Speech Communication 84 (2016) 83–96

Contents lists available at ScienceDirect

Speech Communication

journal homepage: www.elsevier.com/locate/specom

Learning cooperative persuasive dialogue policies using framing

Takuya Hiraoka[a,*], Graham Neubig[a], Sakriani Sakti[a], Tomoki Toda[b], Satoshi Nakamura[a]

[a] Graduate School of Information Science, Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma, Nara 603-0192, Japan
[b] Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464–8601, Japan

ARTICLE INFO

Article history:
Received 2 December 2014
Revised 24 August 2016
Accepted 12 September 2016
Available online 13 September 2016

Keywords:
Cooperative persuasive dialogue
Framing
Reinforcement learning
Dialogue modeling
Dialogue system

ABSTRACT

In this paper, we propose a new framework of cooperative persuasive dialogue, where a dialogue system simultaneously attempts to achieve user satisfaction while persuading the user to take some action that achieves a pre-defined system goal. Within this framework, we describe a method for reinforcement learning of cooperative persuasive dialogue policies by defining a reward function that reflects both the system and user goal, and using framing, the use of emotionally charged statements common in persuasive dialogue between humans. In order to construct the various components necessary for reinforcement learning, we first describe a corpus of persuasive dialogues between human interlocutors, then propose a method to construct user simulators and reward functions specifically tailored to persuasive dialogue based on this corpus. Then, we implement a fully automatic text-based dialogue system for evaluating the learned policies. Using the implemented dialogue system, we evaluate the learned policy and the effect of framing through experiments both with a user simulator and with real users. The experimental evaluation indicates that the proposed method is effective for construction of cooperative persuasive dialogue systems.

1. Introduction

With the basic technology supporting dialogue systems maturing, there has been more interest in recent years about dialogue systems that move beyond the traditional task-based or chatter bot frameworks. In particular there has been increasing interest in dialogue systems that engage in persuasion or negotiation (Georgila, 2013; Georgila and Traum, 2011; Guerini et al., 2003; Heeman, 2009; Mazzotta and de Rosis, 2006; Mazzotta et al., 2007; Nguyen et al., 2007; Paruchuri et al., 2009). In this paper, we propose a method for learning *cooperative* persuasive dialogue systems, in which we place a focus not just on the success of persuasion (the *system* goal) but also user satisfaction (the *user* goal). This variety of dialogue system has the potential to be useful in situations where the user and system have different, but not mutually exclusive goals. An example of this is a sales situation where the user wants to find a product that matches their taste, and the system wants to successfully sell a product, ideally one with a higher profit margin.

Creating a system that both has persuasive power and is able to ensure that the user is satisfied is not an easy task. In order to tackle this problem with the help of recent advances in statistical dialogue modeling, we build our system upon the framework of reinforcement learning and specifically partially observable Markov decision processes (POMDP) (Levin et al., 2000; Williams and Young, 2007; 2007), which we describe in detail in Section 2. In the POMDP framework, it is mainly necessary to define a *reward* representing the degree of success of the dialogue, the set of *actions* that the system can use, and a *belief state* to keep track of the system beliefs about its current environment. Once these are defined, reinforcement learning enables the system to learn a policy maximizing the reward.

In this paper, in order to enable the learning of policies for cooperative persuasive dialogue systems, we tailor each of these elements to the task at hand (Section 4):

Reward: We present a method for defining the reward as a combination of the user goal (user satisfaction), the system goal (persuasive success), and naturalness of the dialogue. This is in contrast to research in reinforcement learning for slot-filling dialogue, where the system aims to achieve only the user goal (Levin et al., 2000; Williams and Young, 2007; 2007), or for persuasion and negotiation dialogues, where the system receives a reward corresponding to only the system goal (Georgila, 2013; Georgila and Traum, 2011; Heeman, 2009; Paruchuri et al., 2009). We use a human-to-human persuasive dialogue corpus (Section 3, Hiraoka et al.,

* Corresponding author.
E-mail addresses: takuya-h@is.naist.jp, hiraoka.et.al@gmail.com (T. Hiraoka), neubig@is.naist.jp (G. Neubig), ssakti@is.naist.jp (S. Sakti), tomoki@icts.nagoya-u.ac.jp (T. Toda), s-nakamura@is.naist.jp (S. Nakamura).

http://dx.doi.org/10.1016/j.specom.2016.09.002
0167-6393/

2014a) to train predictive models for achievement of a human persuadee's and a human persuader's goals, and introduce these models to reward calculation to enable the system to learn a policy reflecting knowledge of human persuasion.

- **System Action:** We introduce framing (Irwin et al., 2013), which is known to be important for persuasion, as a system action (i.e., system dialogue act). Framing uses emotionally charged words (positive or negative) to explain particular alternatives. In the context of research that applies reinforcement learning to persuasive (or negotiation) dialogue, this is the first work that considers framing in this way. In this paper the system controls the polarity (positive or negative) and the target alternative of framing (see Table 3 for an example of framing).
- **Belief State:** As the belief state, we use the dialogue features used in calculating the reward function. For example, whether the persuadee has been informed that a particular option matches their preference was shown in human dialogue to be correlated with persuasive success, which is one of the reward factors. Some of the dialogue features reward calculation can not be observed directly by the system, and thus we incorporate them into the belief state.

Based on this framework, we construct the first fully automated text-based cooperative persuasive dialogue system (Section 5). To construct the system, in addition to the policy module, natural language understanding (NLU), and natural language generation (NLG) are required. We construct an NLU module using the human persuasive dialogue corpus and a statistical classifier. In addition, we construct an NLG module based on example-based dialogue, using a dialogue database created from the human persuasive dialogue corpus.

Using this system, we evaluate the learned policy and the utility of framing (Section 6). To our knowledge, in context of the research for persuasive and negotiation dialogue, it is first time that a learnt policy is evaluated with fully automated dialogue system. The evaluation is done both using a user simulator and real users.

This paper comprehensively integrates our work in Hiraoka et al. (2014b) and Hiraoka et al. (2015), with a more complete explanation and additional experiments. Specifically regarding the additional experimental results, in this paper we additionally perform 1) experimental evaluation using a reward function which exactly matches the learning phase (Section 6.1.1, 6.2), and 2) an evaluation of the effect of NLU error rate (Section 6.1.2).

2. Reinforcement learning

In reinforcement learning, policies are updated based on exploration in order to maximize a reward. In this section, we briefly describe reinforcement learning in the context of dialogue. In dialogue, the policy is a mapping function from a dialogue state to a particular system action. In reinforcement learning, the policy is learned to maximize the reward function, which in traditional task-based dialogue system is user satisfaction or task completion (Walker et al., 1997). Reinforcement learning is often applied to models based on the frameworks of Markov decision processes (MDP) or partially observable Markov decision processes (POMDP).

In this paper, we follow a POMDP-based approach. A POMDP is defined as a tuple $\langle S, A, P, R, O, Z, \gamma, b_0 \rangle$ where S is the set of states (representing different contexts) which the system may be in (the system's world), A is the set of actions of the system, P: $S \times A \rightarrow P(S, A)$ is the set of transition probabilities between states after taking an action, R: $S \times A \rightarrow \Re$ is the reward function, O is a set of observations that the system can receive about the world, Z is a set of observation probabilities Z: $S \times A \rightarrow Z(S, A)$, and γ a discount factor weighting longterm rewards. At any given time step i the world is in some unobserved state $s_i \in S$. Because s_i is not known exactly, we keep a hypothesis over states called a belief state b.[1] When the system performs an action $\alpha_i \in A$ based on b, following a policy $\pi: b \rightarrow A$, it receives a reward $r_i(s_i, \alpha_i) \in \Re$ and transitions to state s_{i+1} according to $P(s_{i+1}|s_i, \alpha_i) \in P$. The system then receives an observation o_{i+1} according to $P(o_{i+1}|s_{i+1}, \alpha_i)$. The quality of the policy π followed by the agent is measured by the expected future reward, also called the Q-function, $Q^{\pi}: b \times A \rightarrow \Re$.

In this framework, we use Neural fitted Q Iteration (Riedmiller, 2005) for learning the system policy. Neural fitted Q Iteration is an offline value-based method, and optimizes the parameters to approximate the Q-function. Neural fitted Q Iteration repeatedly performs 1) sampling training experience using a POMDP through interaction and 2) training a Q-function approximator using training experience. Neural fitted Q Iteration uses a multi-layered perceptron as the Q-function approximator. Thus, even if the Q-function is complex, Neural fitted Q Iteration can approximate the Q-function better than using a linear approximation function. In a preliminary experiment, we confirmed that this is true in our domain as well. Once the Q-function is learned, the system creates the policy based on the Q-function. In our research, we use the ϵ-greedy policy. Namely, the system randomly selects an action with a probability of ϵ, otherwise selects the action which maximizes the Q-function given the current state.

As Porta et al. noted, (discrete-state) POMDPs can be seen as MDPs with continuous state space that has one dimension per state, which represents the probability of each state in original POMDP (Porta et al., 2006). More concretely, assuming the state space of POMDPs is the discrete set $S = \{s_1, \ldots, s_n, \ldots, s_N\}$, the state s_i' in corresponding MDPs at time step i can be represented as follows:

$$s_i' = (b_i(s_1), \ldots, b_i(s_n) \ldots, b_i(s_N)),$$

where b_i represents belief state at turn i. In our paper, we follow that discrete-state POMDPs, and treat it as MDPs with continuous state space. So neural fitted Q iteration should be an appropriate method to solve this problem.

3. Cooperative persuasive dialogue corpus

In this section, we give a brief overview of cooperative persuasive dialogue, and a human dialogue corpus that we use to construct the dialogue models and dialogue system described in later sections. Based on the persuasive dialogue corpus (Section 3.1), we define and quantify the actions of the cooperative persuader (Section 3.2). In addition, we annotate persuasive dialogue acts of the persuader from the point of view of framing (Section 3.3).

3.1. Outline of persuasive dialogue corpus

The cooperative persuasive dialogue corpus (Hiraoka et al., 2014a) consists of dialogues between a salesperson (persuader) and customer (persuadee) as a typical example of persuasive dialogue. The salesperson attempts to convince the customer to purchase a particular product (decision) from a number of alternatives (decision candidates). We define this type of dialogue as "sales dialogue." More concretely, the corpus assumes a situation where the

[1] Note that, in this paper we use "belief state" to refer to both 1) known information about a part of the dialogue state (e.g., the most recent system action), and 2) a distribution over all possible hypotheses regarding a part of the dialogue state (e.g., the most recent users' dialogue act). We explain about how we define this belief state in our domain in Section 4.2.3.