Application of Hidden Topic Markov Model on Dialogues for Learning Dialogue POMDP Models from Data

Hamid R. Chinaei, Brahim Chaib-draa
Computer Science Department, Laval University
Quebec, Canada
hrchinaei@damas.ift.ulaval.ca
Brahim.Chaib-Draa@ift.ulaval.ca

***Abstract*—In this paper, we apply Hidden Topic Markov Model (HTMM) for learning the components of dialogue POMDP models from data. In particular, using unannotated human-human dialogues we learn the states, observations, as well as transition and observation functions based on HTMM. First, we go through the HTMM and its application for dialogues in order to learn the intentions behind user's utterances. As proposed earlier for dialogue POMDPs, we also use the discovered user's intentions for the states of dialogue POMDPs. However, as opposed to previous works, instead of using state keywords as POMDP observations, we use some meta observations based on the learned user's intentions. Since the number of meta observations is much less than the actual observations, i.e. the number of words in the dialogue set, the POMDP learning and planning becomes tractable. The experimental results on real dialogues show that the quality of the learned models increases by increasing the number of dialogues as training data. Moreover, the experiments based on simulation show that the introduced method is robust to the ASR noise level. Furthermore, we briefly discuss about related work to this paper followed by our conclusion and our future research of learning parameters of dialogue POMDPs at the end.**

I. Introduction

Consider the example in Table I taken from the dialogue set SACTI-2 [12], where SACTI stands for Simulated ASR-Channel: Tourist Information. The first line of the table shows the first user's utterance, $U1$. Because of Automatic Speech Recognition (ASR) this utterance is corrupted and is received by the system as $U'1$ in the following line in braces. $M1$ in the next line shows the system's response to the user.

For each dialogue utterance, the system's goal is first to capture the user's intention and then to perform the best action which satisfies the user's intention. For instance, for the first received user's utterance *U'1 [Is there a good restaurant week an hour tonight]*, the system may be able to predict the user's intention as request information for *restaurant* with a high probability, as the utterance contains the only keyword *restaurant*. However, in the second received user's utterance, *U'2 [No I think late like uh museum price restaurant]*, the system has more difficulty to find the user's intention. In fact, in $U'2$, the system is required to understand that the user is looking for a *restaurant*; though this utterance is highly corrupted. Specifically, it contains misleading words such as *museum* that can be strong observations for another user's intention, i.e. user's intention for museums.

TABLE I: Sample dialogue from SACTI-2.

U1	*Is there a good restaurant we can go to tonight*
U'1	*[Is there a good restaurant week an hour tonight]*
M1	*Would you like an expensive restaurant*
U2	*No I think we'd like a medium priced restaurant*
U'2	*[No I think late like uh museum price restaurant]*
M2	*Cheapest restaurant is eight pounds per person*
U3	*Can you tell me the name*
U'3	*[Can you tell me the name]*
M3	*bochka*
M4	*b o c h k a*
U4	*Thank you can you show me on the map where it is*
U'4	*[Thank you can you show me i'm there now where it is]*
M5	*It's here*
U5	*Thank you*
U'5	*[Thank u]*
U6	*I would like to go to the museum first*
U'6	*[I would like a hour there museum first]*
	...

Recently, there has been a great interest for modelling the dialogue manager of spoken dialogue systems using Partially Observable Markov Decision Processes (POMDPs) [15]. However, in POMDPs, similar to many other machine learning frameworks, estimating the environment dynamics is a significant issue; as it has direct impact on their applicability in the domain of interest. In other words, the POMPD models highly impact the planned strategies. Moreover, a good learned model can be used as a prior model in all Bayesian approaches so that the model be further updated and enhanced. As such, in this work we are interested in learning proper POMDP models for dialogue POMDPs based on human-human dialogues.

In this paper, we present a method for learning the components of dialogue POMDP models using unannotated data available in SDSs. In fact, using an unsupervised method based on Dirichlet distribution, one can learn states and observations as well as transition and observation POMDP functions. In addition, we develop a simple idea for reducing the number of observations while learning the model, and define a small practical set of observations for the designed dialogue POMDP.

In the rest of the paper, we briefly present POMDP background in Section II. Then, in Section III we introduce a variant of Latent Dirichlet Allocation for dialogues. Section IV introduces our method for learning dialogue POMDP models followed by the experiments in Section V. Discussions about our method of learning dialogue POMDP models is presented in Section VI followed by conclusion and future work in Section VII.

II. Background

A POMDP can be represented by n-tuple $\{\mathcal{S},\mathcal{A},O,T,R,\Omega,\gamma,H\}$ where $\mathcal{S}$ is the set of some discrete states, $\mathcal{A}$ is the set of some discrete actions, $R(s,a)$ is the reward of taking action a in the state s, and T the transition function which consists of the probability of state transitions:

$$T(s,a,s') = P(s_{t+1} = s'|a_t = a, s_t = s)$$

where s is for the current state of the system and s' for the next state of the system.

Moreover, O is the set of some discrete observations which is used for estimating the current state using the model Ω, with $\Omega(o,s,a) = P(o|a,s)$, i.e. the probability of observing o after taking the action a which results in the state s. The system's initial belief state is b_0, and the belief state at time t is derived from $b_t = P(s_t|b_0,a_0,o_1,\ldots,b_{n-1},a_{n-1},o_n)$, γ is a discount factor, and H is the planning horizon. The belief at time $t+1$, b_{t+1}, can be computed from the previous belief, b_t, the last action a, and observation o, by applying Bayes rule:

$$b^{a,o}_{t+1}(s') = \frac{1}{P(o|b_t,a)}\Omega(o,s',a)\sum_{s\in\mathcal{S}} T(s,a,s')b_t(s)$$

where $P(o|b_t,a) = \sum_{s'\in\mathcal{S}}\Omega(s',a,o)\sum_{s\in\mathcal{S}}T(s,a,s')b_t(s)$ is the probability of observing o after doing action a in belief b_t, this acts as a normalizing constant such that b_{t+1} remains a probability distribution:

In POMDPs, the system's goal is to find an optimal policy, π, where $\pi : \mathcal{S}\times\mathcal{A}\to[0,1]$, that maximizes the expected discounted rewards. That is, the policy which has the maximum value. The value of policy π is defined as:

$$V^{\pi}(s) = \sum_{t=0}^{\infty} E_{s_t,a_t}[\gamma^t R(s_t,a_t)|s_0 = s,\pi]$$

If the environment dynamics are not known, i.e. transition probabilities, observation probabilities, or the reward function are not known in advance, the system has to assume the environment as an unknown POMDP and during interaction with the user tries to approximate the unknown function(s). Conversely, if the dynamics are known to the system then the classic dynamic programming techniques, such as point based value iteration [9], can be used to find an optimal policy π^*. In this work, we would like to learn the dialogue POMDP from data, to be able to learn the POMDP policy from available model-based algorithms.

III. Hidden Topic Markov Models for Dialogues

In this section, we briefly describe our previous work for learning states, as well as transition and observation functions from dialogues [1]. First, we briefly explain Hidden Topic Markov Model (HTMM) [6] that we use to learn hidden intentions behind user's utterances.

HTMM is a method which combines Hidden Markov Model (HMM) and Latent Dirichlet Analysis (LDA) for obtaining some topics for documents [6]. HMM is a framework for obtaining the hidden states based on some observations in Markovian domains such as part-of-speech tagging [2]. In LDA, similar to Probabilistic Latent Semantic Analysis (PLSA), the observations are explained by groups of latent variables. For instance, if we consider observations as uttered words in a dialogue, then the dialogue is considered as a bag of words with mixture of some intentions, where intentions are represented by the words with higher probabilities. In LDA as opposed to PLSA, the mixture of intentions is generated from a Dirichlet prior mutual to all dialogues in a dialogue set. Since HTMM adds the Markovian property inherited in HMM to LDA, in HTMM the dependency between successive words is regarded, and the dialogue is no longer seen as a bag of words. Notice that HTMM can be considered as a clustering method such as LSA, however, it is probabilistic similar to PLSA, and moreover, the Markovian property between the utterances are considered as opposed to PLSA or LSA.

Hidden intentions can be used as user's hidden states in a dialogue POMDP [3]. For each dialogue in the dialogue set, we assign its hidden state as the maximum likely state, then we estimate the transition function using maximum likelihood with a smoothing method to make it more robust. To construct an observation function for the dialogue POMDP, we use the learned hidden states for each utterance as its meta observation. This meta observations are used as the POMDP observation set rather than the whole words in the utterance to be able to reduce the size of observation set significantly. However, this model is a fully observable Markov Decision Process (MDP). To do it in POMDPs, the observations are reduced to the number of states, and the observation function is estimated by taking average over belief of states given each action and next state. This allows that each meta observation be allowed in other states with a small probability.

In HTMM model, latent topics are found using Latent Dirichlet Allocation. The topics for a document are generated using a multinomial distribution, defined by a vector θ. The vector θ is generated using the Dirichlet prior α. Words for all documents in the corpus are generated based on multinomial distribution, defined by a vector β. The vector β is generated using the Dirichlet prior η. Figure 1 shows that the dialogue d in a dialogue set D can be seen as a sequence of words (w) which are observations for some hidden topics (z). Since hidden topics are equivalent to user intentions in our work, from now on, we call hidden topics as user intentions. The vector β is a global vector that ties all the dialogues in a dialogue set D, and retains the probability of words given user

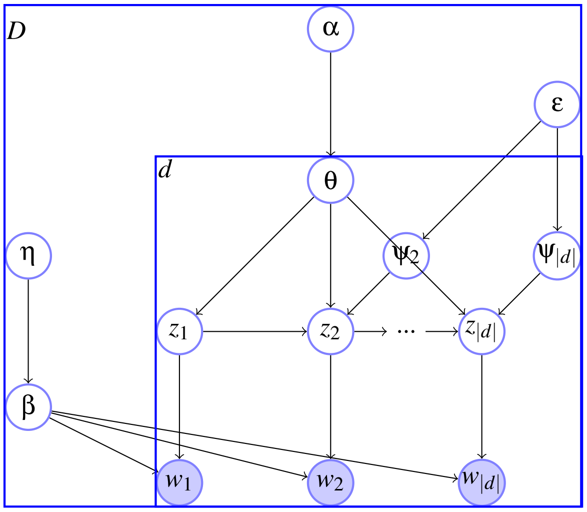


Fig. 1: The HTMM model adapted from [6], the shaded nodes are observations (w) used to capture intentions (z).

intentions. The vector θ is a local vector for each dialogue d, and retains the probability of intentions in a dialogue.

Algorithm 1 shows the process of generating and updating the parameters. First, for all possible user intentions β is drawn using the Dirichlet prior η. Then, for each dialogue, θ is drawn using the Dirichlet prior α.

The parameter ψ_i is for adding the Markovian property in dialogues since successive utterances are more likely to include the same user intention. The assumption here is that an utterance represents only one user intention, so all the words in an utterance are representative for the same user intention. To formalize that, the algorithm assigns $\psi_i = 1$ for the first word of an utterance, and $\psi_i = 0$ for the rest. Then, the intention transition is possible just when $\psi = 1$. This is represented in the algorithm between lines 6 and 18. Moreover, ε is used as a prior over ψ which controls the probability of transition between utterances in dialogues.

HTMM uses Expectation Maximization (EM) and forward backward algorithm [10], the standard method for approximating the parameters in HMMs. It is because of the fact that conditioned on θ and β, HTMM is a special case of HMMs. In HTMM, the latent variables are user intentions z_i and ψ_i which determines if the intention for the word w_i is drawn from w_{i-1}, or a new intention will be generated. In the expectation step, for each user intention z, we need to find the expected count of intention transitions to intention z.

$$E(C_{d,z}) = \sum_{j=1}^{|d|} Pr(z_{d,j} = z, \psi_{d,j} = 1 | w_1, \ldots, w_{|d|})$$

where d is a dialogue in the dialogue set D.
Moreover, we need to find expected number of co-occurrence of a word w with an intention z.

$$E(C^{z,w}) = \sum_{i=1}^{|D|} \sum_{j=1}^{|d_i|} Pr(z_{i,j} = z, w_{i,j} = w | w_1, \ldots, w_{|d|})$$

```
   Input: Set of transcribed dialogues $D$, N number of
          intentions
   Output: Finding intentions for D
1  foreach intention $z$ in the set of $N$ intentions do
2      Draw $\beta_z \sim Dirichlet(\eta)$;
3  end
4  foreach dialogue $d$ in $D$ do
5      Draw $\theta \sim Dirichlet(\alpha)$;

6      foreach $i = 1 \ldots |d|$ do
7          if beginning of an utterance then
8              $\psi_i = bernoli(\varepsilon)$
9          else
10             $\psi_i = 0$
11         end
12     end
13     foreach $i = 1 \ldots |d|$ do
14         if $\psi_i = 0$ then
15             $z_i = z_{i-1}$
16         else
17             $z_i = multinomial(\theta)$
18         end

19         Draw $w_i \sim multinomial(\beta_{z_i})$ ;
20     end
21 end
```

Algorithm 1: The HTMM generative algorithm adapted from [6].

In the Maximization step, the MAP (Maximum A Posteriori) for θ and β is computed using Lagrange multipliers:

$$\theta_{d,z} \propto E(C_{d,z}) + \alpha - 1$$

$$\beta_{z,w} \propto E(C^{z,w}) + \eta - 1$$

The random variable $\beta_{z,w}$ gives the probability of an observation w given the intention z.
The parameter ε denotes the dependency of the utterances on each other, i.e. how likely it is that two successive uttered utterance of the user have the same intention.

$$\varepsilon = \frac{\sum_{i=1}^{|D|} \sum_{j=1}^{|d|} Pr(\psi_{i,j} = 1 | w_1, \ldots, w_{|d|})}{\sum_{i=1}^{|D|} N_{i,utt}}$$

where $N_{i,utt}$ is the number of utterances in the dialogue i.

In this method, EM is used for finding MAP estimate in hierarchical generative model similar to LDA. [4] argued that Gibbs sampling is preferable than EM since EM can be trapped in local minima. Also, [8] argued that EM suffer from local minima. However, they suggested methods for getting away from local minima. Furthermore, they also proposed that EM can be accelerated based on the type of the problem. In HTMM, the special form of the transition matrix enables a reduced time complexity of $O(|d|N^2)$, where $|d|$ is the length of the dialogue d, and N is the number of desired user's intentions, given to the algorithm [5].

Intention 0:	visits			Intention 1:	transports			Intention 2:	foods		
the	0.08	like	0.01	the	0.08	a	0.02	you	0.06	um	0.02
i	0.06	**hotel**	0.01	to	0.04	does	0.02	the	0.04	and	0.02
to	0.05	for	0.01	is	0.04	**road**	0.02	i	0.04	thank	0.01
um	0.02	would	0.01	how	0.03	and	0.01	a	0.03	to	0.01
is	0.02	i'm	0.01	um	0.02	on	0.01	me	0.03	of	0.01
a	0.02	**tower**	0.01	it	0.02	long	0.01	is	0.02	**restaurant**	0.01
and	0.02	**castle**	0.01	uh	0.02	of	0.01	uh	0.02	there	0.01
you	0.02	go	0.01	i	0.02	much	0.01	can	0.02	do	0.01
uh	0.02	do	0.01	from	0.02	**bus**	0.01	tell	0.02	could	0.01
what	0.01	me	0.01	**street**	0.02	there	0.01	please	0.02	where	0.01

Fig. 2: Intentions learned by HTMM for SACTI-1, with their 20-top words and their probabilities.

IV. Capturing Dialogue POMDP Model for SACTI-1

This section describes the method for learning POMDP transition and observation functions. Using HTMM, we designed a dialogue POMDP, for SACTI-1 dialogues [14], publicly available at: http://mi.eng.cam.ac.uk/projects/sacti/corpora/. There are about 144 dialogues between 36 users and 12 experts who play the role of a DM for 24 total tasks on this data set. Similar to SACTI-2, the utterances here are also first confused using a speech recognition error simulator, and then are sent to the human experts.

Figure 2 shows 3 captured user's intentions and their top 20 words with their probabilities learned by HTMM. For each intention, we have highlighted the keywords which best distinguish the intention. These intentions are for the user's intentions for request information about some visiting places, the transportation, and food places, respectively.

Without loss of generality, we can consider the user's intention as the system's state [3]. Based on the above captured intentions, we defined 3 primary states for the SACTI-1 DM as follows: *visits (v)* , *transports (t)* , and *foods (f)*. Moreover, we defined two absorb states, i.e., *Success (S)* and *Failure (F)* for dialogues which end successfully and unsuccessfully, respectively. The notion of successful or unsuccessful dialogue is defined by user. After finishing each dialogue, the user assigns the level of precision and recall. These are the only explicit feedback which we require from the user, to be able to define absorb states of dialogue POMDP. A dialogue is successful if its precision and recall is above a predefined threshold. Figure 3 shows defined states for SACTI-1 dialogue POMDP.

The set of actions are coming directly from SACTI-1 dialogue set, and they include: *GreetingFarewell*, *Inform*, *StateInterp*, *IncompleteUnknown*, *Request*, *ReqRepeat*, *RespondAffirm*, *RespondNegate*, *ExplAck*, *ReqAck*, etc. For instance *GreetingFarewell* is used for initiating or ending a dialogue, *Inform* is for giving information for a user's intention, *ReqAck* is for the DM's request for user's acknowledgement, *StateInterp* for interpreting the intentions of user, and it can be considered as implicit confirmation, etc.

The transition function is calculated using maximum likelihood with add-one smoothing to make a more robust transition model:

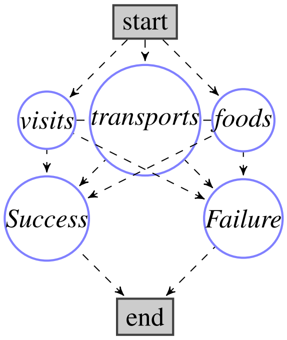


Fig. 3: The dialogue POMDP states for the DM based on SACTI-1 dialogues.

$$T(s_1,a_1,s_2) \quad = \quad \frac{Count(s_1,a_1,s_2)+1}{Count(s_1,a_1)+K}$$

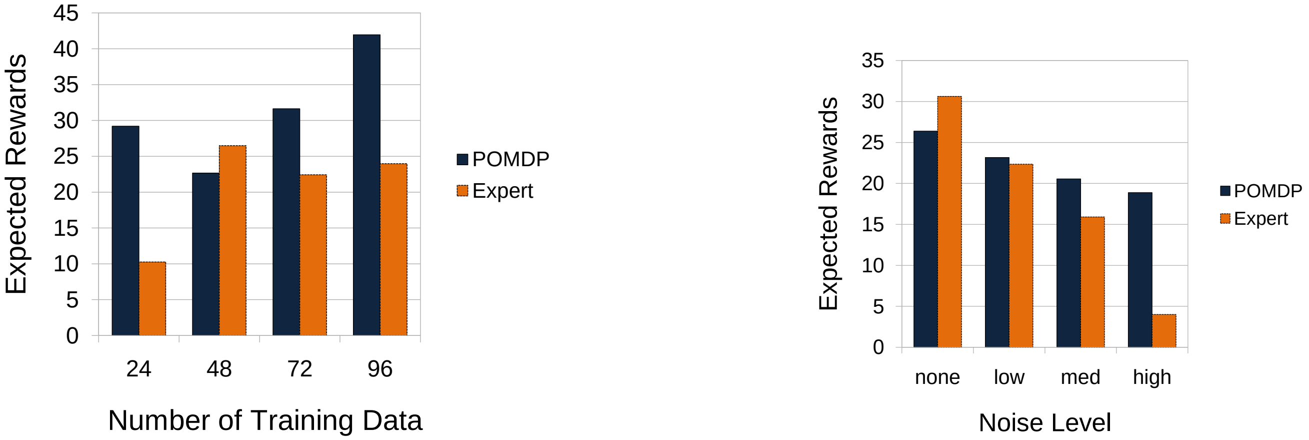
where $K = |S|^2|A|$, S is the state set, and S equals to number of intentions N which is 5 in our example. For each utterance U its corresponding state is the intention with highest probability.

For the choice of observation function, we assumed 5 observations, each one is specific for one state, i.e. user's hidden intention. we use the notation O= { *VO, TO, FO, SuccessO, FailureO* } for the meta observations for *visits, transports, foods, Success*, and *Failure*, respectively. For each user's intention, one can capture POMDP observations given each utterance $W = \{w_1, \ldots, w_{|W|}\}$ using vector β. Notice that we have already captured the probability of each word given each user's intention in $\beta_{w_i z}$ in time of model learning. Then, in dialogue POMDP interaction, given any arbitrary user's utterance POMDP observation o is captured as:

$$o = argmax_z \prod_i \beta_{w_i z} \qquad (1)$$

Then, the observation function is estimated by taking average over belief of states given each action and state.

As stated earlier, in HTMM, the special form of the transition matrix enables a reduced time complexity of $O(|d|N^2)$, where $|d|$ is the length of the dialogue d, and N is the number of desired user's intentions, given to the algorithm [5]. The small time complexity of the algorithm enables the dialogue



(a)

(b)

Fig. 4: (a): Comparison of performance in dialogue POMDPs v.s. experts with respect to the number of expert dialogues. (b): Comparison of performance in dialogue POMDPs v.s. experts with respect to the noise level.

system to apply it at any time to update the observation function based on its recent observations. In fact, the algorithm took real time (less than a second) to converge on the authors machine with 1.7 MHz cpu and 4 gigabytes memory. This fact suggests use of the algorithm after finishing some tasks by the system to learn new states, observations, and hopefully a better strategy.

For the choice of reward model, similar to previous works we penalized each action in primary states by -1, i.e. -1 reward for each dialogue turn [15]. Moreover, actions in *Success* state get $+50$ as reward, and those which lead to *Failure* state get -50 reward. The choice of success reward as 50 is because the largest successful expert dialogue in training data includes 50 turns (actions). As the final reward is the absorb state reward minus number of turns, we chose 50 to avoid negative reward for successful dialogues.

V. Experiments

We generated dialogue POMDP models as described in the previous section for SACTI-1. The automatic generated dialogue POMDP models consist of 5 states, 14 actions and 5 meta observations (each of which is for one state) which are drawn by HTMM using 817 primitive observations (words).

We solved our POMDP models, using ZMDP software available online at: http://www.cs.cmu.edu/~trey/zmdp/. We set a uniform distribution on 3 primary states (*visits, transports, and foods*), and set discount factor to 90%. Based on simulation, we evaluated the performance of dialogue POMDP by increasing the number of expert dialogues based on the gathered rewards.

Figure 4 (a) shows that by increasing expert dialogues the dialogue POMDP models perform better. In other words, by increasing data the introduced method learns better dialogue POMDP models. The only exception is when we use 48 dialogues where the dialogue POMDP performance decreases compared to when 24 dialogues were used, and it has average performance worse than performance of experts in corresponding 48 dialogues. The reason could be use of EM for learning the model which is depended on priors α and η. Moreover, EM is prone to local optima. In this work, we set the priors based on heuristic given in [6], and our trial and error experiments, which is indeed a drawback for use of parametric models in real applications.

Furthermore, based on our simulations, we evaluated the robustness of generated POMDP models to ASR noise. There are four levels of ASR noise: no noise, low noise, medium noise, and high noise. For each noise level, we randomly took 24 expert dialogues and made a dialogue POMDP model. Then, for each POMDP we performed 24 simulations and gathered their expected rewards, and compared to corresponding expert dialogues. Figure 4 (b) shows the results of these experiments. As the figure shows the dialogue POMDP models are more robust to ASR noise levels compared to expert dialogues. The only exception is with the presence of no noise, where the experts perform better. This also might be because of use of EM for learning model, where the model can converge in local minima. Nevertheless, our preliminarily results based on simulation shows that dialogue POMDP models are much more robust to higher levels of noise compared to expert performance.

Moreover, Table II shows a sample dialogue from SACTI-1 dialogue set after applying HTMM on dialogues. In fact, this is a sample of data used for learning dialogue POMDP model. The first line of the table shows the first user's utterance ($U1$). Because of ASR this utterance is corrupted which is the following line in braces, $U'1$. The next line $o1$ is the observation behind $U'1$ which is used in the time of dialogue POMDP interaction. Note that it is assumed that each user utterance corresponds to one user's intention. So, for each system's observation the values in the following line show the system's belief over possible hidden intentions ($B1$). The next line, $a1$ shows the DM's action in the form of dialogue acts. For instance, *Inform(foods)* is the dialogue act for the actual DM's utterance in the following line, i.e. *M1: cafe blu is on alexander street.*

Furthermore, Table III shows samples of our simulation of dialogue POMDP. In the simulation time, for instance action $a1$, *GreetingFarewell* is generated by dialogue POMDP manager, the description of this action is shown in $M1$, *How can I help you?*. Then, the observation $o2$ is generated by environment, VO. For instance, the received user's utterance could have been something like *U'1=I would like a hour there museum first*, which easily the intention behind this can be calculated using β_{ws} and equation 1. However, notice that these results are only based on dialogue POMDP simulation; where there is no actual user's utterance, but only simulated meta observations o_i. As the table shows, dialogue POMDP performance seems intuitive. For instance, in $a4$ the dialogue POMDP requests for acknowledgement that the user actually looks for *transports*, since dialogue POMDP already informed the user about *transports* in a_3.

VI. Discussion

A common problem in dialogue POMDP frameworks is finding the dialogue POMDP policy. If we can estimate the POMDP model in particular the transition, observation, and reward functions then we are able to use common dynamic programming approaches for finding POMDP policies. In this context, [15] used POMDPs for modelling a DM and defined the observation function based on confidence score which is in turn based on some recognition features. However, the work here is tackled differently. We consider all the words in an utterance and consider the highest intention under the utterance as the meta observation for the POMDP.

Similar to [3], in this work, we used user's intentions as POMDP states. However, here we are interested in modelling realistic transition and observation functions based on real dialogues and considering all the words in the user's utterances as observations, as opposed to [3], as well as [15]. In this way, we consider all the words in an utterance as observations at the time of model learning. These observations represent one intention, however, at the time of POMDP interaction, given a user's utterance the underlying user's intention is estimated using the learned intention model, and is used as the meta observation in the observation function of the dialogue POMDP. This allows us to be able to deal with larger spoken dialogue domains, where we need to reduce the observations for the sake of applicability of POMDPs.

In [11], the authors also defined a similar dialogue POMDP model and used transcribed data and EM algorithm for learning part of the observation function i.e. the probability of user's utterance given user's intention and system's action. However, their defined model is slightly different from ours, in particular in that they require a model of ASR which we abstract away from it here. This is because of the fact that our model basically is defined for slightly different domains where the user is looking for information for some topics i.e. user's intentions in a small domain. In [13], the author introduced a generative model for learning ASR-N-best list, i.e., given user's utterances, the probability of received utterances and their recognized features. This model can also be considered as part of an observation function in dialogue POMDP modelling, however, it needs annotating dialogues.

On the other hand, our model is unsupervised and requires unannotated dialogues. Notice that the work presented here differs from all other mentioned works, in terms of minimizing the manual work for learning a POMDP model in early design of POMDP dialogue managers. In fact, our method is able to generate dialogue POMDP states, transition, and observation functions only based on some unannotated dialogues. However, the work presented here is only a preliminarily work towards learning a complete POMDP model from unannotated data, and there are many directions for future work which is presented in the following section.

VII. Conclusion and Future Work

In this work, we applied Hidden Topic Markov Model for learning the components of dialogue POMDP model from unannotated dialogues of SACTI-1 corpus. The learned model includes states, observations, as well as transition and observation functions. Our preliminary experiments show that the quality of the learned model increases by increasing the dialogues as training data. Moreover, our simulation based experiments show that the learned models are robust to ASR noise level.

However, the evaluation done here is in a rather small domain for real dialogue systems. The number of states needs to be increased and the learned model should be evaluated accordingly. Moreover, the definition of states here is a simple intention state whereas in real dialogue domains the information or dialogue states are more complex. Then, the challenge would be to compare in particular the learned observation function presented here with confidence score based ones such as in in [15], as well as keyword based ones as presented in [3]. Then, one may try to learn the possible actions also from the dialogue set. There are other manual work which is worth of future research. First, the number of available intentions in the dialogue set was chosen based on trial and error. Then, the description of each learned topic was done manually here, which can be done in an algorithmic way. Moreover, learning the reward model remains a challenge. In our current work, we manually set the immediate rewards. As a matter of fact, capturing proper reward model is an open problem in inverse reinforcement learning in general [7]. We are also interested in learning rewards for state action pairs based on expert dialogues, which is a future direction for this work.

TABLE II: Sample results of applying HTMM on SACTI-1

	...
$U1$	yeah hello this is johan schmulka uh and i'm uh searching for a bar in this town can you may be tell me where the cafe blu is
U'1	[hello this is now seven four bus and do you tell me where to cafe blu is]
$o1$	*FO*
$B1$	*t:0.000000 v:0.000000 f:1.000000*
$a1$:	*Inform(foods)*
$M1$	cafe blu is on alexander street
$U2$	oh um yeah how can i get to alexander street and where exactly is it i know there a shopping area on alexander street um
U'2	[i am yeah i am at the alexander street and where is it was on a the center of alexander street]
$o2$	*TO*
$B2$	*t:0.999992 v:0.000008 f:0.000000*
$a2$:	*Inform(transports)*
	...

TABLE III: Sample results of simulation for SACTI-1 dialogue POMDP

	...
$a1$:	*GreetingFarewell*
$M1$:	How can I help you?
$o2$:	*VO*
$B1$:	*t:0.048145 v:0.912760 f:0.039093*
$a2$:	*Inform(visits)*
$M2$:	Here is information about visiting areas
$o2$:	*TO*
$B2$:	*t:0.967322 v:0.008186 f:0.024490*
$a3$:	*Inform(transports)*
$M3$:	*Here is information about transportation*
$o3$:	*TO*
$B3$:	*t:0.993852 v:0.000314 f:0.005833*
$a4$:	*ReqAck(transports)*
$M4$:	*Are you looking for transportation*
$o4$:	*TO*
$B4$:	*t:0.945658 v:0.048333 f:0.006008*
$a5$:	*Inform(transports)*
$M5$:	Here is information about transportation

REFERENCES

[1] Hamid R. Chinaei, Brahim Chaib-draa, and Luc Lamontagne. Learning user intentions in spoken dialogue systems. In *Proceedings of the International Conference on Agents and Artificial Intelligence (ICAART)*, pages 107–114, Porto, Portugal, January 2009.

[2] Kenneth Ward Church. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of conference on Applied Natural Language Processing (ANLP)*, pages 136–143, Morristown, NJ, USA, 1988.

[3] Finale Doshi and Nicholas Roy. Spoken language interaction with model uncertainty: an adaptive human-robot interaction system. *Connection Science*, 20(4):299–318, 2008.

[4] Thomas Griffiths and Jark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Science*, 101:5228–5235, January 2004.

[5] Amit Gruber and Ashok Popat. Notes regarding computations in openhtmm, 2007.

[6] Amit Gruber, Michal Rosen-Zvi, and Yair Weiss. Hidden topic markov models. In *Artificial Intelligence and Statistics (AISTATS)*, San Juan, Puerto Rico, 2007.

[7] Andrew Ng and Stuart Russell. Algorithms for Inverse Reinforcement Learning. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*, pages 663–670, 2000.

[8] Luis E. Ortiz and Leslie Pack Kaelbling. Accelerating EM: An empirical study. In *Proceedings of the fifteenth conference on Uncertainty in Artificial Intelligence (UAI '99)*, pages 512–521, Stockholm, Sweden, 1999.

[9] Joelle Pineau, Geoffrey Gordon, and Sebastian Thrun. Point-based value iteration: An anytime algorithm for pomdps. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1025 – 1032, August 2003.

[10] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[11] Umar Syed and Jason D. Williams. Using automatically transcribed dialogs to learn user models in a spoken dialog system. In *Proceedings of annual meeting of the association for computational linguistics on Human Language Technologies, (HLT)*, pages 121–124, Morristown, NJ, USA, 2008.

[12] Karl Weilhammer, Jason D. Williams, and Steve Young. The SACTI-2 Corpus: Guide for Research Users, Cambridge University. Technical report, 2004.

[13] Jason D. Williams. Exploiting the ASR N-best by tracking multiple dialog state hypotheses. *Proc ICSLP, Brisbane, Australia*, 2008.

[14] Jason D. Williams and Steve Young. The SACTI-1 Corpus: Guide for Research Users. Cambridge University Department of Engineering. Technical report, 2005.

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