A Recursive Dialogue Game Framework with Optimal Policy Offering Personalized Computer-Assisted Language Learning

Pei-hao Su [#1], Yow-Bang Wang [*], Tsung-Hsien Wen [*], Tien-han Yu [#], and Lin-shan Lee [#*2]

[#] Graduate Institute of Communication Engineering, National Taiwan University
[*]Graduate Institute of Electrical Engineering, National Taiwan University
[1] r00942135@ntu.edu.tw, [2] lslee@gate.sinica.edu.tw

Abstract

This paper introduces a new recursive dialogue game framework for personalized computer-assisted language learning. A series of sub-dialogue trees are cascaded into a loop as the script for the game. At each dialogue turn there are a number of training sentences to be selected. The dialogue policy is optimized to offer the most appropriate training sentence for an individual learner at each dialogue turn considering the learning status, such that the learner can have the scores for all 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. Experiments demonstrate promising results for the approach.

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

1. Introduction

Education and learning has long been the most important way for individuals to improve their quality of life [1, 2]. With the explosive development of technologies including computers, hand-held devices, the Internet, and social networks, learners today can absorb knowledge not only from printed materials in classrooms, but also benefit more from efficient and effective learning processes such as distance learning [3] and peer discussion [4] with people worldwide. "Coursera" [5] and "edX" [6] are two good examples. Second language learning is a very important subfield of education in today's world of rapid globalization. In this subfield, effective approaches, immersive environments, and experienced teachers are needed but expensive. The use of speech processing technologies has been considered a good solution to overcome these difficulties [7, 8, 9, 10, 11].

"Rosetta Stone" [12] and "byki" [13] are useful applications that provide multifaceted functions, including pronunciation evaluation and corrective feedback. Nevertheless, sentence-level practice lacks opportunities for language interaction and an immersive language learning environment [14, 15]. Spoken dialogue systems [16, 17, 18, 19, 20] are regarded as excellent solutions to provide language interaction scenarios. Recently we presented a dialogue game framework [21] in which proper training sentences 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 [22, 23], and the learning status was based on NTU Chinese [24], a Mandarin Chinese pronunciation evaluation tool. One limitation of this framework is that its training assumes a fixed number of dialogue turns; this is impractical and inflexible.

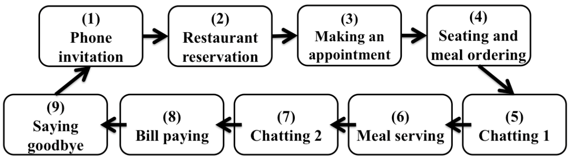


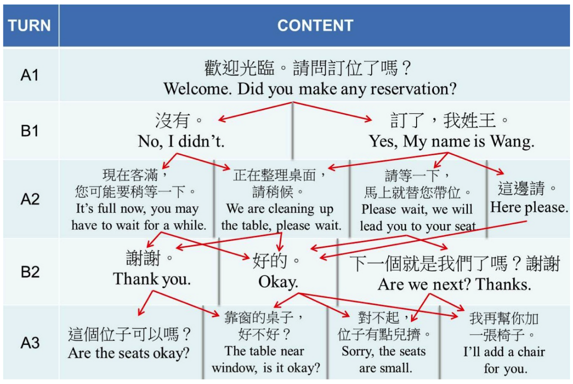
Figure 1: The script of the recursive dialogue game in the restaurant scenario from experiments: starting from (1) Phone invitation and (2) Restaurant reservation, after (9) Saying goodbye returning to (1) for next meal.

In this paper, we propose a new dialogue game framework for language learning. A series of sub-dialogue trees are cascaded into a loop. At any dialogue turn there are several training sentences that can be selected. The leaves of the last tree are linked to the root of the first tree, making the dialogue paths infinitely long. 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 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 considering every possible distribution of scores for all pronunciation units. Fitted value iteration (FVI) [25, 26, 27] is adopted for reinforcement learning to train the policy. Simulated learners with incrementally improved pronunciation scores generated from real learner data are used in policy training. Preliminary experimental results indicate the effectiveness of the approach and the usability of the framework in practice.

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



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.

sub-dialogue again for the next meal. Essentially, then, the dialogue can continue infinitely. Figure 2 is a segment of the sub-dialogue "Seating and meal ordering", in which 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 receives pronunciation performance feedback immediately after each utterance pronounced.

The advantage of using such a recursive script with tree-structured sub-dialogues is that the learner can have diversified interactions with the system in an immersive environment; the learner can practice poorly produced pronunciation units many times in different sentences, rather than use the same set of sentences repeatedly. This also provides flexibility for learners to have personalized sentence practice opportunities.

The above recursive dialogue game is modeled by an MDP with the desired optimal policy trained with the 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 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 pronouncing 30 sentences selected by language teachers. NTU Chinese, a Mandarin pronunciation evaluation tool developed at NTU [24], 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 [28, 29]. 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.

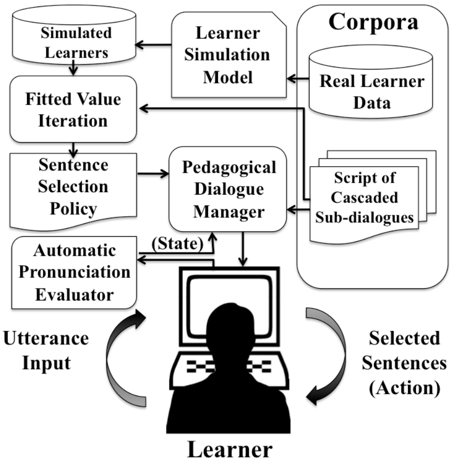

Simulated Learners
Learner Simulation Model
Corpora
Real Learner Data
Fitted Value Iteration
Sentence Selection Policy
Pedagogical Dialogue Manager
Script of Cascaded Sub-dialogues
Automatic Pronunciation Evaluator
(State)
Utterance Input
Selected Sentences (Action)
Learner

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

The GMM not only aggregates the utterance-wise score distribution statistics of the real users, but also reflects the utterance-wise correlation of scores across different pronunciation units within different contexts. For example, some learners have difficulties pronouncing all retroflexed phonemes (these occur in Mandarin but not necessarily in other languages) with contexts of certain vocal tract articulation: this may be reflected in the GMM. Therefore each mixture of this GMM could represent the pronunciation error distribution patterns for a certain group of learners with similar native language backgrounds.

For MDP policy training, when starting a new dialogue game, we randomly select a Gaussian mixture component as a simulated learner [30, 31, 32]. The mean vector of the mixture stands for the simulated learner's level on each pronunciation unit, while the covariance matrix represents the score variation within and between each unit. 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 need to develop an incremental pronunciation improvement model for the simulated learners. When the i-th pronunciation unit in PCV has been practiced C times by a simulated learner, the i-th component of the mean vector in the Gaussian mixture is increased by α and the (i, i)-th element in the covariance matrix of the Gaussian mixture is decreased by β. Thus the scores produced by the simulated learner improve and stabilize with practice. Here

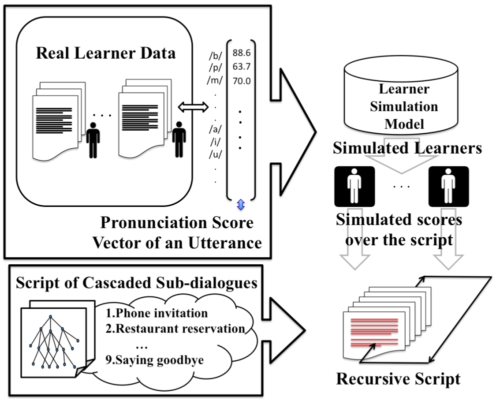


Figure 4: Learner Simulation Model for simulated learner generation.

$\mathcal{C}$, α, and β are all Gaussian random variables with means and variances assigned according to the overall pronunciation performance of the simulated learner. Thus the more units in the mean vector reach a certain threshold, the smaller and more stable the variable $\mathcal{C}$ is and the bigger and more stable the variables α and β are: the pronunciation performance of the simulated learner improves incrementally along the dialogue path.

2.3. Markov decision process

A Markov decision process (MDP) [33] is a mathematical framework for modeling sequential decision making problems, formally represented by the 5-tuple $\{S, A, R, T, \gamma\}$, which contains 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.

Since these Q functions are updated iteratively toward optimal values, this process is also known as *value iteration* and can be solved as a *Dynamic Programming* (or *Bellman*) Equation:

$$[B^\pi(Q)](s, a) = E_{s' \sim T(s,a)}[R(s, a, s') + \gamma Q(s', \pi(s'))], \tag{1}$$

where B^π is the Bellman backup operator and $s' \sim T(s, a)$ stands for next state s' following probability distribution T from state s with action a taken, and $R(s, a, s')$ is the reward gained from state s to s' by taking action a.

2.4. MDP framework on dialogue game

Here we describe how the dialogue game is modeled using MDP.

2.4.1. Continuous state space

The state represents the system's perspective towards the environment, that is, 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 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 choose for the learner at the turn.

2.4.3. Reward definition

A dialogue *episode* E contains a sequence of state transitions $\{s_0, a_0, s_1, a_1, ..., s_K\}$, where s_K represents the terminal state. As mentioned above, 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 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.

2.5. Fitted value iteration

For the high-dimensional continuous state space, we use the function approximation method [34, 35, 36] to approximate the exact Q value function with a set of m basis functions:

$$Q(s, a) = \sum_{i=1}^{m} \theta_i \phi_i(s, a) = \underline{\theta}^T \underline{\phi}(s, a), \tag{2}$$

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 $\hat{Q}_\theta(s, a)$ 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

$$\hat{B}(Q(s_i, a_i)) = r_i + \gamma \max_{a \in A} Q({s_i}', a_i). \tag{3}$$

With a batch of transition samples $\{s_j, a_j, r_j, s_j' | j = 1, ..., N\}$, least-squares regression can be performed to find the new parameter vector $\underline{\theta}_n$ 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

$$\underline{\theta}_{n+1} = \arg \min_{\underline{\theta} \in \mathbb{R}^M} \sum_{j=1}^{N} (\hat{Q}_{\theta_n} - \hat{B}(Q(s_i, a_i))^2 + \frac{\lambda}{2} \|\underline{\theta}\|^2, \tag{4}$$

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

3. Experiment

3.1. Experimental Setup

Experiments were performed on the complete script of nine sub-dialogue trees for the Mandarin Chinese learning (Section 2.1). The results below are for the learner as role B and the computer as role A. In all, 82 Mandarin pronunciation units including 58 phonetic units (Initial/Finals) and 24 tone patterns (uni/bi-tone) were considered. NTU Chinese was used as the automatic pronunciation evaluator for unit scoring and immediate feedback for the learners. In the MDP setting, 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 (2) to represent the Q value function. Five-fold cross-validation was used here: 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. The Bayesian information criterion (BIC) [37, 38] was employed on GMM to control the model likelihood and parameter complexity.

3.2. Experimental Result

3.2.1. *Number of dialogue turns needed*

In this experiment, simulated learners were generated to go through the nine sub-dialogue trees sequentially and recursively until the terminal state s_K was reached, which was defined such that all 82 pronunciation units were produced with scores over 75 more than seven times. The number of Gaussian basis function m in (2) was set to 5,10,15 respectively, where these Gaussian functions were spread evenly on the state space.

In Figure 5, we plot the number of turns to reach the terminal state as a function of the number of training iterations. Clearly the three solid curves for different values of the Gaussian basis function m yielded promising results. The number of needed turns converged around 162 to 177. Slight differences among the three curves showed over-fitting for higher parameter complexities. Since there were 84 turns in all for role B in the nine consecutive sub-dialogues, the results for 162 to 177 turns indicated that going through all nine trees and restarting from the first sub-dialogue was necessary for the testing simulated learners here.

The dashed curve ("Arbitrary") showed the result of using the nine sub-dialogue trees in a different scenario. In this scenario, we assumed that the learner choose to practice the sub-dialogue trees in an arbitrary order. For example, the learner could jump to tree four after finishing trees one and two (after restaurant reservation, the learner wishes to learn how to order meals first). Based on this scenario, we aim to test the trained policy obtained in the blue line ($m = 5$). The number of needed turns reached 212 as shown in Figure 5. The extra turns compared to the normal order scenario shows the trade-off between the user's free will to interact with the dialogue game and the minimum number of dialogue turns needed.

3.2.2. *Policy and learning status for an example learner*

Using the policy learned in normal order scenario in Section 3.2.1 ($m = 5$, blue curve in Figure 5), Figure 6 shows how the system offered practice opportunities for every pronunciation unit for an example testing simulated learner in the sub-dialogue trees four to five after finishing the first three trees (20 turns in all after tree three). In this case, there were $K = 171$ total dialogue turns. The horizontal axis is the Initial/Finals on the left and tone patterns on the right sorted by the average scores (left vertical scale) of this simulated learner (green bars). The red curve indicates the *percentage of extra practice* for each unit over those by random policy (right vertical scale) using the proposed policy, while the blue line is zero for random policy.

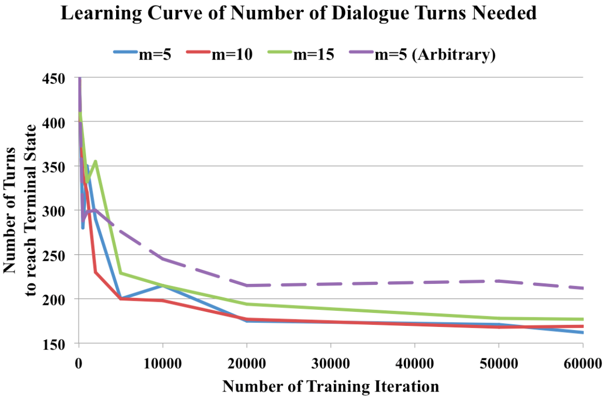


Figure 5: Number of dialogue turns needed with respect to different number of training iterations.

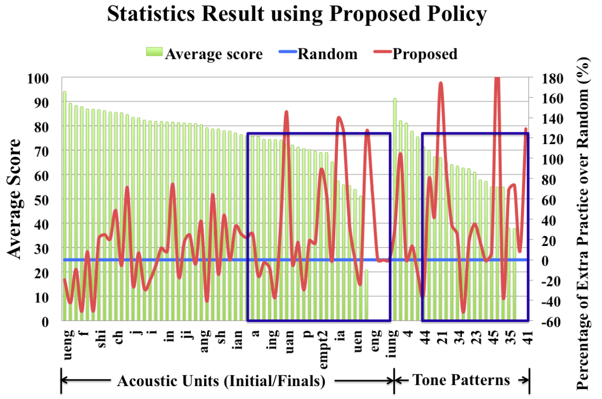


Figure 6: Average scores (left scale) of an example testing simulated learner for all pronunciation units after finishing the first three sub-dialogue trees, and percentage of extra practice over those by random policy (right scale) for this simulated learner to practice in the next two sub-dialogue trees.

Since the policy goal was for each pronunciation unit to be produced with score 75 over seven times, we focused on the units within two dark blue blocks, which have scores below 75. Clearly our proposed approach resulted in a greater number of practice opportunities than random policy on these units in 20 turns after tree three. This means the policy was efficient to provide what the simulated learner needed.

4. Conclusions

We presented a new recursive dialogue game framework with an optimal policy offering personalized learning materials for CALL. A series of sub-dialogue trees are cascaded into a loop. 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. Experimental results showed promising results, and the work to implement a real system for real learners is in progress.

5. References

[1] S. B. Merriam, *Qualitative Research and Case Study Applications in Education. Revised and Expanded from "Case Study Research in Education."*. Jossey-Bass Publishers, 1998.

[2] J. Dewey, "Experience and education," *The Educational Forum*, 1986.

[3] B. Holmberg, *The evolution, principles and practices of distance education*. Bis, 2005.

[4] M. K. Smith, W. B. Wood, W. K. Adams, C. Wieman, J. K. Knight, N. Guild, and T. T. Su, "Why peer discussion improves student performance on in-class concept questions," *Science*, 2009.

[5] (2012) Coursera. [Online]. Available: https://www.coursera.org/

[6] (2012) edx. [Online]. Available: https://www.edx.org/

[7] M. Eskenazi, "An overview of spoken language technology for education," in *Speech Communication*, vol. 51, 2009, pp. 832–844.

[8] C. Cucchiarini, J. van Doremalen, and H. Strik, "Practice and feedback in l2 speaking: an evaluation of the disco call system," in *Interspeech*, 2012.

[9] Y. Xu, "Language technologies in speech-enabled second language learning games: From reading to dialogue," Ph.D. dissertation, Massachusetts Institute of Technology, 2012.

[10] 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.

[11] 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.

[12] (1999) Rosetta Stone. [Online]. Available: http://www.rosettastone.com/

[13] (2013) byki. [Online]. Available: http://www.byki.com/

[14] D. Christian, *Profiles in Two-Way Immersion Education. Language in Education: Theory and Practice 89.*, 1997.

[15] W. L. Johnson, "Serious use of a serious game for language learning," in *International Journal of Artificial Intelligence in Education*, 2010.

[16] 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.

[17] 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.

[18] 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.

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

[20] S. Lee and M. Eskenazi, "Incremental sparse bayesian method for online dialog strategy learning," *Journal of Selected Topics Signal Processing*, 2012.

[21] 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.

[22] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1999.

[23] R. Bellman, *Dynamic programming*. Princeton University Press, 1957.

[24] (2009) NTU Chinese. [Online]. Available: http://chinese.ntu.edu.tw/

[25] S. Chandramohan, M. Geist, and O. Pietquin, "Optimizing spoken dialogue management from data corpora with fitted value iteration," in *Interspeech*, 2010.

[26] A. s Antos, R. mi Munos, and C. S. ri, "Fitted q-iteration in continuous action-space mdps," in *NIPS*, 2007.

[27] A. massoud Farahmand, M. Ghavamzadeh, C. Szepesvari, and S. Mannor, "Regularized fitted q-iteration for planning in continuous-space markovian decision problems," in *ACC*, 2009.

[28] R. Hogg, J. McKean, and A. Craig, *Introduction to Mathematical Statistics*. Pearson Prentice Hall, 2005.

[29] 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.

[30] 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.

[31] H. Ai and F. Weng, "User simulation as testing for spoken dialog systems," in *SIGdial*, 2008.

[32] 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.

[33] A. N. Burnetas and M. N. Katehakis, "Optimal adaptive policies for markov decision processes," *Mathematics of Operations Research*, 1995.

[34] L. Daubigney, M. Geist, and O. Pietquin, "Off-policy learning in large-scale pomdp-based dialogue systems," in *ICASSP*, 2012.

[35] Y. Engel, S. Mannor, and R. Meir, "Bayes meets bellman: The gaussian process approach to temporal difference leraning," in *ICML*, 2003.

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

[37] W. Zucchini, "An introduction to model selection," in *Journal of Mathematical Psychology*, vol. 44, 22006, pp. 41–61.

[38] 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.