Dialogue Management based on Multi-domain Corpus

Wendong Ge
Institute of Automation
Chinese Academy of Sciences
Beijing, China
`wendong.ge@ia.ac.cn`

Bo Xu
Institute of Automation
Chinese Academy of Sciences
Beijing, China
`xubo@ia.ac.cn`

Abstract

Dialogue Management (DM) is a key issue in Spoken Dialogue System. Most of the existing data-driven DM schemes train the dialogue policy for some specific domain (or vertical domain), only using the dialogue corpus in this domain, which might suffer from the scarcity of dialogue corpus in some domains. In this paper, we divide Dialogue Act (DA), as semantic representation of utterance, into DA type and slot parameter, where the former one is domain-independent and the latter one is domain-specific. Firstly, based on multiple-domain dialogue corpus, the DA type prediction model is trained via Recurrent Neutral Networks (RNN). Moreover, DA type decision problem is modeled as a multi-order POMDP, and transformed to be a one-order MDP with continuous states, which is solved by Natural Actor Critic (NAC) algorithm and applicable for every domain. Furthermore, a slot parameter selection scheme is designed to generate a complete machine DA according to the features of specific domain, which yields the Multi-domain Corpus based Dialogue Management (MCDM) scheme. Finally, extensive experimental results illustrate the performance improvement of the MCDM scheme, compared with the existing schemes.

1 Introduction

With the fast development of Automatic Speech Recognition (ASR) and Natural Language Processing (NLP), a lot of Spoken Dialogue Systems (SDS) appear in our lives as information assistants. In SDS, Dialogue Management (DM), as one of the most important modules, not only determines the current machine reaction, but also controls the process of future dialogue. Thus, it is important to study DM in the establishment of SDS.(Michael, 2002)

A lot of studies have been done on DM. (Thomson, 2010) introduces a new POMDP-based framework for building spoken dialogue systems by using Bayesian updates of the dialogue state. (Olivier, 2011) explores the possibility of using a set of approximate dynamic programming algorithms for policy optimization in SDS, which are combined to a method for learning a sparse representation of the value function. (Annemiek, 2012) analyzes current dialogue management in operating unmanned systems and develops a more advanced way of dialogue management and accompanying dialogue manager. (Yuan, 2012) proposes a task ontology model for domain independent dialogue management, where the knowledge of a specific task is modeled in its task ontology which is independent from dialogue control. (Daubigney, 2012) proposes to apply the Kalman Temporal Differences (KTD) framework to the problem of dialogue strategy optimization so as to address all these issues in a comprehensive manner with a single framework. (Emmanuel, 2013) proposes a scheme to utilize a socially-based reward function for Reinforcement Learning and uses it to fit the user adaptation issue for dialogue management. (Daniele, 2013) describes an architecture for a dialogue management system to be employed in serious games for natural language interaction with non-player characters. (Young et al., 2013) provides an overview of the current state of the art in the development of POMDP-based spoken dialog systems. (Hao, 2014) presents a dialogue manager based on a log-linear probabilistic model and uses context-free grammars to impart hierarchical structure to variables and features. (Kallirroi, 2014) uses single-agent Reinforcement Learning and multi-agent Reinforcement Learning for

Proceedings of the SIGDIAL 2015 Conference, pages 364–373,
Prague, Czech Republic, 2-4 September 2015.

learning dialogue policies in a resource allocation negotiation scenario. To sum up, most of these previous studies establish a specific-domain DM model, only using the dialogue corpus in this domain, which might suffer from scarcity of dialogue corpus in some vertical domains.

In this paper, we mainly consider the domains about slot-filling tasks such as hotel reservation, flight ticket booking, and shopping guidance. We utilize dialogue act (DA) as semantic representation of utterance, and divide it into DA type and slot parameter, where the former one is domain-independent and the latter one is domain-specific. Based on the dialogue corpus in multiple domains, we train the current machine DA type prediction model and the next user DA type prediction model via Recurrent Neutral Networks (RNN). With these two prediction models, the current machine DA type decision problem is modeled as a multi-order POMDP, and transformed to be a one-order MDP with continuous states, which could be solved by Natural Actor Critic (NAC) algorithm. This general DA type decision model could be applied to multiple domains. After calculating the machine DA type, we design a slot parameter selection scheme to generate a complete machine DA according to the features of vertical domain, which yields the Multi-domain Corpus based Dialogue Management (MCDM) scheme. The advantages of this scheme are as follows.

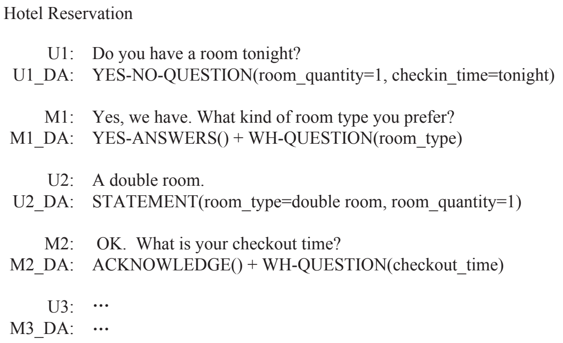
- The MCDM scheme separates DA into DA type and slot parameter, where DA type is domain-independent. It utilizes multi-domain corpus to train a general DA type decision model that is applicable to every domain. Namely, it extracts general dialogue knowledge from all the domains and put it into vertical domain DM model. Even for some vertical domain with insufficient dialogue corpus, it could work well.
- The MCDM scheme encodes the dialogue historical information into history vector via RNN, and utilizes this history vector to estimate the distribution over possible current machine DA type and the distribution over possible next user DA. Theoretically, the history vector contains the whole dialogue history, even the information of utterances in the first turn.
- The MCDM scheme models the machine DA type decision problem as a POMDP, which makes a decision in the limitation of unreliable ASR and NLP, and achieves a tradeoff between dialogue popularity (frequency of dialogue pattern) and slot-filling efficiency.
- The MCDM scheme designs a slot parameter selection method for generated machine DA type, according to the features of vertical domain.

The rest of this paper is organized as follows. In Section 2, system model is introduced. Section 3 establishes the current machine DA type prediction model and the next user DA type prediction model via RNN, and Section 4 models the DA type decision problem as a POMDP. Section 5 describes slot selection scheme for the given DA type and slot filling process. Extensive experimental results are provided in Section 6 to illustrate the performance comparison, and Section 7 concludes this study.

2 System Model

Generally, the SDS operates as follows. Receiving user voice input, Natural Language Understanding (NLU) module transforms it into semantic representation such as DA. There are two steps in NLU: the first is Automatic Speech Recognition (ASR) that turns voice into text (Willie, 2004) (Vinyals, 2012); the second is Semantic Decoder (SD) that extracts DA from text (Hea, 2006) (Mairesse, 2009). NLU is hardly able to analyze the exact DA of user input due to inevitable ambiguity, uncertainty and errors in ASR and SD. Thus, the distribution of possible DAs is utilized to represent the result of NLU. According to this distribution and dialogue history, Dialogue Management (DM) module calculates the optimal output DA. Finally, Natural Language Generation (NLG) module transforms output DA into voice, including sentence generation that generates sentence based on DA (Mairesse, 2007) and Text To Speech (TTS) that turns sentence text into output voice (Clark, 2004) (Zen, 2007).

In this paper, we focus on DM in SDS for the slot-filling task. Firstly, we collect the dialogue corpus in multiple domains such as hotel reservation, flight ticket booking and shopping guidance. We label the dialogue corpus with DA set introduced in (Stolcke, 2000). This set includes 42 DA labels, which is wildly used and cited over 600



Hotel Reservation

U1: Do you have a room tonight?
U1_DA: YES-NO-QUESTION(room_quantity=1, checkin_time=tonight)

M1: Yes, we have. What kind of room type you prefer?
M1_DA: YES-ANSWERS() + WH-QUESTION(room_type)

U2: A double room.
U2_DA: STATEMENT(room_type=double room, room_quantity=1)

M2: OK. What is your checkout time?
M2_DA: ACKNOWLEDGE() + WH-QUESTION(checkout_time)

U3: ···
M3_DA: ···

Figure 1: an example of labeled dialogue

times in Google Scholar. Fig.1 is an example of labeled dialogue. Additionally, DA is divided into two parts: DA type and slot parameters. For example, for the DA "WH-QUESTION (room_type)", the DA type is "WH-QUESTION", and the slot parameter is "room_type". It is observed that DA type is domain-independent while slot parameter is domain-specific.

Based on these labeled dialogues, we design the multi-domain DM scheme, which could be divided into two steps:

- DA type decision: The dialogue historical information is encoded into history vector via RNN. Based on this vector, we estimate the possible current machine DA types and possible next user DA types, which will be introduced in section 3. With these DA type estimations, the DA type decision process is modeled as a POMDP, which will be introduced in section 4. This DA type decision model is applicable to every vertical domain.

- Slot parameter selection: After determining the DA type, the slot parameter selection scheme is designed according to the features of vertical domain, in order to generate a complete machine output DA.

3 RNN based Prediction Model

In this section, we introduce current machine DA type prediction model and next user DA type prediction model. (Nagata, 1994) utilizes N-gram Language Model to predict DA type. As quantity of DA type combination in historical epoches grows exponentially with the increment of N, the parameter N could not be too big, namely Bigram and Tri-gram are usually used. Thus, N-gram based DA type prediction model could not consider the dialogue historical information efficiently. In order to solve this problem, we utilize RNN to predict the DA type. The details of prediction model are as follows.

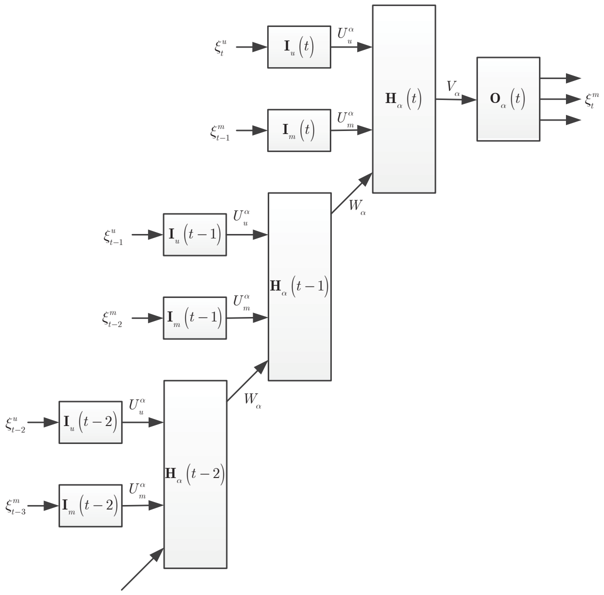


Figure 2: RNN for the current machine DA prediction

Firstly, the sentences in dialogue corpus are divided into two sets: sentence set spoken by machine (or service provider such as customer service representative in hotel reservation) and sentence set spoken by user (customer). We count the DA type combination in these two sets respectively, where the machine DA type combination set is denoted as $\mathscr{C}_m$ and the user DA type combination set is denoted as $\mathscr{C}_u$.

Secondly, we predict the probability distribution over current machine DA types. We denote the combination of DA type corresponding to user and machine sentences in t-th turn as ξ_t^m and ξ_t^u, where $\xi_t^m \in \mathscr{C}_m$ and $\xi_t^u \in \mathscr{C}_u$. The probability distribution over current machine DA types is determined by the current user DA type, the last machine DA type and the previous dialogue historical information, which is denoted as

$$\Pr\left\{\xi_t^m \left| \xi_t^u, \xi_{t-1}^m, \xi_{t-1}^u, \cdots, \xi_1^m, \xi_1^u \right.\right\} \qquad (1)$$

We utilize RNN to estimate the conditional probability in equation (1). The architecture of this RNN is illustrated in Fig. 2. The inputs of RNN in the t-th turn are ξ_t^u and ξ_{t-1}^m. The input layers in this turn are one-hot representations (Turian, 2010) of ξ_t^u and ξ_{t-1}^m, denoted as $\mathbf{I}_u(t)$ and $\mathbf{I}_m(t)$. (The size of $\mathbf{I}_u(t)$ or $\mathbf{I}_m(t)$ is equivalent to $|\mathscr{C}_u|$

or $|\mathscr{C}_m|$. There is only one 1 in $\mathbf{I}_u(t)$ or $\mathbf{I}_m(t)$ corresponding to the ξ_t^u or ξ_{t-1}^m position, and other elements are zeros.) We denote hidden layer as $\mathbf{H}_\alpha(t)$ and output layer as $\mathbf{O}_\alpha(t)$. Thus, $\mathbf{O}_\alpha(t)$ is the probability distribution of current machine DA type combination, which could be calculated as (Mikolov, 2010)

$$\begin{aligned}\mathbf{H}_\alpha(t) = f\,(\mathbf{U}_u^\alpha \mathbf{I}_u(t) + \mathbf{U}_m^\alpha \mathbf{I}_m(t) \\ + \mathbf{W}_\alpha \mathbf{H}_\alpha(t-1))\end{aligned} \tag{2}$$

and

$$\mathbf{O}_\alpha(t) = g\,(\mathbf{V}_\alpha \mathbf{H}_\alpha(t)) \tag{3}$$

where $f(\cdot)$ is a sigmoid function, namely $f(x) = 1/(1+e^{-x})$ and $g(\cdot)$ is a soft-max function, namely $g(x_i) = e^{x_i} \Big/ \sum_{i=1}^{N_g} e^{x_i}$. The parameters of this RNN could be trained by the Back Propagation Through Time (BPTT) algorithm (Mikolov, 2012).

Thirdly, we predict the probability distribution over next user DA types based on the current machine DA type, the current user DA type and the previous dialogue historical information, which is denoted as

$$\Pr\left\{\xi_{t+1}^u \left| \xi_t^m, \xi_t^u, \xi_{t-1}^m, \xi_{t-1}^u, \cdots, \xi_1^m, \xi_1^u \right.\right\} \tag{4}$$

We also utilize the RNN with the same architecture mentioned above to predict this conditional probability, but inputs and outputs are different. The inputs in the t-th turn are ξ_t^m and ξ_t^u, and the output is the probability distribution of ξ_{t+1}^u, which is illustrated in Fig 3. The parameters of this RNN could be also trained by BPTT.

Besides, in different vertical domains, the pattern of DA type evolution might be different. For example, there might be a lot of question-answer exchanges in hotel reservation domain, because machine needs to collect a lot of information about reservation such as room type, check-in time and client name, and user also needs to inquire a lot of information about room and hotel such as room price and hotel address. While in other domains such as restaurant catering, the slots requested by machine are more than slots requested by user, which might lead to less question-answer exchanges. Thus, in order to solve this problem, when training some specific domain (target domain), we copy the dialogue corpus in the target domain repeatedly and control the size of target-domain corpus to be K_d times than the size of corpus in other domains, which increases the size of the corpus in the target domain and makes DA type prediction model fit for the features of the target domain.

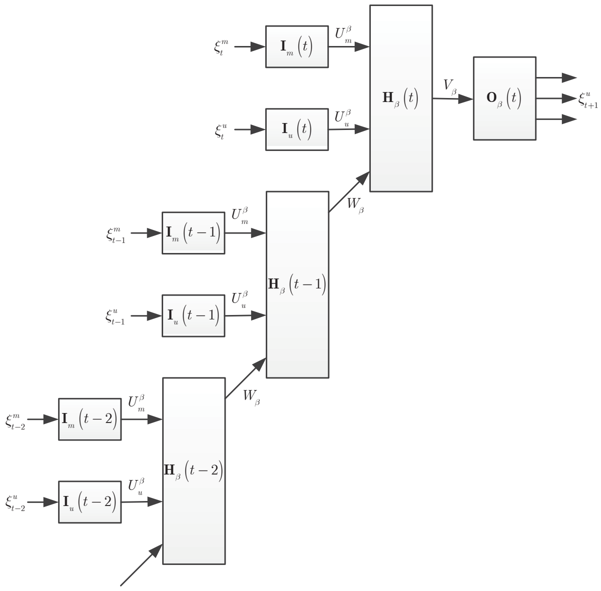


Figure 3: RNN for the next user DA prediction

4 Model DM as POMDP

In this section, we use POMDP (Littman, 2009) to model DM problem, illustrated in Fig.4. State is defined as the combination of user DA types in each turn, namely $s_t = \xi_t^u \in \mathscr{C}_u$. Action is defined as the combination of machine DA types in each turn, namely $a_t = \xi_t^m \in \mathscr{C}_m$. As the user DAs in $(t+1)$-th turn are not only determined by the user and machine DAs in t-th turn, but also related to the previous DAs, we define τ as a window size for this kind of relevance. Thus, the state transition probability could be represented as

$$\begin{aligned}&\Pr\left\{s_{t+1} \left| a_t, s_t, \cdots, a_1, s_1 \right.\right\} \\ &= \Pr\left\{s_{t+1} \left| a_t, s_t, \cdots, a_{t-\tau+1}, s_{t-\tau+1} \right.\right\} \\ &= \Pr\left\{\xi_{t+1}^u \left| \xi_t^m, \xi_t^u, \cdots, \xi_{t-\tau+1}^m, \xi_{t-\tau+1}^u \right.\right\}\end{aligned} \tag{5}$$

This conditional probability could be estimated by RNN in section 3, which is denoted as π_{t+1}^u. Observation is defined as user input voice in each turn, denoted as $o_t \in \mathscr{O}$. As s_t could not be observed directly, o_t is utilized to estimate s_t, namely $\Pr\{s_t \left| o_t \right.\}$, which could be obtained from ASR and SD and denoted as p_t^o. The reward function includes two parts: slot-filling efficiency and dialogue popularity, which is denoted as

$$\begin{aligned}r_t(s_t, a_t, s_{t+1}) = \lambda_1 F(s_t, a_t, s_{t+1}) \\ + \lambda_2 G(s_t, a_t, s_{t+1})\end{aligned} \tag{6}$$

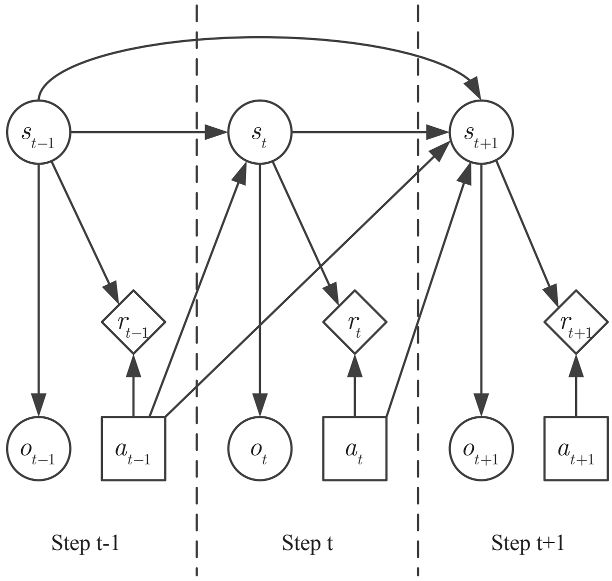


Figure 4: POMDP

where $F(\cdot)$ is a function mapping from the current user DA type, the current machine DA type and the next user DA type to the normalized quantity of filled slots that will be introduced in section 5, $G(\cdot)$ is the normalized quantity of sequence (s_t, a_t, s_{t+1}) that could be counted from dialogue corpus and represent dialogue popularity, λ_1 and λ_2 are the weights of slot filling reward and popularity reward, and $\lambda_1 + \lambda_2 = 1$. The policy is defined as a mapping from observation to action, which is denoted as $\zeta \in \mathscr{X} : \mathscr{O} \to \mathscr{A}$. Thus, the DM problem is to find out the optimal policy to maximize the total expected discount reward, which is shown as

$$\begin{array}{l} \max\limits_{\zeta \in \mathscr{X}} E_\zeta \left[\sum\limits_{t=1}^{T} \beta r_t (s_t, a_t, s_{t+1}) \right] \\ s.t. \\ \Pr\{s_{t+1} | a_t, s_t, \cdots, a_{t-\tau+1}, s_{t-\tau+1}\} = \pi_{t+1}^{u} \\ \Pr\{s_t | o_t\} = p_t^o \end{array} \tag{7}$$

where β is a time discount factor. This problem is a τ order POMDP, which is difficult to slove directly. In the following, it will be transformed to be a MDP with continuous states.

We define belief state as $b_t \in \mathscr{B}$ to represent the distribution over possible states in the t-th turn, not only based on the current voice input, but also based on the dialogue history. The belief state updating process is the process of calculating b_{t+1} according to $\{b_t, b_{t-1}, \cdots, b_{t-\tau+1}\}$, which could be represented as

$$\begin{array}{l} b_{t+1} = \kappa \cdot \Pr\{o_{t+1} | s_{t+1}\} \sum\limits_{s_t} \cdots \sum\limits_{s_{t-\tau+1}} \Pr\{s_{t+1} \\ \qquad | s_t, a_t, \cdots, s_{t-\tau+1}, a_{t-\tau+1}\} \prod\limits_{i=t-\tau+1}^{t} b_i \end{array} \tag{8}$$

where κ is normalization constant. The deduction of this updating processing will be found in Appendix A. As user input voice is a continuous signal and different people have different habits of pronunciation and semantic representation, it is hard to estimate $\Pr\{o_{t+1} | s_{t+1}\}$ directly. Thus, according to Bayes Rules, $\Pr\{o_{t+1} | s_{t+1}\}$ could be shown as

$$\Pr\{o_{t+1} | s_{t+1}\} = \frac{\Pr\{s_{t+1} | o_{t+1}\} \Pr\{o_{t+1}\}}{\Pr\{s_{t+1}\}} \tag{9}$$

where $\Pr\{s_{t+1} | o_{t+1}\}$ could be estimated by ASR and SD, $\Pr\{s_{t+1}\}$ is prior distribution that could be counted in corpus, denoted as p_{t+1}^s, and $\Pr\{o_{t+1}\}$ is the same for different s_{t+1} that could be deleted. For belief state, the reward function could be redefined as

$$\begin{array}{l} r_t (b_t, a_t, \cdots, b_{t-\tau+1}, a_{t-\tau+1}) = \\ \sum\limits_{s_t} \cdots \sum\limits_{s_{t-\tau+1}} \Big(r_t (s_t, a_t, s_{t+1}) \cdot \Pr\{s_{t+1} \\ | s_t, a_t, \cdots, s_{t-\tau+1}, a_{t-\tau+1}\} \prod\limits_{i=t-\tau+1}^{t} b_i + \\ \Pr\{a_t | s_t, a_{t-1}, \cdots, s_{t-\tau+1}, a_{t-\tau}\} \prod\limits_{i=t-\tau+1}^{t} b_i \Big) \end{array} \tag{10}$$

where the first part is the belief form of state reward and the second part is the expectation of the current machine DA type probability estimated by RNN in the section 3. We redefine the policy as a mapping from belief state to action, which is denoted as $\zeta' \in \mathscr{X}' : \mathscr{B} \to \mathscr{A}$. Thus, the problem (7) could be reformulated as

$$\begin{array}{l} \max\limits_{\zeta' \in \mathscr{X}'} E_{\zeta'} \left[\sum\limits_{t=1}^{T} \beta r_t (b_t, a_t, \cdots, b_{t-\tau+1}, a_{t-\tau+1}) \right] \\ s.t. \\ b_{t+1} = \kappa \cdot \frac{p_t^o}{p_{t+1}^s} \sum\limits_{s_t} \cdots \sum\limits_{s_{t-\tau+1}} \Pr\{s_{t+1} | s_t, a_t, \\ \cdots, s_{t-\tau+1}, a_{t-\tau+1}\} \prod\limits_{i=t-\tau+1}^{t} b_i, \\ b_0 = p_0^o. \end{array} \tag{11}$$

This problem is a τ order MDP with continuous states, which will be transformed to be one order MDP.

We redefine new state as the sequence of belief state and action from the $(t-\tau+1)$-th turn to the t-th turn, which is denoted as $\bar{s}_t = \{b_t, a_{t-1}, b_{t-1}, \cdots, a_{t-\tau+1}, b_{t-\tau+1}\}$ and $\bar{s}_t \in \bar{\mathscr{S}}$. Thus, the state transition probability could be shown as

$$\begin{aligned}&\Pr\{\bar{s}_{t+1}\,|\bar{s}_t\ , a_t\}\\&= \Pr\{b_{t+1}, a_t, b_t, \cdots, a_{t-\tau+2}, b_{t-\tau+2}\\&\quad |b_t, a_{t-1}, b_{t-1}, \cdots, a_{t-\tau+1}, b_{t-\tau+1}\ , a_t\}\\&= \Pr\{b_{t+1}, |b_t, a_t, \cdots, b_{t-\tau+1}, a_{t-\tau+1}\,\}\end{aligned} \tag{12}$$

which could be obtained from equation (8) and denoted as $\bar{\pi}_{\bar{s}_t,a_t}^{\bar{s}_{t+1}}$. The reward function could be rewritten as

$$\bar{r}_t\left(\bar{s}_t, a_t\right) = r_t\left(b_t, a_t, \cdots, b_{t-\tau+1}, a_{t-\tau+1}\right) \tag{13}$$

We redefine the policy as a mapping from new state to action, which is denoted as $\bar{\zeta} \in \bar{\mathscr{Z}} : \bar{\mathscr{S}} \to \mathscr{A}$. Thus, the problem (11) could be reformulated as

$$\begin{aligned}&\max_{\bar{\zeta}\in\bar{\mathscr{Z}}} E_{\zeta}\left[\sum_{t=1}^{T} \beta\bar{r}_t\left(\bar{s}_t, a_t\right)\right]\\&s.t.\\&\Pr\{\bar{s}_{t+1}\,|\bar{s}_t\ , a_t\} = \bar{\pi}_{\bar{s}_t,a_t}^{\bar{s}_{t+1}}\end{aligned} \tag{14}$$

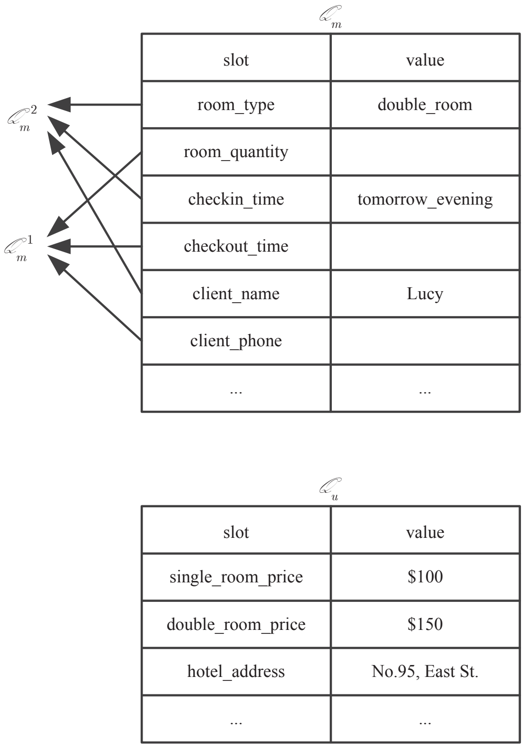
This problem is a one order MDP with continuous states, which could be solved by Natural Actor Critic algorithm (Peters, 2008) (Bhatnagar, 2009).

5 Slot Selection and Slot-filling

After determining the DA type of machine, the next step is selecting slot parameter for it to yield a complete output DA. Firstly, the parameters for DAs could be classified as follows.

- $\emptyset$: some DAs have no parameters, such as YES-ANSWERS ()
- $slot$: parameter of some DA is a slot, such as WH-QUESTION (room_type)
- $slot = value$: parameter of some DA is a slot value pair , such as STATEMENT (double_room_price= $100)

Additionally, The slots in human-machine dialogue could be divided into two categories, illustrated in Fig.5:



slot	value
room_type	double_room
room_quantity	
checkin_time	tomorrow_evening
checkout_time	
client_name	Lucy
client_phone	
...	...

slot	value
single_room_price	$100
double_room_price	$150
hotel_address	No.95, East St.
...	...

Figure 5: slot classification

- Slots requested from machine to users, such as room_type, checkin_time, which is denote as $\mathscr{Q}_m$. The values of these slots are unknown for machine before the dialogue. In dialogue processing, we denote unfilled slots as $\mathscr{Q}_m^1$ and filled slots as $\mathscr{Q}_m^2$.
- Slots requested from users to machine, such as double_room_price, hotel_address, which is denote as $\mathscr{Q}_u$. The values of theses slot are known for machine before the dialogue.

The purpose of human-machine dialogue is to exchange these slot information. For example, in hotel reservation, machine is to request values of slots in $\mathscr{Q}_m$, while user is to request values of slots in $\mathscr{Q}_u$ in order to determine the values of slots in $\mathscr{Q}_m$ that user will inform to machine. Besides, it is obvious that $\mathscr{Q}_m^1$ is a set of slots, $\mathscr{Q}_m^2$ and $\mathscr{Q}_u$ are sets of slot value pairs.

Thus, there are three situations in the slot selection for a machine DA type

- If the parameter classification corresponding to the machine DA type is $\emptyset$, it is no need to select slot.
- If the parameter classification corresponding to the machine DA type is a slot, it is selected from $\mathscr{Q}_m^1$.
- If the parameter classification corresponding to the machine DA type is a slot value

pair, it is selected from $\mathscr{Q}_m^2$ and $\mathscr{Q}_u$. For example, for “STATEMENT”, it is selected from $\mathscr{Q}_u$; for “DECLARATIVE YES-NO-QUESTION”, it is selected from $\mathscr{Q}_m^2$.

In slot selection process, the orders of slots in $\mathscr{Q}_m^1$, $\mathscr{Q}_m^2$ and $\mathscr{Q}_u$ ought to be learned based on the dialogue corpus in vertical domain such as slot sequence in the task, slot dependency, slots that user request, domain expertise knowledge and so forth.

After obtaining a complete the machine, the last task is filling the slots according to the current DA and historical DA sequence. In this paper, we use handcrafted rules to fill the slots. For example, according to the DA sequence

user: STATEMENT (room_type = double room)
machine: DECLARATIVE YES-NO
-QUESTION (room_type = double room)
user: YES ANSWER ()
machine: ACKNOWLEDGE ()

The slot “room_type” is filled by the value “double room”. This knowledge could be represented by the first order logic (Smullyan, 1995) as follow.

STATEMENT $(X = A) \wedge$ DECLARATIVE YES-NO-QUESTION $(X = A) \wedge$ YES ANSWER () $\wedge$ ACKNOWLEDGE () $\Rightarrow \mathit{fill}\,(X, A)$

6 Experimental Results

In this section, we compare the performance of the proposed DM schemes and the existing DM scheme. The DM scheme proposed in this paper is named as the RNN-MCDM scheme. In the N-Gram-MCDM scheme, the DA type is estimated by N-gram model, and other parts are the same as the RNN-MCDM scheme. In the existing scheme, the DM model in each domain is designed according to (Young et al., 2013), using the dialogue corpus in its own domain. Namely, for a given domain, the existing scheme does not utilize dialogue corpus in other domains.

The dialogue corpus for experiments covers five vertical domains, including hotel reservation (171 dialogues), shopping guidance (71 dialogues), banking service (64 dialogues), restaurant catering (46 dialogues), and taxi service (33 dialogues). Several slots are defined for each vertical domain. For example, in hotel reservation, the slots requested from machine to users include “room type”, “room quantity”, “client quantity”, “checkin time”, “checkout time”, “breakfast demand”(yes or no), “breakfast type”, “client name” and “client phone”, while the slots requested from users to machine include “hotel address = No.95 East St.”, “room type set = single room, double room, and deluxe room”, “single room price = $80”, “double room price = $100”, “deluxe room price = $150”, “breakfast type set = Chinese breakfast, American breakfast, and Japanese breakfast”, “Chinese breakfast price = $12”, “American breakfast price = $15” and “Japanese breakfast price = $10”. Besides, we also define 8 slots for shopping guidance, 9 slots for banking service, 6 slots for restaurant catering and 4 slots for taxi service. The details of these slots are not described due to the limitation of pages. Besides, K_d is set to be 10.

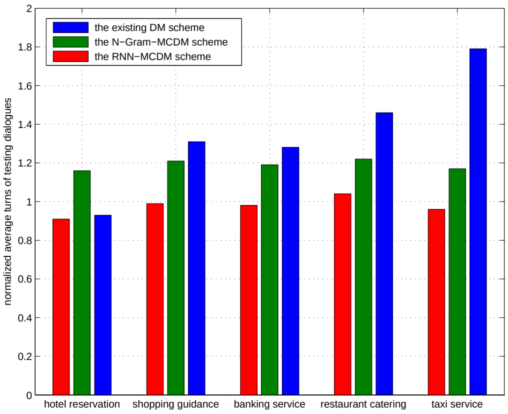


Figure 6: comparison of normalized average turn in different domains

The dialogues in corpus are divided into two parts: 70% corpus for training the DM model and 30% corpus for user simulation to test the systems. The simulated users are built via Bayesian Networks according to (Pietquin, 2005). There are two performance indices for SDS evaluation: average turn and success rate. Average turn is defined as the average dialogue turn cost for task completion. Generally, in different vertical domains, the dialogue turns are directly proportional to the quantities of slots. Thus, we define the normalized average turn as the ratio of average dialogue turn to slot number. In addition, success rate is defined as the ratio of the dialogues that complete the task in the threshold turns to all the dialogues. Here, we define the threshold as double of slot number.

Fig. 6 illustrates the normalized average turn in the RNN-MCDM scheme, the N-Gram-MCDM scheme and the existing DM scheme. The ver-

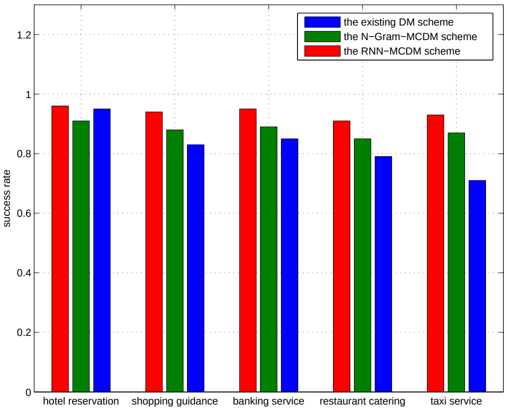


Figure 7: comparison of success rate in different domains

tical domains for comparison are hotel reservation, shopping guidance, banking service, restaurant catering and taxi service. From this picture, we have the following conclusions. For the existing DM scheme, in the vertical domain with more dialogue corpus it has lower normalized average turn, while it has higher normalized average turn in the vertical domain with less dialogue corpus. The reasons are as follows. The existing scheme only uses the dialogue corpus in one domain. Its trained DM model might not contain the abundant states if the size of dialogue corpus is small. Thus, when being in a unknown state, it could not calculate the optimal action, which might be detrimental to the efficiency of slot filling. However, the MCDM schemes have stable and better performance of normalized average turn, which should be ascribed to the fact that the proposed schemes train the general DM model based on the dialogue corpus in all the domains, and leaning general dialogue knowledge to guide the dialogue evolution. In addition, the N-Gram-MCDM scheme has lower normalized average turn than the existing scheme. Especially in the vertical domain with less dialogue corpus, performance improvement is more obvious. The reason is that the N-Gram-MCDM scheme could learn the general dialogue knowledge from all the domains, especially in the domain with less corpus it could use a part of other domain knowledge to train its optimal dialogue policy. Furthermore, the RNN-MCDM scheme has the lowest normalized average turn in every domain, because the RNN-MCDM scheme use RNN to learning history vector for DA prediction that takes the whole dialogue history into account. Namely, the RNN-MCDM scheme utilizes dialogue historical information more efficiently than the N-Gram-MCDM scheme, and RNN-based prediction model is smoother than N-Gram-based prediction model.

Fig. 7 compares the success rate among the RNN-MCDM scheme, the N-Gram-MCDM scheme and the existing DM scheme. From this picture, we can find out that the RNN-MCDM scheme has the highest success rate, and the success rate in the existing scheme is lower than the N-Gram-MCDM scheme, the gap become huge in the vertical domain with less dialogue corpus, which should be ascribed to the same reasons in Fig. 6.

7 Conclusion

In this paper, we proposed the DM scheme based on Multi-domain Corpus. In this scheme, DA is divided into DA type and slot parameter, where the former one is domain-independent and the latter one is domain-specific. We used RNN to estimate the probability distributions of next user DA type and current machine DA type with dialogue corpus in all the domains, and established a POMDP-based current machine DA type decision model that is applicable to all the vertical domains. Additionally, we designed a slot parameter selection scheme to generate a complete machine DA according to the features of vertical domain, which yields the MCDM scheme. Finally, extensive experimental results indicated that the proposed DM scheme is superior to the existing scheme.

Acknowledgments

This work is supported by the National Program on Key Basic Research Project (973 Program), basic theories and methods of Chinese Language Processing and Deep Computing in Internet environment, multi-lingual Automatic Speech Recognition for complex environments. (No. 2013CB329302)

Appendix A

In this section, we deduce the belief state updating process in equation (8). The belief state in the $(t+1)$-th turn could be represented as

$$b_{t+1} = \Pr\left\{s_{t+1} \left| o_{t+1}, b_t, a_t, \cdots, b_{t-\tau+1}, a_{t-\tau+1}\right.\right\} \tag{15}$$

If we denote $b_t, a_t, \cdots, b_{t-\tau+1}, a_{t-\tau+1}$ as φ, b_{t+1} could be written as

$$b_{t+1} = \frac{\Pr\{s_{t+1}, o_{t+1}, \varphi\}}{\Pr\{o_{t+1}, \varphi\}}$$

$$= \frac{\Pr\{o_{t+1}\,|s_{t+1}, \varphi\}\Pr\{s_{t+1}\,|\varphi\}\Pr\{\varphi\}}{\Pr\{o_{t+1}\,|\varphi\}\Pr\{\varphi\}}$$

$$= \frac{\Pr\{o_{t+1}\,|s_{t+1}, \varphi\}\Pr\{s_{t+1}\,|\varphi\}}{\Pr\{o_{t+1}\,|\varphi\}} \qquad (16)$$

According to (Thomson, 2009), $\Pr\{o_{t+1}\,|s_{t+1}, \varphi\} = \Pr\{o_{t+1}\,|s_{t+1}\}$. In addition, $\Pr\{s_{t+1}\,|\varphi\}$ could be shown as

$$\Pr\{s_{t+1}\,|\varphi\} = \sum_{s_t}\cdots\sum_{s_{t-\tau+1}}\Pr\{s_{t+1}\,|s_t, a_t, \cdots, s_{t-\tau+1}, a_{t-\tau+1}\}\Pr\{s_t, \cdots, s_{t-\tau+1}\,|\varphi\} \qquad (17)$$

where

$$\Pr\{s_t, \cdots, s_{t-\tau+1}\,|\varphi\} = \prod_{i=t-\tau+1}^{t} b_i \qquad (18)$$

Besides, $\Pr\{o_{t+1}\,|\varphi\}$ could be shown as

$$\Pr\{o_{t+1}\,|\varphi\} = \sum_{s_{t+1}}\Pr\{o_{t+1}\,|s_{t+1}\}\Pr\{s_{t+1}\,|\varphi\} \qquad (19)$$

Accordingly,

$$b_{t+1} = \frac{\Pr\{o_{t+1}\,|s_{t+1}\}\Pr\{s_{t+1}\,|\varphi\}}{\sum\limits_{s_{t+1}}\Pr\{o_{t+1}\,|s_{t+1}\}\Pr\{s_{t+1}\,|\varphi\}}$$

$$= \kappa\cdot\Pr\{o_{t+1}\,|s_{t+1}\}\sum_{s_t}\cdots\sum_{s_{t-\tau+1}}\Pr\{s_{t+1}\,|s_t, a_t, \cdots, s_{t-\tau+1}, a_{t-\tau+1}\}\prod_{i=t-\tau+1}^{t} b_i \qquad (20)$$

where

$$\kappa = \frac{1}{\sum\limits_{s_{t+1}}\Pr\{o_{t+1}\,|s_{t+1}\}\Pr\{s_{t+1}\,|\varphi\}} \qquad (21)$$

is a normalization factor.

References

S. Bhatnagar, R. S. Sutton, M. Ghavamzadeh, M. Lee. 2009 Natural actorCcritic algorithms. *Automatica*, 45(11), 2471-2482, 2009.

R. A. Clark, K. Richmond, S. King. 2004 Festival 2-Build Your Own General Purpose Unit Selection Speech Synthesiser. *In Fifth ISCA Workshop on Speech Synthesis*, 2004.

L. Daubigney, M. Geist, S. Chandramohan, O. Pietquin. 2012. A Comprehensive Reinforcement Learning Framework for Dialogue Management Optimization. *IEEE Journal of Selected Topics in Signal Processing*, Vol. 6 No.8, pp: 891-902, 2012.

Annemiek van Drunen. 2012. Dialogue management and automation in interaction with unmanned systems. *Proceedings of the 2nd International Conference on Application and Theory of Automation in Command and Control System*, May 2012.

Emmanuel Ferreira, Fabrice Lefvre. 2013. Social signal and user adaptation in reinforcement learning-based dialogue management. *MLIS '13*, August 2013.

Kallirroi Georgila, Claire Nelson, David Traum. 2014. Single-Agent vs. Multi-Agent Techniques for Concurrent Reinforcement Learning of Negotiation Dialogue Policies. *The 52nd Annual Meeting of the Association for Computational Linguistics*, Jun, 2014.

Yulan Hea, Steve Young. 2006. Spoken language understanding using the Hidden Vector State Model. *Speech Communication*, Volume 48, Issues 3C4, Pages 262C275, 2006

M. L. Littman. 2009 A tutorial on partially observable Markov decision processes. *Journal of Mathematical Psychology*, 53(3), 119-125, 2009.

F. Mairesse, M. Gasic, F. Jurcicek, S. Keizer, B. Thomson, K. Yu, S. Young. 2009. Spoken language understanding from unaligned data using discriminative classificati.on models. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009.

F. Mairesse, M. Walker. 2007 PERSONAGE: Personality generation for dialogue. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2007.

Michael F. Mctear. 2002. Spoken dialogue technology: enabling the conversational user interface. *ACM computing Survey*, volume 34, No. 1, pages: 90-169, 2002.

T. Mikolov. 2012 Statistical language models based on neural networks. *Presentation at Google, Mountain View*, 2012.

T. Mikolov, M. Karafit, L. Burget, J. Cernocky, S. Khudanpur. 2010 Recurrent neural network based language model. *11th Annual Conference of the International Speech Communication Association*, Sep, 2010.

Daniele Mori, Riccardo Berta, Alessandro De Gloria, Valentina Fiore, Lauto Magnani. 2013. An easy to author dialogue management system for serious games. *Journal on Computing and Cultural Heritage*, Vol.6 no.2 May 2013.

M. Nagata, T. Morimoto. 1994 First steps towards statistical modeling of dialogue to predict the speech act type of the next utterance. *Speech Communication*, 15(3), 193-203.

Reithinger Norbert, Elisabeth Maier. 1995 Utilizing statistical dialogue act processing in verbmobil. *Proceedings of ACL*, Jun, 1995.

J. Peters, S. Schaal. 2008 Natural actor-critic. *Neurocomputinge*, 71(7), 1180-1190, 2008.

O. Pietquin and T. Dutoit. 2005 A probabilistic framework for dialog simulation and optimal strategy learning. *IEEE Transactions on Speech and Audio Processing*, Special Issue on Data Mining of Speech, Audio and Dialog, 2005.

Olivier Pietquin, Matthieu Geist, Senthilkumar Chandramohan, Herv Frezza-Buet. 2011. Sample-efficient batch reinforcement learning for dialogue management optimization. *ACM Transactions on Speech and Language Processing*, Vol. 7 No. 3, May 2011.

R. M. Smullyan. 1995 First-order logic. *Courier Corporation*, 1995

A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky. 2000 Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3), 339-373.

Hao Tang, S. Watanabe, T.K. Marks, J.R. Hershey. 2014. Log-linear dialog manager. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp: 4092 C 4096, May 2014.

B. Thomson. 2009 Statistical methods for spoken dialogue management. *Ph.D. dissertation, Cambridge*, 2009.

B. Thomson and S. Young. 2010. Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems. *Comput. Speech Lang*, vol. 24, no. 4, pp. 562C588, 2010.

J. Turian, L. Ratinov, Y. Bengio. 2010 Word representations: a simple and general method for semi-supervised learning. *In Proceedings of the 48th annual meeting of the association for computational linguistics*, Jul, 2010.

O. Vinyals, S.V. Ravuri, D. Povey. 2012. Revisiting Recurrent Neural Networks for robust ASR. *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012.

Willie Walker, Paul Lamere, Philip Kwok. 2004. Sphinx-4: a flexible open source framework for speech recognition. *Technical Report, Sun Microsystems*

S. Young, M. Gasic, B. Thomson, J. D. Williams. (2013). 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5), pages: 1160-1179, 2013.

Xiaobu Yuan, Guoying Liu. 2012. A task ontology model for domain independent dialogue management. *IEEE International Conference on Virtual Environments Human-Computer Interfaces and Measurement Systems*, 2012.

H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. W. Black, K. Tokuda. 2007 The HMM-based speech synthesis system version 2.0. *In Proc. 6th ISCA Workshop on Speech Synthesis*, Aug, 2007.