieve.

C. Learning Status and Policy Behavior for a Typical Example Learner

Using the policy learned with the parameters chosen in Section V-A ($v = 0.0625$, $\lambda = 0.05$, $m = 5$, all 101 units, blue

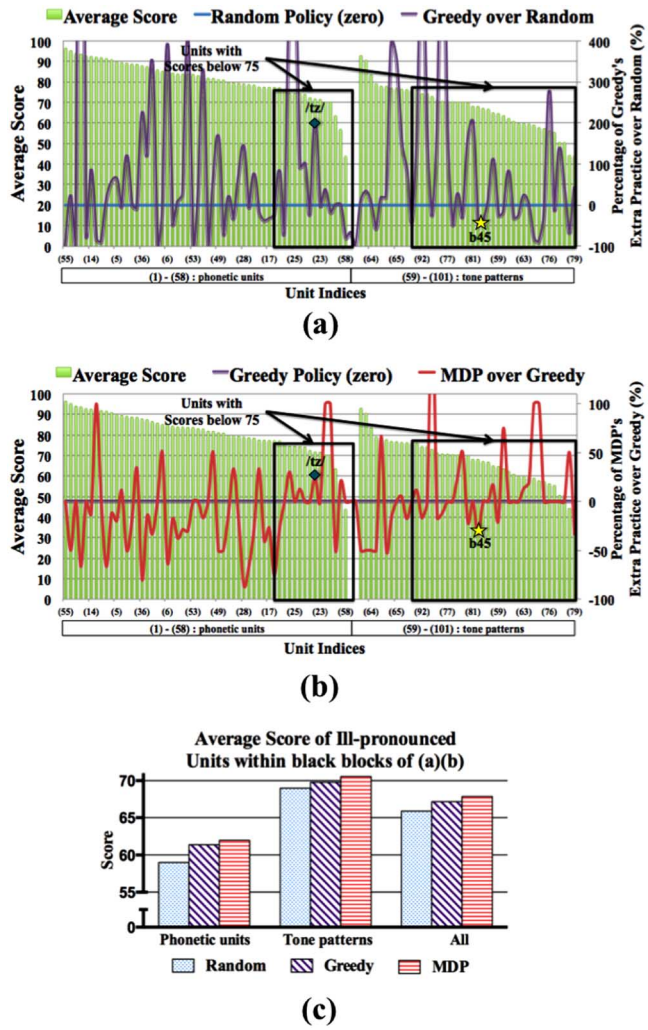

Fig. 6. Statistical analysis for a typical example testing simulated learner. In (a)(b) , all pronunciation units as listed in Table I (58 phonetic units on the left half and 43 tone patterns on the right half) are sorted and shown accordingly based on their average scores after finishing the sub-dialogues (1)–(3) (green bar, left scale). Only one ID index as listed in Table I out of every five units are shown on the horizontal axes. (a) percentage of extra practice (right scale) using greedy method (purple curve) over those by random method (blue line) in the next two sub-dialogues (4)(5), (b) percentage of extra practice using proposed policy with MDP (red curve) over those with greedy method (purple line) in the next sub-dialogues (4)(5), and (c) average scores of different sets of ill-pronounced units (within black blocks of (a)(b)) after finishing sub-dialogues (4)(5) using different approaches.

curve in Fig. 4), Fig. 6 shows how two straightforward heuristic methods (random and greedy) and the proposed policy using MDP offered practice opportunities for every pronunciation unit for a typical example testing simulated learner when going through the sub-dialogues (4) and (5) (a total of 18 turns) after finishing the first three sub-dialogues (1)–(3). For this simulated learner the terminal state s_K was reached with $K = 265$ dialogue turns. Listed on the horizontal axes in Fig. 6 are the 101 pronunciation units including Initial/Finals (left half) and tone patterns (right half) by their ID indices as listed in Table I (only 1 out of 5 for the limited space), respectively sorted by their average scores (represented by green bars based on the left vertical scale) received by this simulated learner after finishing the sub-dialogues (1)–(3). In Fig. 6(a), we first compare two heuristic methods (greedy and random). The purple curve

indicates the *percentage of extra opportunities for practice* for each unit offered by the greedy method, as compared and normalized to those offered by the random method represented by the blue line or zero level (based on the right vertical scale). Here the greedy method chose at each turn the sentence among those available which has the most practice opportunities for those units with average scores below 75, while the random method simply made the choice randomly.

Since the policy goal of the proposed approach was for 95% of the pronunciation units to be produced with scores above 75 over five times, we should focus on the units within the two black blocks in Fig. 6(a), which are those with average scores below 75. Clearly we can see from Fig. 6(a) the greedy method resulted in much more practice opportunities than the random method on most units within the black blocks (more than 70%), although only 18 turns in sub-dialogues (4)(5) were considered. These results show the effectiveness of the greedy method over the random method, and we expect the difference would be more obvious if more dialogue turns were considered and at each dialogue turn more candidate sentences were available.

In Fig. 6(b), we further compare the proposed policy using MDP represented by the red curve with the greedy method represented by the purple line in exactly the same way as in Fig. 6(a), except the purple curve for greedy method in Fig. 6(a) is taken as the baseline for comparison and normalization here and shown as a straight purple line (or zero level) in Fig. 6(b). Notice that the greedy method chose the sentences simply based on the practice opportunities among the next sentence candidates, it was not able to look ahead over the future dialogue paths. High counts of practicing opportunities for some units in the next sentence do not imply the same for the future. In contrast, the proposed policy with MDP has already been trained to consider every possible dialogue path till the terminal state using the approximation method mentioned in Section III-F with huge number of simulated learners. Thus in the two black blocks of Fig. 6(b) we can notice that even much more practice opportunities on most units (almost 80%) were offered by the proposed policy. Again we expect the difference would be more obvious if given a dialogue script with more dialogue turns and more candidate sentences available at each dialogue turn. Note that in Fig. 6(b) the red curve is below the zero level of the greedy method for most units (almost 50%) outside the two black blocks (their scores were above 75 so no further practices were needed), indicating that the proposed policy focused properly on the poorly-pronounced units. But this phenomenon is not seen in Fig. 6(a), in which much of the purple curve representing greedy is above the zero level of random for units outside the two black blocks, or many units with scores above 75 were actually given more practice opportunities, which is in fact useless. These results verified that the proposed policy was efficient to provide better personalized learning materials to the specific simulated learner than hand-craft heuristic methods, either greedy or random.

Besides, we clearly see in Fig. 6 that each pronunciation unit was given very different practice opportunities by various methods. Take the phonetic unit /tz/ (green diamond, whose score was below 75) for example. The greedy method provided more practice than random, while the proposed policy using MDP offered even much more. This is exactly the objective here. On the other hand, however, there also existed some difficult units, although very few. For example, both the greedy method and the proposed policy with MDP gave less practice opportunities than random method for the inter-word bi-tone b45 (yellow star), whose score was also below 75. This is probably because this unit appears with very low frequency in the dialogue script (unit b45 is the tone pattern containing a syllable of tone 4 before the word boundary with a syllable of tone 5 after the word boundary, but tone 5 almost never appears in the first syllable of a word except in mono-syllable words). Even though the system noticed it was poorly-pronounced, the system tended to choose the sentences that offered more practice opportunities for other poorly-produced units and most of them didn't contain the unit b45. Hence the unit b45 was somehow less focused by the MDP policy. This is also why the present objective is only to have 95% of units receiving scores above 75, or allowing for giving up 5% of low frequency units that the MDP policy cannot take care of. The above examples explain the difference between approaches on practice opportunities for different units.

It is also interesting to see in both Fig. 6(a) and (b) that most of the tone patterns are within the black block while most of the Initial/Finals are outside of the black block. This reflects the behavior of the real learners for the data collected and used here, i.e., tone patterns are often difficult for most real learners, while most real learners are relatively good at the pronunciation of many phonetic units.

To further investigate the effectiveness of three approaches compared above with respect to different sets of pronunciation units, we analyzed the average scores of those ill-pronounced units in black blocks of Fig. 6(a)(b) after practicing with sub-dialogues (4)(5). The scores were estimated using the incremental improvement model as mentioned in Section III-G, for which more practice opportunities brought about better performance. The results are in Fig. 6(c) respectively for phonetic units, tone patterns and all (phonetic units plus tone patterns). Naturally the outcomes in Fig. 6(c) are proportional to the results in 6(a)(b) in all cases, indicating the proposed policy using MDP is more effective and efficient than the other two methods.

D. Focused Learning on Selected Subsets of Units

Sometimes learners may already know their pronunciation status in advance and wish to focus their learning on a specific subset of units using the dialogue game. We consider this in two different scenarios: Using the policies learned with a larger training target set, or using a policy specially trained focused on the selected units. These show how the dialogue game could be developed and utilized in different ways. For the first experiment below, we assume the simulated learners decided to focus on a smaller number of units that were randomly selected from a larger set of target units, while the policy were trained using larger set of target units. The scores of other units were completely ignored during testing.

Table II shows the number of dialogue turns needed averaged over 100 simulated learners for such focused learning of 10, 20, 30 Initial/Finals (Part (A)), 5, 10, 15 tone patterns (Part (B)),

TABLE II
NUMBER OF DIALOGUE TURNS NEEDED FOR FOCUSED LEARNING ON SUBSETS OF RANDOMLY SELECTED SMALLER NUMBER OF PRONUNCIATION UNITS, USING POLICIES TRAINED WITH A LARGER SET OF TARGET UNITS

Target units for Policy Training	Number of units focused	Number of turns needed	
(A) 58 Initial/Finals	10	Sequential	73.99
		Arbitrary	89.61
	20	Sequential	127.94
		Arbitrary	138.56
	30	Sequential	133.9
		Arbitrary	142.98
(B) 43 Tone Patterns	5	Sequential	96.57
		Arbitrary	100.5
	10	Sequential	132.09
		Arbitrary	138.05
	15	Sequential	211.08
		Arbitrary	217.82
(C) All 101 units ((A) plus (B))	10	Sequential	106.35
		Arbitrary	108.03
	20	Sequential	136.67
		Arbitrary	143.75
	30	Sequential	156.5
		Arbitrary	156.78

and 10, 20, 30 Initial/Finals or tone patterns (Part (C)) respectively, using the policies learned when considering all the 58 Initial/Finals, all the 43 tone patterns, or the complete set of 101 units. The results were tested by following the sub-dialogues in either sequential order recursively (labeled "Sequential") or in arbitrary order (labeled "Arbitrary").

From Table II we can see that a significant number of turns were needed even if only 5 or 10 units were focused on. Also, the policies became more efficient when more units were considered. This is obviously because the training utterances automatically carried many different units for practice even if the learner wished to focus on only a small number of them, so trying to learn more units was in general more efficient. Also, some low frequency units, if selected by the learner, might require more turns to be practice in the dialogue. This is probably also the reason why in many cases in Table II the extra turns needed for taking arbitrary order of sub-dialogues as compared to the sequential order seemed to be relatively limited. On the other hand, it is also interesting to notice that learning 10 tone patterns required much more turns than learning 10 Initial/Finals (middle of Part (B) vs. top of Part (A)), obviously because learning intra-word bi-tones were much more difficult as discussed previously, as reflected in the real learner data and the simulated learners.

In the next experiment we assume the learners chose to focus on a small set of units, say the Retroflexed units (the four Initials /j/, /ch/, /sh/, and /r/), which are special in Mandarin Chinese and usually difficult for many non-native speakers to learn and pronounce. We trained a policy specifically focused on these four Initials and then it was tested by 100 simulated learners with different pronunciation performance on these Retroflexed units. The red curves as shown in Fig. 7 are when the policies were trained and tested both on the specific set of units. The blue curves are exactly those in Fig. 5 when all 58 Initial/Finals were considered for both policy training and testing, in both sequential and arbitrary case for comparison. The red curves show that the system policies which took the Retroflexed units into consideration yielded promising performance.

All the above results demonstrate that the proposed framework could provide personalized policies which were specially

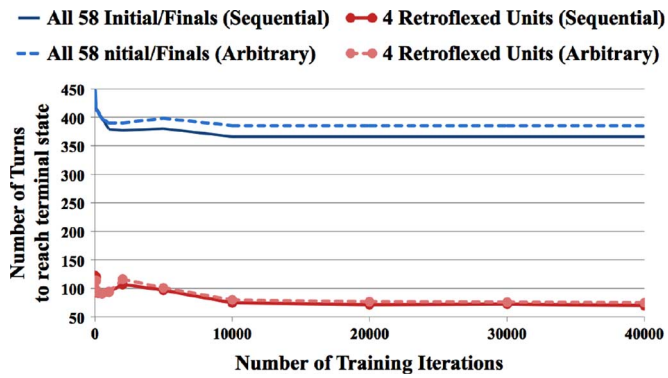

Fig. 7. Learning curves for number of dialogue turns needed when policies were trained and tested both on two specific set of units: all 58 Initial/Finals and 4 Retroflexed units in Mandarin Chinese respectively.

trained for each learner considering the personalized learning goals such as focusing on a specific set of units.

E. Consideration for Different Learner Levels and Different Native Language Backgrounds

Note that the policy in the proposed approach was trained with simulated learners generated from real learner data. In the initial experiments we had only very limited data, so we used these data to construct a single learner simulation model (GMM), from which all simulated learners generated are used to train a single set of policy. The set of 278 real learners described in Section II-B included beginner, intermediate and advanced levels of learners with varying native language backgrounds. All these varying learning levels and different native language backgrounds were automatically reflected by the different Gaussian mixtures in the learner generation model described in Section III-G, so the single set of trained policy is assumed to take care of learners with varying levels and different native language backgrounds. The beginners will need more number of turns to reach the system objective, while the advanced learners may need much less. All the results regarding the number of turns needed reported in above experiments are actually the averages over large number of simulated learners, and we may assume the distributions of the different levels and different native language backgrounds for these simulated learners were similar to those for the 278 real learners. On the other hand, if more real learner data are available and can be divided into different sets of data for different levels and different native language backgrounds, it is certainly possible to use the different sets of real learner data to build different learner simulation models and in turn to train different sets of policies for different levels of learners and different native language backgrounds. Experiments on simulated and real learners at different levels can then also be further tested based on these policies. But these are out of the scope of this paper.

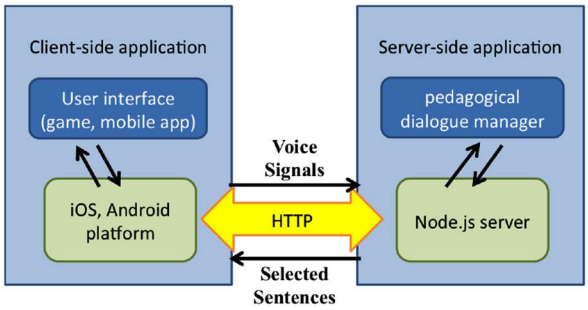

Fig. 8. System architecture for the cloud-based system.

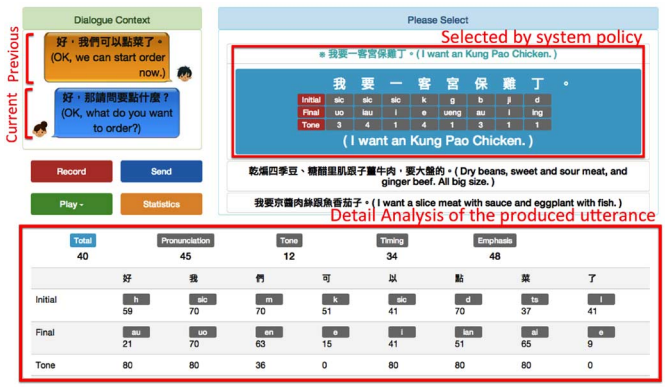

Fig. 9. The initial user interface of the real system.

VI. Real System Implementation

A. System Overview

Based on the framework proposed above, we have implemented the dialogue game core engine as a cloud-based system using NTU Chinese as the automatic pronunciation evaluator. To provide good operability, the system is exposed through REST API. Fig. 8 shows the system architecture. A Node.js server accepts HTTP requests with a URL mapping to the dialogue system service. The web server then passes the HTTP requests including the parameters and voice signals to the pedagogical dialogue manager. When the next sentence for practice is selected by the dialogue manager, this selected sentence is packed into a HTTP response and sent to the client. User-specific data, such as the pronunciation scores and profile, are stored in a separate database. In this way, developers can further build applications on various platforms, such as a web page, a mobile app or a flash game, using any HTTP library that can issue the REST calls. .

B. Initial User Interface

Fig. 9 is the initial user interface of the real system for the dialogue game illustrating the basic functionalities. The upper left part shows the dialogue context, including recent dialogue history alternating between the system and the learner. The last sentence is the current sentence produced by the system. The upper right section contains a set of next sentence candidates for the learner to choose, in which the first choice including Initial/Final/Tone for each syllable in the relatively larger blue box is the sentence recommended by the policy, in contrast to other sentences in black font. After the learner chooses and produces the utterance by clicking on the "Record" and "Send" buttons in the middle left, the server responses by offering the

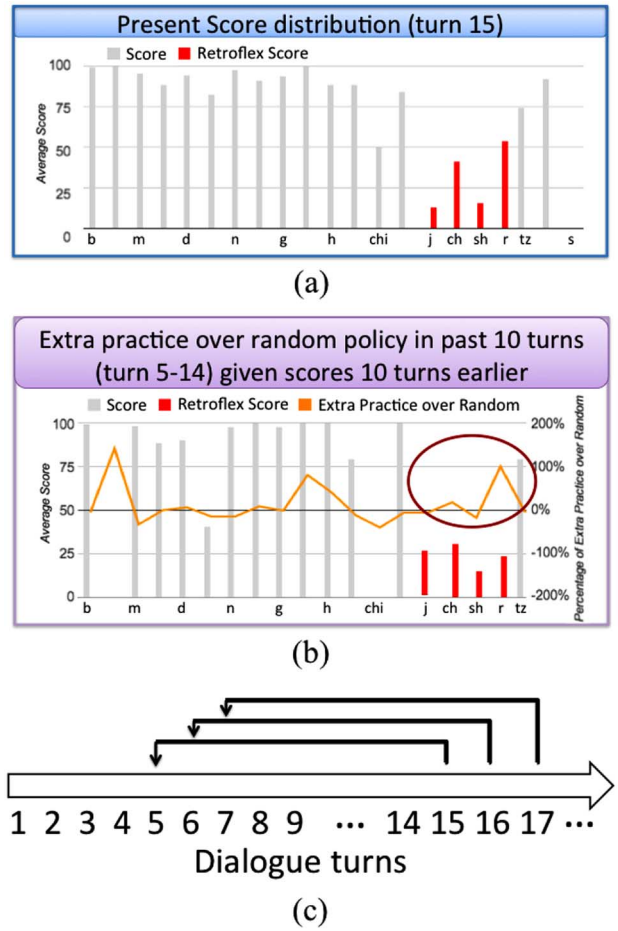

Fig. 10. On-line statistical analysis for the learning status: (a) Score distribution updated at each turn, (b) percentage of extra practice over random method in past 10 turns (yellow curve, right scale) given score distribution 10 turns earlier (gray/red bars, left scale), and (c) look-back analysis over the 10 past turns along the dialogue path.

pronunciation evaluation results for the practiced utterance and the next sentence candidates for selection. The evaluation results including the overall scores of the whole utterance and those of each Mandarin syllable are shown in the bottom half of the Fig., which includes evaluation details such as Initial/Finals, tones, timing and emphasis. The learners can also listen to their own pronunciation on the practiced sentence or that produced by a language teacher by clicking "Play" button. Other kinds of feedback such as descriptive judgments, corrective suggestions, or animation showing the correct articulation are also provided when needed.

By clicking on the "Statistics" button, the system offers online statistical analysis on the learning status for all pronunciation units including Initial/Finals and tone patterns as shown in Fig. 10. The average scores distributed over all pronunciation units as shown in Fig. 10(a) (not completely shown) are updated at each turn along with the dialogue progress. In Fig. 10 we assume the learner wishes to focus on the Retroflexed units, so the bars for those units are in red. Whenever the count of dialogue turn exceeds 11 (e.g. at turn 15 in Fig. 10), the system starts to look back and offer the extra Fig. in Fig. 10(b), that is, the percentage of extra practice opportunities compared and normalized to the random method in the past 10 turns (e.g. turns 5-14 in Fig. 10) given the score distribution of 10 turns earlier (e.g.

turn 5 in Fig. 10). Here we chose random method rather than the greedy method mentioned in section 5.3, since the latter may often result in over-emphasis on ill-pronounced units detected by the system in early dialogue turns while ignoring the other unseen units. In practical usage the random method may automatically avoid this dilemma. Such look-back analysis along the dialogue path continues at each turn as shown in Fig. 10(c). The part of the yellow curve in the brown circle in Fig. 10(b) illustrates that the policy offered more opportunities on some of the personally selected units (Retroflexed units here) than random method in the last 10 turns (turns 5-14 here), and as a result the scores of some of these units were improved as in Fig. 10(a) at turn 15 compared to 10 turns before. Such on-line analysis including updated scores and look-back analysis at each turn has been successfully implemented in the real system.

.

VII. Initial Test Results from Human Learners Using the Real System

All the experimental results reported in Section V are based on simulated learners. People may wonder if the results based on simulated learners really reflect the situations for real human learners. With the real system successfully implemented, we were actually able to evaluate the effectiveness of the dialogue game with real human learners. In the initial tests, we invited two Mandarin Chinese learners to practice their pronunciation using the working real system. Both subjects have Japanese as their native language and have learned Mandarin Chinese for 2 years (intermediate level). They are not among the 278 real learners participating in the construction of the real learner data as mentioned in Section II-B. The experiment was conducted as follows.

1) Both learners were asked to read 30 Mandarin Chinese sentences, which was the phonetically balanced and prosodically rich sentence set covering almost all frequently used Mandarin syllables and tone patterns (used in Section II-B) for constructing the read learner data. We then used NTU Chinese as the pronunciation evaluator to obtain the initial scores of all the pronunciation units in every sentences for both learners. For those pronunciation units appeared more than once, we took the averages. All the 101 units used here obtained their scores in this way.
2) The learners were asked to play with sub-dialogues (3)–(5) twice.
3) The learners were asked to read the same 30 Mandarin Chinese sentences (as in step 1) again, and we evaluated their pronunciation scores in exactly the same way.
4) We compared the scores for all the pronunciation units before and after playing with the dialogue game.

From the results for all the 101 units for the two real learners, in Fig. 11(a), we can see clearly the difference between the scores after playing with the dialogue game system (orange curves) and those before (blue bars) for both learners. Good progress was made for most of the pronunciation units. The trends for the two learners look somewhat similar, probably because they both have Japanese as their native language. Fig. 11(b) shows the average scores for each set of pronunciation units averaged from the two learners: Initials, Finals, tone

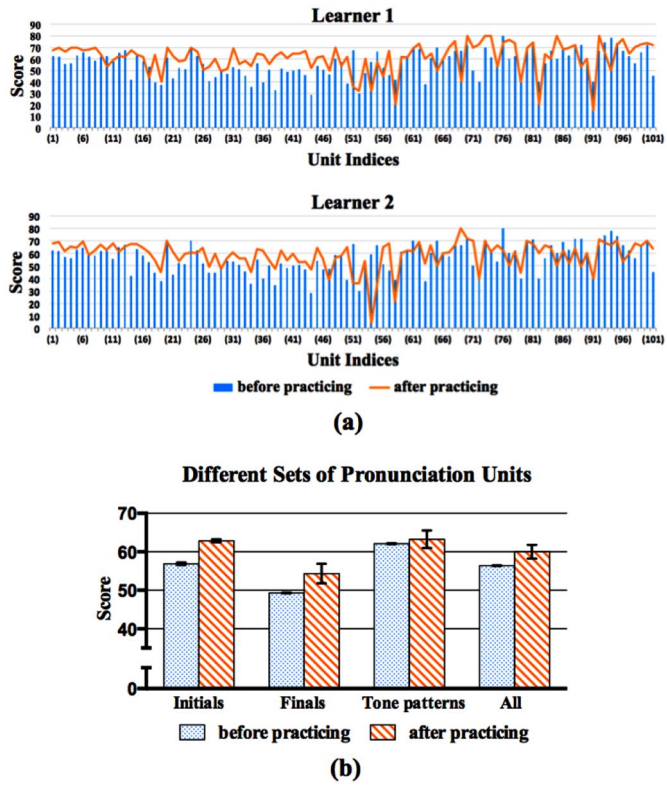

Fig. 11. (a) Detailed scores of the two human learners on all the 101 pronunciation units (listed in the order of the ID indices in Table I, different from that in Fig. 6) before and after playing with the dialogue game. (b) Average performance of the two human learners on different sets of pronunciation units before and after playing with the dialogue game.

patterns and all pronunciation units. We see good improvements for Initials and Finals (also seen for units (1)-(58) in Fig. 11(a)), probably because most Mandarin phonetic units are not too difficult for Japanese native speakers, therefore the dialogue game helped very well. The average improvement for tone patterns was much less (also seen for units (59)-(101) in Fig. 11(a)), obviously because the tone patterns are more difficult for learners and playing with only a part of the game twice may not be very useful. The overall improvement achieved here in Fig. 11(a) and (b) seems very good for the real learners playing with the sub-dialogues (3)–(5) twice only. One possible reason may be that the two real learners already have intermediate level of Mandarin, so learning may be easier for them. Another possible reason may be that the 30 testing sentences were exactly those used for constructing the real learner data and generating the simulated learners for policy training in Section II-B. Therefore all the unit co-existence relationships between phonetic units, and tone patterns, and all context relationships among units, were previously seen somehow when training the system policy. Note that this set of 30 sentences was the only smallest phonetically balanced and prosodically rich sentence set we had at the moment. These results demonstrate that the approach and framework proposed in this paper can help real learners improve their pronunciation. Further user studies on real learners with different learning levels and using different testing sentences can be conducted in the future.

VIII. Conclusion Remarks

In this paper we presented a framework of recursive dialogue game for personalized CAPT with an example of learning Mandarin Chinese. A series of tree-structured sub-dialogues were linked recursively as the script for the game. The policy was to select the best sentence for practice at each turn during the dialogue considering the learning status of the learner and the future paths within the dialogue script. It was optimized by an MDP with a high-dimensional continuous state space of pronunciation units and trained using fitted value iteration. A GMM with BIC was proposed to construct learner simulation model to generate simulated learners for training the MDP. A series of experiments conducted in different ways of using the dialogue game (sequential or arbitrary order of the sub-dialogues, aiming for learning all pronunciation units or focusing on selected subsets of units) showed very promising results and the effectiveness of the proposed approach. The framework was also successfully implemented as a real cloud-based system based on NTU Chinese, which was tested by human learners and good improvements were obtained.

References

[1] S. Witt and S. Young, "Phone-level pronunciation scoring and assessment for interactive language learning," *Speech Commun.*, vol. 30, pp. 95–108, 2000.

[2] J. Zheng, C. Huang, M. Chu, F. Soong, and W.-P. Ye, "Generalized segment posterior probability for automatic mandarin pronunciation evaluation," in *Proc. ICASSP*, 2007, pp. 201–204.

[3] A. M. Harrison, W.-K. Lo, X.-J. Qian, and H. Meng, "Implementation of an extended recognition network for mispronun- ciation detection and diagnosis in computer-assisted pronunciation training," in *Proc. SLaTE*, 2009.

[4] M. Suzuki, Y. Qiao, N. Minematsu, and K. Hirose, "Pronunciation proficiency estimation based on multilayer regression analysis using speaker-independent structural features," in *Proc. Interspeech*, 2010.

[5] N. Minematsu, "Yet another acoustic representation of speech sounds," in *Proc. ICASSP*, 2004.

[6] M. Suzuki, Y. Qiao, N. Minematsu, and K. Hirose, "Integration of multilayer regression analysis with structure-based pronunciation assessment," in *Proc. Interspeech*, 2010.

[7] T. Zhao, A. Hoshino, M. Suzuki, N. Minematsu, and K. Hirose, "Automatic Chinese pronunciation error detection using SVM trained with structural features," in *Proc. IEEE Workshop Spoken Lang. Technol.*, 2012, pp. 473–478.

[8] H. Strik, F. Cornillie, J. Colpaert, J. van Doremalen, and C. Cucchiarini, "Developing a call system for practicing oral proficiency: How to design for speech technology, pedagogy and learners," in *Proc. SLaTE*, 2009.

[9] H. Strik, P. Drozdova, and C. Cucchiarini, "GOBL: Games online for basic language learning," in *Proc. Interspeech*, 2011.

[10] K. Zechner, D. Higgins, X. Xi, and D. M. Williamson, "Automatic scoring of non-native spontaneous speech in tests of spoken English," *Speech Commun.*, vol. 51, no. 10, pp. 883–895, 2009.

[11] W. Xiong, K. Evanini, K. Zechner, and L. Chen, "Automated content scoring of spoken responses containing multiple parts with factual information," in *Proc. SLaTE*, 2013.

[12] L. Chen and K. Zechner, "Applying rhythm features to automatically assess non-native speech," in *Proc. Interspeech*, 2011.

[13] Y. Xu, "Language technologies in speech-enabled second language learning games: From reading to dialogue," Ph.D. dissertation, Mass. Inst. of Technol., Cambridge, MA, USA, 2012.

[14] S. Seneff, C. Wang, and C. Y. Chao, "Spoken dialogue systems for language learning," in *Proc. HLT-NAACL*, 2007.

[15] C. Y. Chao, S. Seneff, and C. Wang, "An interactive interpretation game for learning Chinese," in *Proc. SLaTE*, 2007.

[16] C. Wang and S. Seneff, "A spoken translation game for second language learning," in *Proc. AIED*, 2007.

[17] Y. Xu and S. Seneff, "A generic framework for building dialogue games for language learning: Application in the flight domain," in *Proc. SLaTE*, 2011.

[18] (2009) NTU Chinese. [Online]. Available: http://chinese.ntu.edu.tw/

[19] (1999) Rosetta Stone. [Online]. Available: http://www.rosettastone.com/

[20] English Town (1996). [Online]. Available: http://www.englishtown.com.tw/

[21] T. Misu, K. Sugiura, K. Ohtake, C. Hori, H. Kashioka, H. Kawai, and S. Nakamura, "Modeling spoken decision making dialogue and optimization of its dialogue strategy," in *SIGdial*, 2010.

[22] D. J. Litman and S. Silliman, "ITSPOKE: An intelligent tutoring spoken dialogue system," in *HLT-NAACL*, 2004.

[23] S. Young, M. Gasic, B. Thomson, and J. D. Williams, "POMDP-based statistical spoken dialogue systems: A review," *Proc IEEE*, vol. 101, no. 5, pp. 1160–1179, May 2013.

[24] J. D. Williams, I. Arizmendi, and A. Conkie, "Demonstration of AT&T "let's go": A production-grade statistical spoken dialogue system," in *Proc. SLT*, 2010.

[25] A. Raux and M. Eskenazi, "Using task-oriented spoken dialogue systems for language learning: Potential, practical applications and challenges," in *Proc. InSTIL/ICALL Symp.*, 2004.

[26] W. L. Johnson, "Serious use of a serious game for language learning," *Int. J. Artif. Intell. Educat.*, vol. 20, pp. 175–195, 2010.

[27] P.-H. Su, Y.-B. Wang, T.-H. Yu, and L.-S. Lee, "A dialogue game framework with personalized training using reinforcement learning for computer-assisted language learning," in *Proc. ICASSP*, 2013, pp. 8213–8217.

[28] R. Bellman, *Dynamic programming*. Princeton, NJ, USA: Princeton Univ. Press, 1957.

[29] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY, USA: Wiley Interscience, 1994.

[30] P.-H. Su, Y.-B. Wang, T.-H. Wen, T.-H. Yu, and L.-S. Lee, "A recursive dialogue game framework with optimal policy offering personalized computer-assisted language learning," in *Proc. Interspeech*, 2013.

[31] R. Ellis, S. Loewen, and R. Erlam, "Implicit and explicit corrective feedback and the acquisition of l2 grammar," *Studies Second Lang. Acquisit.*, vol. 28, pp. 339–368, 2006.

[32] A. Ferreira, "An experimental study of effective feedback strategies for intelligent tutorial systems for foreign language," *Adv. Artif. Intell. -IBERAMIA-SBIA*, pp. 27–36, 2006.

[33] T. Heift, "An experimental study of effective feedback strategies for intelligent tutorial systems for foreign language," in *Proc. ReCALL*, 2004, pp. 416–431.

[34] R. M. DeKeyser, *Practice in Second Language - Perspectives from Applied Linguistics and Cognitive Psychology*. Cambridge, U.K.: Cambridge Univ. Press, 2007.

[35] Y. Yoshimura and B. MacWhinney, "The effect of oral repetition on l2 speech fluency: An experimental tool and language tutor," in *Proc. SLaTE*, 2007.

[36] My Language Exchange. [Online]. Available: http://www.mylanguageexchange.com/

[37] Riswanto and E. Haryanto, "Improving students' pronunciation through communicative drilling technique at senior high school (SMA) 07 South Bengkulu, Indonesia," *Int. J. Human. Soc. Sci.*, vol. 2, no. 21, pp. 82–87, 2012.

[38] C. Maxwell, "Role play and foreign language learning," in *Proc. Annu. Meting Jpn. Assoc. Lang. Teachers*, 1997.

[39] A. N. Burnetas and M. N. Katehakis, "Optimal adaptive policies for Markov decision processes," *Math. Operat. Res.*, 1995.

[40] S. Singh, M. Kearns, D. Litman, and M. Walker, "Reinforcement learning for spoken dialogue systems," in *Proc. NIPS*, 1999.

[41] L. Daubigney, M. Geist, and O. Pietquin, "Off-policy learning in large-scale POMDP-based dialogue systems," *ICASSP*, pp. 4989–4992, 2012.

[42] Y. Engel, S. Mannor, and R. Meir, "Bayes meets Bellman: The Gaussian process approach to temporal difference learning," in *Proc. ICML*, , 2003.

[43] F. S. Melo, S. P. Meyn, and M. I. Ribeiro, "An analysis of reinforcement learning with function approximation," in *Proc. ICML*, 2008.

[44] R. Hogg, J. McKean, and A. Craig, *Introduction to Mathematical Statistics*. Upper Saddle River, NJ, USA: Pearson Prentice-Hall, 2005.

[45] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. R. Statist. Soc.. Ser. B (Methodol.)*, vol. 39, pp. 1–38, 1977.

[46] W. Zucchini, "An introduction to model selection," *J. Math. Psychol.*, vol. 44, no. 22006, pp. 41–61.
[47] K. Hirose, S. Kawano, S. Konishi, and M. Ichikawa, "Bayesian information criterion and selection of the number of factors in factor analysis models," *J. Data Sci.*, vol. 9, pp. 243–259, 2011.
[48] J. Schatzmann, K. Weilhammer, M. Stuttle, and S. Young, "A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies," *Knowl. Eng. Rev.*, vol. 21, pp. 97–126, 2006.
[49] H. Ai and F. Weng, "User simulation as testing for spoken dialog systems," in *Proc. SIGdial*, 2008.
[50] J. Schatzmann, M. N. Stuttle, K. Weilhammer, and S. Young, "Effects of the user model on simulation-based learning of dialogue strategies," in *Proc. ASRU*, 2005.
[51] M. A. Walker, D. J. Litman, C. A. Kamm, and A. Abella, "Paradise: A framework for evaluating spoken dialogue agents," in *Proc. ACL*, 1997.
[52] F. Jurc1cek, M. G. S. Keizer, F. Mairesse, B. Thomson, K. Yu, and S. Young, "Real user evaluation of spoken dialogue systems using Amazon Mechanical Turk," in *Proc. Interspeech*, 2011.

Pei-hao Su received his B.S. degree from the Department of Electrical Engineering, National Taiwan University, in 2012 and the M.S. degree from the Graduate Institute of Communication Engineering, National Taiwan University, in 2013. While at NTU, he worked on dialogue game systems for computer-assisted language learning in the Speech Processing Laboratory.

He is currently a Ph.D. student in the Dialogue Systems Group at the University of Cambridge. His research interests include spoken language understanding and dialogue modeling.

Chuan-hsun Wu was born in 1991. He received his B.S. degree from the Department of Computer Science and Information Engineering, National Taiwan University, in 2013.

He is currently an M.S. student in the Speech Processing Laboratory, National Taiwan University. His research interests include computer-assisted language learning applications.

Lin-shan Lee (F'93) received the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA.

He has been a Professor of electrical engineering and computer science at the National Taiwan University, Taipei, Taiwan, since 1982 and holds a joint appointment as a Research Fellow of Academia Sinica, Taipei. His research interests include digital communications and spoken language processing. He developed several of the earliest versions of Chinese spoken language processing systems in the world including text-to-speech systems, natural language analyzers, dictation systems, and voice information retrieval systems.

Dr. Lee was on the Board of Governors (1995), the Vice President for International Affairs (1996–1997) and the Awards Committee chair (1998–1999) of the IEEE Communications Society. He was a member of the Board of International Speech Communication Association (ISCA 2002–2009), a Distinguished Lecture (2007–2008) and a member of the Overview Paper Editorial Board (2009–2010) of the IEEE Signal Processing Society, and the general chair of ICASSP 2009 in Taipei. He is a fellow of ISCA since 2010, and received the Meritorious Service Award from IEEE Signal Processing Society in 2011, and IEEE ComSoc/KICS Exemplary Global Service Award from IEEE Communication Society in 2014.