A DIALOGUE GAME FRAMEWORK WITH PERSONALIZED TRAINING USING REINFORCEMENT LEARNING FOR COMPUTER-ASSISTED LANGUAGE LEARNING

Pei-hao Su [#1], *Yow-Bang Wang* [*2], *Tien-han Yu* [#], *and Lin-shan Lee* [#*3]

[#] Graduate Institute of Communication Engineering, National Taiwan University
[*]Graduate Institute of Electrical Engineering, National Taiwan University
[1] r00942135@ntu.edu.tw, [2] piscesfantasy@gmail.com, [3] lslee@gate.sinica.edu.tw

ABSTRACT

We propose a framework for computer-assisted language learning as a pedagogical dialogue game. The goal is to offer personalized learning sentences on-line for each individual learner considering the learner's learning status, in order to strike a balance between more practice on poorly-pronounced units and complete practice on the whole set of pronunciation units. This objective is achieved using a Markov decision process (MDP) trained with reinforcement learning using simulated learners generated from real learner data. Preliminary experimental results on a subset of the example dialogue script show the effectiveness of the framework.

Index Terms— Computer-Assisted Language Learning, Dialogue Game, Markov Decision Process, Reinforcement Learning

1. INTRODUCTION

In this globalized world, second language acquisition is becoming increasingly important. Although one-on-one tutoring is the most effective method for language learning, it comes at a high cost. As speech processing technology matures, computer-assisted language learning (CALL) is becoming more and more attractive [1, 2, 3]. Computer-aided pronunciation training (CAPT) [4, 5] has focused on pronunciation evaluation [6, 7] and error pattern detection [8, 9]. Pronunciation evaluation provides an overall assessment of a speaker's proficiency in the produced speech. NTU Chinese [10], for instance, is a successfully operating online Chinese pronunciation learning application developed in National Taiwan University (NTU). Error pattern detection attempts to identify specific erroneous pronunciation patterns at the word or subword level [11, 12].

The use of spoken dialogue technologies in language learning has also been extensively investigated [13, 14]. Traditionally, most spoken dialogue systems have been developed to serve specific purposes [15, 16], for example slot-filling tasks such as city information querying [17]. When spoken dialogue technologies are used for language learning, slot-filling tasks have evolved into dialogue games [18, 19] or task-based language learning processes [20, 21] by which learners practice the target language.

We here propose a dialogue game framework for language learning, which combines pronunciation scoring and a statistical dialogue manager based on a tree-structured dialogue script designed by language teachers. Sentences to be learned can be adaptively selected for each learner, based on the pronunciation unit practiced and scores obtained along with the dialogue progress. The dialogue manager is modeled as a Markov decision process (MDP) [22, 23] which is trained with reinforcement learning using simulated learners generated from real learner data.

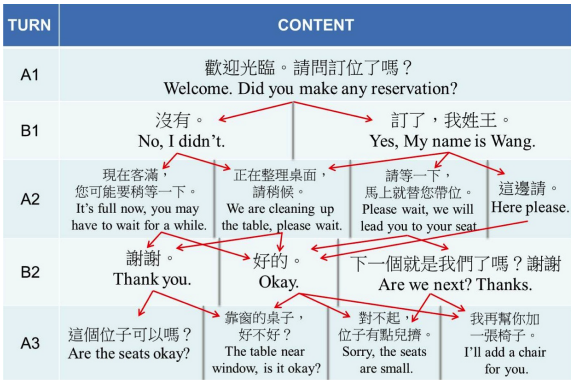

Fig. 1. A segment of the dialogue script for the example of dialogue game in a restaurant conversation scenario.

2. PROPOSED DIALOGUE GAME FRAMEWORK

2.1. Dialogue Script

The progress of the dialogue game is based on a tree-structured dialogue script. In preliminary experiments, this dialogue contains conversations in restaurant scenario between two roles A and B – one the computer and the other the learner. After each utterance produced by one speaker, there are several choices for the other speaker's next sentence. The contents of the dialogue script are designed by language teachers to be phonetically balanced and prosodically rich with good coverage of commonly used words at the proper level. The script includes 9 short dialogues with a total of 176 turns. Figure 1 is a segment of the example dialogue in which A is the waiter and B the customer.

Since both the computer and the learner have multiple sentence choices in each dialogue 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 pronunciation units considered are the context-independent Initial/Finals and tone patterns in Mandarin Chinese. An Initial is the onset of a syllable, while the Final is the rime part of a syllable. Tone patterns include uni-tone and bi-tone (within word) patterns. Figure 2 shows the normalized count distributions of the Initial/Finals and tone patterns between two example paths in the dialogue script. Different paths yield quite different learning opportunities.

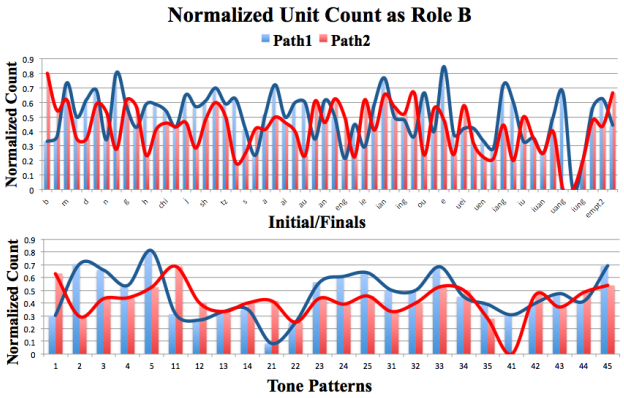

Fig. 2. Example of acoustic and prosodic pattern distributions between two paths for a learner as role B.

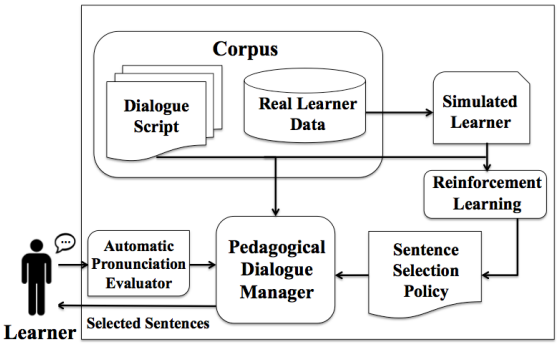

Fig. 3. Proposed framework of the pedagogical dialogue game.

2.2. Proposed Framework – "Practice Makes Perfect"

Numerous studies [24, 25] cite the necessity of repeated practice in language learning. Here we aim to design a pedagogical dialogue manager that adaptively selects sentences for each individual learner along with the progress of the dialogue based on the learning status of each pronunciation unit, such that more practice is offered for poorly-pronounced units while most units are still pronounced and learned. This problem is considered as an optimization problem of a sequential stochastic decision, solved by reinforcement learning based on an MDP model. Figure 3 shows the system diagram of the proposed framework.

When the learner produces an utterance, the Automatic Pronunciation Evaluator (NTU Chinese) scores each pronunciation unit in the utterance. The Pedagogical Dialogue Manager then selects the next sentence for the learner to practice based on the Sentence Selection Policy. A set of simulated learners generated from Real Learner Data is used in Reinforcement Learning to train the Sentence Selection Policy.

Below are the detailed parameter definitions and settings thereof.

2.3. State Space

The MDP model contains a set of states. The state represents the system's perspective to the environment, which in our task is the learner's learning status. Three variables describe the state: the sentence index in the dialogue (describing the present dialogue turn), the quantized percentage of poorly-pronounced units (units with scores under a predefined threshold), and the indices of the worst-pronounced units.

2.4. Action Set

Given the present state, the actions to be taken are the sentences to be selected for the learner to practice.

2.5. Reward Function

In general, there is a reward when an action is taken at one state to transmit to another state. The final return is the cumulative result of all rewards gained in an *episode*, which represents a complete dialogue, including all turns.

Our goal is to provide the proper learning sentences to each specific learner along with dialogue path, considering the learner's learning status, and offering the best learning opportunities among all pronunciation units. The reward serves as the objective of the system: we set to 0 the reward gained in each intermediate state transition. The reward in the last state transition of the dialogue episode is defined as a combination as follows.

1. *More Practice Needed*: Selecting sentence for learner which contains more poorly-pronounced units.

$$R_1 = \frac{1}{|W|} \sum_{i \in W} \frac{\psi_i - C_i}{C_i} (1 - s_i)^v, \tag{1}$$

where W is the set of pronunciation units with average scores below a pre-defined threshold at the end of an episode, i is the index of a pronunciation unit in W, ψ_i accounts for the occurrence count of the unit i in the whole dialogue episode, C_i is the average count of the unit i of all possible dialogue paths, and s_i is the average score of unit i normalized between 0 and 1. The term $(1 - s_i)$ emphasizes units with lower scores s_i; v is a weight parameter. This objective represents the *weighted average percentage of extra practice for poorly-pronounced units* against randomly offered sentences. A higher value means more such opportunities.

2. *Practice Completeness*: We also wish to make sure the learner practices the entire set of pronunciation units.

$$R_2 = \frac{N_p}{N_o}, \tag{2}$$

where N_p stands for the number of units the learner has practiced when the dialogue finishes, and N_o is the total number of units in the specific language. Thus this objective is simply the percentage of pronunciation units in the language which have been practiced, regardless of the scores.

The overall objective function is then the weighted sum of the above two objectives:

$$R = w \cdot R_1 + (1 - w) \cdot R_2, \tag{3}$$

where w is the weight between the two objectives.

3. LEARNER SIMULATION FROM REAL DATA

Reinforcement learning uses a training set to learn the sentence selection policy of the pedagogical dialogue manager. Since it is practically infeasible to collect "enough" real dialogue episodes for policy training, studies have focused on generating simulated users to interact with the dialogue manager [26, 27, 28, 29]. We propose an approach to generate simulated learners from real learner data.

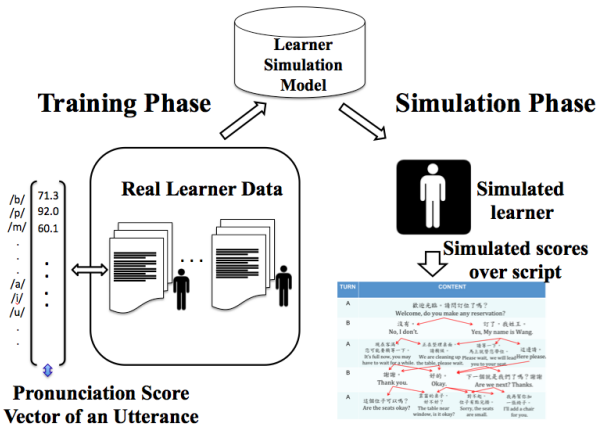

Fig. 4. Pronunciation score vector, simulated learner modeling and creation.

3.1. Real Learner Data

For the experiments below, we use a read speech corpus collected in 2008 and 2009 from real learners practicing their Mandarin with NTU Chinese [11, 12]. A total of 278 learners from 36 different countries, balanced by gender and with 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, each of which contained 6 to 24 Chinese characters. These 30 sentences came from the learning materials designed by ICLP language teachers and used in NTU Chinese. The data set covers almost all frequently used Mandarin syllables and tone patterns.

3.2. Simulated Learner Creation

Figure 4 shows the training (left) and simulation phases (right) of learner simulation. In the training phase, we first evaluate all utterances produced in the real learner data using the automatic pronunciation evaluator (NTU Chinese in our experiment), which assigns to each pronunciation unit (Initial/Finals and tone patterns in the experiment) a score from 0 to 100. For each utterance, we construct a pronunciation score vector (PSV), the dimensionality of which is the number of pronunciation units considered. For the units that appear in a given utterance, the corresponding component in the PSV is the average score in the utterance; for those units that do not appear, we treat them as missing (latent) data. The PSVs from all utterances produced by all real learners are used to train a learner simulation model as a Gaussian mixture model (GMM). The missing data problem is dealt with by using the expectation-maximization (EM) algorithm [30, 31].

The GMM not only aggregates the utterance-wise score distribution statistics of the real users, but reflects the utterance-wise correlation of scores across different pronunciation units within different contexts. For example, some learners have difficulties pronouncing all retroflexed phonemes (present 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 group of learners with similar native language backgrounds.

In the simulation phase, when starting a new dialogue episode, we first choose one mixture component according to the probability of the mixture weights as a simulated learner. Then for each utterance in the script we generate a sampled PSV from this Gaussian mixture component, taking its corresponding components as the scores for those units needed in the utterance, to form a "simulated utterance" produced by this simulated learner. In this way the simulated learners can behave very similarly to real learners.

4. REINFORCEMENT LEARNING POLICY

Policy is a mapping from state to action. Here, the policy defines what sentence the learner will practice next. The optimal policy maximizes reward function [32]. We use reinforcement learning with sampling method to train the policy [33], adopting the temporal difference (TD) Q-learning algorithm. For each simulated learner's training episode, we update the Q value of each state-action pair as

$$Q(s_t, a_t) \leftarrow (1-\alpha)Q(s_t, a_t) + \alpha[r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})], \quad (4)$$

where s_t is the state the training simulated user is in and a_t is the action the system has taken at state s_t. α is the learning rate, interpolating the present Q value and the updating information. γ is the discount factor, which determines the influence of the next state-action pair (a_{t+1}, s_{t+1}) on the current Q value, and r_t is the reward gained from the t-th state transition.

After the Q value is updated, the optimal policy $\pi(s)$ – the optimal system action taken in each state – is updated:

$$\pi(s) = arg \max_a Q(s, a). \quad (5)$$

In addition, *Exploratory actions* are taken to avoid reaching local optimal results [33]. We choose the action with the highest Q value in (5) with probability $1-\epsilon$ and the remaining actions with probability ϵ, since actions with lower Q's may yield better results eventually. Careful selection of ϵ can lead to better convergence of the optimal policy. The reward in our task is gained only at the end (the last state transition) of each episode. This means that after our first training simulated learner has finished one episode, only r_T is non-zero (s_T is the last state in this training episode), and thus only $Q(s_T, a_T)$ and $Q(s_{T-1}, a_{T-1})$ are updated, while the other states are still based on random policy. Thus the training process requires many iterations to propagate the Q values all over the whole state-action space to reach proper convergence.

5. EXPERIMENT

5.1. Environmental Setup

A short dialogue of 62 turns from the whole dialogue script was evaluated in preliminary experiments. The results shown are the case when learner played as role B in the dialogue game (role A yielded similar results). We used NTU Chinese as the automatic pronunciation evaluator to score the Initial/Finals and tone patterns of each utterance. The system's initial policy was always to choose the first sentence among the candidate sentences. Real learner data was used to generate the simulated learners for reinforcement learning.

We compared the proposed approach with the following two heuristic policies and one approach for combining the two on MDP:

1. Always select the sentence with the most diverse pronunciation units from learner's practiced units.
2. Always select the sentence with the most count of worst-pronounced units.
3. Cast the above two heuristic policies as two actions in an MDP. The system learns to choose between these two actions via reinforcement learning.

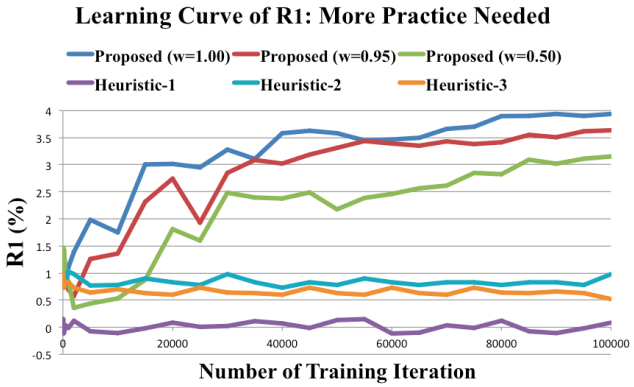

Fig. 5. R_1 learning curves.

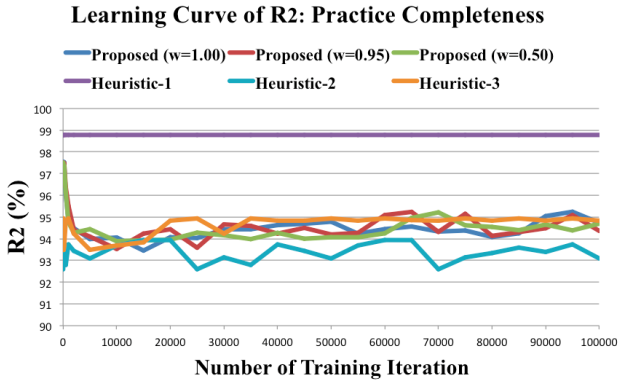

Fig. 6. R_2 learning curves.

The following experiments were performed using 5-fold cross-validation. 80% of the real learner data was used to train a GMM to generate simulated learners for model training, while simulated learners for model testing were generated by a GMM trained with the remaining 20%. We used 3-mixture GMMs. This number of mixtures we determine using the Bayesian information criterion (BIC) [34, 35], which jointly takes into account likelihood and parameter complexity. In each training iteration, one simulated learner – a sampled mixture component of a GMM – was selected and proceeded through one dialogue episode for policy updating. Immediately after each training iteration, the obtained policy was tested on 50 simulated learners from the testing GMM. We averaged the five learning curves of the five validation sets to obtain the final result.

5.2. Experimental Result

5.2.1. More Practice Needed v.s. Practice Completeness

In this experiment we set w=1.00, 0.95, and 0.50 in (3), yielding different emphases on R_1 and R_2. Factor v in (1) was set to 1. Figures 5 and 6 are the learning curves of R_1 and R_2 for different values of w for different numbers of training iterations. Heuristic-1, 2, and 3 correspond to the heuristic approaches listed in section 5.1.

Figure 5 shows that the proposed approaches significantly outperformed the three heuristic policies. $w = 1.00$ clearly yielded the best value of R_1, since in this case the learning processes focused only on maximizing R_1. With $w = 0.95$ or 0.50 the values of R_1 were lower, due to consideration of R_2. Moreover, although Heuristic-3 "learned" from predefined actions (Heuristic-1,2), the result shows no improvement.

Figure 6 shows the effectiveness of Heuristic-1, since it optimized R_2 directly. Only subtle differences are observed among the other approaches, despite the various weights set in the proposed approaches. Note that a total of 58 Initial/Finals and 24 tone patterns were considered, and all six results were distributed within 93–98.9% of R_2; this implies that 77–81 of them were practiced with small disparity. This may be because the dialogue script was already well designed to be phonetically balanced and prosodically rich.

5.2.2. Emphasis on Low-scoring Pronunciation Units

Figure 7 shows how the system offered opportunities to practice each pronunciation unit for one example simulated learner with a well-trained policy ($w = 1$) and 80000 training iterations. The horizontal axis is the Initial/Finals and tone patterns (partially listed) sorted by the average scores of this simulated learner (green bars). The blue, red, and black curves indicate the *percentage of extra practice over random* offered by the random policy, the proposed policy with $v = 0$, and that with $v = 1$, respectively. Note that the blue line is always zero since the random policy offers no extra practice over random.

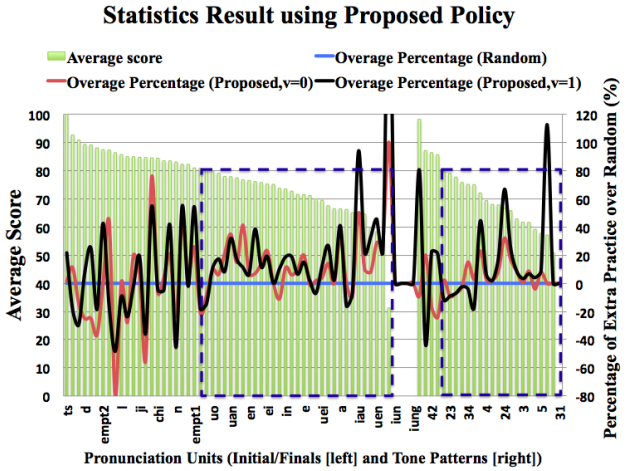

Fig. 7. Average scores and overage percentages of pronunciation units for an example testing simulated learner with random and proposed policies (v=0,1).

Units with scores lower than 80 were defined to be poorly-pronounced. The area within the dashed line contains the weak units of the learner. The adaptive policy (red line) clearly provides more practice opportunities on weak units than the random policy (blue line), and that an adaptive policy that emphasizes low scores (black line, $v = 1$) further provides more opportunities to practice units with lower score in weak units. This shows the effectiveness of the proposed approach.

6. CONCLUSIONS

We described a new perspective for CALL combining pronunciation evaluation and a statistical dialogue system. We proposed a dialogue game framework based on a Markov decision process model trained with reinforcement learning. GMMs were used in simulated learner creation for MDP policy training. Preliminary experimental results on an extracted short dialogue of the script showed the effectiveness of the proposed approach.

7. REFERENCES

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

[2] M. Levy, *Computer-Assisted Language Learning: Context and Conceptualization.* Oxford University Press, 1997.

[3] A. Neri, C. Cucchiarini, and H. Strik, "The effectiveness of computer-based speech corrective feedback for improving segmental quality in l2 dutch," in *European Association for Computer Assisted Language Learning*, 2008.

[4] M. A. Peabody, "Methods for pronunciation assessment in computer aided language learning," Ph.D. dissertation, Massachusetts Institute of Technology, 2011.

[5] S. M. Witt, "Use of speech recognition in computer-assisted language learning," Ph.D. dissertation, University of Cambridge, 1999.

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

[7] C. Hacker, *Automatic Assessment of Children Speech to Support Language Learning*. Logos, 2009.

[8] A. M. Harrison, W.-K. Lo, X.-J. Qian, and H. Meng, "Implementation of an extended recognition network for mispronunciation detection and diagnosis in computer-assisted pronunciation training," in *Sig-SLATE*, 2009.

[9] C. Cucchiarini, H. van den Heuvel, E. Sanders, and H. Strik, "Error selection for asr-based english pronunciation training in 'my pronunciation coach'," in *Interspeech*, 2011.

[10] (2009) NTU chinese. [Online]. Available: http://chinese.ntu.edu.tw/

[11] Y.-B. Wang and L.-S. Lee, "Improved approaches of modeling and detecting error patterns with empirical analysis for computer-aided pronunciation training," in *ICASSP*, 2012.

[12] ——, "Error pattern detection integrating generative and discriminative learning for computer-aided pronunciation training," in *Interspeech*, 2012.

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

[14] K. VanLehn, P. Jordan, and D. Litman, "Developing pedagogically effective tutorial dialogue tactics: Experiments and a testbed," in *SLaTe*, 2007.

[15] 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.

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

[17] 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.

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

[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] 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.

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

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

[23] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Interscience, 1994.

[24] R. M. DeKeyser, *Practice in Second Language - Perspectices from Applied Linguistics and Cognitive Psycology*. Cambridge University Press, 2007.

[25] My language exchange. [Online]. Available: http://www.mylanguageexchange.com/

[26] W. Eckert, E. Levin, and R. Pieraccini, "User modeling for spoken dialogue system evaluation," in *ASRU*, 1997.

[27] 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.

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

[29] 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.

[30] J. A. Bilmes, "A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models," *U.C. Berkely*, vol. TR-97-021, 1998.

[31] 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.

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

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

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

[35] 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.