Master Thesis

Applying Transfer Learning to the Implementation of an Interactive FAQ Search System

Supervisor Associate Professor Shigeo Matsubara

Department of Social Informatics
Graduate School of Informatics
Kyoto University

Shingo KITAMOTO

February 8, 2011

Applying Transfer Learning to the Implementation of an Interactive FAQ Search System

Shingo KITAMOTO

Abstract

In information retrieval field, keyword retrieval is general. But when a user inputs inappropriate keyword in keyword retrieval, he/she cannot find the contents which the user wants. There are various contents today, so finding an appropriate keyword becomes difficult. So interactive search, through which users dialog with a system to retrieve the contents, is needed

In keyword retrieval on FAQ search systems, users who are not familiar with the domain or products cannot find appropriate keywords. So the users dialog with call center operatives to search FAQ. But it is impossible for call center operatives to work all day and the educational cost for call center operatives is high. Based on the situation above, this research focused following problem

dialog control in an interactive FAQ search system In FAQ search, FAQs are attached with attribute values and the FAQ search is done in order to get attribute values. The system can get a FAQ as the result of FAQ search through asking all attribute values to the user. But when the number of attributes gets larger, it is difficult for a user to answer all attribute values. One method to reduce the amount of questions is to use dialog scenario. But the cost of creating dialog scenario gets higher if the number of attributes or the number of FAQs gets larger.

Addition of new FAQs to an interactive FAQ search system Now we think about an interactive FAQ search system about some products. When new version of the products is released, new FAQs corresponding to the new version have to be added to an interactive FAQ search system. If dialog scenario for controlling dialog of an interactive FAQ search system is used, it's necessary to create dialog scenario once again when new FAQs are added. On the other hand, when machine learning for dialog control is used, dialog log between users and call center operatives becomes training data. But dialog log about new FAQs between users and call center operatives does not exist. It is possible to create dialog log through dialog about new FAQs between users and call center operatives. But the cost of creating dialog log is problematic.

In this research, we used the model called Partially Observable Markov Decision Process (POMDP) to solve the first problem. POMDP gives the system reward for the system action and learn the policy which maximizes the total expected reward. It got possible to achieve effective dialog control to set up the system reward for the system action in an interactive FAQ search.

We used transfer learning to solve the second problem. Transfer learning is a method to transfer knowledge which is obtained from some domain to another domain. Training data about new FAQs does not exist. So, an interactive FAQ search after adding new FAQs controls dialog through using transfer learning to transfer the knowledge of the policy which is learned before. To use transfer learning makes it possible to implement an interactive FAQ Search at a low cost.

Contributions of this research are as follows:

Implementation of dialog control in an interactive FAQ search system Dialog control in an interactive FAQ search system was achieved by using POMDP. In dialog control using POMDP, POMDP gives the system reward for the system action and learns the policy which maximizes the total expected reward. We achieved the dialog control which reduces the number of asking an attribute value to the user through giving positive reward when showing a FAQ which the user needs and giving negative reward when asking an attribute value to the user. POMDP makes it possible to implement dialog control in an interactive FAQ search system at a low cost because POMDP uses the existing dialog log between users and call center operatives.

Reduction of implementation cost of an interactive FAQ search system after adding new FAQs We proposed transfer learning after new FAQs were added to an interactive FAQ search system. The system's action is obtained by mapping the optimal action which is learned from POMDP before. It becomes possible not only to use the knowledge of dialog control before but also to show new FAQs. Furthermore, there is no need to create dialog log about new FAQs, and it becomes possible to implement an interactive FAQ search system after adding new FAQs at low cost.

対話型FAQ検索システムの構築における転移学習の適用

北本 進悟

内容梗概

コンテンツ検索においてはキーワード検索が一般的である．しかし，キーワード検索では適切なキーワードを入力しなければ求めているコンテンツを発見できない．多種多様なコンテンツが存在する現在では適切なキーワードを見つけることは困難である．そこでコンテンツの検索手法として，ユーザがシステムと対話を行い自身の求めているコンテンツを検索する対話型検索が必要となっている．

Frequently Asked Questions(FAQ) におけるキーワード検索ではユーザがその製品やドメインについて詳しくない場合，適切なキーワードを見つけることが難しい．そこで従来ではユーザはコールセンターのオペレータと対話することでFAQ検索を行ってきた．しかし人が対話を行う場合，24時間対応が難しい，教育コストがかかるという問題がある．そこでコールセンターのオペレータをシステム化した対話型FAQ検索システムが望まれている．そのような背景を踏まえ本研究では以下の課題に取り組んだ．

対話型FAQ検索システムの対話制御　FAQ検索ではFAQに属性が与えられており，その属性の値を特定していくことでFAQの検索を行う．全ての属性についてその値をユーザに質問することでFAQの検索は可能であるが，属性の数が増えてくると全ての属性の値をユーザが回答するのは困難になる．1つの解決策として人手で対話シナリオを与えるという方法がある．しかし対話シナリオを与えた場合，属性やFAQの数の増加に伴い対話シナリオの数も増加し対話シナリオ作成のコストが大きくなってしまう．

対話型FAQ検索システムへの新たなFAQの追加　ある製品についての対話型FAQ検索システムを考える．新たなバージョンのリリースがあった場合，対話型FAQ検索システムは新たなバージョンに対する新たなFAQにも対応する必要がある．対話型FAQ検索システムの対話制御において対話シナリオを用いる場合，新たなFAQが追加された際に再度対話シナリオを与える必要がある．一方，対話制御に機械学習を用いた場合ユーザとコールセンターのオペレータとの対話ログが学習データとなる．しかし新たなFAQに関するユーザとコールセンターのオペレータの対話ログは存在しない．そのため学習データ作成のコ

ストの問題がある．

本研究では第1の課題を解決するためにPartially Observable Markov Decision Process(POMDP)と呼ばれるモデルを用いた．POMDPではシステムの行動に報酬を与えることで一連の行動の報酬の期待値を最大化するような方策を学習する．そこで対話型FAQ検索システムの行動に報酬を設定することで効率の良い対話制御を行うことが可能となる．

第2の課題を解決するために転移学習を用いた．転移学習とはあるドメインで得られた知識を別ドメインに転移させる手法である．本研究では新たなFAQが追加される前のPOMDPの知識を用いることで，新たなFAQが追加された後の対話制御を実現した．転移学習を用いることで低コストでの対話型FAQ検索システムの構築が可能となる．

本研究の貢献は以下の2つである．

対話型FAQ検索システムの対話制御の実現　POMDPを用いて対話型FAQ検索システムの対話制御を実現した．POMDPを用いた対話制御では学習においてシステムの行動に報酬を与えることで一連のシステムの行動の期待報酬を最大化する方策を学習させる．ユーザへの属性の質問に負の報酬，ユーザが求めるFAQを提示した際には正の報酬を与えることで，ユーザへの質問数が少なくなるような対話制御を実現した．またPOMDPの学習ではすでに存在するユーザとコールセンターのオペレータとの対話ログを用いるため低コストで対話制御を構築可能となる．

新たなFAQが追加された際の対話型FAQ検索システム構築のコスト削減　対話型FAQ検索システムに新たなFAQが追加された場合に転移学習を用いることを提案した．転移学習を用いた対話制御では，システムの行動はFAQを追加する前のPOMDPにおける最適行動をマッピングしたものとなる．マッピングを行うことで，FAQが追加される前の対話制御の知識を用いつつ新たなFAQも提示可能になった．また，新たなFAQに関する学習データを作成する必要がなくなり，新たなFAQが追加された場合でも低コストな対話型FAQ検索システムの構築が期待できる．

Applying Transfer Learning to the Implementation of an Interactive FAQ Search System

Contents

Chapter 1 Introduction

Recently, various contents exist by development of the Internet. When a user searches for contents, keyword retrieval has been used in the past. However, there are problems such as difficulty to find and input appropriate keywords due to variety of contents. So, a laddering search service system called LadaSearch [1] has been developed. The user searches for the contents through dialog with a LadaSearch system. A system asks gradually deep questions to a user and he/she can notice the true needs which he/she had not noticed.

When using keyword retrieval in a Frequently Asked Questions (FAQ) search system , users who are not familiar with the domain cannot find appropriate keywords. Users who cannot use keyword retrieval had found the FAQ by asking an operator in a call center. However, there are problems when operators in a call center search FAQs. First, operators in a call center cannot work all day. Second, educating operators in a call center costs high. So, development of an interactive FAQ search system has been desired.

FAQs have attributes such as version or device, and each FAQs are attached attribute values. FAQ search is performed by getting attribute values of the FAQ which the user wants. The system's action of an interactive FAQ search system is separated two types. First type is asking an attribute value to the user. Second type is showing FAQ as a search result. The problem to implement an interactive FAQ search system is which action the system should take in a dialog history with the user. So this research focuses on dialog control in an interactive FAQ search system.

Dialog scenario is one method to achieve dialog control. Dialog scenario is the set of rules about the system's action. However, experts are needed to create high-quality dialog scenario. As the number of FAQs and the number of attributes become larger, the number of rules in dialog scenario becomes larger, and creating dialog scenario takes a lot of times. So, implementing dialog control using dialog scenario costs a lot.

Thus, this research had implemented dialog control using SDS-POMDP, which is one of Partially Observable Markov Decision Process (POMDP) specialized in a spoken dialog system. POMDP is often used in an interactive system. LadaSearch can be regarded as an interactive system because LadaSearch finds contents through dialog with the user.

POMDP is one of the machine learning, and POMDP learns a policy which maximizes an expected total reward in partially observable environment. In other words, the most optimal action in some state can be obtained by calculation. Training data of SDS-POMDP in an interactive FAQ search system is dialog log between experts such as operators in a call center and users. Using existing dialog log as training data reduces cost of an implementation of dialog control.

Now, we consider an interactive FAQ search system in some product. When a new version of the product is released, new FAQs about the version have to be added to an existing interactive FAQ search system. But dialog log about new FAQs between users and operators in a call center does not exist. So, it is impossible to learn a policy of SDS-POMDP.

It is possible to create training data through dialog about new FAQs between users and operators in a call center. But creating training data takes a lot of time and costs a lot. So we apply transfer learning to dialog control implementation when new FAQs are added. Transfer learning makes it possible to implement dialog control without creating training data, and reduce cost to create training data.

Transfer learning is an approach to solve a problem through transferring the knowledge of similar problems. The problem which we would like to solve is SDS-POMDP learning after adding new FAQs, and similar problem is SDS-POMDP learning before adding new FAQs. Transfer learning is studied in reinforcement learning domain. However, transfer learning in POMDP is not studied. Thus, this paper proposed the method of transfer learning in POMDP through an interactive FAQ search system.

This paper is organized as follows: In Chapter 2, laddering search is introduced. We also introduce the problem of applying laddering search to a FAQ search system. In Chapter 3, we introduce related works like POMDP and transfer learning. In Chapter 4, we introduce an interactive FAQ search system which we assume in this research. We describe data structure of FAQ and the flow of an interactive FAQ search system. In Chapter 5, we apply SDS-POMDP to an interactive FAQ search system. In Chapter 6, we propose the method of transfer learning to reduce cost of implementation when new FAQs are added. In Chapter 7, we discuss the problem of proposed methods. Chapter 8 represents conclusion.

Chapter 2 Laddering Search

2.1 Overview of Laddering Search

Laddering search is a search method which makes the user's need clear by dialog and finds services or contents which the user wants. One of the systems using a laddering search is LadaSearch. A user dialogs with a system in a laddering search system. Users reconsider their needs, and can get the contents which they cannot find by themselves.

Figure 1 shows the architecture of a laddering search system. A laddering search system should consider following four elements.

First element is understands the user's utterance. A laddering search system gets the information to search contents through user's utterance. Second element is responding to the user. Laddering search system makes the user feel relieved and comfortable by giving appropriate responses to the user. Third element is controlling the dialog. A laddering search system chooses what to ask or what to do next. Fourth element is making recommendation. Recommendation means the result of the matching to the user. A laddering search system chooses what contents to show to the user through the information from

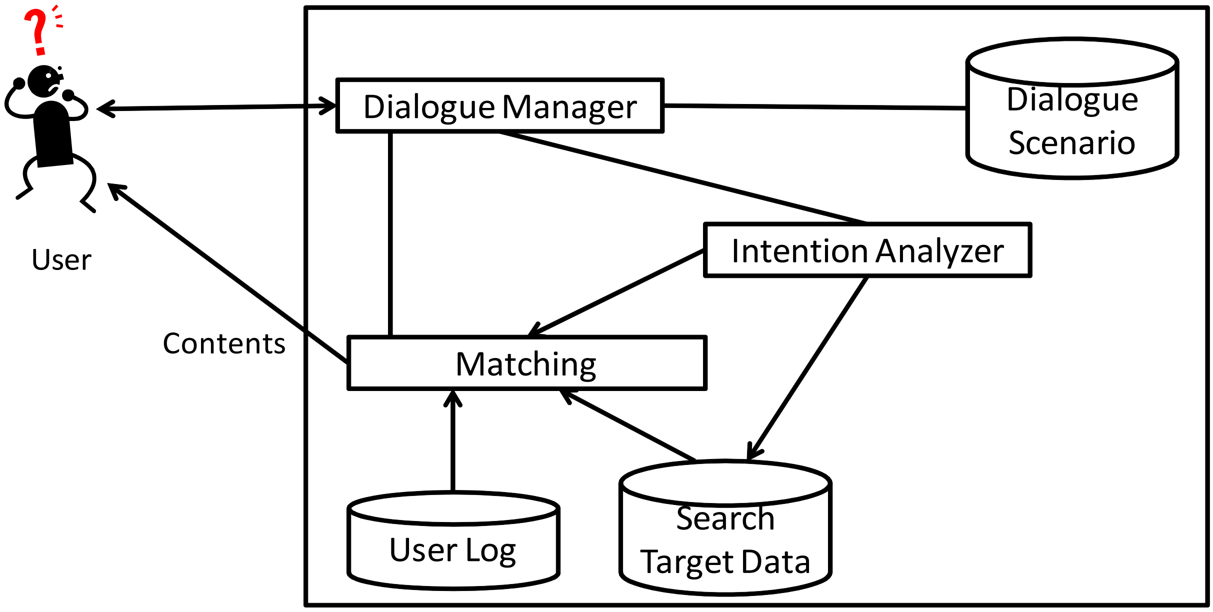

Figure 1: LadaSearch

dialog.

First, intention analyzer performs understanding the user's utterance. Intention analyzer extracts the user's intention using natural language processing. Then, the values extracted from the user's utterance are stored in user data.

Second, dialog manager performs responding to the user and controlling the dialog. Dialog manager controls dialog using the values from intention analyzer based on dialog scenario. Dialog scenario is the set of rules about the system's action.

Third, matching controls the dialog and makes recommendations. Matching shows the result of contents which match the user's need extracted through dialog.

2.2 FAQ Search using Laddering Search

If using keyword retrieval in FAQ search, users who are not familiar with the domain cannot find appropriate keywords. Such users have asked operators in call center. However, there are some problems such as educational cost, the quality difference among operators and limitation of working time if operators in call center search FAQ. Oki Electric Industry Co., Ltd has been developing a FAQ search system using laddering search.

The flow of FAQ search is as follows. First, FAQ attributes are decided. Next, each FAQs are attached attribute values. If a user uses a FAQ search system, a system extracts the attribute values of the FAQ which the user wants through dialog with the user. Then the system shows the FAQ if the number of candidates of FAQ which the user wants becomes small.

Figure 2 shows an example of dialog. First, the system asks "What is your device" to the user. If the user's answer is "I have trouble of IP phone", the system extracts that the value of device is IP phone using intention analyzer. Then, the system asks new question such as "Your question is about IP phone, right? Then choose below services". To continue the dialog, the system extracts values of FAQs which the user wants. If the system extracts adequate information, the system shows a FAQ as the result of FAQ search.

The problem of FAQ search using laddering search is which action the system

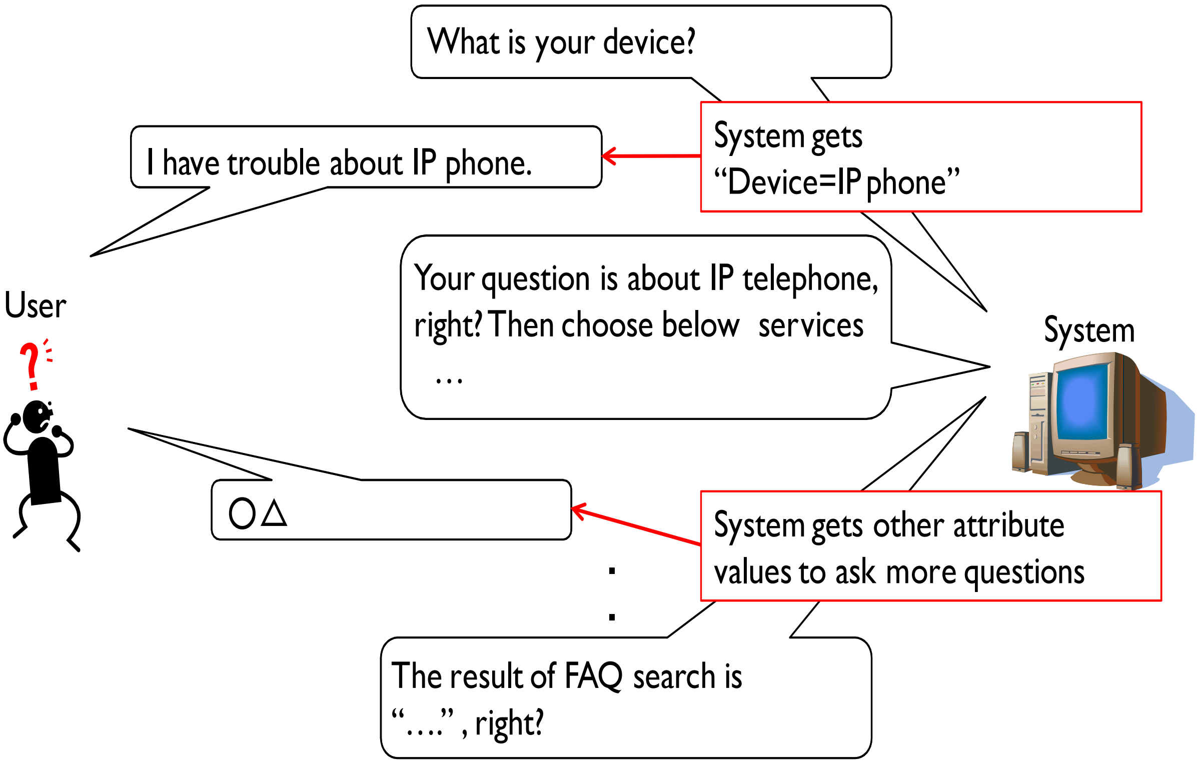
What is your device?
I have trouble about IP phone.
System gets "Device=IP phone"
User
Your question is about IP telephone, right? Then choose below services …
System
○△
System gets other attribute values to ask more questions
The result of FAQ search is "…." , right?

Figure 2: Example of dialog

should take by using dialog history. In figure 2, the user answers ”I have trouble of IP phone”, then the system asks the value of service version. But if the user answers ”I have trouble of LAN cable”, asking another value may be more important. Which attribute the system asks affects the number of the questions to a user.

This research focuses on dialog management in laddering search which decides the system's action in some time step. Intention analyzer which understands a user's answer is also a problem, but this research does not deal in intention analyzer. Present FAQ search system using laddering search asks an attribute value which maximizes expected number of FAQ candidate after getting the attribute value. This is because the system avoids getting an ineffective attribute value which does not affect FAQ search. But present method tends to increase the number of asking questions to a user. The system should ask an attribute value which decreases the total number of asking. So, this research focuses on the effective dialog control using dialog history.

Chapter 3 Related Works

3.1 Interactive System

Users search contents through dialog with a system in laddering search system. So, laddering search can be regarded as an interactive system.

Interactive system is the system which controls dialog automatically. The two problems to implement an interactive system is as follows[2]. First problem is how to choose an appropriate system's action to the user's answer. Second problem is that there are recognition error and understanding error.

A method to solve the first problem is attaching the system's reward as the result of each actions. The system learns a policy which maximizes the expected total reward by using reinforcement learning. Policy is mapping from states to actions. Learning policy means how to select an appropriate system's action to the user's answer. A method to solve the second problem is introducing uncertainty.

3.2 POMDP

The model which solves the problems in an interactive system is Partially Observable Markov Decision Process (POMDP)[3][4]. POMDP is one of the models which can use reinforcement learning. In reinforcement learning, agent gets reward from environment depending on current state when agent takes some action. The learning result of reinforcement learning is a policy which maximizes expected total reward. Agent cannot fully observe state in POMDP. In other words, agent observes states partially in POMDP and a policy which maximizes expected total reward. Unobservable states are inferred from observation values. The model where all states are fully observable is Markov Decision Process (MDP)[5].

Figure 3 shows the structure of POMDP. POMDP is defined as $(S, A, O, T, Z, R, \gamma)$. S represents the set of states. A represents the set of actions. O represents the set of observations.

In each time step, the agent lies in some state $s \in S$, and the agent takes some action $a \in A$ and moves from s to s'. Due to the uncertainty, the state

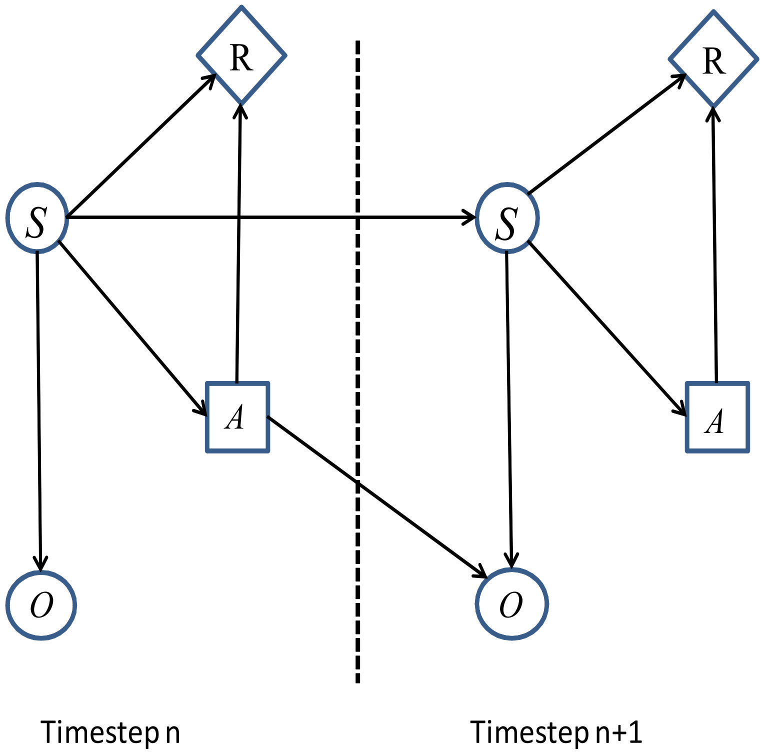

Figure 3: POMDP

s' is modeled as a conditional probability function $T(s, a, s') = p(s'|s, a)$, which gives the probability that the agent lies in s', after taking action a in state s. The agent then makes an observation to gather information on its state. Due to the uncertainty, the observation result $o \in O$ is modeled as a conditional probability function $Z(s, a, o) = p(o|s, a)$.

In each time step, agent receives reward $R(s, a)$, if the agent takes action a in state s. The goal of the agent is to maximize its expected total reward by choosing suitable sequence actions. However, the time step in POMDP is infinite. Thus, POMPD introduces the discount factor $\gamma \in [0, 1)$ so that the total reward is finite. Then expected total reward is given by $E[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)]$, where s_t and a_t represent the agent's state and action at time step t.

The parameters in POMDP are as follows.

- S : the set of states
- A : the set of actions
- O : the set of observations

- T : the set of transition probabilities
  $T(s, a, s') = p(s'|s, a)$
- Z : the set of observation probabilities
  $Z(s, a, o) = p(o|s, a)$
- R : the reward function
  $R(s, a) \to \mathbf{R}$
- γ : the discount factor
  $\gamma \in [0, 1)$

In figure 3, starting nodes affect ending nodes. The two nodes which have no edge between them are independence.

In POMPD, the agent observes observation o not state s. So, current state s is inferred from current observation o and previous action a. POMDP introduces belief b and current state is represented by $b(s)$. The probability $b(s')$, which represents the probability that state in time step $n+1$ is s', is calculated as follow equation using observation o' in time step $n+1$, action a in time step n and belief $b(s)$ in time step n.

$$\begin{aligned} b(s') &= p(s'|o', a, b) \\ &= \frac{p(o'|s', a, b)p(s'|a, b)}{p(o'|a, b)} \\ &= \frac{p(o'|s', a) \sum_{s \in S} p(s'|a, b, s)p(s|a, b)}{p(o'|a, b)} \\ &= \frac{p(o'|s', a) \sum_{s \in S} p(s'|a, s)b(s)}{p(o'|a, b)} \end{aligned} \tag{1}$$

The probability of state s in each time step can be calculated using equation 1.

3.3 SDS-POMDP

POMDP for spoken dialog system is called SDS-POMDP The features of a spoken dialog system are following three points. First, there is a user, who dialogs with the spoken dialog system. Second, users dialog with the spoken dialog system to achieve their goals. For example, users dialog with an interactive FAQ search system to search FAQs. Third, the user's action and the system's action are affected by dialog history. SDS-POMDP is POMDP which covers these features of a spoken dialog system.

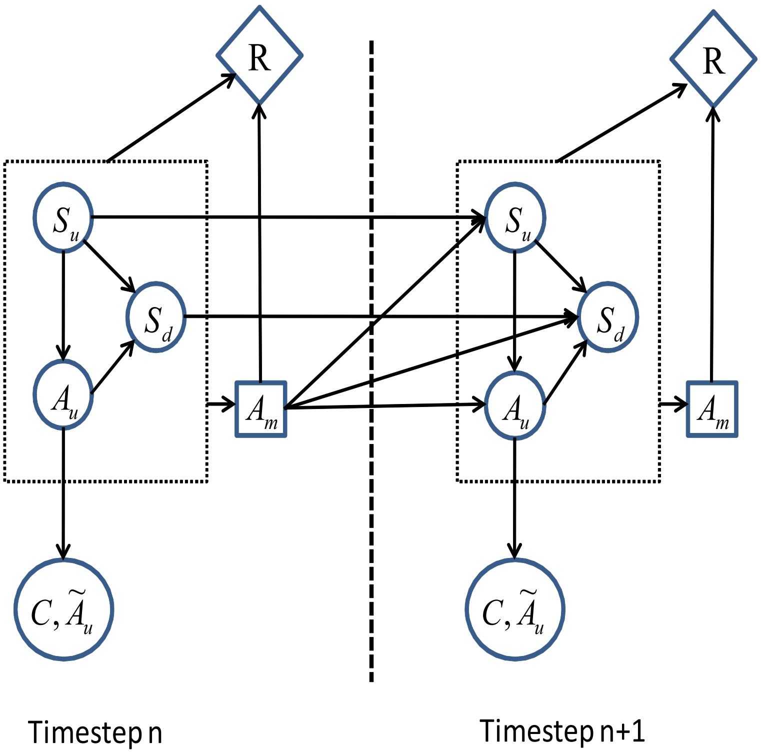

Figure 4: Typical SDS-POMDP structure

In SDS-POMDP, state S is separated into the user's goal S_u, the user's action A_u and the dialog history. Observation O is separated into the recognition result $\tilde{A}_u$ and the confidence score C. The action A in POMDP is represented as the system's action A_m. Figure 4 shows the structure of SDS-POMDP. Figure 4 is one of SDS-POMDP structures, so the structure will change depending on the domain of dialog

The user's goal S_u, the user's action A_u and the dialog history S_d cannot be observed directory. So, SDS-POMDP also introduces belief b. The probability which the agent is in state s is inferred from the observation. The state s' in time step $n+1$ is separated into the user's goal s'_u in time step $n+1$, the user's action a'_u in time step $n+1$ and the dialog history s'_d in time step $n+1$. The probability $b(s')$, which represents the probability that state in time step $n+1$ is s', is represented by equation 2 using the user's goal s_u in time step n, the dialog history s_d in time step n, the user's action a_u in time step n and the system's action a_m in time step n

$$\begin{aligned} p(s'|s,a) &= p(s'_u, s'_d, a'_u | s_u, s_d, a_u, a_m) \\ &= p(s'_u | s_u, s_d, a_u, a_m) \cdot \\ &\quad p(a'_u | s'_u, s_u, s_d, a_u, a_m) \cdot \\ &\quad p(s'_d | a'_u, s'_u, s_u, s_d, a_u, a_m) \end{aligned} \tag{2}$$

Then we assume the SDS-POMDP structure is the structure shown by figure 4. So, the user's goal s'_u in each time step depends on the previous user's goal s_u and the system's action a_m.

$$p(s'_u | s_u, s_d, a_u, a_m) = p(s'_u | s_u, a_m) \tag{3}$$

The user's action a'_u in each time step depends on the current user's goal s_u and the previous system's action a_m.

$$p(a'_u | s'_u, s_u, s_d, a_u, a_m) = p(a'_u | s'_u, a_m) \tag{4}$$

The dialog history s_d in each time step depends on the current user's action a'_u, the current user's goal s'_u, the previous dialog history s_d and previous system's action a_m.

$$p(s'_d | a'_u, s'_u, s_u, s_d, a_u, a_m) = p(s'_d | a'_u, s'_u, s_d, a_m) \tag{5}$$

So, SDS-POMDP transition function is given by equation 6.

$$\begin{aligned} p(s'|a,s) &= p(s'_u | s_u, a_m) \cdot \\ &\quad p(a'_u | s'_u, a_m) \cdot \\ &\quad p(s'_d | a'_u, s'_u, s_d, a_m) \end{aligned} \tag{6}$$

SDS-POMDP observation function is given by equation 7.

$$\begin{aligned} p(o'|s',a) &= p(\tilde{a}'_u, c' | s'_u, s'_d, a'_u, a_m) \\ &= p(\tilde{a}'_u, c' | a'_u) \end{aligned} \tag{7}$$

Table 1 shows the parameter differences between POMDP and SDS-POMDP.

Table 1: Parameter differences between POMDP and SDS-POMDP

	POMDP	SDS-POMD
State	S	(S_u, A_u, S_d)
Observation	0	$(\tilde{A}_u, C)$
Action	A	A_m
Transition function	$p(s'\|s, a)$	$p(s'_u\|s_u, a_m)p(a'_u\|s'_u, a_m)p(s'_d\|a'_u, s'_u, s_d, a_m)$
Observation function	$p(o'\|s', a)$	$p(\tilde{a}'_u, c'\|a'_u)$
Reward	$R(s, a)$	$R(s_u, a_u, s_d, a_m)$
Belief	$b(s)$	$b(s_u, a_u, s_d)$

3.4 SARSOP

When we apply POMDP to real-world application, there are problem of computing time to learn a policy. If the number of the user's goal S_u is 10, the number of the user's action A_u is 10 and the number of the dialog history is 100 in SDS-POMDP, then the dimension of state S is $10 \times 10 \times 100$. Thus, the dimension of state S tends to large and the computing time increases explosively.

SARSOP[6] is one of algorithms to solve the problem of computing time to learn a policy. SARSOP calculates only about the subset of belief points reachable from a given initial point in order to avoid calculating all belief space, and reduce the computing time.

The result of policy learning is the set of α-vectors. Each α-vectors are associated with an action a. The best action in state s can be calculated as follows. First, finding the α-vector which maximizes inner products of α-vectors and b. Then, the best action is the action a which is associated with the α-vector.

3.5 Transfer Learning

Transfer learning is the method to transfer knowledge given by some task to another task. The insight behind transfer learning is that generalization may occur not only within tasks, but also across tasks.

The domain using knowledge is called source domain, and the domain transferred knowledge is called target domain. Many transfer learning methods are studied in MDP where the agent observes state directory. The merits of applying transfer learning to MDP is as follows.

- The initial performance may be improved.
- The final performance may be improved.
- The learning time may be reduced.

Many transfer learning methods are known. One of the methods is task mapping. In task mapping, the parameters of source domain such as the state S or the action A are associated with the parameters of target domain such as the state S' or the action A'. The parameters of target domain are mapped into the parameters of source domain, and a policy is learned in source domain. Then the result of learning is mapped into target domain. Thus, the target domain is learned.

Chapter 4 Interactive FAQ Search System

4.1 Interactive FAQ Search System Overview

This research deals with FAQs attached attributes. Table 2 shows the examples of FAQs about CTStage, which is one of the products of Oki Electric Industry Co., Ltd.

Table 2 represents that the FAQ attributes are "Version" and "Device". The FAQ which is associated with FAQ_1 is "What is the recommended setting of codec when using MKT/IP", the version of FAQ_1 is 4i or 5i, and the device of FAQ_1 is IP phone. The device of FAQ_3 is null, and this means the value of device is not attached. In FAQs about CTStage, the number of FAQs is 259, the number of attributes is 15 and the number of each attribute values is from 3 to 35.

When the system searches FAQ, the system specifies the attribute values. For example, the FAQs are FAQ_1, FAQ_2 and FAQ_3 in table 2. If the attribute values of FAQ which a user wants are "version = 5i "and "device = IP phone", then FAQ_1 is uniquely identified as the search result. Here, the dialog control

Table 2: Example of FAQs

	FAQ	Attribute	
		Version	Device
FAQ_1	What is the recommended setting of codec when using MKT/IP	4i,5i	IP phone
FAQ_2	There is no reply if I ping Audio Codes board. LAN cable is connected to Audio Codes board. What is the reason?	4i	LAN cable
FAQ_3	The administrative privileges of operators in call center can be set in OPC. What are the administrative privileges of operators in call center.	4i,5i	null

becomes problem. Dialog control in an interactive FAQ search system is which attribute the system should ask or which FAQ the system should show in each time step.

First, this paper introduces dialog control about an attribute. We assume the FAQ which a user wants is FAQ_1. If the system first ask version, then the user's answer is "version = 4i" or "version=5i". We assume that the user's answer is "version = 5i", then the candidate FAQs of the search result are FAQ_1 and FAQ_3, and the candidates FAQs are not unique. Thus, the system needs to ask the device to the user. As the result, the system gets "device = IP phone", and the candidate FAQ of the search result becomes only FAQ_1.

In contrast, if the system first ask device to the user, he/she answers "device = IP phone". Then the candidate FAQ becomes only FAQ_1, and the FAQ search finishes. The number of asking attribute becomes 2 if the system first asks version, but the number of asking attribute becomes 1 if the system first asks device. Thus, the order of asking attribute affects the number of asking attribute.

The degree of satisfaction of an interactive system is depends on quality and efficiency[7]. The quality of dialog is the system response delay or inappropriate utterance ratio, and the efficiency of dialog is the number of utterances or dialog time. So, the number of asking attribute should be reduced as many as possible in an interactive FAQ search system. Thus, the order of asking attribute is important problem.

Second, we introduce dialog control about showing FAQ. For example, the attribute value given by user is "version = 4i". The candidate FAQs which satisfy "version = 4i" are FAQ_1, FAQ_2 and FAQ_3. If we can get the probability that users want FAQ_2 when the dialog history is "version = 4i" is 99% from dialog log, then the inaccuracy rate is 1% if the system shows FAQ_2 to users, and showing FAQ_2 may not be problem even though the candidate FAQs which satisfy "version = 4i" are not unique. Thus, the system does not have to uniquely identify FAQ candidate. So, the system should show a FAQ in appropriate time step using dialog control.

The method to implement dialog control is dialog scenario. Dialog scenario is the set of rules of the system's responses. But the experts in the domain are needed in order to create dialog scenario. Moreover, if the number of FAQs or attributes becomes larger, then the number of rules becomes larger and it takes much time to create dialog history. Thus, the implementation of dialog control using dialog scenario costs a lot.

So, this research applies SDS-POMDP, which is POMDP specialized in spoken dialog system to dialog control in an interactive FAQ search system. The optimal system's action in each time step can be calculated by using SDS-POMDP. A policy of SDS-POMDP can be learned by machine learning, so the implementation cost becomes low if training data exists.

Furthermore, the feature of a FAQ search system is adding new FAQs. We now consider about a FAQ search system about some products. If new versions or new products are released, then the new FAQs about new versions or new products have to be added to existing FAQ database. In addition, the FAQs which are not frequently asked before may be added to FAQ database.

The problem of adding new FAQs is the lack of training data. The training data of SDS-POMDP in an interactive FAQ search system is dialog log with users and operators in call center. But the dialog log about new FAQs does not exist. Thus, learning a policy is impossible.

One of the methods to solve the lack of training data is to create training data. Users dialog with operators in call center about new FAQs, and create the training data including new FAQs. But creating training data cost a lot.

So, this research applies transfer learning to implementation of an interactive FAQ search when new FAQs are added. Applying transfer learning makes it possible to learn a policy of SDS-POMDP in an interactive FAQ search system including new FAQs.

4.2 Flow of Interactive FAQ Search System

Figure 5 shows the flow of an interactive FAQ search system. The processes except dialog with users are preprocessing, and the process which the user searches FAQ is dialog with user.

First of all, we describe the preprocessing. First, all FAQs are attached to attribute values. Then, the system learns a policy of SDS-POMDP. The data for leaning a policy is FAQ data and dialog log. The parameters in SDS-POMDP such as the user's goal S_u are defined by using FAQ data. The parameters in SDS-POMDP such as transition function and observation function are defined by using dialog log.

Then, the process is done when new FAQs are added. If new FAQs are added, then the FAQs are attached to attribute values. After attaching attribute values, the FAQ added to FAQ database. Note that the process is only adding FAQs to FAQ database, and leaning a policy of SDS-POMDP in an interactive FAQ search system including new FAQs is not done.

Figure 6 shows the flow of the dialog with a user.

First, we describe the processes when FAQ database does not include new FAQs. The system reads the FAQ data and the result of policy learning. The FAQ data is used in order to show a FAQ and in order to apply transfer learning. The result of policy learning is used in order to decide the system action in each time step.

Then the system initializes parameters in SDS-POMDP such as the user's goal s_u, the dialog history s_d, the user's action a_u. Next, the system calculates an optimal system's action a_m in current state using a policy. The system's action includes asking an attribute value and showing a FAQ. Then, the system takes action a_m, and observes the user's action $\tilde{a}_u$. The current belief is updated by using the parameters such as the system action a_m and $\tilde{a}_u$. Then, the system calculates an optimal action a_m in updated current belief. The system continues the series of above processes. The dialog finishes when the system shows a FAQ which the user wants.

Second, we describe the processes when FAQ database includes new FAQ. The difference between the process before adding new FAQs and the process after adding new FAQs is how to decide the system's action a_m. The system's action a_m before adding new FAQs is the optimal system's action calculated from a policy. But the system's action does not include the action showing a new FAQ. Thus, the system does not show a new FAQ to user if the system's action a_m is the optimal system's action calculated from a policy of SDS-POMDP.

So, the system mapped the optimal system's action calculated from a policy to some action. The system can show a new FAQ through action mapping. The detail of mapping is described in Chapter 6.

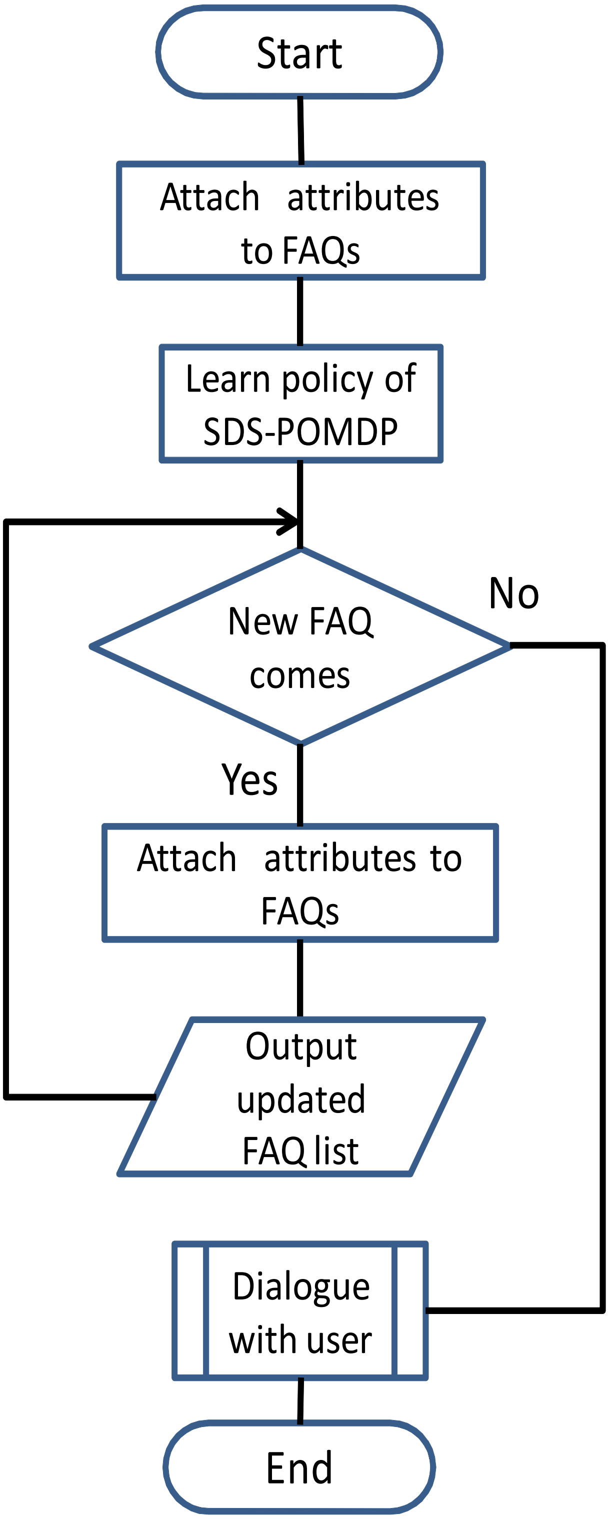
Start
Attach attributes to FAQs
Learn policy of SDS-POMDP
New FAQ comes
No
Yes
Attach attributes to FAQs
Output updated FAQ list
Dialogue with user
End

Figure 5: System flow

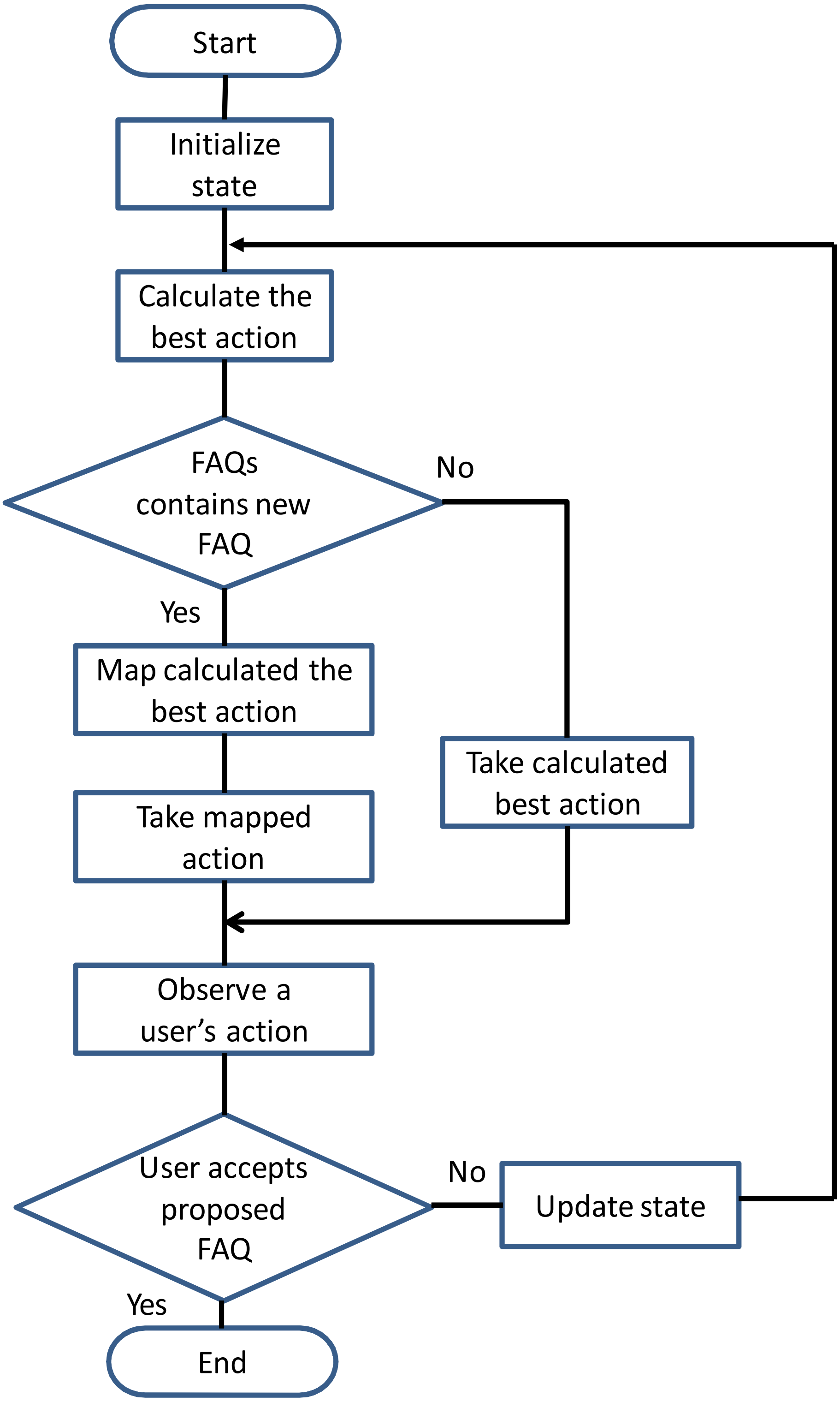

Figure 6: Dialog flow

Chapter 5 Applying SDS-POMDP to Interactive FAQ Search System

5.1 Paramters of SDS-POMDP in an Interactive FAQ Search System

This research uses SDS-POMDP, which is POMDP specialized in spoken dialog system to implement dialog control in an interactive FAQ search system. The system can decide which attribute the system should ask or which FAQ should the system ask by using SDS-POMDP.

We represent the parameters in SDS-POMDP in an interactive FAQ search system. In SDS-POMDP, state S consists of the user's goal S_u, the user's action A_u and the dialog historyS_d.

First, we consider the user's goal S_u. A user dialogs with an interactive FAQ search system in order to search a FAQ. Thus, the user's goal S_u is the FAQ which the user wants. The set of user's goal S_u is $S_u = \{FAQ_1, ..., FAQ_n\}$, where FAQ_i represents the state where FAQ_i is the FAQ which the user wants, and n represents the number of FAQ.

Second, we consider the user's action A_u. There are two types of the user's action. First type is answering the attribute value which the system asks. The example of the attribute value answer is "device = IP phone". Second type is showing a FAQ. If the system shows a FAQ which the user wants, the user accepts the FAQ. If the system shows a FAQ which the user does not wants, the user declines the FAQ. The set of user's action A_u is $A_u = \{v_{11}, ..., v_{n_{att}n_{n_{att}}}, accept, decline\}$, where v_{ij} represents the user's action where the user answers the value of attribute$i$ is v_j, n_{att} represents the number of attribute, n_i represents the number of values of attribute$i$, *accept* represents the action to accept a shown FAQ, *decline* represents the action to decline shown FAQ.

Third, we consider the dialog history S_d. The dialog history S_d is the set of the tuples of attribute values given from the user. Note that the dialog history S_d does not include the information whether the user declines the FAQ or not in order to reduce the dimension of the dialog history S_d. The information whether the user declines the FAQ or not does not directory included in the dialog history but affects the belief of user's goal $b(s_u)$ through changing the structure of SDS-POMDP

Fourth, we describe the system's action A_m. There are two types of the system's action. First type is asking an attribute. For example, the system asks "What is your device" to the user. Second type is showing a FAQ which the user may want. The set of user's action A_m is $A_m = \{ask_1, ..., ask_{n_{att}}, show_1, ..., show_n\}$, where ask_i represents the action to ask attribute$i$ to user, and $show_i$ represents the action to show FAQ_i.

Fifth, we consider the reward R. The user's satisfaction in spoken dialog system depends on the quality and the efficiency[7]. The quality of dialog is the inappropriate utterance ratio, and the efficiency of dialog is the number of utterances or dialog time. Thus, the reward of SDS-POMDP in an interactive FAQ search system depends on the quality and efficiency. From the point of view of the quality, the system gets large positive reward if the system shows the FAQ which the user wants, and the system gets large negative reward if the system shows the FAQ which the user does not wants. From the point of view of the efficiency, the system gets small negative reward if the system asks an attribute value to the user.

Sixth, we consider the observation O. The observation O consists of the observed user's action $\tilde{A}_u$ and the confidence score C in SDS-POMDP. The observed user's action $\tilde{A}_u$ is the same of the user's action A. So, the set of observed user's action $\tilde{A}_u$ is $\tilde{A}_u = \{v_{11}, ..., v_{n_{att}n_{n_{att}}}, accept, decline\}$. The error of natural language processing may occur in an interactive FAQ search system. Furthermore, the error of the user's answer also may occur. FAQ attribute value of behavior in CTStage includes the contents creating and job creating. The user who is not familiar with the CTStage may choose the job creating even though the user should choose contents creating. So, the confidence score C is the error ratio of natural language processing and the user's answer.

The parameters of SDS-POMDP in an interactive FAQ search system are as follows.

- S_u : The set of FAQ which the user wants

$S_u = \{FAQ_1, ..., FAQ_n\}$

- A_u : The attribute value if the system asks an attribute value Acceptance if the system shows the FAQ which the user wants Declinature if the system shows the FAQ which the user does not want
  $A_u = \{v_{11}, ..., v_{n_{att}n_{n_{att}}}, accept, decline\}$
- S_d : The set of tuples of attribute values given from the user
- A_m : Asking an attribute value Showing a FAQ as the result of FAQ search
  $A_m = \{ask_1, ..., ask_{n_{att}}, show_1, ..., show_n\}$
- R : Large positive reward if the system shows the FAQ which the user wants Large negative reward if the system shows the FAQ which the user does not wants Small negative reward if the system asks an attribute value to the user
- $\tilde{A}_u$: The set of the observed user's action
  $\tilde{A}_u = \{v_{11}, ..., v_{n_{att}n_{n_{att}}}, accept, decline\}$
- C : The error ratio of natural language processing and the user's answer.

5.2 Structure of SDS-POMDP in an Interactive FAQ Search System

This section represents the structure of SDS-POMDP in an interactive FAQ search system. The structure of SDS-POMDP affects the transition function and observation function, which is used in state updating. Figure 7 shows the structure of SDS-POMDP in an interactive FAQ search system. Nodes represent parameters, and edges represent the dependency relation.

First, we explain the user's goal S_u. The user's goal s'_u in time step $n+1$ is updated by the dialog history s'_d in time step $n+1$, the user's action a'_u in time step $n+1$ and the system action a_m in time step n. The user's goal S_u in an interactive FAQ search system is the set of FAQs which the user wants. So, $p(FAQ_i)$, which represents the probability that the user wants FAQ_i, depends on the dialog history s_d in the time step. The probability distribution of a FAQ which the user wants is defined by using dialog log.

Here, we consider the situation that the system shows FAQ_i, then the user declines FAQ_i. At this time, $p(FAQ_i)$ should become 0. So the user's goal

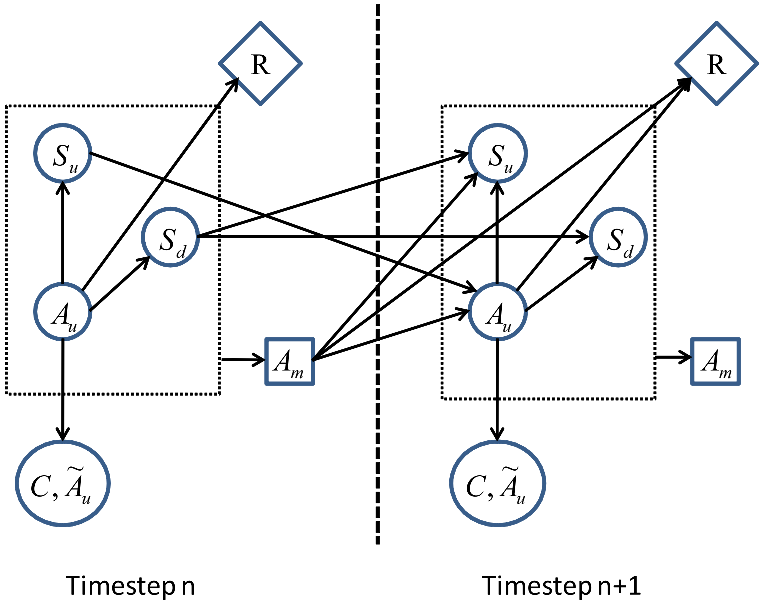

Figure 7: SDS-POMDP in an interactive FAQ search system

s_u depends not only on the dialog history s_d, but also on the user's action a_m and the system's action a_m. Thus, the transition function of the user's goal is $p(s'_u|s'_d, a'_u, a_m)$.

Second, we explain the dialog history S_d. The dialog history s'_d in time step $n+1$ is updated by the dialog history s_d in time step n and the user's action a'_u in time step $n+1$. The dialog history S_d in an interactive FAQ search system is the set of tuples of attribute values given from the user. So, the dialog history s'_d can be get through adding the attribute value getting from the user's action a'_u in time step $n+1$ to the dialog history s_d in time step n. Thus, the transition function of the dialog history is $p(s'_d|a'_u, s_d)$.

Third, we explain the user's action A_u. The user's action a'_u in time step $n+1$ is updated by the user's goal s_u in time step n and the system action a_m in time step n. The user's action A_u in an interactive FAQ search system is the set of actions such as the attribute value if the system asks an attribute value, acceptance if the system shows the FAQ which the user wants, declinature if the system shows the FAQ which the user does not want. So, if the system's action a_m is asking an attribute value, the attribute value of the user's answer is depends on the user's goal s_u, which represents the FAQ which the user wants.

If the system's action a_m is showing a FAQ_i, the user accepts the FAQ when the user's goal s_u is FAQ_i, and the user declines the FAQ when the user's goal s_u is not FAQ_i. Thus, the transition function of the user's action is $p(a'_u|a_u, a_m)$.

Finally, we explain the reward R. The parent nodes of the reward represent the parameters which affect the reward. The reward R in time step $n+1$ depends on the user's action a'_u in time step $n+1$ and the system's action a_m in time step n. The reward R in an interactive FAQ search system is large positive reward if the system shows the FAQ which the user wants, large negative reward if the system shows the FAQ which the user does not wants, and small negative reward if the system asks an attribute value to the user. So, the reward becomes large positive value if the user's action and the system's action are $a'_u = accept, a_m = show_i$, where the system shows FAQ_i and the user accepts FAQ_i. The reward R becomes large negative value if the user's action and the system's action are $a'_u = decline, a_m = show_i$, where the system shows FAQ_i and the user declines FAQ_i. The reward R becomes small negative value if the user's action and the system's action are $a'_u = v_{ij}, a_m = ask_i$, where the system asks attribute i and the user answers v_j. The reward R becomes 0 if the the user's action and the system's action do not satisfy above condition.

5.3 Evaluation

We evaluated the interactive FAQ search system using SDS-POMPD. We describe the experiment data. We used not real FAQ data but virtual FAQ data which we created. The number of FAQs is 10, the number of attributes is 5, and each attributes are binary variables. The attribute values of each FAQs are randomly selected.

The probabilistic distribution of FAQs in some dialog history s_d decides transition function of the user's goal. We decide the probabilistic distribution as follows. First, we decide the probabilistic distribution in initial dialog history. Then, the probabilistic distribution in dialog history s_u is given by the ratio of probabilistic distributions of FAQs which satisfies dialog history s_u.

The reward is 10 if the system shows the FAQ which the user wants, the reward is -10 if the system shows the FAQ which the user does not want, and

Table 3: Evaluation result of SDS-POMDP by changing FAQ probabilities

	Uniform	Biased
The number of asking an attribute	4.33	3.35

the reward is -1 if the system asks attribute value.

We evaluate the affects of two parameters. First parameter is the probabilistic distribution of FAQs. We evaluated the performance difference between when all probabilities that the user wants each FAQs are equal and when the two probabilities that the user wants a FAQ are high. Second parameter is the similarity in FAQs. The similarity between two FAQs is the ratio of the same attribute values between the two FAQs. The similarity in FAQs is the average of the similarities between all combination of two FAQs. We evaluated the performance difference between when the similarity in FAQs is high and when the similarity in FAQs is low.

First, We evaluate two cases. First, the probabilistic distribution is uniform. Second, some FAQs are frequently asked (bias). The two probabilities that the user wants the two FAQs are 0.42, the eight probabilities that the user wants the other eight FAQs are 0.01. The evaluation criterion is the number of asking attribute value to the user. Table 3 shows the result of evaluation.

The number of asking an attribute is 5 in this experiment, so the maximum number of asking an attribute is 5. But the number of attributes is 4.33 when the FAQ distribution is uniform, and is near 5. In contrast, the number of attribute is 3.35 when the FAQ distributions are based. This means that the performance can be improved when the FAQ distributions is based.

We describe why the FAQ distributions affect the number of asking an attribute. Table 4 shows one of the experiment FAQ data. For example, we assume that the user wants FAQ_2.

First, we explain the case when the FAQ distribution is uniform. The reason why the number of asking an attribute is large is the attribute importance which the system should asks to search FAQ are equal. The all probabilities of FAQ which the user wants are equal. So, the total expected reward is the same which

Table 4: Experiment FAQ Data

	att_1	att_2	att_3	att_4	att_5
FAQ_1	1	1	1	1	1
FAQ_2	1	1	1	1	2
FAQ_3	1	1	1	2	1
FAQ_4	1	1	2	1	1
FAQ_5	1	2	1	1	1
FAQ_6	2	1	1	1	1
FAQ_7	2	2	1	1	1
FAQ_8	2	1	2	1	1
FAQ_9	2	1	1	2	1
FAQ_{10}	2	1	1	1	2

attribute the system asks.

Then, we assume the system asks att_1. The user answers that the value of att_1 is 1 because the user wants FAQ_2. The FAQs which satisfy current dialog history are $\{FAQ_1, FAQ_2, FAQ_3, FAQ_4, FAQ_5\}$. The all probabilities of FAQ_1, FAQ_2, FAQ_3, FAQ_4 and FAQ_5 which the user wants are also equal even if the user answer that att_1 is 1. So, the total expected reward is the same which attribute the system asks. Then, the number of asking an attribute gets 5 if the system asks $att_1, att_2, att_3, att_4$, and att_5.

Second, we explain the case when the FAQ distribution is biased. We assume that FAQ_2 and FAQ_9 are frequently asked and other FAQs are not frequently asked. Then, the expected number of asking an attribute gets small if the system asks the attributes which value is different between FAQ_2 and FAQ_9. So, the system asks att_1, att_4 and att_5, which values are different between FAQ_2 and FAQ_9. We assume that the system asks attribute att_1. Then, the user answers that att_1 is 1 because the user wants FAQ_2. The FAQs which satisfy current dialog history are $\{FAQ_1, FAQ_2, FAQ_3, FAQ_4, FAQ_5\}$. The FAQ which is frequently asked is FAQ_2 within $\{FAQ_1, FAQ_2, FAQ_3, FAQ_4, FAQ_5\}$. So, the system asks att_5 which values are different between $\{FAQ_2\}$ and

Table 5: Evaluation result of SDS-POMDP by changing similarity in FAQs

	High similarity	Low similarity
The number of asking an attribute	4.08	3.20

$\{FAQ_1, FAQ_2, FAQ_3, FAQ_4\}$, and the expected number of asking an attribute gets small. Then, the user answers att_5 is 2, and the FAQ which satisfies current dialog history gets only $\{FAQ_2\}$. So, the number of asking an attributes is 2.

Next, we evaluated the affect of similarity in FAQs. The similarity $sim(i, j)$ between FAQ_i and FAQ_j is the ratio of the same attribute values between FAQ_i and FAQ_j. The similarity in FAQs is calculated by $\sum_{i=1}^{n} \sum_{j=1}^{n} sim_{i,j}/n^2$. Table 5 shows the number of asking an attribute when the similarity in FAQs is high (0.624) and when the similarity in FAQs is low (0.504). The two probabilities that the user wants the two FAQs are 0.42, the eight probabilities that the user wants the other eight FAQs are 0.01.

Table 5 shows that the number of asking an attribute is smaller when the similarity in FAQs is low. It's because the number of asking an attribute to make the FAQs which satisfy current dialog history unique tends to be larger when the similarity in FAQs is large. The number of same attribute values is large when the similarity in FAQs is high So, the number of FAQs which satisfy current dialog history tends to be large. In contrast, the number of same attribute values is small when the similarity in FAQs is low. So, the number of FAQs which satisfy current dialog history tends to be small, and the number of asking an attribute to make the FAQs which satisfy current dialog history unique tends to be low.

Chapter 6 Applying Transfer Learning to Interactive FAQ Search System

6.1 Parameter Differences

The new FAQs are added to existing FAQ database after new versions or products are released. So, the adding new FAQs to existing system is needed in order to implement new FAQ search system. But, the training data about new FAQs does not exist. Creating training data is possible for users to dialog about new FAQs with operators in call center. But creating new training data costs a lot. So, we apply transfer learning when new FAQs are added. The policy learning of SDS-POMDP after adding new FAQs becomes possible through using the learning result of SDS-POMDP before adding new FAQs.

Many methods of transfer leaning are studied. But the method of transfer learning in POMDP does not studied. So, we propose the method of transfer learning in POMDP. One of the methods of transfer learning is task mapping. Task mapping is mapping parameters in source domain such as action A or S to parameters in target domain such as action A' and state S'.

We have to consider the differences between target domain and source domain to task mapping transfer learning. So, we describe the parameter differences between target domain and source domain.

The parameters of SDS-POMDP is the user's goal S_u, the user's action A_u, the dialog history S_d, the system's action A_m, the reward R, the observation of the user's action $\tilde{A}_u$ and the confidence score C.

The user's goal S_u in an interactive FAQ search system is the set of FAQs which the user wants. So, the new FAQs are added to the user's goal S_u after adding new FAQs. Thus, the user's goal S_u is different between target domain and source domain.

The user's action A_u in an interactive FAQ search system is the set of actions such as the attribute value if the system asks an attribute value, acceptance if the system shows the FAQ which the user wants, declinature if the system shows the FAQ which the user does not want. This research assumes that the attributes and the attribute values do not change after adding new FAQs. So, the user's action A_u is not different between target domain and source domain. The dialog history S_d in an interactive FAQ search system is the set of tuples of attribute values given from the user. The attributes and attribute values do not change after adding new FAQs, so the dialog history S_d is not different between target domain and source domain.

The system action A_m in an interactive FAQ search system is the set of actions such as asking an attribute value and showing a FAQ as the result of FAQ search. The actions showing new FAQs have to be added to the system's action after adding new FAQs. So, the system's action A_m is different between target domain and source domain.

The reward R in an interactive FAQ search system is large positive reward if the system shows the FAQ which the user wants, large negative reward if the system shows the FAQ which the user does not wants, and small negative reward if the system asks an attribute value to user. The reward can be changed after adding new FAQs, but this research assumes that the reward R does not change after adding new FAQs.

The observation of the user's action $\tilde{A}_u$ in an interactive FAQ search system is the set of the observed user's action. The elements of the observation of the user's action $\tilde{A}_u$ is equal to the elements of the user's action A_u. So, the observation of the user's action $\tilde{A}_u$ is not different between target domain and source domain. Finally, the confidence score C is the error ratio of natural language processing and the user's answer. The confidence score C does not depends on the addition of new FAQs. So, the confidence score C is not different between target domain and source domain.

Table 6 shows the change of SDS-POMDP parameters by adding new FAQs. The parameters which are change after adding new FAQs are the user's goal S_u and the system's action.

6.2 Method of Transfer Learning

We describe the method of the transfer learning in an interactive FAQ search. The elements of $show_{new}$, which represents that the system shows a new FAQ FAQ_new, are added to the system's action A_m after adding new FAQs. Hence,

Table 6: Changes of SDS-POMDP parameters by adding new FAQs

User's goal S_u	change
User's action A_u	no change
Dialog history S_d	no change
System's action A_m	change
Reward R	no change
Observation of the user's action $\tilde{A}_u$	no change
Confidence score C	no change

the system does not show new FAQs to the user if the system decides the system's action a_m by using only SDS-POMDP before adding new FAQs. The system maps the system's action a_m before adding new FAQs to the system's action a'_m after adding new FAQs, and makes it possible to show new FAQs.

We represent the basic idea of transfer learning in an interactive FAQ search system. We assume that the attribute value which the system should ask in some state does not change after adding a few new FAQs to many existing FAQs. So, the system's actions where the system asks attributes do not mapped.

However, an interactive FAQ search system after adding new FAQs has to show new FAQs as the result of FAQ search. So, the system's actions are mapped if the optimal action which is calculated from the policy of SDS-POMDP before adding new FAQs is show a FAQ. The FAQ which the system shows is selected randomly from not only existing FAQs but also new FAQs. The distribution of selected each FAQs is not uniform but depends on $b(s'_u)$, which represents the probabilities where the user wants each FAQs after adding new FAQs. $b(s'_u)$ is mapped from the belief of the user's goal $b(s_u)$ before adding new FAQs.

Figure 8 shows the sequence diagram of dialog using transfer learning. The beliefs of the user's goal, the dialog history and the user's goal are initialized when the user starts dialog with an interactive FAQ search system. Then the system calculates an optimal action by using a policy of SDS-POMDP before adding new FAQs, and takes the optimal action.

Next, the system continues below processes until the user accepts a FAQ which the system shows. First, the system observes the user's action $\tilde{A}_u$. Then the system updates the beliefs of each parameters by using the system's action and the observation of the user's action. The system calculates an optimal action under the updated beliefs. Then, the belief of the user's goal $b(s_u)$ is mapped to the belief $b(s'_u)$ after adding new FAQs. The belief mapping of the user's goal is described in 6.2.1. Then, the system's action is mapped using the belief $b(s'_u)$. The system takes the mapped action. The system continues above processes until the user accepts a FAQ which the system shows.

We consider belief update. The system can show the new FAQs as the result of FAQ search by using the system's action mapping. But, the transition function when the system shows a new FAQ does not exist in SDS-POMDP before adding new FAQ. So, the belief update is impossible. We describe the method to solve belief update problem in 6.2.3.

6.2.1 Mapping Belief of the User's Goal

The user's goal S'_u in SDS-POMDP after adding new FAQs is $\{FAQ_1, ..., FAQ_n, FAQ_{n+1}, ..., FAQ_{n+m}\}$, where FAQ_1,...,FAQ_n represent existing FAQs and FAQ_{n+1},...,FAQ_{n+m} represent new FAQs. The user's goal S_u in SDS-POMDP before adding new FAQs is $\{FAQ_1, ..., FAQ_n\}$.

In POMDP, the agent cannot observe the state directory, so the user's goal is represented by using uncertainty. The belief of the user's goal $b(s_u)$ in an interactive FAQ search system consists of $p(FAQ_i)$, which represents that the probability which the user wants FAQ_i. The belief of the user's goal before adding new FAQs is represented by

$$b(s_u) = (p(FAQ_1), ..., p(FAQ_n))$$

and the belief of the user's goal after adding new FAQs is represented by

$$b(s'_u) = (p(FAQ_1), ..., p(FAQ_n), p(FAQ_{n+1}, ..., p(FAQ_{n+m})$$

The system maps the belief of the user's goal $b(s_u)$ before adding new FAQs to the belief of the user's goal $b(s'_u)$ after adding new FAQs. In mapping, we propose to distribute $p(FAQ_i)$ of existing FAQs to FAQ_1,...,FAQ_{n+m}. The

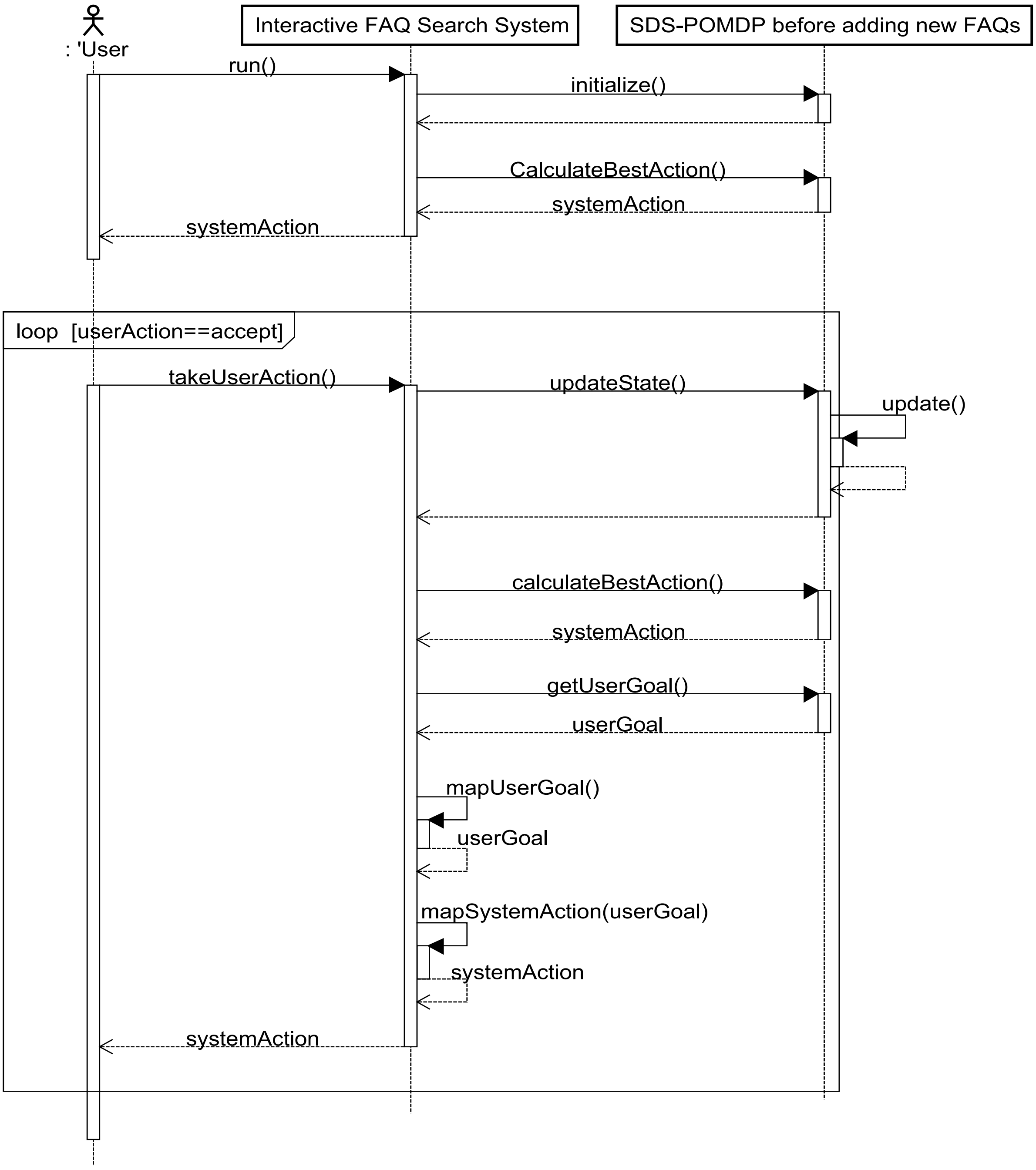

Figure 8: Flow of transfer learning

distribution weights have to be decided in order to to distribute $p(FAQ_i)$.

Here, we use the similarities between two FAQs. We define sim_{ij}, which is the similarity between FAQ_i and FAQ_j, as the ratio of the same attribute number between FAQ_i and FAQ_j. Moreover, the user does not want the FAQs

Algorithm 1 The process of mapping belief of the user's goal

```
$n$ /* the number of existing FAQs */
$m$ /* the number of new FAQs */
$sim_{ij}$ /* the ratio of the same attribute values between $FAQ_i$ and $FAQ_j$ */
$\Gamma \leftarrow \emptyset$
Insert FAQ indexes which satisfy current dialog history into $\Gamma$
$b(s'_u) \leftarrow \emptyset$
for $i = 1$ to $n + m$ do
  if $i \in \Gamma$ then
    $p'(FAQ_i) \leftarrow \sum_{j=1}^{n} \frac{sim_{ij}}{\sum_{k \in \Gamma} sim_{jk}} \cdot p(FAQ_j)$
  else
    $p'(FAQ_i) \leftarrow 0$
  end if
  $b(\mathbf{s})' \leftarrow b(\mathbf{s})' \cup p(FAQ_i)'$
end for
return $b(\mathbf{s})'$
```

which do not satisfy current dialog history. So, we do not distribute the value to the FAQs which do not satisfy current dialog history. We define Γ is the set of FAQ indexes which satisfy current dialog history, and the mapping $b(s_u)$ to $b(s'_u)$ is below updating equation.

$$p'(FAQ_i) = \begin{cases} \sum_{j=1}^{n} \frac{sim_{ij}}{\sum_{k \in \Gamma} sim_{jk}} \cdot p(FAQ_j) \; (i \in \Gamma) \\ 0 \; (i \notin \Gamma) \end{cases} \tag{8}$$

Here, $p(FAQ_i)$ is probability, so the sum of $p(FAQ_1), ..., p(FAQ_n), p(FAQ_{n+1}), ..., p(FAQ_{n+m})$ is 1. So, we use normalized FAQ similarity as the distribution weight.

Figure 7 shows the example of FAQs to explain belief mapping of the user's goal. FAQ_1, FAQ_2 and FAQ_3 are existing FAQs, and FAQ_4, FAQ_5, FAQ_6 are new FAQs. The FAQ similarities of each FAQs are below.

Table 7: Example of FAQs

	Existing FAQ			New FAQ		
	FAQ_1	FAQ_2	FAQ_3	FAQ_4	FAQ_5	FAQ_6
Att_1	1	1	1	1	2	1
Att_2	1	2	1	2	1	1
Att_3	1	1	2	2	1	2
Att_4	1	2	2	1	1	2
Att_5	1	2	1	2	1	2

$$\begin{bmatrix} sim_{11} = 5/5 & sim_{12} = 2/5 & sim_{13} = 3/5 & sim_{14} = 2/5 & sim_{15} = 4/5 & sim_{16} = 2/5 \\ sim_{21} = 2/5 & sim_{22} = 5/5 & sim_{23} = 2/5 & sim_{24} = 3/5 & sim_{25} = 1/5 & sim_{26} = 3/5 \\ sim_{31} = 3/5 & sim_{32} = 2/5 & sim_{33} = 5/5 & sim_{34} = 2/5 & sim_{35} = 2/5 & sim_{36} = 4/5 \end{bmatrix}$$

Now we consider the situation that the belief of the user's goal is $b(s_u) = (0.8, 0.0, 0.2)$ and the dialog history is that the value of attribute2 is 1. Γ, which is the set of FAQ indexes which satisfy current dialog history, is $\Gamma = \{1, 3, 5, 6\}$. So, the result of the belief mapping of the user's goal is below equation.

$$\begin{aligned} p(FAQ_1)' &= \frac{sim_{11}}{sim_{11} + sim_{13} + sim_{15} + sim_{16}} \cdot p(FAQ_1) \\ &\quad + \frac{sim_{12}}{sim_{21} + sim_{23} + sim_{25} + sim_{26}} \cdot p(FAQ_2) \\ &\quad + \frac{sim_{13}}{sim_{31} + sim_{33} + sim_{35} + sim_{36}} \cdot p(FAQ_3) \\ &= \frac{5}{5+3+4+2} \cdot 0.8 + \frac{3}{3+5+2+4} \cdot 0.2 \\ &= 0.329 \\ p(FAQ_2)' &= 0.0 \\ p(FAQ_3)' &= \frac{sim_{31}}{sim_{11} + sim_{13} + sim_{15} + sim_{16}} \cdot p(FAQ_1) \\ &\quad + \frac{sim_{32}}{sim_{21} + sim_{23} + sim_{25} + sim_{26}} \cdot p(FAQ_2) \\ &\quad + \frac{sim_{33}}{sim_{31} + sim_{33} + sim_{35} + sim_{36}} \cdot p(FAQ_3) \\ &= \frac{3}{5+3+4+2} \cdot 0.8 + \frac{5}{3+5+2+4} \cdot 0.2 \end{aligned}$$

$$
\begin{aligned}
&= 0.243 \\
p(FAQ_4)' &= 0.0 \\
p(FAQ_5)' &= \frac{sim_{51}}{sim_{11}+sim_{13}+sim_{15}+sim_{16}} \cdot p(FAQ_1) \\
&\quad + \frac{sim_{52}}{sim_{21}+sim_{23}+sim_{25}+sim_{26}} \cdot p(FAQ_2) \\
&\quad + \frac{sim_{53}}{sim_{31}+sim_{33}+sim_{35}+sim_{36}} \cdot p(FAQ_3) \\
&= \frac{4}{5+3+4+2} \cdot 0.8 + \frac{2}{3+5+2+4} \cdot 0.2 \\
&= 0.257 \\
p(FAQ_6)' &= \frac{sim_{61}}{sim_{11}+sim_{13}+sim_{15}+sim_{16}} \cdot p(FAQ_1) \\
&\quad + \frac{sim_{62}}{sim_{21}+sim_{23}+sim_{25}+sim_{26}} \cdot p(FAQ_2) \\
&\quad + \frac{sim_{63}}{sim_{31}+sim_{33}+sim_{35}+sim_{36}} \cdot p(FAQ_3) \\
&= \frac{2}{5+3+4+2} \cdot 0.8 + \frac{4}{3+5+2+4} \cdot 0.2 \\
&= 0.171
\end{aligned}
$$

So the mapped belief of the user's goal $b(s_u')$ is

$$b(s_u') = (0.329, 0.000, 0.243, 0.000, 0.257, 0.171)$$

6.2.2 Mapping the System's Action

The elements of the system's action A_m are separated into two types. First type is ask_i, where the system asks att_i to the user. Second type is $show_i$, where the system shows FAQ_i to the user as the result of FAQ search.

The number of the FAQs increases when new FAQs are added. So, the system's action A_m' in SDS-POMDP after adding new FAQs includes the actions such as $show_{n+1}$,...,$show_{n+m}$, where the system shows the new FAQs. But, the system's action A_m in SDS-POMDP before adding new FAQs does not include the actions such as $show_{n+1}$,...,$show_{n+m}$. This means that the system does not show new action by using SDS-POMDP before adding new FAQs. So, we map the system's action a_m after adding new FAQs to the system's action a_m' after adding new FAQs, and make it possible for the system to show new FAQs.

There are two cases in action mapping. First, we consider the case when the system's action a_m before adding new FAQs is ask_i, where the system asks an att_i. We assume that the attribute value which the system should ask in some state is not change after adding a few new FAQs to many existing FAQs. So, the mapping result a' is ask_i.

Second, we consider the case when the system's action a_m before adding new FAQs is $show_i$, where the system shows FAQ_i as the result of search result. Here, we select an showing a FAQ action randomly using roulette wheel selection. The possible FAQs which are selected by roulette wheel selection is the FAQ_i and the new FAQs which satisfy current dialog history. The ratio of selecting each FAQs depends on $b(s'_u)$ which is calculated in 6.2.1. Γ' is the set of FAQ index which is selected as the optimal action to show and FAQ indexes which satisfy current dialog history. Then, $p(show_i)$, which represents the probability that $show_i$ is a mapped action, is calculated as below equation.

$$p(show_i) = \begin{cases} \dfrac{p'(FAQ_i)}{\sum\limits_{j \in \Gamma'} p'(FAQ_j)} \; (i \in \Gamma') \\ 0 \; (i \notin \Gamma') \end{cases} \tag{9}$$

Here, we use the FAQ examples in 6.2.1. Now we consider the situation that the belief of the mapped user's goal is $b(s'_u) = (0.4, 0.0, 0.1, 0.0, 0.4, 0.1)$, the dialog history is that the value of attribute2 is 1, and the optimal action before adding new FAQs is $show_1$. Γ', which is the set of FAQ index which is selected as the optimal action to show and FAQ indexes which satisfy current dialog history is $\Gamma' = \{1, 5, 6\}$. So, the probabilities that showing each FAQs are selected as the optimal action is calculated by using equation 9.

$$\begin{aligned} p(show_1) &= \frac{p(FAQ_1)}{p(FAQ_1) + p(FAQ_5) + p(FAQ_6)} \\ &= \frac{0.4}{0.4 + 0.4 + 0.1} \\ &= \frac{4}{9} \\ p(show_2) &= 0 \\ p(show_3) &= 0 \\ p(show_4) &= 0 \end{aligned}$$

Algorithm 2 The process of mapping the system's action

```
$a$ /*best system's action of POMDP before adding new FAQs*/
$a'$ /*mapped system's action*/
$\Gamma' \leftarrow \emptyset$
$\mathbf{w} \leftarrow \emptyset$
Insert new FAQ indexes which satisfy current dialog history into $\Gamma'$
if $a$ is show $FAQ_i$ then
  Insert $i$ into $\Gamma'$
  for $j = 1$ to $n + m$ do
    if $j \in \Gamma'$ then
      $w_j \leftarrow \frac{p'(FAQ_j)}{\sum_{k \in \Gamma'} p'(FAQ_k)}$
    else
      $w_j \leftarrow 0$
    end if
  end for
  Randomly select $l \in \Gamma'$ using roulette wheel selection which fitness function is $\mathbf{w}$
  $a' \leftarrow show_l$
else
  $a' \leftarrow a$
end if
return $a'$
```

$$
\begin{aligned}
p(show_5) &= \frac{p(FAQ_5)}{p(FAQ_1) + p(FAQ_5) + p(FAQ_6)} \\
&= \frac{0.4}{0.4 + 0.4 + 0.1} \\
&= \frac{4}{9} \\
p(show_6) &= \frac{p(FAQ_6)}{p(FAQ_1) + p(FAQ_5) + p(FAQ_6)} \\
&= \frac{0.1}{0.4 + 0.4 + 0.1}
\end{aligned}
$$

$$= \frac{1}{9}$$

So, the ratio of selecting $show_1$,$show_5$ and $show_6$ is 4:4:1.

6.2.3 Belief Update

The transition function is used when the belief is updated. But, the transition function when the system shows a new FAQ does not exist in SDS-POMDP before adding new FAQ, and the belief update is impossible.

So, the belief is updated by not using the transition functions when the system shows FAQ. If the user accepts a FAQ, the dialog is finished. So, the belief update not using the transition function is needed when the user's action is *decline*.

We consider the situation when the system's action is $show_i$ and the user's action is *decline*. The beliefs which should be update are the belief of the user's goal $b(s_u)$, the belief of the dialog history $b(s_d)$ and the belief of the user's action $b(a_u)$.

The dialog history is the set of tuples of attribute values given from the user. So, the dialog history does not updated when the user's action does not answer attribute value. Thus, the belief of the dialog history $b(s_d)$ is not updated when the system's action is $show_i$ and the user's action is *decline*. The belief of the user's action $b(a_u)$ is not updated either when the system's action is $show_i$ and the user's action is *decline*.

In contrast, the belief of the user's goal $b(s_u)$ have to be updated when the system's action is $show_i$ and the user's action is *decline*. $p(FAQ_i)$, which represents the probability that the user wants FAQ_i, have to be 0 when the system's action is $show_i$ and the user's action is *decline*. But, the belief of the user's goal $b(s_u)$ before adding new FAQs does not include $p(FAQ_{new})$. So, $b(s_u)$ is updated below process.

We introduce $b_t(s_u)$, which is the belief of the user's goal in time step t, and $b_{t+1}(s_u)$, which is the belief of the user's goal in time step $t + 1$. Then we consider the situation when the mapped system's action is $show_j$ and the user's action is *decline*. Here, we scale down the FAQ probabilities using FAQ similarity. When scaling down, the more similar to FAQ_j the FAQ is, the more

Algorithm 3 The process of belief update

```
$j$ /* New FAQ index which the system shows in time step t*/
$s_{t+1}(s_u) \leftarrow \emptyset$
if the system's action is $show_j$ and the user's action is $decline$ then
  for $i = 1$ to $n$ do
    $s_{t+1}(s_u) \leftarrow p_t(FAQ_i) \cdot (1 - sim_{ij})$
  end for
  Normalize $s_{t+1}(s_u)$
else
  $s_{t+1}(s_u) \leftarrow s_t(FAQ_i)$
end if
return $s_{t+1}(s_u)$
```

the FAQ probability reduces. So, the belief is updated as below equation.

$$p_{t+1}(FAQ_i) = \alpha p_t(FAQ_i) \cdot (1 - sim_{ij}) \tag{10}$$

α is the normalizing constant which makes the sum of $p(FAQ_1), ..., p(FAQ_n)$ is 1.

6.3 Evaluation

We evaluated the performance of an interactive FAQ search system using transfer learning. We describe the experiment data. The number of FAQs is 10, the number of attributes is 5, and each attributes are binary variable as the same as Chapter 5 evaluation. The attribute values of each FAQs are decided randomly. The probabilities that the user wants each FAQs are biased as described in 5.3. So, the probabilities that the user wants the 8 FAQs are 0.01, and the probabilities that the user wants the other 2 FAQs are 0.42. The evaluation criteria are the number of asking attribute value and the accuracy to show a FAQ which the user wants. Accuracy is the ratio of showing a FAQ which the user wants to showing a FAQ.

We evaluated two setting. First, we evaluated the performance when the number of new FAQs is changed. The number of existing FAQs is constantly

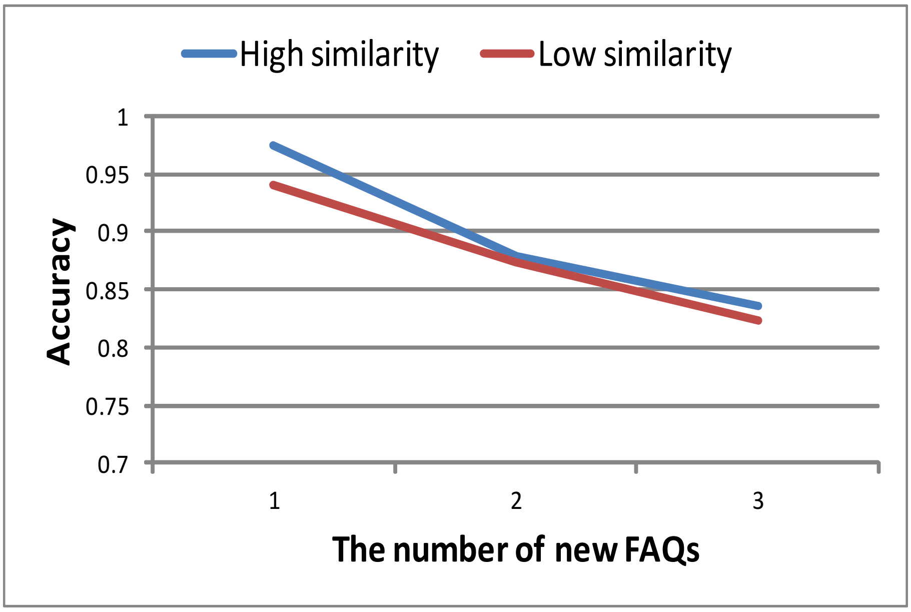

Figure 9: Accuracy of all FAQs

10.

Second, we evaluated the performance when the similarities between new FAQs and existing FAQs are changed. Here, we calculated the similarity between new FAQs and existing FAQs as $\sum_{i=1}^{n} \sum_{j=n+1}^{n+m} sim(i,j)/n \cdot m$.

First, we evaluated the situation where the user wants not only new FAQs but also existing FAQs. Here, the probability that the user wants a new FAQ is the ratio of the number of new FAQs to the number of new FAQs and existing FAQs. So, the probability that the user wants new FAQs is 1/11 when the number of new FAQs is 1, 2/12 when the number of new FAQs is 2, and 3/13 when the number of new FAQs is 3. The probabilities that the user wants existing FAQs are downscaled in order to make the sum of probabilities 1, where the proportion of the probabilities is saved. So, the probability of a FAQ which is 0.01 before adding new FAQs becomes $0.01 \cdot 10/11$ when the number of new FAQs is 1.

Figure 9 shows the accuracy to show a FAQ which the user wants. The accuracy is 0.975 when the similarity between new FAQs and existing FAQs is high, and the accuracy is 0.94 when the similarity is low. The system can frequently show the FAQ which the user wants. But the accuracy becomes low

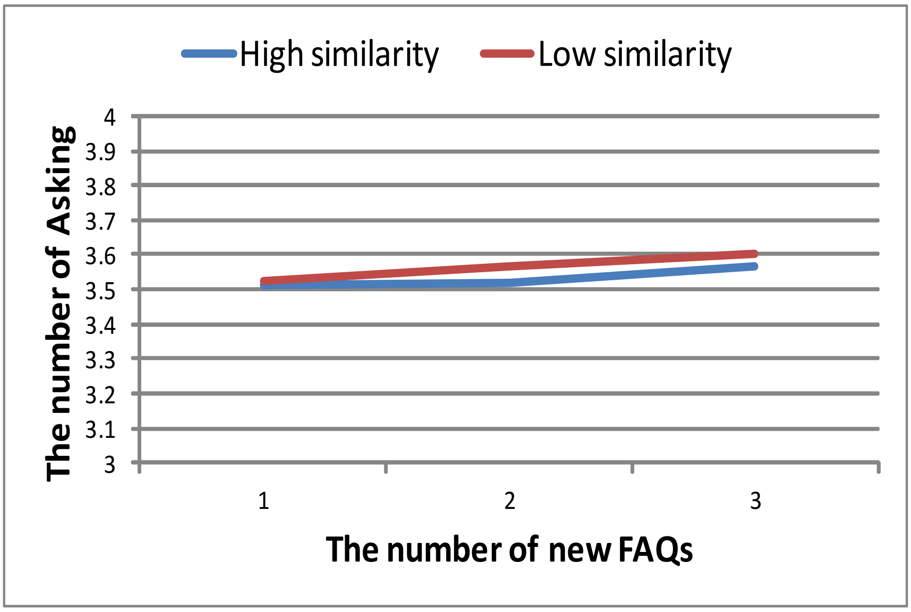

Figure 10: The number of asking an attribute of all FAQs

if the number of new FAQs is large. The accuracy is 0.83 when the number of new FAQs is 3 and the similarity is high, and the accuracy is 0.82 when the number of new FAQs and the similarity is low. The accuracy declines due to the increase of new FAQs because the system shows a FAQ randomly in the system's action mapping. The number of new FAQs becomes large, and the probability of showing a FAQ which the user does not want becomes high.

The accuracy is higher when the similarity is high. It's because the timing of the system's action mapping. The system's action is mapped when the system shows a FAQ. But the number of same attribute value between new FAQ and existing FAQs tends to small when the similarity is low. So, there tend to be no existing FAQs which satisfy current dialog history if the user which wants a low similarity FAQ answers the attributes. So, the system does not show the FAQs during dialog, and the accuracy declines.

Next, figure 10 shows the number of asking an attribute. Figure 10 shows that the number of new FAQs does not affect the number of asking. It's because the system uses the policy of SDS-POMDP before in dialog control of asking an attribute.

Figure 10 also shows that the similarity between new FAQs and existing

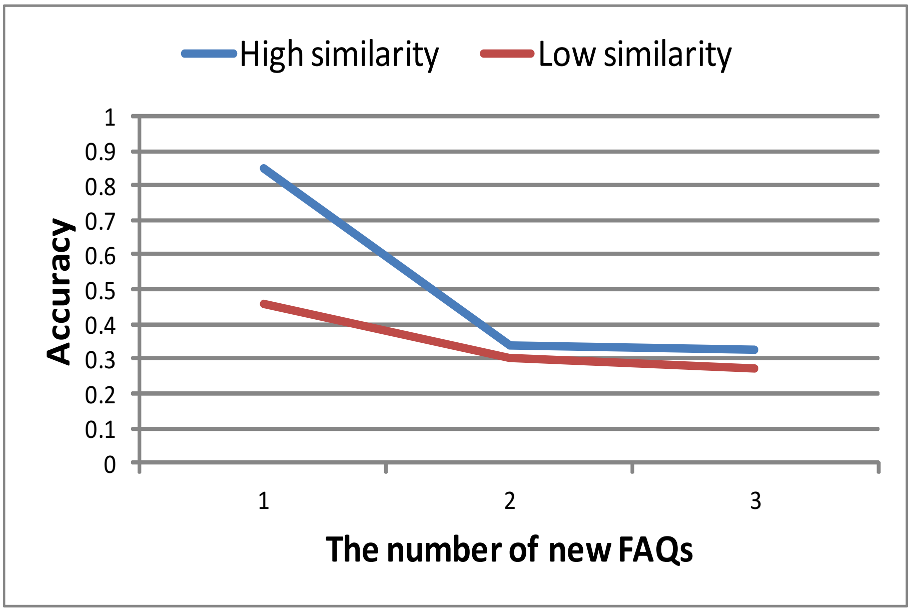

Figure 11: Accuracy of new FAQs

FAQs does not affect the number of asking. As we described the accuracy result, when the user wants a low similarity FAQ, there tend to be no existing FAQs which satisfy the current dialog history. As the result, the system continues to ask an attribute. But, we does not count the such asking. So, the number of asking an attribute does not depends on the similarity.

The probabilities that the user wants new FAQs depend on the performance. So, we evaluated the case that the user wants only new FAQs.

Figure 11 shows the accuracy when the user wants only new FAQs. If the similarity is high, the accuracy is 0.85 when the number of new FAQ is 1, and the accuracy is 0.32 when the number of new FAQs is 3. If the similarity is low, the accuracy is 0.46 when the number of new FAQ is 1, and the accuracy is 0.27 when the number of new FAQs is 3. So, the accuracy becomes low.

The reason why the accuracy becomes low is the FAQ candidates which the system shows in the system's mapping. The FAQ candidates which the system shows is the existing FAQ which the system shows is the optimal system's action in SDS-POMDP before adding new FAQs and the new FAQs which satisfy the current dialog history in the system's action mapping. New FAQs which satisfy current dialog history does not always exists. But, the existing FAQ which the

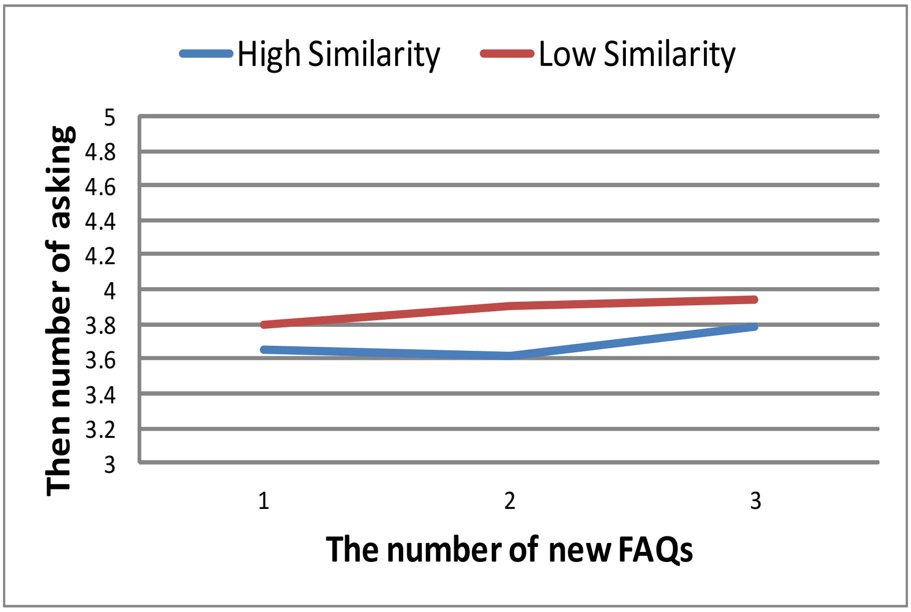

Figure 12: The number of asking an attribute of new FAQs

system shows is the optimal system's action in SDS-POMDP before adding new FAQs always exists. So, the systems may show a existing FAQ to the user who wants a new FAQ even if the mapping depends on the mapped belief of user's goal $b(s'_u)$, and the accuracy declines when the user wants only new FAQs.

Finally, figure 12 shows the number of asking an attribute when the user wants only new FAQs The number of asking an attribute is not so different between figure 12 and figure 10.

Chapter 7 Discussion

7.1 Scaling

The number of attributes is 5 and each attribute values are binary variable in evaluation. But, the number of attributes is 15 and the number of attribute values is from 3 to 35 in CTStage's FAQ, which is real FAQ data. So, the number of attributes and the number of attribute values in evaluation are smaller than the number of attributes and the number of attribute values in real world. The reason why we set the number of attributes is 5 and each attribute values are binary variable is scaling problem.

We consider the dimension of SDS-POMDP in an interactive FAQ search system. The dimension of the user's goal is n, the dimension of the dialog history is o^m and the dimension of the user's goal is $2 + m \cdot o$, where the number of FAQ is n, the number of attributes is m and the number of attribute values is o. So, the dimension of the state is $n \cdot o^m \cdot (2 + m \cdot o)$, and the dimension of the state increase exponentially by increasing the number of attributes m. So, the computation of SDS-POMDP is impossible if the number of attributes m becomes large.

This research assumes that the attributes are independent in order to simplify the problem. In other words, a attribute value is not affected by other attribute values. But, some attributes in CTStage are dependent. So, the attribute can be inferred from other attribute values without asking to the user. So, the number of attributes m can be reduced by not including the attribute in the dialog history.

7.2 Error Process in Transfer Learning

As we described in 6.3, there are case that there are no existing FAQs which satisfy current dialog history if the user wants a new FAQ. If this happens, the system continues to ask an attribute without showing a new FAQ. So, we need to add error process in transfer learning when there are no existing FAQs which satisfy current dialog history.

We propose the method to show a new FAQ which satisfies current dialog history randomly when there are no existing FAQs which satisfy current dialog history. This method makes it possible to show new FAQs.

In the system's action mapping, we use roulette wheel selection which weight is the mapped belief of the user's goal $b(s'_u)$. But the belief of the user's goal $b(s_u)$ is uniform when there are no existing FAQs which satisfy current dialog history, and we cannot use the mapped belief of the user's goal $b(s'_u)$ as the weight. So, we have to select new FAQs which satisfy current dialog history randomly assumed the uniform distribution.

Chapter 8 Conclusion

First, this research applies SDS-POMDP to dialog control in an interactive FAQ search system. The dialog control using dialog scenario costs a lot. So, we reduced the cost of implement dialog control through using already existing dialog log to learn a policy learning in SDS-POMDP.

Second, this research applies transfer learning to implement dialog control after adding new FAQs. The dialog log about new FAQs is needed when learning a policy of SDS-POMDP. But, the dialog log about new FAQs does not existing, so it is impossible to learn a policy. Creating dialog log about new FAQs is possible, but it costs a lot. So, we transferred the learning result of SDS-POMDP before adding new FAQs to SDS-POMDP after adding new FAQs, and reduced the costs of implementation due to needless of creating dialog log.

This research assumes that the number of attributes is not changed after adding new FAQs. But new attributes may be added when new FAQs are added. The future work is the transfer learning when new attributes are added. The system's action is mapped only when the system shows a FAQ if the new attributes are not added. But the system's action is mapped not only when the system shows FAQ but also when the system asks an attribute value if the new attributes are added.

Acknowledgments

The author would like to express sincere gratitude to the supervisor, Associate Professor Shigeo Matsubara at Kyoto University, for his consecutive instruction, pertinent advice and earnest argument. The author would like to express sincere appreciations to the advisers, Professor Tetsuro Sakai at Kyoto University, Associate Professor Michinori Hatayama at Kyoto University for his valuable advice. The author would like to express deep gratitude to Oki Electric Industry Co., Ltd for giving the start of this research. The author would like to tender his acknowledgments to Professor Toru Ishida, Assistant Professor Hiromitsu Hattori, the member of Ishida Matsubara Laboratory at Kyoto University, for his technical and constructive advice. Finally the author would like to thank all member of Ishida and Matsubara laboratory for their various support and discussion. Especially the author would like to thank Noriyuki Ishida to help implementing an evaluation program.

References

[1] M. Kitamura, S. Shimohata, T. Sukehiro, A. Ikeno, M. Sakamoto, I. Orihara, and T. Murata. Laddering search service system: “ ladasearch ” . In *Universal Communication, 2008. ISUC'08. Second International Symposium on Universal Communication.*, pages 382–389. IEEE, 2008.

[2] B. Thomson and S. Young. Bayesian update of dialogue state: A pomdp framework for spoken dialogue systems. *Computer Speech & Language*, 24(4):562–588, 2010.

[3] A.R. Cassandra. A survey of pomdp applications. In *Working Notes of AAAI 1998 Fall Symposium on Planning with Partially Observable Markov Decision Processes*, pages 17–24, 1998.

[4] B. Thomson, F. Jurcicek, M. Gasic, S. Keizer, F. Mairesse, K. Yu, and S. Young. Parameter learning for pomdp spoken dialogue models. In *Spoken Language Technology Workshop (SLT), 2010 IEEE*, pages 271–276. IEEE, 2010.

[5] R. Bellman. A markovian decision process. Technical report, DTIC Document, 1957.

[6] H. Kurniawati, D. Hsu, and W.S. Lee. Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces. In *Proc. Robotics: Science and Systems*, 2008.

[7] M.A. Walker, D.J. Litman, C.A. Kamm, and A. Abella. Paradise: A framework for evaluating spoken dialogue agents. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pages 271–280. Association for Computational Linguistics, 1997.

[8] J.D. Williams and S. Young. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422, 2007.

[9] M.E. Taylor and P. Stone. Transfer learning for reinforcement learning domains: A survey. *The Journal of Machine Learning Research*, 10:1633–1685, 2009.

[10] D. Foster and P. Dayan. Structure in the space of value functions. *Machine*

Learning, 49(2):325–346, 2002.

[11] J.D. Williams and S. Young. Scaling pomdps for dialog management with composite summary point-based value iteration (cspbvi). In *AAAI Workshop on Statistical and Empirical Approaches for Spoken Dialogue Systems*, pages 37–42, 2006.

[12] S. Young. Using pomdps for dialog management. In *Spoken Language Technology Workshop, 2006. IEEE*, pages 8–13. IEEE, 2006.