Affordable On-line Dialogue Policy Learning

Cheng Chang*, **Runzhe Yang***, **Lu Chen, Xiang Zhou and Kai Yu**
Key Lab. of Shanghai Education Commission for Intelligent Interaction and Cognitive Eng.
SpeechLab, Department of Computer Science and Engineering
Brain Science and Technology Research Center
Shanghai Jiao Tong University, Shanghai, China
{cheng.chang,yang_runzhe,chenlusz,owenzx,kai.yu}@sjtu.edu.cn

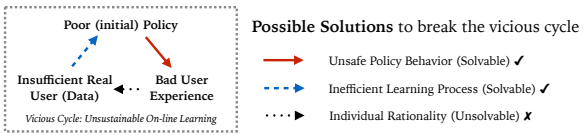
Abstract

The key to building an evolvable dialogue system in real-world scenarios is to ensure an affordable on-line dialogue policy learning, which requires the on-line learning process to be safe, efficient and economical. But in reality, due to the scarcity of real interaction data, the dialogue system usually grows slowly. Besides, the poor initial dialogue policy easily leads to bad user experience and incurs a failure of attracting users to contribute training data, so that the learning process is unsustainable. To accurately depict this, two quantitative metrics are proposed to assess safety and efficiency issues. For solving the unsustainable learning problem, we proposed a complete *companion teaching* framework incorporating the guidance from the human teacher. Since the human teaching is expensive, we compared various *teaching schemes* answering the question *how* and *when* to teach, to economically utilize teaching budget, so that make the online learning process affordable.

1 Introduction

A *task-oriented* dialogue system is designed for interacting with humans users to accomplish several predefined domains or tasks (Young et al., 2013; Daubigney et al., 2012). *Dialogue Manager* is the core component in a typical dialogue system, which controls the flow of dialogue by a *state tracker* and a *policy module* (Levin et al., 1997). The state tracker tracks the internal state of the system while the policy module decides the response to the user according to the status of states (Sun et al., 2014a; Thomson and Young, 2010). The approaches of constructing a policy module can be divided into two categories: rule-based and statistical. Rule-based policies are usually hand-crafted by domain experts which means they are inconvenient and difficult to be optimized (Williams and Young, 2007; Wang and Lemon, 2013). In recent mainstream statistical studies, *Partially Observable Markov Decision Process* (POMDP) framework has been applied to model dialogue management with unobservable states, where policy training can be formulated as a *Reinforcement Learning* (RL) problem, which enables the policy to be optimized automatically (Kaelbling et al., 1998; Arnold, 1998; Young et al., 2013).

Though RL-based approaches have the potential to improve themselves as they interact more with human users and achieve better performance than rule-based approaches, they are rarely used in real-world applications, especially in on-line scenarios, since the training process is unsustainable.



The main causes of unsustainable on-line dialogue policy learning are two-fold:

- *Safety issue:* the initial policy trained from scratch may lead to terrible user experience at the early training period, thus fail to attract sufficient users for more dialogues to do further policy training.
- *Efficiency issue:* if the progress of policy learning is not so efficient, it will exhaust users' patience before the policy reaches a desirable performance level.

* Both authors contributed equally to this work.

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2190–2199
Copenhagen, Denmark, September 7–11, 2017.

Prior works have mainly focused on improving *efficiency*, such as Gaussian Processes RL (Gašić et al., 2010), deep RL (Fatemi et al., 2016), etc. For deep RL approaches, recent researches on the *student-teacher RL framework* have shown prominent acceleration to policy learning process (Torrey and Taylor, 2013; Williams and Zweig, 2016; Amir et al., 2016). In such framework, the *teacher agent* instructs the *student agent* by providing suggestions on what actions should be taken next (Clouse, 1996).

For the *safety* issue, Chen et al. (2017) developed several teaching strategies answering "how" the human teacher guide the learning process.

However, those previous teaching methods exclude "when" to teach from concern. They simply exhaust all the budget continuously from the beginning, which is wasteful and causes a heavy workload of the human teacher. An *affordable* dialogue policy learning with human teaching should require a lighter workload and economically utilize teaching budget.

Furthermore, as for safety and efficiency evaluation, previous works have been observing the training curves and testing curves to tell which one is better, or evaluate policy performance after certain dialogues of training, which are subjective and error prone (Chen et al., 2015a; Su et al., 2016; Chen et al., 2017).

Our contribution is to address the above problems. We propose a complete framework of companion teaching, and develop various *teaching schemes* which combine different *teaching strategies* and *teaching heuristics* together, to answer the questions of "how" and "when" to teach to achieve affordable dialogue policy learning (section 2). Specifically, a novel *failure prognosis* based teaching heuristic is proposed, where MultiTask Learning (MTL) is utilized to predict the dialogue success reward (section 3). To avoid the drawbacks of traditional subjective measurements, we propose two evaluation metrics, called *Risk Index* (RI) and *Hitting Time* (HT), to quantify the safety and efficiency of on-line policy learning respectively (section 4). Simulation experiments showed, with the proposed companion teaching schemes, sustainable and affordable on-line dialogue policy learning has been achieved (section 5).

2 Companion Teaching Framework

The *companion teaching* framework is an on-line policy training framework with three intelligent participants: machine dialogue manager, human user, and human teacher (Chen et al., 2017). Under this framework, the human teacher is able to accompany the dialogue manager to guide policy learning with a limited teaching budget. By investigating the real work mode in a call center, this framework makes a reasonable assumption that human teacher has access to the extracted dialogue states from the dialogue state tracker as well as the system's dialogue act, and can also reply in the same format.

However, there are two major problems in the previous framework. First, the system will judge whether a dialogue session succeeds by several simple rules and then determine whether to feed a success reward signal to dialogue manager for reinforcement learning. Actually, the success feedback made by the system lacks flexibility and credibility, and it could mislead the policy learning. A more suitable judge should be the user or the human teacher. Second, the previous framework only answers in which way the human teacher can guide the online dialogue policy learning, but another essential question, *when* should the human teacher give guidance, remains undiscussed.

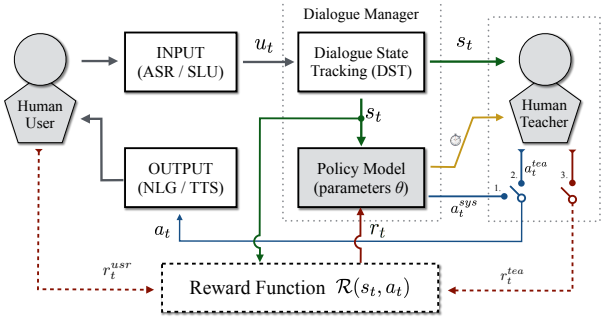


Figure 1: Companion Teaching Framework for On-line Policy Learning

In this paper, we proposed a complete framework of *companion teaching*, depicted as Figure 1. At each turn, the input module receives a speech input from the human user, then produces possible utterances u_t of the speech in text. After that, the dialogue state tracker extracts the dialogue state s_t from possible utterances. This dialogue state will be shared with policy model and human teacher if needed. When the final response a_t, has been

determined, the output module will translate this dialogue act to the natural language and reply to the human user. The success signal will be fed back by the user or the human teacher as an important part of system reward, at the end of each session. The human teacher can take the initiative or be activated by *student initiated heuristic* to give the dialogue guidance with strategies corresponding to different configurations of switches in the illustration. We call the combination of strategy and heuristic as *teaching scheme*.

2.1 Teaching Strategies

The teacher can choose among three teaching strategies corresponding to different configurations of switches in a wiring diagram as Figure 1 shows: The left switch is a Single-Pole, Double-Throw (SPDT) switch, which controls whether the answer is made by the system (connected to 1) or given by the teacher as an example (connected to 2). The right switch is a simple on-off switch, which represents whether there is an extra reward signal from the teacher (ON) or not (OFF). The strategy related to the right switch is called *teaching via Critic Advice* (CA), also known as turn-level reward shaping (Thomaz and Breazeal, 2008; Judah et al., 2010). When the switch at position 3 is turned on, the teacher will give the policy model extra turn-level reward to distinguish the student's actions between good and bad actions. Besides, the left switch corresponds to *teaching via Example Action* (EA), which means the teacher gives example action for the student to take according to the student's state.

The other strategy is proposed by Chen et al. (2017), which take the advantages of both EA and CA, named *teaching via Example Action with Predicted Critique* (EAPC). With this strategy, the human teacher gives example actions, meanwhile, a weak action predictor is trained using this teaching information to provide the extra reward even in teacher's absence.

2.2 Teaching Heuristics

The strategies only answer how the human teacher can offer companion teaching to the system. However, the timing of teaching should not be ignored for the sake of utilizing the limited teaching budget better. Exhausting all the budget at early training stage, named *Early teaching heuristic* (Early), is simple and straightforward but wastes teaching opportunities on unnecessary cases. Thus, it is imperative to design some effective heuristics to instruct when the teacher should give a hand to the student.

In addition to early teaching, the teaching heuristics can be broadly divided into two categories: *teacher-initiated* heuristics and *student-initiated* heuristics (Amir et al., 2016). However, the teacher-initiated approaches require the constant long-term attention of the teacher to monitor the dialogue process (Torrey and Taylor, 2013; Amir et al., 2016), which is costly and impractical for real applications. Therefore, in this paper, we only discuss student-initiated heuristics, shown as the line with a stopwatch in Figure 1, which means that the student agent decides when to ask for the teacher's help.

Previous works have presented several effective heuristics based on *state importance*, $I(s)$, which is determined by the Q-values of the RL agent:

$$I(s) = max_a Q_{(s,a)} - min_a Q_{(s,a)}$$

Torrey and Taylor (2013) proposed *State Importance based Teaching heuristic* (SIT) which make the student ask for advice only when the current state is important:

$$I(s) > t_{si}, \tag{1}$$

where t_{si} is a fixed threshold for importance. And Clouse (1996) proposed an *State Uncertainty based Teaching heuristic* (SUT) which ask for advice when the student is uncertain about which action to take:

$$I(s) < t_{su}, \tag{2}$$

where t_{su} is a given threshold for uncertainty.

Though teaching effort can be conserved by only applying to those important or uncertain states, it may end up wasting advice if the dialogue is likely to be successful without teaching. In this paper, we propose a novel *Failure Prognosis based Teaching heuristic* (FPT) for on-line policy learning to reduce that unnecessary advice. The details are given in section 3. For comparison, we will also investigate *Random teaching heuristic* (Rand) which means the student seek for advice with a fixed probability p_r.

3 Failure Prognosis Based Teaching Heuristic

To make better use of teaching advice, we propose to use an on-line turn-level *task success predictor*

to predict whether the ongoing dialogue will end in success and ask for advice only when the current prediction is a failure. The proposed approach utilizes MultiTask Learning (MTL) for the policy model to estimate future dialogue success reward and is compatible with various RL algorithms. In this paper, we implement the policy model with a Deep Q-Network (DQN), in which a neural network function approximator, named *Q-network*, is used to estimate the action-value function (Mnih et al., 2013).

3.1 Multitask Deep Q-Network

The goal of the policy model is to interact with human user by choosing actions in each turn to maximize future rewards. We define the dialogue state shared by dialogue state tracker in the t-th turn as s_t, the action taken by policy model under current policy π_θ with parameters θ in the t-th turn as a_t, and $a_t \sim \pi_\theta(\cdot|s_t)$. In an ideal dialogue environment, once the policy model emit an action a_t, the human user will give an explicit feedback, like a normal response or a feedback of whether the dialogue is successful, which will be converted to a reward signal r_t delivering to the policy model immediately, and then the policy model will transit to next state s_{t+1}. The reward r_t is composed of two parts:

$$r_t = r_t^{turn} + r_t^{\texttt{succ}},$$

where r_t^{turn} is the turn penalty reward and $r_t^{\texttt{succ}}$ is the dialogue success reward. Typically, r_t^{turn} is fixed for each turn as a negative constant R^{turn}, while $r_t^{\texttt{succ}}$ equals to a predefined positive constant $R^{\texttt{succ}}$ only when the dialogue terminates and receives a successful user feedback otherwise zero.

In DQN algorithm, all these transitions (s_t, a_t, r_t, s_{t+1}) will be stored in a replay memory $\mathcal{D}$. And the objective is to optimize MSE between *Q-network* $Q(s, a; \theta)$ and *Q-learning target* Q_e. The loss function $L(\theta)$ is defined as:

$$L(\theta) = \mathbb{E}_{s,a\sim\pi_\theta}\left[(Q_e - Q(s, a; \theta))^2\right]. \quad (3)$$

During the training period, Q_e is estimated with old fixed parameter θ^- and sampled transitions $e \sim \mathcal{D}$:

$$Q_e = r + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta^-), \quad (4)$$

where γ is the discount factor.

The reward $Q(s, a)$ estimated by original Q-learning algorithm is essentially a combination of future turn penalty reward $Q^{\texttt{turn}}(s, a)$ and future dialogue success reward $Q^{\texttt{succ}}(s, a)$. For a task-oriented dialogue system, the prediction of $Q^{\texttt{succ}}(s, a)$ is much more important because it reflects the possibility of the dialogue to be successful. If these two rewards are estimated separately, the objective of $Q^{\texttt{succ}}(s, a)$ can be optimized explicitly, and we can get more insights into the estimated future. We found that in practice, optimizing these two objectives with *MultiTask Learning* (MTL) converges faster and more stable compared with two separate models, the reason of which may lie in that MTL can learn different related tasks in parallel using shared representations, which will be helpful for each task to be learned better (Caruana, 1997). The structure of proposed MTL-DQN is depicted in Figure 2.

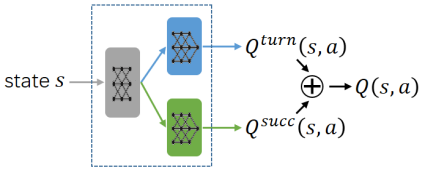


Figure 2: MTL-DQN structure

3.2 Failure Prognosis

In the proposed multitask DQN, we define *on-line task success predictor* $T(s_t)$ as:

$$T(s_t) = Q^{\texttt{succ}}(s_t, a_t),$$

where a_t is the action taken under state s_t. It is reasonable to assume that the dialogue is going to fail if $T(s_t)$ is relatively small. Based on the task success predictor, we propose a novel student-initiated heuristic, named *Failure Prognosis based Teaching heuristic* (FPT).

The key to the proposed heuristic is to define *failure prognosis* quantitatively. A straight way is to set a ratio threshold α, and consider it to be *failure prognosis* when $T(s_t) < \alpha R^{\texttt{succ}}$. However, this assumes that the numerical scale of $Q^{\texttt{succ}}$ is consistent through the training period, which is not always the case. And the student's noisy estimation of $Q^{\texttt{succ}}$ at early training period will make the learning process unstable. To smooth the teaching, we consider using a turn-level sliding window near the current state to calculate an average value

as the replacement of the fixed $R^{\texttt{succ}}$. So in the t-th turn, the *failure prognosis* for the student to be *true* is equivalent to:

$$T(s_t) < \alpha \frac{1}{w} \sum_{j=t-w}^{t-1} T(s_j), \tag{5}$$

where w is the size of the sliding window.

4 Quantitative Measurements for Safety and Efficiency

The performance of different teaching strategies and heuristics should be measured in both the *safety* and *efficiency* dimension. However, the measurements in these two dimensions are subjective and error prone in the previous work (Chen et al., 2017). Especially for assessing the degree of safety of various teaching strategies and heuristics, we simply obverse the training curves so that we cannot tell of two interleaving curves which training process is safer. Thus, it is imperative to set up some quantitive measurements for both *safety* and *efficiency* evaluations. In this paper, we propose two scalar criteria: *Risk Index* (RI) and *Hitting Time* (HT).

4.1 Risk Index

The *Risk Index* is a nonnegative index designed to indicate how risky the training process could be for evaluating the safety issue during the whole online dialogue policy learning process. Because we expect that the system satisfies the quality-of-service requirement in the early training period, specifically, we hope it can keep a relatively acceptable success rate. It is straightforward to set a success rate *threshold* for the training process. In a real application scenario, this threshold can be obtained by an appropriate user study.

If the success rate over a training process keeps above this threshold all the time, we will think this training process is absolutely safe. Therefore, its RI should equal to zero.

On the other hand, if the success rate over a training process rises and falls and sometimes is below the threshold, it is risky. The riskiness consists of two parts:

- **Disruptiveness**: Sometimes the success rate during a certain period will fall much lower than the threshold, which could be very disruptive. To quantify the disruptiveness, we consider the function

  $$\texttt{dis}(t) = \texttt{threshold} - \%\texttt{succ}(t)$$

  over the training process. The higher the value of $\texttt{dis}(t')$ is, the riskier the training process could be during the period of a certain length centered with time t'.

- **Persistence**: Another thing we should take into account is the duration of the time at high risk. Let $\delta_{\texttt{risk}}(t)$ be the indicator of whether $\texttt{threshold} \geq \%\texttt{succ}(t)$. Then the persistence can be quantified as total risky time

  $$\texttt{per}(T) = \int_{t=0}^{T} \delta_{\texttt{risk}}(t) dt$$

  The longer the danger persists over the training process, the value of persistence of the training process will be, and the riskier it is.

Our *Risk Index* integrate these two contents of riskiness. That is, a nonnegative scalar

$$\texttt{RI} = \int_{t=0}^{T} \texttt{dis}(t) \delta_{\texttt{risk}}(t) dt,$$

which measures the integrated riskiness for the online training process of total length T. The RI also has an intuitive interpretation as the area of the region which is below the threshold line and above training curve. Straightforwardly, high RI indicates poor safety.

4.2 Hitting Time

To measure the efficiency, we proposed the *Hitting Time* in order to show how fast the system learns and reaches the satisfactory performance.

The difficulty of designing such a criterion lies in the dramatic and undamped fluctuation of the test curves, which is inherent in the instability dialogue task. Therefore, many popular criteria for the evaluating dynamic performance in control theory, such as "settling time" and "rise time", cannot be applied to evaluate efficiency here.

We use *Hitting Time* to evaluate efficiency over the fluctuant testing curve first by fitting it to the empirical learning curve

$$f(t) = a - b \cdot e^{-(t/c)^2},$$

where the parameter a is the stationary performance which is predicted as the asymptotic goal of the system, b relates to the initial performance,

and c relates to the climbing speed. This empirical model forces our fitted learning curve be an "S" shape curve satisfying constraints $f'(0) = 0$ and $f''(c/2) = 0$. Then we observe when this fitted learning curve hits the target performance τ and this time (measured in sessions) is *Hitting Time*. It can be calculated analytically as follow

$$\texttt{HT} = c\sqrt{\ln\left(\frac{b}{a-\tau}\right)}.$$

Ideally, the ultimate success rate a should be very close under different settings because of the sufficient training. However, if the success rate keeps very poor during the given sessions, the fitted a will be very low, and even less than the target satisfactory performance τ. In this situation, a is meaningless, and HT becomes a complex number. And this indicates the real hitting time is far larger than given number of sessions T. We will note the HT in this case as **ULT** (Unacceptably Large Time).

In this way, we overcome the fluctuation and make the HT tell us how much time the system takes to hit and surpass the target success rate.

5 Experiments and Results

Three objectives are set for our experiments: (1) Observing the effect of multitask DQN; (2) Contrasting the performances of different teaching schemes (strategies and heuristics) under the companion teaching framework; (3) Observing the safety and efficiency issues under sparse user feedback scenarios.

5.1 Experimental Setup

Our experiments are conducted with the Dialogue State Tracking Challenge 2 (DSTC2) dataset, which is on restaurant information domain (Henderson and Thomson, 2014). The human user is emulated by an agenda-based user simulator with error model (Schatzmann et al., 2007), while the human teacher is emulated by a pre-trained policy model with success rate of about 0.78 through multitask DQN approach without teaching. A rule-based tracker is used for dialogue state tracking (Sun et al., 2014b). The semantic parser is implemented according to an SVM-based method proposed by Henderson et al. (2012). The natural language generator is implemented and modified based on an RNNLG toolkit (Wen et al., 2016, 2015a,c,b).

Early	Rand	SIT	SUT	FPT
None	$p_r = 0.6$	$t_{si} = 5$	$t_{su} = 10$	$\alpha = 1.2$ $w = 25$

Table 1: Experimental configurations of teaching heuristics introduced in section 2.2 and 3.2.

In our experiments, all dialogues are limited to twenty turns. The "dialogue success" is judged by the user simulator according to whether all user goals are satisfied. And for policy learning, we set a small per-turn penalty of one to encourage short interactions, i.e. $R^{\texttt{turn}} = -1$, and a large dialogue success reward of thirty to appeal to successful interactions, i.e. $R^{\texttt{succ}} = 30$, and the discount factor γ is set to one. Table 1 summarizes the heuristics studied in our experiments, together with corresponding configurations which are chosen empirically.

5.2 Observing the Effect of MTL-DQN

The MTL-DQN described in section 3.1 can estimate the prediction of Q^{turn} and Q^{succ} respectively. In our experiments, it was implemented with one shared hidden layer and two dependent hidden layers for two different tasks using MXNet (Chen et al., 2015b).

Figure 3 shows a typical failure in dialogue policy training. The policy showed in the example hasn't been trained well, and it tends to ask the user to repeat over and over again when the confidence score of the user utterance is not high, which causes the user to terminate the dialogue impatiently.

TASK: ask for *moderate spanish* restaurant & request its *address*

		Dialogue Turn	Score	Q^{turn}	Q^{succ}	FP
[1]	System	[SLU] welcomemsg()				
	User	[Top ASR] I would like it to be moderate.	0.68	-6.05	27.34	False
[2]	System	[SLU] repeat()				
	User	[Top ASR] I would like it to be moderate.	0.81	-5.35	26.37	False
[3]	System	[SLU] repeat()				
	User	[Top ASR] Moderate.	0.57	-3.31	20.43	**True**
[4]	System	[SLU] repeat()				
	User	[Top ASR] Bye.	0.61	-0.19	17.96	**True**

Figure 3: An example of failed dialogue while training without teaching. The labels "Score" and "FP" represent for the confidence score of user utterance and the value of *failure prognosis* of the current turn respectively.

This kind of failure can be predicted and corrected in advance. By equation 5, the third turn will be estimated to be *failure prognosis*, which can be a sign for the teacher to intervene and correct the following actions to avoid dialogue failure. Besides, the explicit separate estimation of Q^{turn} and Q^{succ} provides a better understanding of the state of the current turn. For example, although the first turn and second turn have similar Q-values ($Q^{turn}+Q^{succ}$), the latter turn is predicted with less future turns and less possibility to lead to dialogue success. See appendix A for additional successful example.

5.3 Comparing Different Teaching Schemes

Our proposed complete companion teaching framework allows us to teach dialogue systems with different teaching schemes, which consists various strategies and heuristics. In our experiments, we compared 18 schemes consisting of three teaching strategies (CA, EA and EAPC), and six teaching heuristics (Early, Rand, SIT, SUT, FPT and SUT&FPT). The *SUT&FPT* heuristic means the student only ask for advice when equation 2 and 5 are both satisfied. For comparison, we use *No Teaching* (NoTeaching) as the baseline.

To verify the effects of different companion *teaching schemes*, we conduct a set of experiments to see their performances on safety and efficiency dimensions. During training, the teacher can only teach for a limited budget of 1000 turns. All the training curves shown in this paper are moving average curves with a window of size 250 dialogues and over eight runs with an endurable standard error.

5.3.1 Safety Evaluation

To compare effects of different teaching schemes on safety dimension, we use the *Risk Index* (RI) in section 4.1 to quantitatively measure each training process. We set the empirical safety threshold as 65% here. The results are shown in Table 2.

As RIs implies, schemes composed with EAPC strategy is much safer than those composed with other strategies. As for teaching heuristics, FPT, SUT and SUT&FPT are three relatively safer heuristic accompanying different strategies. One exception is that Early teaching looks more suitable for CA. A possible explanation is that when the teacher gives critique earlier, the student will mind its behavior earlier so that increase safety. Figure 4 shows the training curves of on-line learning process under EAPC with various heuristics. Among all 18 teaching schemes, EAPC+SUT is the safest teaching scheme which reduces about 78% risk of no-teaching learning.

	CA	EA	EAPC
Early	**98.5**	110.6	56.1
Rand	193.4	102.4	65.5
FPT	154.4	**86.2**	**53.6**
SIT	230.8	121.7	66.0
SUT	183.5	**95.8**	**44.5***
SUT&FPT	**131.6**	101.8	54.6
NoTeaching	202.9		

Table 2: RIs of learning processes under different teaching schemes. The least risky teaching scheme is annotated with *. For comparing different teaching heuristics with fixed teaching strategy, the smallest RIs in each column are bold and underlined, the 2^{nd} smallest ones are bold only, and the 3^{rd} smallest ones are underlined only. See abbreviations of schemes in section 2.1 and 2.2.

5.3.2 Efficiency Evaluation

We use *Hitting Time* (HT) in section 4.2 to measure the efficiency of learning process under different teaching schemes. The empirical satisfactory target success rate for the student is 70% in our experimental settings.

	CA	EA	EAPC
Early	3390.9	3479.4	4354.7
Rand	3669.0	3518.5	2979.2
FPT	**3089.4**	**2921.1**	**2798.4**
SIT	3576.4	4339.7	3768.7
SUT	3230.4	**2954.5**	3300.2
SUT&FPT	**2890.7**	3393.0	**2702.2***
NoTeaching	3204.1		

Table 3: HTs of test curves of different teaching schemes. The most efficient teaching scheme is annotated with *. For comparing different teaching heuristics with fixed strategy on efficiency issue, the smallest HTs in each column are bold and underlined, and the 2^{nd} smallest bold only. See abbreviations of schemes in section 2.1 and 2.2.

Table 3 contains all HTs of learning process under 18 teaching schemes. Intuitively, The number in the table reflect the number of sessions at which the model achieves target success rate. As it shows, not any teaching scheme will improve the learning efficiency. If the teacher intervenes at an improper time, it will distract system or confuse system even with a right guidance. But teaching when a potential failure exists (F-

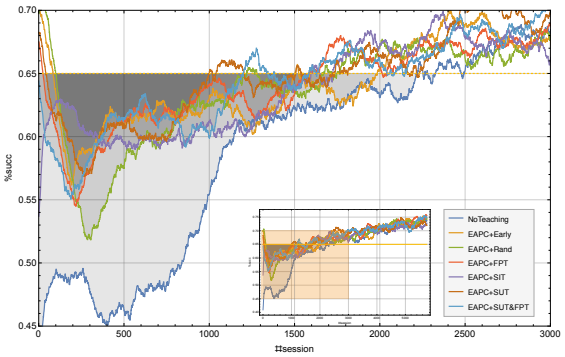


Figure 4: On-line learning process under different teaching schemes (EAPC + different heuristics). The yellow dashed line indicates safe success rate threshold. The area in gray indicates how risky a training process is. See abbreviations of schemes in section 2.1 and 2.2.

PT) is always good for improving learning efficiency. EAPC+SUT&FPT is the teaching scheme that leads to the most efficient learning process in our experiments. Figure 6 gives some example test curves and fitted empirical learning curves of learning process under EAPC with various heuristics.

5.3.3 Teacher's Workload

We also observe teacher's workload of all the teaching schemes since economically utilizing teaching budget is one of our goals.

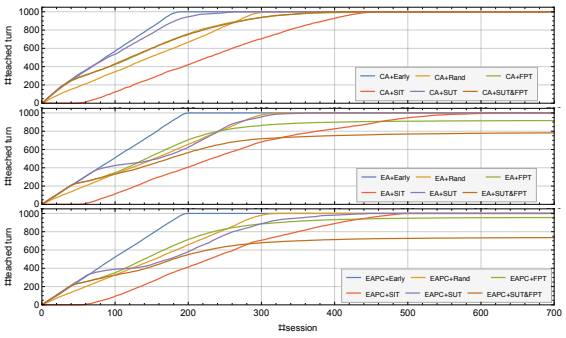


Figure 5: Cumulative usage of teaching budget. The total teaching budget is 1000 for every teaching scheme. See abbreviations of schemes in section 2.1 and 2.2.

Figure 5 illustrates the cumulative usage of teaching budget of 18 teaching schemes. It shows that early teaching is the most costly teaching heuristic so that the teaching budget is soon used up. SIT looks a bit lazy at the beginning and consumes teaching budget slowly. When the teaching

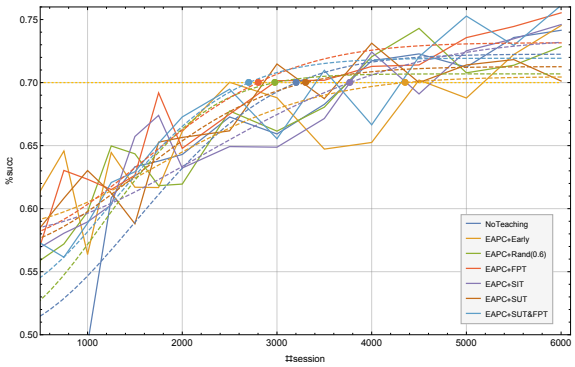


Figure 6: Test curves and fitted empirical learning curves of learning process with different teaching schemes (EAPC+different heuristic). See abbreviations of schemes in section 2.1 and 2.2.

strategy is EA or EAPC, FPT-based schemes do not use up full teaching budget in our experiments. Combine SUT and FPT, the workload is relatively lighter than that of teaching in other heuristics. And through proper teaching schemes, we can make better use of the teaching budget and reduce teacher's workload.

5.4 Safety and Efficiency Issues under Sparse User Feedback Scenarios

In real application scenarios, the user rarely provides feedback at the end of the dialogue, so that safety and efficiency issues are even more serious. To observe the effectiveness of different teaching schemes under sparse user feedback, we conducted experiments with sparse user feedback.

The user feedback rate is set to 30% empirically and experiments are carried out under teaching schemes consisting of EAPC strategies and different heuristics, since EAPC is much safer and more efficient than other teaching strategies.

	RIs	HTs
NoTeaching	608.2	ULT
Early	223.0	6881.8
Rand	226.6	ULT
FPT	**171.5**	**6753.0**
SIT	308.8	7868.4
SUT	183.3	**5876.9**
SUT&FPT	**155.4**	8420.9

Table 4: RIs & HTs of learning processes under EAPC strategy and different heuristics when user feedback rate is 30%. See abbreviations of schemes in section 2.1 and 2.2.

Table 4 records the RIs and HTs of those different learning process when user feedback is sparse.

We can see that when the user feedback rate drops from 100% to 30%, the RIs and HTs increase dramatically. The NoTeaching baseline is very risky and inefficient (its hitting time is even unpredictable within 10000 sessions learning). However, with teaching scheme such as EAPC+FPT, both safety and efficiency can be improved a lot.

6 Conclusions and Future Work

This paper addressed the *safety* and *efficiency* issues of sustainable on-line dialogue policy learning with different teaching schemes, which answer both "how" and "when" to teach, within the complete companion teaching framework. To evaluate the policy learning process precisely, we proposed two measurements, Risk Index (RI) and Hitting Time (HT), to quantify the degree of safety and efficiency. Particularly, through multitask learning, we managed to optimize the predicted remaining turns and dialogue success reward explicitly, based on which we developed a novel *Failure Prognosis based Teaching* (FPT) heuristic to better utilize the fixed teaching budget and make the teaching affordable.

Experiments showed that different teaching schemes have different effects on safety and efficiency dimension. And they also require different workload of the teacher. Among 18 compared teaching schemes, FPT-based heuristics combined with EAPC strategy achieved promising performance on RI and HT, and required relatively slight workload. This result indicates a proper teaching scheme under the companion teaching framework is able to guarantee a sustainable and affordable on-line dialogue policy learning process.

There are several directions for our future work. We expect to deploy our proposed framework in real-world scenarios collaborating with real human teachers to verify the results presented in this paper and discover more potential challenges of on-line dialogue system development. Furthermore, the current study is focused on dialogue success rate, which is a simplification of the human satisfaction evaluation. So future work is needed to take more qualities into consideration to achieve better user experience.

Acknowledgments

This work was supported by the Shanghai Sailing Program No. 16YF1405300, the China NSFC projects (No. 61573241 and No. 61603252) and the Interdisciplinary Program (14JCZ03) of Shanghai Jiao Tong University in China. Experiments have been carried out on the PI supercomputer at Shanghai Jiao Tong University.

References

Ofra Amir, Ece Kamar, Andrey Kolobov, and Barbara Grosz. 2016. Interactive teaching strategies for agent training. In *IJCAI*. International Joint Conferences on Artificial Intelligence.

Benedict Arnold. 1998. Reinforcement learning: An introduction (adaptive computation and machine learning). *IEEE Transactions on Neural Networks*, 9(5):1054.

Rich Caruana. 1997. Multitask learning. *Machine Learning*, 28(1):41–75.

Lu Chen, Pei Hao Su, and Milica Gasic. 2015a. Hyperparameter optimisation of gaussian process reinforcement learning for statistical dialogue management. In *Meeting of the Special Interest Group on Discourse and Dialogue*, pages 407–411.

Lu Chen, Runzhe Yang, Cheng Chang, Zihao Ye, Xiang Zhou, and Kai Yu. 2017. On-line dialogue policy learning with companion teaching. In *EACL*.

Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. 2015b. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *Statistics*.

Jeffery Allen Clouse. 1996. On integrating apprentice learning and reinforcement learning. University of Massachusetts.

L Daubigney, M Geist, S Chandramohan, and O Pietquin. 2012. A comprehensive reinforcement learning framework for dialogue management optimization. *IEEE Journal of Selected Topics in Signal Processing*, 6(8):891–902.

Mehdi Fatemi, Layla El Asri, Hannes Schulz, Jing He, and Kaheer Suleman. 2016. Policy networks with two-stage training for dialogue systems. *arXiv.org*.

Milica Gašić, Filip Jurčíček, Simon Keizer, François Mairesse, Blaise Thomson, Kai Yu, and Steve Young. 2010. Gaussian processes for fast policy optimisation of pomdp-based dialogue managers. pages 201–204.

Matthew Henderson, Milica Gai, Blaise Thomson, and Pirros Tsiakoulis. 2012. Discriminative spoken language understanding using word confusion networks. In *Spoken Language Technology Workshop*, pages 176–181.

Matthew Henderson and Blaise Thomson. 2014. The second dialog state tracking challenge. In *SIGDIAL*, volume 263, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kshitij Judah, Saikat Roy, Alan Fern, and Thomas G Dietterich. 2010. Reinforcement learning via practice and critique advice. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, pages 481–486.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. 1998. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1):99–134.

E Levin, R Pieraccini, and W Eckert. 1997. Learning dialogue strategies within the markov decision process framework. In *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*, pages 72–79.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *Computer Science*.

Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007. Agenda-based user simulation for bootstrapping a pomdp dialogue system. In *NAACL*, pages 149–152, Morristown, NJ, USA. Association for Computational Linguistics.

Pei Hao Su, Milica Gasic, Nikola Mrksic, Lina Rojas-barahona, Stefan Ultes, David Vandyke, Tsung Hsien Wen, and Steve Young. 2016. On-line active reward learning for policy optimisation in spoken dialogue systems. In *Meeting of the Association for Computational Linguistics*, pages 2431–2441.

Kai Sun, Lu Chen, Su Zhu, and Kai Yu. 2014a. The sjtu system for dialog state tracking challenge 2. In *Meeting of the Special Interest Group on Discourse and Dialogue*, pages 318–326.

Kai Sun, Lu Chen, Su Zhu, and Kai Yu. 2014b. The sjtu system for dialog state tracking challenge 2. In *SIGDIAL*, pages 318–326, Stroudsburg, PA, USA. Association for Computational Linguistics.

A. L. Thomaz and C. Breazeal. 2008. Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence*, 172(6):716–737.

Blaise Thomson and Steve Young. 2010. Bayesian update of dialogue state: A pomdp framework for spoken dialogue systems. *Computer Speech and Language*, 24(4):562–588.

Lisa Torrey and Matthew Taylor. 2013. Teaching on a budget: Agents advising agents in reinforcement learning. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 1053–1060. International Foundation for Autonomous Agents and Multiagent Systems.

Zhuoran Wang and Oliver Lemon. 2013. A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information. In *The Sigdial Meeting on Discourse and Dialogue*.

Tsung-Hsien Wen, Milica Gašić, Dongho Kim, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015a. Stochastic Language Generation in Dialogue using Recurrent Neural Networks with Convolutional Sentence Reranking. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*. Association for Computational Linguistics.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, and Steve Young. 2015b. Toward multi-domain language generation using recurrent neural networks. *NIPS Workshop on Machine Learning for Spoken Language Understanding and Interaction*.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, and Steve Young. 2016. Multi-domain neural network language generation for spoken dialogue systems. In *Proceedings of the 2016 Conference on North American Chapter of the Association for Computational Linguistics (NAACL)*. Association for Computational Linguistics.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015c. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Jason D Williams and Steve Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Computer Speech and Language*, 21(2):393–422.

Jason D Williams and Geoffrey Zweig. 2016. End-to-end LSTM-based dialog control optimized with supervised and reinforcement learning. *CoRR*.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.