POLICY COMMITTEE FOR ADAPTATION IN MULTI-DOMAIN SPOKEN DIALOGUE SYSTEMS

M. Gašić, N. Mrkšić, P-H. Su, D. Vandyke, T-H. Wen and S. Young

Cambridge University Engineering Department, Trumpington St, Cambridge CB2 1PZ, UK

ABSTRACT

Moving from limited-domain dialogue systems to open domain dialogue systems raises a number of challenges. One of them is the ability of the system to utilise small amounts of data from disparate domains to build a dialogue manager policy. Previous work has focused on using data from different domains to adapt a generic policy to a specific domain. Inspired by Bayesian committee machines, this paper proposes the use of a *committee* of dialogue policies. The results show that such a model is particularly beneficial for adaptation in multi-domain dialogue systems. The use of this model significantly improves performance compared to a single policy baseline, as confirmed by the performed real-user trial. This is the first time a dialogue policy has been trained on multiple domains on-line in interaction with real users.

***Index Terms*—** Bayesian committee machines, Gaussian processes, reinforcement learning

1. INTRODUCTION

Statistical approaches to dialogue management have been shown to reduce design costs and provide superior performance to hand-crafted systems particularly in noisy environments [1]. Traditionally, spoken dialogue systems were built for limited domains described by an underlying *ontology*, which is essentially a structured representation of the database of entities that the dialogue system can talk about.

The semantic web is an effort to organise a large amount of information available on the Internet into a structure that can be more easily processed by a machine designed to perform reasoning about this data [2]. *Knowledge graphs* are one instance of such structures. They typically consist of a set of triples, where each triple represents two entities connected by a specified relationship. Current knowledge graphs have millions of entities and billions of relations and are constantly growing. There has been a significant amount of work in spoken language understanding focused on exploiting knowledge graphs in order to improve coverage [3, 4]. More recently there have been some initial attempts to build statistical dialogue systems that operate on large knowledge graphs, but limited so far to the problem of belief tracking [5]. In this work, we address the problem of decision-making.

Moving from a limited domain dialogue system that operates on a relatively modest ontology to an open domain dialogue system that can converse about anything in a large knowledge graph is a non-trivial problem. An open domain dialogue system can be seen as a (large) set of limited domain dialogue systems. If each of them were trained separately then an operational system would require sufficient training data for every possible topic in the knowledge graph, which is simply not feasible. What is more likely is that instead we have limited and varied data drawn from different domains. In this paper, we present a dialogue model particularly well suited for this problem.

The architecture of a statistical dialogue system typically provides a single policy model that proposes actions throughout the dialogue [1]. This has so far also been the case for multi-domain systems [6]. Multi-policy models have been proposed in the context of hierarchical modelling, where the decision-making process follows a hierarchy of policies, but at any given time only one policy makes a decision [7]. Concurrent policy models have been studied in [8] where several policies can propose an action at any given time and heuristics are used to decide which policy should be followed. Combining outputs of multiple policies was previously studied only in the context of combining statistical and hand-crafted policies [9, 10]. Previous work on multi-domain dialogue systems has proposed a distributed architecture where a generic policy can be trained on data coming from different domains and later specialised to provide in-domain performance once sufficient data becomes available [11]. In this paper, we enhance this model to operate with multiple policies to further improve the performance when the data is limited and comes from very different domains.

We propose a *policy committee* model, based on a *Bayesian committee machine* (BCM) [12], which consists of a number of policies trained on different, potentially small, datasets. At any given time, when the system needs to make a decision, it consults each committee member and they each propose an action. A data-driven combination method is then used to reach the consensus. The idea behind this is that each committee member can be trained on different datasets and therefore the expertise of different members varies, so collaboratively they can reach a better decision. We examine two committee architectures. In the first, the committee al-

978-1-4799-7291-3/15/$31.00 ©2015 IEEE

ways consists of two members: *generic* – trained on data coming from a variety of domains and *specific* – trained only on in-domain data. In the second, each domain has a separate committee member trained only on data for that domain. We examine a number of methods for combining committee members' decisions.

The experiments are presented within the context of Gaussian process reinforcement learning (GPRL) [13, 14], as this method not only provides the estimate of the objective function, but also the confidence in its estimate. This information is essential for combining committee members' decisions in a data-driven manner. Another reinforcement learning method with this property to which the policy committee could be applied is Kalman temporal difference reinforcement learning [15].

The main contributions of the proposed policy committee model are:

1. Efficient use of data for building statistical multi-domain dialogue systems;

2. significant improvements in performance compared to the traditional one-policy model; and

3. a flexible architecture which allows for both adding and removing committee members, which is particularly useful for extending multi-domain systems to new unseen domains.

The rest of the paper is organised as follows. In Sections 2 and 3 we review Gaussian process reinforcement learning and the Bayesian committee machine respectively. Following that, in Section 4, we describe the multi-domain dialogue manager. Section 5 presents the experimental set-up, followed by an evaluation of several committee combination models using a simulated user (Section 6) and real users (Section 7). We conclude the paper in Section 8 with a summary and future work directions.

2. GAUSSIAN PROCESS REINFORCEMENT LEARNING

The input to a statistical dialogue manager is typically an N-best list of scored hypotheses obtained from the spoken language understanding unit. Based on this input, at every dialogue turn, a distribution of possible dialogue states called the *belief state*, an element of *belief space* $\boldsymbol{b} \in \mathcal{B}$, is estimated. The quality of a dialogue is defined by a *reward function* $r(\boldsymbol{b}, a)$ and the role of a dialogue policy π is to map the belief state $\boldsymbol{b}$ into a system action, an element of *action space* $a \in \mathcal{A}$, at each turn so as to maximise the expected cumulative reward.

The expected cumulative reward for a given belief state $\boldsymbol{b}$ and action a is defined by the Q-function:

$$Q(\boldsymbol{b}, a) = E_\pi\left(\sum_{\tau=t+1}^{T} \gamma^{\tau-t-1} r_\tau | b_t = \boldsymbol{b}, a_t = a\right) \tag{1}$$

where r_τ is the immediate reward obtained at time τ, T is the dialogue length and γ is a discount factor, $0 < \gamma \leq 1$. Optimising the Q-function is then equivalent to optimising the policy π.

GP-Sarsa is an on-line reinforcement learning algorithm that models the Q-function as a Gaussian process [13]:

$$Q(\boldsymbol{b}, a) \sim \mathcal{GP}\left(0, k((\boldsymbol{b}, a), (\boldsymbol{b}, a))\right) \tag{2}$$

where the kernel $k(\cdot, \cdot)$ is factored into separate kernels over belief and action spaces $k_{\mathcal{B}}(\boldsymbol{b}, \boldsymbol{b}')k_{\mathcal{A}}(a, a')$.

For a training sequence of belief state-action pairs $\boldsymbol{B} = [(\boldsymbol{b}^0, a^0), \ldots, (\boldsymbol{b}^t, a^t)]^\intercal$ and the corresponding observed immediate rewards $\boldsymbol{r} = [r^1, \ldots, r^t]^\intercal$, the posterior of the Q-function for any belief state-action pair $(\boldsymbol{b}, a)$ is given by:

$$Q(\boldsymbol{b}, a)|\boldsymbol{r}, \boldsymbol{B} \sim \mathcal{N}(\overline{Q}(\boldsymbol{b}, a), cov((\boldsymbol{b}, a), (\boldsymbol{b}, a))) \tag{3}$$

where the posterior mean and covariance take the form:

$$\begin{aligned} \overline{Q}(\boldsymbol{b}, a) &= \boldsymbol{k}(\boldsymbol{b}, a)^\intercal \boldsymbol{H}^\intercal (\boldsymbol{H}\boldsymbol{K}\boldsymbol{H}^\intercal + \sigma^2 \boldsymbol{H}\boldsymbol{H}^\intercal)^{-1} \boldsymbol{r}, \\ cov((\boldsymbol{b}, a), (\boldsymbol{b}, a)) &= k((\boldsymbol{b}, a), (\boldsymbol{b}, a)) - \\ &\quad \boldsymbol{k}(\boldsymbol{b}, a)^\intercal \boldsymbol{H}^\intercal (\boldsymbol{H}\boldsymbol{K}\boldsymbol{H}^\intercal + \sigma^2 \boldsymbol{H}\boldsymbol{H}^\intercal)^{-1} \boldsymbol{H}\boldsymbol{k}(\boldsymbol{b}, a) \end{aligned} \tag{4}$$

where $\boldsymbol{k}(\boldsymbol{b}, a) = [k((\boldsymbol{b}^0, a^0), (\boldsymbol{b}, a)), \ldots, k((\boldsymbol{b}^t, a^t), (\boldsymbol{b}, a))]^\intercal$, $\boldsymbol{K}$ is the Gram matrix [16], $\boldsymbol{H}$ is a band matrix with diagonal $[1, -\gamma]$ and σ^2 is an additive noise factor which controls how much variability in the Q-function estimate is expected during the learning process. Since the Gaussian process for the Q-function defines a Gaussian distribution for every belief state-action pair (3), when a new belief point $\mathbf{b}$ is encountered, for each action $a \in \mathcal{A}$, there is a Gaussian distribution over Q-values. Sampling from these Gaussian distributions gives Q-values $\hat{Q}(\mathbf{b}, a) \sim \mathcal{N}(\overline{Q}(\mathbf{b}, a), \Sigma^Q(\boldsymbol{b}, a))$ where $\Sigma^Q(\boldsymbol{b}, a) = cov((\boldsymbol{b}, a), (\boldsymbol{b}, a))$ from which the action with the highest sampled Q-value can be selected:

$$\pi(\mathbf{b}) = \arg\max_a \left\{\hat{Q}(\mathbf{b}, a) : a \in \mathcal{A}\right\}. \tag{5}$$

The likelihood of the sampled $\hat{Q}$ value is given by:

$$\mathcal{L}(\hat{Q}) = \frac{1}{\sqrt{2\pi\Sigma^Q(\boldsymbol{b}, a))}} \exp\left(\frac{-|\overline{Q}(\boldsymbol{b}, a) - \hat{Q}|^2}{2\Sigma^Q(\boldsymbol{b}, a)}\right). \tag{6}$$

To use GPRL for dialogue, a kernel function must be defined on both the belief state space $\mathcal{B}$ and the action space $\mathcal{A}$. Here we use the Bayesian Update of Dialogue State (BUDS) dialogue model [17]. The action space consists of a set of slot-dependent and slot-independent summary actions which are mapped to master actions using a set of rules and the kernel is defined as:

$$k_{\mathcal{A}}(a, a') = \delta_a(a') \tag{7}$$

where $\delta_a(a') = 1$ iff $a = a'$, 0 otherwise. The belief state consists of the probability distributions over the Bayesian network hidden nodes that relate to the dialogue history for each slot and the user goal for each slot. The dialogue history nodes can take a fixed number of values, whereas user goals range over the values defined for that particular slot in the ontology and can have very high cardinalities. User goal distributions are therefore sorted according to the probability assigned to each value since the choice of summary action does not depend on the values but rather on the overall shape of each distribution. The kernel function over both dialogue history and user goal nodes is based on the expected likelihood kernel [18], which is a simple linear inner product. The kernel function for belief space is then the sum over all the individual hidden node kernels:

$$k_{\mathcal{B}}(\boldsymbol{b}, \boldsymbol{b}') = \sum_h \langle \boldsymbol{b}_h, \boldsymbol{b}'_h \rangle \tag{8}$$

where $\boldsymbol{b}_h$ is the probability distribution encoded in the h^{th} hidden node.

3. BAYESIAN COMMITTEE MACHINE

The Bayesian committee machine is an approach to combining estimators that have been trained on different datasets and can be applied to Gaussian process regression [12]. Here we apply this method to combine the outputs of multiple estimates of Q-values Q_i with mean $\overline{Q}_i$ and covariance Σ_i^Q, given by Eq. 4 and trained on a set of rewards and belief-state action pairs $\boldsymbol{r}_i, \boldsymbol{B}_i$ for $i \in \{1, \ldots, M\}$, where M is the number of policies in the policy committee.

Following the description in [12], the combined mean $\overline{Q}$ and covariance Σ^Q are calculated as:

$$\begin{aligned} \overline{Q}(\boldsymbol{b}, a) &= \Sigma^Q(\boldsymbol{b}, a) \textstyle\sum_{i=1}^M \Sigma_i^Q(\boldsymbol{b}, a)^{-1} \overline{Q}_i(\boldsymbol{b}, a), \\ \Sigma^Q(\boldsymbol{b}, a)^{-1} &= -(M-1) * k((\boldsymbol{b}, a), (\boldsymbol{b}, a))^{-1} \\ &\qquad + \textstyle\sum_{i=1}^M \Sigma_i^Q(\boldsymbol{b}, a)^{-1}. \end{aligned} \tag{9}$$

Fig 1 depicts a Bayesian committee machine consisting of three estimators.

4. MULTI-DOMAIN DIALOGUE MANAGER

In this work we assume that the spoken language understanding unit, the belief tracker and the natural language generator can deal with multiple domains and we examine how to define a policy model that can support multiple domains. Previous work [11] has introduced the notion of a *generic* policy, which can be trained from data coming from different domains, and a *specific* policy that can be derived from a generic policy using additional in-domain data. In order to produce a generic policy that works across multiple domains, a kernel function must be defined on belief states and actions that come from different domains. In [11] this is done by finding an appropriate mapping between slots. Namely, portions of the belief relating to shared slots are directly mapped to each other and for slots which are different the mapping is defined manually.

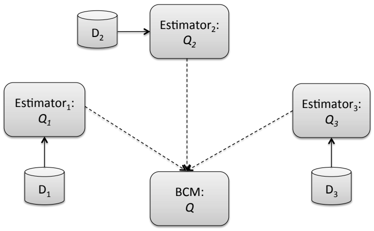

Fig. 1. Bayesian committee machine: Committee members consist of estimators trained on different datasets D_i. At every turn their estimated Q-values, Q_i, are combined to propose the final Q-value estimate.

Here we take a different approach for building policies that can operate (and be trained on) belief states and actions that come from different domains. The approach is as follows

1. For each slot in every domain calculate the *normalised entropy* η given by

$$\eta(s) = -\sum_{v \in \mathcal{V}_s} \frac{p(s = v) \log(p(s = v))}{|\mathcal{V}_s|}, \tag{10}$$

   where s is a slot that takes values v from a set $\mathcal{V}_s$ and where $p(s = v)$ is the empirical probability that an entity in the database with slot s takes value v for that slot. For example, if all entities in the database for the restaurant domain have allowedforkids=False, then that slot has a normalised entropy 0. The measure is normalised so that slots that take different numbers of values can be compared. This measure provides an indicator of how useful each slot is in the dialogue. For instance, in this case it is not useful for the system to ask the user about their preference for slot allowedforkids since the answer provides no information gain.

2. For each domain, sort slots based on their normalised entropy and give them abstract names $slot_1, slot_2, \ldots$ so that $\eta(slot_i) \geq \eta(slot_j)$ for $i \leq j$.

3. When calculating the kernel function between belief states and actions which come from different domains $\mathcal{M}$ and $\mathcal{N}$, for each i try to match portions of belief states and actions related to $slot_i^{\mathcal{M}}$ from domain $\mathcal{M}$ to $slot_i^{\mathcal{N}}$ in domain $\mathcal{N}$, if both domains have the same slot $slot_i$. Otherwise, disregard the portion of the belief

state relating to $slot_i$ and if one of the actions relates to $slot_i$, consider the action kernel to be 0.

This approach has three important properties:

- it does not require human intervention to define the relationship between different domains;
- it provides a well-defined computable relationship between any two domains; and
- the kernel function that is defined in step 3 is positive definite so the Gaussian process is well-defined.

5. EXPERIMENTAL SET-UP

In order to examine the ability of the proposed method to operate on multiple domains we examine four domains: SFR consisting of restaurants in San Francisco, SFH consisting of hotels in San Francisco, L6 consisting of laptops with 6 properties that the user can specify and L11 which is the same as L6 but with 11 user-specifiable properties. A description of each domain with slots sorted according to their normalised entropy is given in Table 1.

Table 1. Slots for each domain. The upper half represents slots that can be specified by the user to constrain a search. They are ranked according to normalised entropy (Eq. 10). The remainder are informable slots which can be queried by the user.

SFR	SFH	L6	L11
name	name	name	name
allowedforkids	allowedforkids	isforbusiness	isforbusiness
pricerange	pricerange	batteryratings	batteryrating
near	near	pricerange	pricerange
goodformeal	takescreditcards	draverange	draverange
food	hasinternet	weightrange	weightrange
area	area	family	family
-	-	-	platform
-	-	-	utility
-	-	-	processorclass
-	-	-	sysmemory
addr	addr	price	weight
price	phone	drive	battery
phone	postcode	dimension	price
postcode	-	-	dimension
-	-	-	drive
-	-	-	display
-	-	-	graphadaptor
-	-	-	design
-	-	-	processor

6. SIMULATION RESULTS

In order to investigate the effectiveness of the proposed policy committee model, a variety of contrasts were examined using an agenda-based simulated user operating at the dialogue act level [19, 20]. The reward function allocates -1 at each turn to encourage shorter dialogues, plus 20 at the end of each successful dialogue. The user simulator includes an error generator and this was set to generate incorrect user inputs 15% of time.

The contrasts studied were as follows:

INDOM **In-domain policy** – trained only on in-domain data, other data is not taken into consideration, action-selection is based only on the in-domain policy. This is the baseline.

GEN **Single generic policy** – one policy trained on all available data. This approach was proposed in [11].

2CQ **Two-policy committee with Q-values** – one generic policy trained on all available data and one specific policy trained only on in-domain data. Action selection depends directly on each Q-value i.e. the action that has the highest Q-value between the two policies is taken.

2CQL **Two-policy committee with Q-values and likelihood** – same as 2CQ but the Q values are scaled by the likelihood (Eq. 6) for action selection.

2BCM **Two-policy Bayesian committee machine** – same as 2CQ but uses a Bayesian committee machine described in Section 3 to provide the consensus estimate of the Q-value for action selection.

MBCM **Multi-policy Bayesian committee machine** – same as 2BCM but has one committee member for each domain. Each committee member is trained only on in-domain data. However, for action-selection, the estimates of all committee members are taken into account via a Bayesian committee machine (Eq 9), both during training and testing.

GOLD **Gold standard** – this is the performance of the single policy where all training data comes from the same domain i.e. for N domains, GOLD has N times the number of in-domain dialogues for training as provided to INDOM.

We examine two cases: when the training data is limited, with only 250 dialogues available for each domain, and when there is more training data available, 2500 for each domain. A previous study [11] considered domains which are relatively similar. Here, we consider two set-ups:

- Multi-domain system for SFR, SFH and L6, where the domains have different slots but each domain has the same number of slots, and

- Multi-domain system for SFR, SFH and L11, where not only are the slots different, but also one of the domains, L11, has many more slots than the others.

For each method described above, 10 policies were trained on the simulated user using different random seeds. Each policy was then evaluated using 1000 dialogues on each domain. The overall average reward, success rate and number of turns are given in Table 2 together with 95% confidence intervals.

There are several important conclusions to be drawn from the results given in Table 2. First, as shown in [11], generic policies make use of data that comes from different domains and this improves performance over an in-domain baseline, even in the case presented here where the domains are very different. Second, having a policy committee that only uses the Q-value estimate for action-selection (2CQ) degrades performance compared to the GEN policy. However, taking into account both the Q-value and the likelihood (2CQL) improves performance and this is consistent across domains. This finding emphasises the importance of maintaining second order information during Q-value estimation. Finally, while the two-policy Bayesian committee machine 2BCM gives somewhat inconsistent results across the domains, the multi-policy MBCM results in performance which is either significantly better than other methods or statistically indistinguishable from other methods. In the case of limited training data, its performance is at least as good as the gold standard. Another advantage of MBCM is that it does not require storing a separate generic policy model but only ever produces in-domain models that have the ability to contribute to action-selection for other domains.

Unlike other multi-policy models, MBCM allows flexible selection of committee members. The usefulness of each committee member in the MBCM multi-policy model is explored in Table 3 for the SFR domain. As can be seen from the results, all committee members contribute to performance gains. However not all committee members are equally important. In this case, for good performance on the SFR domain, the SFH committee member is more useful than the L11 committee member.

Table 2. Comparison of strategies for multi-domain adaptation. In-domain performance is measured in terms of reward, success rate and the average number of turns per dialogue. Results are given with 95% confidence intervals.

Strategy	Reward	Success	#Turns
L6 trained on 750 dialogues from SFR, SFH, L6			
INDOM	7.92 ± 0.20	72.64 ± 0.87	6.56 ± 0.07
GEN	9.34 ± 0.19	79.43 ± 0.80	6.49 ± 0.06
2CQ	8.95 ± 0.20	78.91 ± 0.80	6.77 ± 0.07
2CQL	9.69 ± 0.18	81.87 ± 0.76	6.65 ± 0.07
2BCM	10.22 ± 0.17	84.30 ± 0.71	6.62 ± 0.07
MBCM	9.89 ± 0.18	82.95 ± 0.74	6.68 ± 0.07
GOLD	9.25 ± 0.19	80.35 ± 0.79	6.77 ± 0.07
L6 trained on 7500 dialogues from SFR, SFH, L6			
INDOM	10.62 ± 0.16	86.04 ± 0.68	6.50 ± 0.06
MBCM	11.60 ± 0.14	90.32 ± 0.58	6.42 ± 0.06
GOLD	11.98 ± 0.13	92.36 ± 0.53	6.42 ± 0.06
SFR trained on 750 dialogues from SFR, SFH, L11			
INDOM	5.73 ± 0.21	68.17 ± 0.92	7.89 ± 0.08
GEN	6.32 ± 0.21	72.04 ± 0.89	8.05 ± 0.08
2CQ	5.73 ± 0.21	70.13 ± 0.90	8.24 ± 0.08
2CQL	6.78 ± 0.21	75.36 ± 0.85	8.26 ± 0.09
2BCM	6.46 ± 0.22	73.80 ± 0.87	8.25 ± 0.09
MBCM	7.37 ± 0.20	76.60 ± 0.83	7.92 ± 0.08
GOLD	7.34 ± 0.20	76.97 ± 0.83	8.01 ± 0.08
SFR trained on 7500 dialogues from SFR, SFH, L11			
INDOM	9.03 ± 0.17	85.16 ± 0.70	7.97 ± 0.08
MBCM	9.67 ± 0.17	88.28 ± 0.66	7.96 ± 0.08
GOLD	9.65 ± 0.16	88.80 ± 0.62	8.05 ± 0.08
L11 trained on 750 dialogues from SFR, SFH, L11			
INDOM	6.46 ± 0.22	67.59 ± 0.92	7.02 ± 0.08
GEN	7.18 ± 0.21	70.91 ± 0.89	6.97 ± 0.08
2CQ	6.91 ± 0.21	70.15 ± 0.90	7.09 ± 0.08
2CQL	7.24 ± 0.22	72.24 ± 0.88	7.17 ± 0.08
2BCM	6.55 ± 0.23	69.11 ± 0.91	7.20 ± 0.09
MBCM	8.52 ± 0.20	77.09 ± 0.82	6.88 ± 0.07
GOLD	8.68 ± 0.20	77.26 ± 0.83	6.74 ± 0.07
L11 trained on 7500 dialogues from SFR, SFH, L11			
INDOM	10.05 ± 0.17	84.58 ± 0.71	6.84 ± 0.07
MBCM	10.73 ± 0.16	87.23 ± 0.66	6.70 ± 0.07
GOLD	11.17 ± 0.15	88.89 ± 0.62	6.57 ± 0.06

Table 3. Selection of committee members for multi-policy Bayesian committee machine for SFR domain.

MBCM – SFR			
Committee members	Reward	Success	#Turns
SFR	7.32 ± 0.22	79.97 ± 0.82	8.51 ± 0.10
SFR+SFH	9.20 ± 0.18	86.51 ± 0.70	8.05 ± 0.09
SFR+L11	8.73 ± 0.19	84.56 ± 0.73	8.12 ± 0.09
SFR+SFH+L11	9.67 ± 0.17	88.28 ± 0.66	7.96 ± 0.08

7. REAL USER EVALUATION

In order to fully examine the effectiveness of the proposed adaptation scheme, policies were also trained in direct interaction with human users. We compare two set-ups: one where an in-domain L6 policy is trained on-line and another where a multi-policy Bayesian committee machine is trained from scratch using data from the SFR, SFH and L6 domains, which produces a policy committee which can operate on all three domains. We deployed the system in a telephone-based set-up, with subjects recruited via Amazon MTurk and given prescribed dialogue tasks to complete in a similar set-up to [11]. At the end of each dialogue, a recurrent neural network (RNN) model was used to predict the dialogue success used as the reinforcement feedback [21]. For each contrast, three instances were trained and the results were averaged.

Fig. 2 shows the moving average reward as a function of the number of training dialogues for the L6 domain comparing the in-domain (INDOM) policy and the multi-policy Bayesian committee machine (MBCM) as defined in Section 6. The performance of the MBCM policy was only shown on training dialogues that came from the L6 domain, but in fact it was also trained on SFR and SFH domains in parallel. The training data across the domains was equally distributed. The moving window was set to 100 dialogues so that after the initial 100 dialogues each point on the graph is an average of 300 dialogues. The shaded area represents a 95% confidence interval. The initial parts of the graph exhibit more randomness in behaviour because the number of training dialogues is small. The results show that the multi-policy Bayesian committee machine consistently outperforms the in-domain policy. To the best of our knowledge, this is the first time a dialogue policy has been trained on multiple domains on-line in interaction with real users.

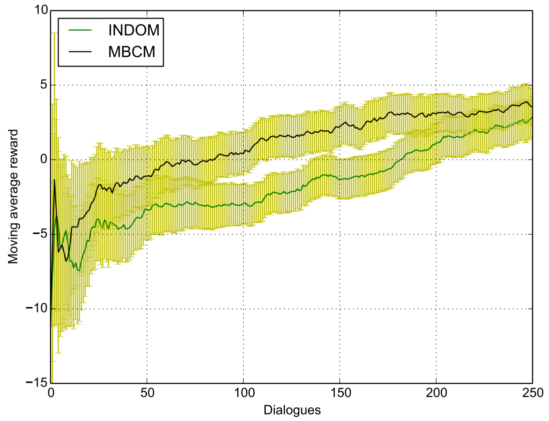

Fig. 2. Training in interaction with human users on L6 domain – moving average reward

8. CONCLUSIONS AND FUTURE WORK

We have presented a policy committee model which uses estimates from different policies for action selection at every dialogue turn. We have demonstrated that this model is particularly useful for training multi-domain dialogue systems where the data is limited and varied. As shown in both simulations and a real user trial, the Bayesian policy committee approach gives superior performance to the traditional one-policy-approach across multiple domains and allows flexible selection of committee members during testing.

In the future, we plan to apply this method to a large knowledge graph. This requires investigating useful committee members for a given topic in the knowledge graph. We have already shown that not all committee members are equally useful for a given domain. Also, the computational cost is linearly dependent on the number of committee members, which is an added incentive carefully choose the committee members for any given topic in the knowledge graph.

9. ACKNOWLEDGEMENTS

The research leading to this work was funded by the EPSRC grant EP/M018946/1 ”Open Domain Statistical Spoken Dialogue Systems”. The data used in these experiments is available at https://www.repository.cam.ac.uk/handle/1810/251169.

10. REFERENCES

[1] SJ Young, M Gašić, B Thomson, and JD Williams, “Pomdp-based statistical spoken dialogue systems: a review,” *Proceedings IEEE*, vol. 101, no. 5, pp. 1160–1179, 2013.

[2] Péter Szeredi, Gergely Lukácsy, and Tamás Benkö, *The Semantic Web Explained: The Technology and Mathematics Behind Web 3.0*, Cambridge University Press, New York, NY, USA, 2014.

[3] Gökhan Tür, Minwoo Jeong, Ye-Yi Wang, Dilek Hakkani-Tür, and Larry P Heck, “Exploiting the semantic web for unsupervised natural language semantic parsing,” in *Proceedings of Interspeech*, 2012.

[4] Larry P Heck, Dilek Hakkani-Tür, and Gökhan Tür, “Leveraging knowledge graphs for web-scale unsupervised semantic parsing.,” in *Proceedings of Interspeech*, 2013, pp. 1594–1598.

[5] Yi Ma, Paul A. Crook, Ruhi Sarikaya, and Eric Fosler-Lussier, “Knowledge graph inference for spoken dialog systems,” in *Proceedings of 40th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2015*. April 2015, IEEE Institute of Electrical and Electronics Engineers.

[6] Z. Wang, H. Cheng, G. Wang, H. Tian, H. Wu, and H. Wang, “Policy learning for domain selection in an extensible multi-domain spoken dialogue system,” in *EMNLP*, 2014.

[7] Heriberto Cuayáhuitl, Steve Renals, Oliver Lemon, and Hiroshi Shimodaira, “Evaluation of a hierarchical reinforcement learning spoken dialogue system,” *Comput. Speech Lang.*, vol. 24, no. 2, pp. 395–429, Apr. 2010.

[8] Pierre Lison, “Multi-policy dialogue management,” in *Proceedings of the SIGDIAL 2011 Conference*, Stroudsburg, PA, USA, 2011, SIGDIAL ’11, pp. 294–300, Association for Computational Linguistics.

[9] Jason D. Williams, “The best of both worlds: Unifying conventional dialog systems and pomdps,” in *Proc Interspeech, Brisbane, Australia*, 2008.

[10] M. Gašić, F. Lefevre, F. Jurcicek, S. Keizer, F. Mairesse, B. Thomson, K. Yu, and S. Young, “Back-off action selection in summary space-based pomdp dialogue systems,” in *Automatic Speech Recognition Understanding, 2009. ASRU 2009. IEEE Workshop on*, Nov 2009, pp. 456–461.

[11] M. Gašić, D. Kim, P. Tsiakoulis, and S. Young, “Distributed dialogue policies for multi-domain statistical dialogue management,” in *Proceedings of ICASSP*, 2015.

[12] Volker Tresp, “A Bayesian Committee Machine,” *Neural Comput.*, vol. 12, no. 11, pp. 2719–2741, Nov. 2000.

[13] Y Engel, S Mannor, and R Meir, “Reinforcement learning with Gaussian processes,” in *Proceedings of ICML*, 2005.

[14] M Gašić and S Young, “Gaussian Processes for POMDP-Based Dialogue Manager Optimization,” *TASLP*, vol. 22, no. 1, 2014.

[15] M Geist and O Pietquin, “Managing Uncertainty within the KTD Framework,” in *Proceedings of the Workshop on Active Learning and Experimental Design*, Sardinia (Italy), 2011.

[16] CE Rasmussen and CKI Williams, *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, Massachusetts, 2005.

[17] B Thomson and S Young, “Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems,” *Computer Speech and Language*, vol. 24, no. 4, pp. 562–588, 2010.

[18] T Jebara, R Kondor, and A Howard, “Probability product kernels,” *J. Mach. Learn. Res.*, vol. 5, pp. 819–844, Dec. 2004.

[19] J Schatzmann, B Thomson, K Weilhammer, H Ye, and SJ Young, “Agenda-Based User Simulation for Bootstrapping a POMDP Dialogue System,” in *Proceedings of HLT*, 2007.

[20] S Keizer, M Gašić, F Jurčíček, F Mairesse, B Thomson, K Yu, and S Young, “Parameter estimation for agenda-based user simulation,” in *Proceedings of SIGDIAL*, 2010.

[21] Pei-Hao Su, David Vandyke, Milica Gašić, Dongho Kim, Nikola Mrkšić, Tsung-Hsien Wen, and Steve Young, “Learning from real users: Rating dialogue success with neural networks for reinforcement learning in spoken dialogue systems,” in *Proceedings of Interspeech*, 2015.