

A Cloud-based Personalized Recursive Dialogue Game System for Computer-Assisted Language Learning

Pei-hao Su ^{#1}, Tien-han Yu [#], Ya-Yunn Su ^{*}, and Lin-shan Lee ^{*2}

[#]Graduate Institute of Communication Engineering, National Taiwan University

^{*}Graduate Institute of Computer Science and Information Engineering, National Taiwan University

¹ r00942135@ntu.edu.tw, ² lslee@gate.sinica.edu.tw

Abstract

In this paper we present the design and experimental results of a cloud-based personalized recursive dialogue game system for computer-assisted language learning. A number of tree-structured sub-dialogues are used sequentially and recursively as the script for the game. The dialogue policy at each dialogue turn is optimized to offer the most appropriate training sentence for every individual learner considering the learning status, such that the learner can have the scores for all selected pronunciation units exceeding a pre-defined threshold in minimum number of turns. The policy is modeled as a Markov Decision Process (MDP) with high-dimensional continuous state space and trained with a huge number of simulated learners generated from a corpus of real learner data. A real cloud-based system is implemented and the experimental results demonstrate promising outcomes.

Index Terms: Computer-Assisted Language Learning, Dialogue Game, Continuous State Markov Decision Process, Fitted Value Iteration, Gaussian Mixture Model

1. Introduction

Computer-assisted language learning (CALL) systems offer various advantages for language learning such as immersive environment and corrective feedback during the learning process. Thanks to the explosive development of technology in recent years, high performance computers, tablets and even smartphones are common nowadays. It is convenient and useful to embed systems into these devices. Also, The use of speech processing technologies has been considered a good approach to provide effective assistance [1, 2, 3, 4, 5].

“Rosetta Stone” [6] and “byki” [7] are useful applications that provide multifaceted functions including pronunciation evaluation and corrective feedback. However, sentence-level practice lacks opportunities for language interaction and an immersive language learning environment [8, 9]. Spoken dialogue systems [10, 11, 12, 13, 14] are regarded as excellent solutions to provide language interaction scenarios. Recently we presented a dialogue game framework [15] in which proper training sentences

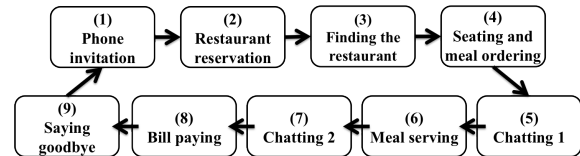


Figure 1: The script of the recursive dialogue game in the restaurant scenario: starting from (1) Phone invitation and (2) Restaurant reservation, after (9) Saying goodbye returning to (1) for next meal.

at each dialogue turn are selected for each individual learner during the interaction based on the learning status. The dialogue framework was modeled as a Markov decision process (MDP) trained with reinforcement learning [16, 17], and the learning status was based on NTU Chinese [18], a Mandarin Chinese pronunciation evaluation tool. One limitation of this framework is that the discrete state representation was in short of full observation of the learner’s learning status. Furthermore, its training assumed a fixed number of dialogue turns; this is impractical and inflexible.

In a companion paper, we propose a new dialogue game framework for language learning [19]. A number of sub-dialogue trees are used sequentially and recursively. The leaves of the last tree are linked to the root of the first tree, making the dialogue paths infinitely long. At any dialogue turn there are a number of training sentences that can be selected. The goal of the policy is to select the training sentence at each dialogue turn based on the learning status of the learner, such that the learner’s scores for all selected pronunciation units exceed a pre-defined threshold in a minimum number of turns. The framework is again modeled as an MDP, but here the MDP is realized in a high-dimensional continuous state space for a more precise representation of the learning status. This framework has been successfully implemented under a cloud-based environment and displayed on the iOS platform. This paper presents the complete design and the preliminary experimental results of this cloud-based dialogue game system.

2. Proposed recursive dialogue game framework

2.1. Recursive dialogue game concept and framework

The progress of the dialogue game is based on the script of a series of tree-structured sub-dialogues cascaded into a loop, with the last sub-dialogue linked to the first. In preliminary experiments, the whole dialogue set contains conversations between roles A and B — one the computer and the other the learner. After each utterance produced by one speaker, there are a number of choices for the other speaker's next sentence. Figure 1 shows the recursive structure of the script in the restaurant scenario. In all, nine sub-dialogues with 176 turns are used in the experiments. The whole dialogue starts with the phone invitation scenario, followed by restaurant reservation and so on, all the way to the last sub-dialogue of saying goodbye. After the last tree, the progress restarts at the first phone invitation sub-dialogue again for the next meal. This makes the dialogue continue infinitely. Figure 2 is a segment of the sub-dialogue “Seating and meal ordering”, where A is the waiter and B the customer.

Since both the computer and the learner have multiple sentence choices in each turn, every choice influences the future path significantly; this results in a very different distribution of pronunciation unit counts for the learners to practice. The dialogue policy here is to select the most appropriate sentence for the learner to practice at each turn considering the learning status, such that more opportunities are given to practice poorly produced pronunciation units along the dialogue path. In this way the learner can achieve the goal of having the scores of all pronunciation units exceed a pre-defined threshold in a minimum number of turns. Also, they receive pronunciation performance feedback immediately after each utterance pronounced.

The above recursive dialogue game is modeled by an MDP with the desired optimal policy trained with the Fitted Value Iteration (FVI) algorithm. A learner generation model is developed to generate simulated learners from real learner data to be used in the FVI algorithm.

The overall system block diagram of the proposed framework is shown in Figure 3. Interaction between the learner and the system involves Utterance Input from the learner and Selected Sentences from the system. The Automatic Pronunciation Evaluator scores the performance of each pronunciation unit in the utterance. These quantitative assessments are sent to the Pedagogical Dialogue Manager, which is driven by the Sentence Selection Policy for choosing the next sentence for the learner. A set of Real Learner Data is used to construct the Learner Simulation Model, which generates the Simulated Learners to train the Sentence Selection Policy based on the Script of Cascaded Sub-dialogues using the Fitted Value Iteration

TURN	CONTENT
A1	歡迎光臨。請問訂位了嗎？ Welcome. Did you make any reservation?
B1	沒有。 No, I didn't. 訂了，我姓王。 Yes, My name is Wang.
A2	現在客滿，您可能要稍等一下。 It's full now, you may have to wait for a while. 正在整理桌面，請稍候。 We are cleaning up the table, please wait. 請等一下，馬上就替您帶位。 Please wait, we will lead you to your seat. 這邊請。 Here please.
B2	謝謝。 Thank you. 好的。 Okay. 下一個就是我們了嗎？謝謝 Are we next? Thanks.
A3	這個位子可以嗎？ Are the seats okay? 靠窗的桌子，好不好？ The table near window, is it okay? 對不起，位子有點兒擠。 Sorry, the seats are small. 我再幫你加一張椅子。 I'll add a chair for you.

Figure 2: A segment of the dialogue script for the dialogue game example in a restaurant conversation scenario.

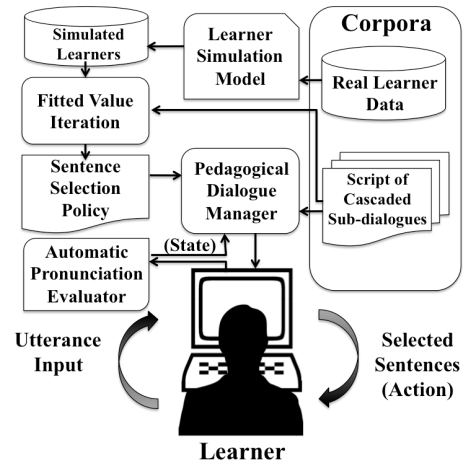


Figure 3: System block diagram of the proposed recursive dialogue game framework.

algorithm.

2.2. Simulated learner generation from real learner data

The real learner data used in these experiments were collected in 2008 and 2009. In total there were 278 Mandarin Chinese learners at the National Taiwan University (NTU) from 36 countries with balanced gender, each pronouncing 30 sentences selected by language teachers. NTU Chinese, a Mandarin pronunciation evaluation tool developed at NTU [18], was used as the Automatic Pronunciation Evaluator in Figure 3. It assigned scores from 0 to 100 to each pronunciation unit in every utterance of the real learner data. The scores of each utterance pronounced by a learner are used to construct a pronunciation score vector (PSV), whose dimensionality is the number of the pronunciation units considered. Every component of the PSV is the average score of the corresponding unit in the utterance; those units unseen in the utterance are viewed as missing data and solved by the expectation-maximization (EM) algorithm [20, 21]. The

PSVs from all utterances produced by all real learners are used to train a Gaussian mixture model (GMM), here referred to as the Learner Simulation Model. This is shown in Figure 4.

For MDP policy training, when starting a new dialogue game, we randomly select a Gaussian mixture component as a simulated learner [22, 23, 24]. When a sentence is to be pronounced, a randomly sampled PSV from this mixture yields the scores for the units in this sentence as the simulated utterance. Since the goal of the dialogue is to provide proper sentences for each learner until their pronunciation performance for every unit reaches a pre-defined threshold, we further develop an incremental pronunciation improvement model for the simulated learners. Details about the simulated learners are in the companion paper [19].

2.3. Markov decision process

A Markov decision process (MDP) [25] is a framework that models decision making problems, represented by the 5-tuple $\{S, A, R, T, \gamma\}$: 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 state s , a reward r is received and the state is transmitted to new state s' . Solving the MDP consists in determining an infinite state transition process called a *policy* that maximizes the expected total discounted reward from state s (or value function): $V^\pi(s) = E[\sum_{k=0}^{\infty} \gamma^k r_k | s_0 = s, \pi]$, where 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 of taking action a at state s : $Q^\pi(s, a) = E[\sum_{k=0}^{\infty} \gamma^k r_k | s_0 = s, a_0 = a, \pi]$. Thus, the optimal policy π^* can be expressed as $\pi^*(s) = \arg \max_{a \in A} Q(s, a)$ by a greedy selection of the state-action pair. The goal of finding the optimal policy is therefore equivalent to maximizing these Q functions.

2.4. MDP framework on dialogue game

We describe how the dialogue game is modeled using MDP.

2.4.1. Continuous state space

The state represents the learner’s learning status. It consists of the scores obtained for every pronunciation unit given by the Automatic Pronunciation Evaluator in Figure 3, each a continuous value ranging from 0 to 100 and directly observable by the system. This results in the high-dimensional continuous state space $s \in [0, 100]^U$, where U is the total number of pronunciation units considered. In addition, as the system must determine which

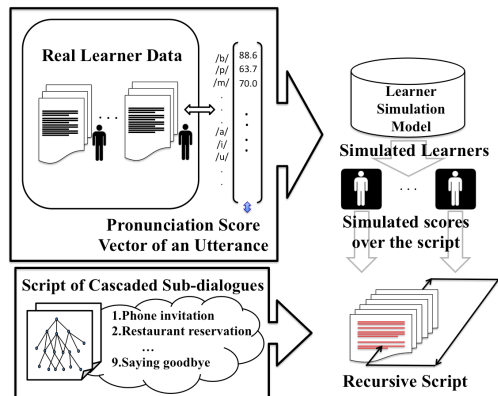


Figure 4: Learner Simulation Model for simulated learner generation.

dialogue turn the learner is in, the index of dialogue turn t is also included in the state space.

2.4.2. Action set

At each state with dialogue turn t , the system’s action is to select one out of a number of available sentence options for the learner to practice. The number of actions is the number of next available sentences to be chosen for the learner at the turn.

2.4.3. Reward definition

A dialogue *episode* E contains a sequence of state transitions $\{s_0, a_0, s_1, a_1, \dots, s_K\}$, where s_K represents the terminal state. As mentioned, the goal here is to train a policy that can at each turn offer the learner the best selected sentence to practice considering the learning status, such that the learner’s scores for all selected pronunciation units exceed a pre-defined threshold 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, where scores of all pronunciation units reach a certain threshold. The final return R is then the sum of the obtained rewards: $R = \sum_{k=0}^K r_k$. In addition, a timeout count of state transitions J is used to limit episode lengths.

2.4.4. Fitted value iteration (FVI) algorithm

For the high-dimensional continuous state space, we use the function approximation method [26, 27, 28] to approximate the exact Q value function with a set of m basis functions:

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

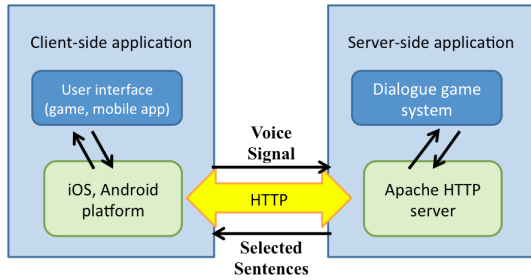


Figure 5: System architecture of the cloud-based system.

where θ is the parameter (weight) vector corresponding to the basis function vector $\phi(s, a)$. The goal of finding the optimal policy can then be reduced to finding the appropriate parameters θ for a good approximation $\hat{Q}_\theta(s, a)$ of $Q(s, a)$. A *sampled* version of the Bellman backup operator \hat{B} is introduced for the i -th sampled transition (s_i, a_i, r_i, s'_i) as

$$\hat{B}(Q(s_i, a_i)) = r_i + \gamma \max_{a \in A} Q(s'_i, a_i). \quad (2)$$

With a batch of transition samples $\{s_j, a_j, r_j, s'_j | j = 1, \dots, N\}$, least-squares regression can be performed to find the new parameter vector θ_n at the n -th iteration so that $\hat{Q}_{\theta_n}(s, a)$ approaches $Q(s, a)$ as precisely as possible. The parameter vector is updated as

$$\theta_{n+1} = \arg \min_{\theta \in \mathbb{R}^M} \sum_{j=1}^N (\hat{Q}_{\theta_n} - \hat{B}(Q(s_j, a_j)))^2 + \frac{\lambda}{2} \|\theta\|^2, \quad (3)$$

where the second term is the 2-norm regularized term determined by λ to prevent over-fitting.

3. Cloud-based system design and implementation

3.1. System overview

We have implemented the Mandarin Chinese dialogue game core engine as a cloud-based system. To provide good operability, the system is exposed through REST API. Figure 5 shows our system architecture. A web server accepts HTTP requests with a URL mapping to our dialogue system service. The web server then passes the HTTP request including the parameters and translates the request into a corresponding voice signal call to the pedagogical dialogue manager. After the next sentence for practice is selected by the dialogue manager, this selected sentence is packed into a HTTP response. User-specific data, such as pronunciation scores and profile, are stored in a separate database. In this way, developers can build applications for various platforms, such as a web page or a mobile app or a flash game, using any HTTP library that can issue the REST calls.



Figure 6: An example view of our designed system interface.

3.2. Initial user interface

Figure 6 is the initial user interface of our dialogue game showing the fundamental functionalities. The left part shows the dialogue progress, which alternates between the waiter and the customer. The last two sections are the current sentence produced by the waiter (system) and the sentence candidates for the customer (learner) to choose, while the other sections list the past sentences spoken. The “Hide/Show” button on the upper right switch off/on the display of the past sentences. The customer chooses to produce one sentence by clicking on the “Start Recording” button. Note that there is a “BEST CHOICE” labeled on one sentence candidate of the customer, it is the one recommended by the optimized sentence selection policy mentioned above. In addition, by clicking on the blue “play” icon we can listen to the sentence spoken by the waiter again. When clicking on the “analysis” icon on the past sentences of the customer, the system shows the evaluation result of each unit within the selected sentence, which is shown on the right part. The evaluation result indicates the pronunciation performance on the whole utterance and on each Mandarin syllable, including scores of Initial/Finals, tone, timing and emphasis. This offers detailed assessment of the learner’s pronunciation.

The complete cloud-based system has been successfully implemented and operated in real time. It is also submitted to the demonstration session of SLATE 2013 [29].

4. Experiments

4.1. Experimental Setup

Experiments were performed on the complete script of nine sub-dialogue trees for Mandarin Chinese learning as described in Section 2.1. The results below are for the computer as role A and the learner as role B. Totally 82 Mandarin pronunciation units including 58 phonetic units (Initial/Finals of Mandarin syllables) and 24 tone

patterns (uni/bi-tone) were considered, and three cases were tested: learning tone patterns only, phonetic units only, and both. NTU Chinese [18] was used as the automatic pronunciation evaluator for unit scoring and immediate feedback for the learners. In the MDP setting, the terminal state s_K was defined as the situation that all pronunciation units considered were produced with scores over 75 more than eight times. The reward at the dialogue terminal state r_K was set to 300 and timeout count J was 500. Multivariate Gaussian functions of 82 dimensions served as the basis function $\phi(s, a)$ in (1) to represent the Q value function. The number of the basis functions was set 5, and these Gaussian functions were spread evenly on the state space. 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 saved for another GMM to generate simulated learners in the testing phase. In our work, the MDP testing result was the average of 50 testing simulated learners. Also, Bayesian information criterion (BIC) [30, 31] was employed on GMM to balance the model likelihood and parameter complexity. In the experiment, simulated learners were generated to go through the nine sub-dialogue trees in either sequential and recursive order or arbitrary order until the terminal state s_K was reached.

4.2. Experimental Result

4.2.1. Number of dialogue turns needed

In Figure 7, we plot the number of turns needed to reach the terminal state as a function of the number of training iterations. Clearly the three solid curves (labeled “Sequential”) for different sets of target units considered yielded promising results. The number of needed turns for learning tone patterns alone, phonetic units alone, and both converged at 179.88, 203.42, 215.58 turns respectively. Clearly as the number of target units is smaller, the needed turns is smaller. Note that the needed turns of considering phonetic units alone (203.42) is only slightly smaller than considering both phonetic units and tone patterns (215.58), while that of considering only tone patterns (179.88) is much smaller. Different sets of pronunciation units are presented in the training sentences in any case with different distributions. The above results indicate that when considering only phonetic units, the practice may cover many tone patterns as well. That is to say, considering a set of units together as target learning units at a time may result in less total number of training sentences than considering the same set of units in separated times. In addition, since there were 84 turns in all for role B in the nine consecutive sub-dialogues, the results indicated that going through all nine trees and restarting

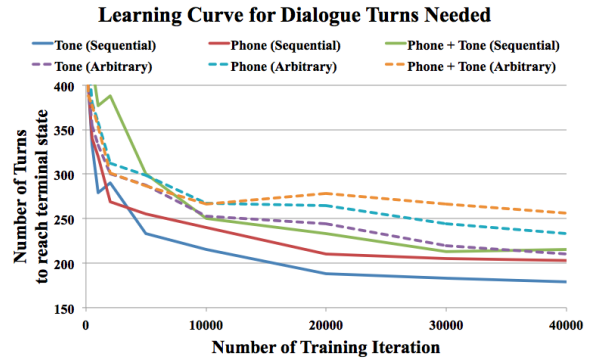


Figure 7: Number of dialogue turns needed with respect to different number of training iterations.

from the first sub-dialogue was necessary for the testing simulated learners here.

The dashed curves (labeled “Arbitrary”) show the results of using the nine sub-dialogue trees in a different scenario, in which the learner chose to practice the sub-dialogue trees in an arbitrary order. For example, the learner could jump to sub-dialogue four after finishing sub-dialogue two (after restaurant reservation, the learner wishes to learn how to order meals first). The same three cases (tone patterns only, phonetic units only, and both) tested in this scenario converged at 210.16, 233.64, and 256.82 turns respectively as shown in Figure 7. The extra turns needed compared to the sequential order scenario shows the trade-off between the user’s free will to interact with the dialogue game and the dialogue turns needed to learn all target units well enough.

4.2.2. Focused learning for specific sets of pronunciation units

From section 4.2.1 we learned the effectiveness of the system policy. The system provided personalized pronunciation unit practice as efficient as possible to each individual learner. However, some language learners might already know their pronunciation status in advance and wished to focus their learning on a specific set of units using the dialogue game system. We therefore would like to test the learned policies considering different target units as discussed in section 4.2.1. In the experiments below, the simulated learner selected certain number of units randomly as the units to be focused on while ignoring the scores of all other units.

Table 1 shows the dialogue turn needed for focused learning of 10 tone patterns (row(1)), 20 phonetic units (row(2)), and 20 phonetic units or tone patterns (row(3)) using the policies learned in section 4.2.1 respectively, either following the sub-dialogue trees sequentially (labeled “Sequential”) or in arbitrary order (labeled “Arbitrary”). Each result is the average over 100 simulated learners. This shows different ways of utilizing the dia-

Table 1: Number of dialogue turns needed for focused learning on a specific number of pronunciation units.

Target units	Number of units focused	Number of turns needed	
(1) Tone patterns	10	Sequential	140.28
		Arbitrary	159.77
(2) Phonetic units	20	Sequential	173.53
		Arbitrary	205.09
(3) Phone + Tone	20	Sequential	179.11
		Arbitrary	209.16

logue game developed here.

From Table 1 we can see that a significant number of turns were needed even if only 10 units are focused on, but the policy 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 a small number of them. Also, some low frequency units, if selected by the learner, may require more turns to be practice in the dialogue.

5. Conclusions

We presented a cloud-based recursive dialogue game with an optimized policy offering personalized learning materials for CALL. A series of recursive tree-structured sub-dialogues are used as the script for the game. The policy is to offer the proper sentence for practice at each turn considering the learning status of the learner. It was optimized by an MDP with a high-dimensional continuous state space and trained using fitted value iteration. The cloud-based system has been successfully completed and operated in real time. Experimental results of sequential and arbitrary order usage showed promising results and the effectiveness of the proposed approach.

6. References

- [1] M. Eskenazi, "An overview of spoken language technology for education," in *Speech Communication*, vol. 51, 2009, pp. 832–844.
- [2] C. Cucchiari, J. van Doremalen, and H. Strik, "Practice and feedback in l2 speaking: an evaluation of the disco call system," in *Interspeech*, 2012.
- [3] Y. Xu, "Language technologies in speech-enabled second language learning games: From reading to dialogue," Ph.D. dissertation, Massachusetts Institute of Technology, 2012.
- [4] X. Qian, H. Meng, and F. Soong, "The use of DBN-HMMs for mispronunciation detection and diagnosis in l2 english to support computer-aided pronunciation training," in *Interspeech*, 2012.
- [5] T. Zhao, A. Hoshino, M. Suzuki, N. Minematsu, and K. Hirose, "Automatic chinese pronunciation error detection using svm trained with structural features," in *Proceedings IEEE Workshop on Spoken Language Technology*, 2012.
- [6] (1999) Rosetta Stone. [Online]. Available: <http://www.rosettastone.com/>
- [7] (2013) byki. [Online]. Available: <http://www.byki.com/>
- [8] D. Christian, *Profiles in Two-Way Immersion Education. Language in Education: Theory and Practice* 89., 1997.
- [9] W. L. Johnson, "Serious use of a serious game for language learning," in *International Journal of Artificial Intelligence in Education*, 2010.
- [10] S. Young, M. Gasic, B. Thomson, and J. Williams, "Pomdp-based statistical spoken dialogue systems: a review," in *Proceedings of the IEEE*, vol. 99, 2013, pp. 1–20.
- [11] A. Raux and M. Eskenazi, "Using task-oriented spoken dialogue systems for language learning: potential, practical applications and challenges," in *InSTIL/ICALL Symposium 2004*, 2004.
- [12] 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.
- [13] Y. Xu and S. Seneff, "A generic framework for building dialogue games for language learning: Application in the flight domain," in *Proc. SLaTE*, 2011.
- [14] S. Lee and M. Eskenazi, "Incremental sparse bayesian method for online dialog strategy learning," *Journal of Selected Topics Signal Processing*, 2012.
- [15] 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 *ICASSP*, 2013.
- [16] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1999.
- [17] R. Bellman, *Dynamic programming*. Princeton University Press, 1957.
- [18] (2009) NTU Chinese. [Online]. Available: <http://chinese.ntu.edu.tw/>
- [19] 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 *Interspeech*, 2013.
- [20] R. Hogg, J. McKean, and A. Craig, *Introduction to Mathematical Statistics*. Pearson Prentice Hall, 2005.
- [21] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," in *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, 1977, pp. 1–38.
- [22] J. Schatzmann, K. Weilhammer, M. Stuttle, and S. Young, "A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies," in *The Knowledge Engineering Review*, vol. 00:0, 2006, pp. 1–24.
- [23] H. Ai and F. Weng, "User simulation as testing for spoken dialog systems," in *SIGdial*, 2008.
- [24] J. Schatzmann, M. N. Stuttle, K. Weilhammer, and S. Young, "Effects of the user model on simulation-based learning of dialogue strategies," in *ASRU*, 2005.
- [25] A. N. Burnetas and M. N. Katehakis, "Optimal adaptive policies for markov decision processes," *Mathematics of Operations Research*, 1995.
- [26] L. Daubigney, M. Geist, and O. Pietquin, "Off-policy learning in large-scale pomdp-based dialogue systems," in *ICASSP*, 2012.
- [27] Y. Engel, S. Mannor, and R. Meir, "Bayes meets bellman: The gaussian process approach to temporal difference learning," in *ICML*, 2003.
- [28] F. S. Melo, S. P. Meyn, and M. I. Ribeiro, "An analysis of reinforcement learning with function approximation," in *ICML*, 2008.
- [29] P.-H. Su, T.-H. Yu, Y.-Y. Su, and L.-S. Lee, "NTU Chinese 2.0: A personalized recursive dialogue game for computer-assisted learning of mandarin chinese," in *Proc. SLaTE (submitted)*, 2013.
- [30] W. Zucchini, "An introduction to model selection," in *Journal of Mathematical Psychology*, vol. 44, 2006, pp. 41–61.
- [31] K. Hirose, S. Kawano, S. Konishi, and M. Ichikawa, "Bayesian information criterion and selection of the number of factors in factor analysis models," in *Journal of Data Science*, vol. 9, 2011, pp. 243–259.