Reinforcement Learning for Personalized Dialogue Management

Floris den Hengst
ING Groep NV
Floris.den.Hengst@ing.com

Mark Hoogendoorn
Vrije Universiteit Amsterdam
M.Hoogendoorn@vu.nl

Frank van Harmelen
Vrije Universiteit Amsterdam
Frank.van.Harmelen@vu.nl

Joost Bosman
ING Groep NV
Joost.Bosman@ing.com

ABSTRACT

Language systems have been of great interest to the research community and have recently reached the mass market through various assistant platforms on the web. Reinforcement Learning methods that optimize dialogue policies have seen successes in past years and have recently been extended into methods that *personalize* the dialogue, e.g. take the personal context of users into account. These works, however, are limited to personalization to a single user with whom they require multiple interactions and do not generalize the usage of context across users. This work introduces a problem where a generalized usage of context is relevant and proposes two Reinforcement Learning (RL)-based approaches to this problem. The first approach uses a single learner and extends the traditional POMDP formulation of dialogue state with features that describe the user context. The second approach segments users by context and then employs a learner per context. We compare these approaches in a benchmark of existing non-RL and RL-based methods in three established and one novel application domain of financial product recommendation. We compare the influence of context and training experiences on performance and find that learning approaches generally outperform a handcrafted gold standard.

CCS CONCEPTS

• **Computing methodologies** → **Reinforcement learning**; *Discourse, dialogue and pragmatics*; • **Human-centered computing** → *Natural language interfaces*;

KEYWORDS

Reinforcement Learning; Dialogue Management; Personalization; Adaptive Virtual Assistants; Recommendation

1 INTRODUCTION

The use of language by machines has been one of the central challenges in Artificial Intelligence since its initiation as a field of research [30] [19]. Decades of research have advanced the state of art to such an extent that major consumer-facing web platforms currently offer text- and voice-based 'assistant' capabilities, such as Tencent's WeChat, Microsoft's Cortana, Google's Assistant etc. These platforms have made access to the web through dialogue ordinary. Although such platforms offer high-quality Automatic Speech Recognition (ASR), Natural Language Understanding (NLU) and audio synthesis modules, Dialogue Management (DM) modules are typically handcrafted and require many non-trivial decisions in design and implementation. *Learned* DM based on the formalism of Partially Observable Markov Decision Processes (POMDPs) has shown promising results in task-oriented dialogue systems, both in simulation and real-life settings [25] [8] [35].

Personal context is understood to be fundamental to efficient human-human communication [4]. As a consequence, recent works have addressed the usage of personal context in DM. For example, [29], [18] and [20] used previous interactions with a user to directly estimate that users' preferences and then used these estimates in policy optimization. An alternative approach based on transfer learning was presented in [6]. It requires a similarity metric and weighting regime and performance degrades when these are not available. None of these methods generalize the usage of context across users and none of them leverages information available prior to some users' first interaction with the system.

We propose two approaches that optimize the DM policies using personal context. Both approaches are based on the POMDP formalism of learned DM. The first approach consists of extending the POMDP state space with features that describe the personal context of the user. The DM module automatically learns how to use this information for both groups, i.e. it learns the task at hand and segmentation of users simultaneously. This approach allows for personalization to emerge gracefully, e.g. only when enough data is present and when the user model is sufficiently informative for personalization. We compare this approach with a method that explicitly segments users and then uses a learner per user segment. The segmentation of interactions with different user groups mitigates the issue of a 'mixed' signal but leaves less experiences to learn from per learner.

To test our approaches, we extend an existing benchmark for POMDP-based statistical DM for recommendation in three ways [5]. Firstly, we add a novel recommendation task in the financial domain. Here, different user groups have different familiarity with

products and specify their preferences at different levels of detail as a result. Secondly, we change the user simulator in the benchmark to reflect this scenario. Thirdly, we add three non-POMDP based approaches to the benchmark: a randomized approach, an approach with a task-specific heuristic and a state-of-art approach based on entropy minimization [34]. To the best of our knowledge, this comparison between POMDP and non-POMDP based approaches on task-oriented dialog management is novel.

We use the extended benchmark to investigate when each approach is suitable for personalized DM and we investigate the impact of available data to the achieved level of personalization. We first introduce and formalize the recommendation task in Section 2 and survey related work in Section 3. Next, we introduce the generic approach to RL for DM and then introduce our extensions. The experimental setup consists of recommendation in existing and novel domains, a user simulator for personalized DM and a benchmark of POMDP and non-POMDP algorithms, is introduced in Section 5. After describing and analyzing the results in Section 6, we conclude with a discussion in Section 7.

2 TASK DESCRIPTION

This work addresses DM in task-oriented dialogue systems. These systems aim to solve a task by interacting with the user in a conversational style. A popular task for these systems is to recommend a suitable item for a user. The system elicits user preferences or constraints during a dialogue and recommends items from a given item database. We introduce this task formally.

The task addressed in this paper can be formalized as a q-ary two-player interactive search game [23]. In these game, the goal of one player, dubbed Questioner, is to find a target subset $X_{target} \subseteq X = \{x_1, \ldots, x_n\}$ out of a universe of items X of size n by asking questions to the other player, the Responder. In this case, each $x_i \in X$ consists of a vector of values $\langle x_{i1}, \ldots, x_{im} \rangle$ for features $\{f_1, \ldots, f_m\}$. X_{target} is identified by a set of constraints C, in the form of the desired value c_j for some feature f_j. We assume $\forall c_j \in C, \forall x_i \in X_{target}, \forall x_{ij} \in x_i : x_{ij} = c_j$. Each c_j eliminates a part of the search space. We use C_t to denote the set of constraints at game turn t and X_{C_t} to denote the corresponding candidate item set.

Both the typical q-ary search game and our variation are generalizations of the Rényi-Ulam game (RU game), also known as the binary search game or the parlour game '20 questions'. In RU games, Questions are limited to confirmation of a single constraint, i.e. they are all of the form '$c_j \in C$?' In this format, the optimal question halves the candidate item set X_{C_t} in the optimal case. In our setting, however, the optimal decrease in candidate item set size depends on the distribution of values for all f_j's in X_{C_t}. The Questioner may use knowledge about these distributions in selecting a f_j to ask a constraint for. We therefore include a policy that uses knowledge about the distribution of values in all f_j's as a search heuristic. More so, the Responders' tendency to provide constraints for a feature f_j may not be distributed uniformly in realistic settings. A Questioner with access to past plays may use this experience to estimate the likelihood of a constraint for a feature being present to find an item more efficiently. We therefore include approaches that can leverage experience into our benchmark, see Section 5.3 for details.

3 RELATED WORK

Most approaches to personalizing dialogue systems can be categorized as learning-based or rule-based. We provide a brief overview of approaches in both categories. An example of a rule-based approach can be found in [10] and [29]. This system uses a model of user preferences for constraints c_j to weigh factors that determine similarity of a user query to the items in X. The DM policy is hand-crafted, which typically entails many nontrivial decisions that can seriously impact system performance [16]. More recent examples, such as [14], [3] and [27] collect user-related facts in a knowledge graph. These facts are then used to personalize hand-crafted response templates. These approaches focus on personalized natural language generation and have handcrafted DM modules.

Learning-based approaches, on the other hand, optimize the DM policy using experiences with real or simulated users. A conversational shopping recommender is described in [18]. It requires multiple interactions with a specific user and has a query-response interaction style. An example with a natural language interaction style based on transfer learning can be found in [6]. It initializes a policy for the target user by training on data from interactions with similar users. The authors find that it is beneficial to include data from dissimilar users, albeit with lower weights, as this results in better coverage of the state space during training. A drawback of the approach is that it requires a suitable similarity metric. A transfer learning-based approach that does not suffer from this drawback is introduced in [20]. A policy is optimized using a global optimization criterion and all available experiences. Next, the optimization criterion is extended with user-specific slot-value preference estimates which are updated in subsequent interactions. This approach only adapts to individual users in terms of slot-value preferences and requires multiple interactions with a single user. A third transfer learning-based approach is presented in [9]. The selection of experiences to train the model on for a specific user is cast as a multi-armed bandit problem. Finding a source of experiences out of all n users, however, requires at least n bandit trials. This limits applicability to scenarios with a small number of users.

None of the approaches discussed so far leverage information external to the conversation, e.g. context, to optimize the dialogue policy. In non-conversational recommendation, however, numerous works rely on the users' personal contexts. As a full survey is out of scope for this paper, so we focus on generic trends instead. Recommender systems are typically classified as content-based, collaborative filtering or a hybrid of these two. Content-based recommender systems 'exploit the user profile to suggest relevant items by matching the profile representation against that of items to be recommended' and thus rely on the users' personal context [22]. Collaborative filtering selects items for recommendation by looking at past consumption patterns by similar users and personal context can be used to determine similarity of users [13] [17] [1]. Out of these approaches, contextual bandit methods are specifically related to this work. These methods aim to determine how elements of personal context affect relevance of items through subsequent interactions with users [15]. These methods, however, are not suitable for conversational settings as they do not take sparsity of rewards and the sequential nature of these settings into account.
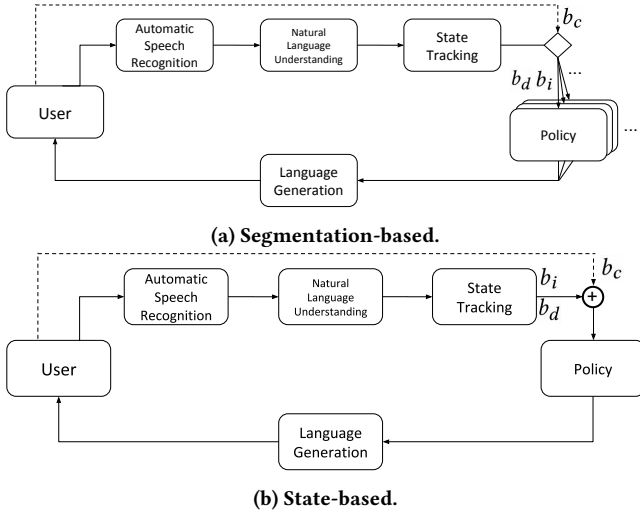
(a) Segmentation-based.

(b) State-based.

Figure 1: RL-based approaches to personalized DM.

4 APPROACH

This section describes two novel approaches to personalized DM for the interactive recommendation task described in Section 2. First, the formalism of Partially Observable Markov Decision Problems is described and it is explained how it it can be applied to DM for the interactive recommendation task.

4.1 RL for DM

State of the art statistical dialogue systems cast DM as a Partially Observable Markov Decision Problem (POMDP) [25] [33]. A POMDP is a generalization of a Markov Decision Process where the true state is not directly observable, but must be estimated through observations. In dialogue systems, the source of uncertainty about the true state stems from errors in Automatic Speech Recognition (ASR) and Natural Language Understanding (NLU) modules. The POMDP is defined as $M = \langle S, A, T, R, \Omega, O \rangle$ where $S \in \{s_1, \ldots, s_n\}$ denotes a finite set of partially observable states representing user intentions and dialogue history, $A \in \{a_1, \ldots, a_m\}$ is a finite set of actions representing system responses, $T \colon S \times A \times S \to [0, 1]$ is a probabilistic transition function over states and $R \colon S \times A \to \mathbb{R}$ denotes a reward function based on number of turns and accuracy of recommendation, $\Omega \in \{o_1, \ldots, o_l\}$ is a finite set of observations available to the system, and $O \colon \Omega \times A \times S \to [0, 1]$ denotes a probabilistic function over observations, actions and states. The true state s is unavailable to the agent, only observations Ω are.

The dependence of O on Ω and A makes the decision process non-Markovian and thus unsuitable for standard RL algorithms. The Markovian property can be regained, however, by maintaining a Bayesian belief over S and substituting the original state space with this belief space. This substitution leaves us with a continuous MDP with an input space $B \in \{b_1, \ldots, b_o\}$ with dimensionality $|S| - 1$, which is too complex for most practical purposes. In practice, however, the belief space can be significantly reduced in size by splitting it into factors and assuming mutual independence between factors. In dialogue systems aimed at the interactive recommendation task

from Section 2, the belief space can be split into a factored belief space B' consisting of dialogue history belief b_d and a user intention belief b_i. The dialogue history b_d describes, for example, whether the system has already recommended an item x_i or requested a constraint for feature f_j. The user intention belief describes preferences of the user w.r.t. the product database. Maintaining this state is a challenge in itself, but outside of the scope of this work. See [32] and [12] for overviews. As B is replaced by B' and not used anymore, we denote B' as B from here on.

Constructing the POMDP involves some design decisions based on the task at hand. Specifically, A should contain actions that are useful or necessary for the agent to achieve its task. For the interactive recommendation task the agent plays the part of Questioner. The available utterances should thus at least reflect requesting a constraint for each feature f_j and recommending an item. Additional actions can make the dialogue more natural and efficient, such as confirmation questions of the form '$c_j \in C$?' and selection questions of the form '$c_j \in C$ or $c_{j'} \in C$?'.

Besides a suitably defined A, the POMDP should be constructed with an R that reflects the goal of the task at hand. This work is based on a benchmark further described in Section 5. In the benchmark R is defined as

$$20 * acc(X_{target}, \langle a^1, \ldots, a^l \rangle) - l \tag{1}$$

for a given X_{target} and trajectory of system actions $\langle a^1, \ldots, a^l \rangle$ of length l. $acc()$ returns 1 if the trajectory contains a recommendation action for an item $x_i \in X_{target}$ and 0 otherwise. The goal is to find the optimal function $\pi^* \colon B \to A$ that maximizes the expected sum of discounted future rewards

$$\pi^*(b) = \arg\max_a Q^{\pi^*}(b, a), \forall b \in B, \forall a \in A \tag{2}$$

where

$$Q^{\pi^*}(b, a) = E_{\pi^*}\left\{ \sum_{k=0}^{\infty} \gamma^k r^{t+k+1} | b^t = b, a^t = a \right\} \tag{3}$$

and $\gamma \in [0, 1]$ is a factor weighing future rewards and b^t and a^t are future beliefs and actions.

4.2 Personalized Dialogue Management

We present two approaches to DM using personal context of the user based on the formalism described. Figure 1 provides an overview of the two methods. Both use a vector describing the agents' belief of *personal context* b_c of the user to optimize the dialogue for specific users. This may include any available information about the user that may aid in policy optimization. Examples of context include demographics, purchase history and previous interactions. Note that context need not be constant during or in between dialogues. This section describes how context is used in both methods.

The method in Figure 1a is based on segmentation of the user population by context. It assumes a function $M \colon B_c \to G$ that maps agent beliefs on user contexts $B_c \in \{b_{c_1}, b_{c_2}, \ldots, b_{c_n}\}$ to segments $g \in G$ (g for 'group'). A separate policy $\pi_g(b_d, b_i)$ is maintained that exclusively interacts with contexts b_c for which $M(b_c) = g$. As the policy interacts with user contexts in a single segment, it learns a policy optimal for that segment using only beliefs on dialogue history b_d and user intentions b_i. The context b_c is not available

to the policy. A benefit of this approach is the absence of negative transfer between segments: behaviors suitable to only a particular segment of users are only learned by that segments' policy and will not be considered suitable by policies serving the other segments. On the other hand, there cannot be any positive transfer either: each policy is exposed to less interactions which may result in poor belief state space coverage and degraded performance. Furthermore, it may be nontrivial to find a suitable segmentation function M as this involves finding an unambiguous context representation and determining the number of segments.

The method in Figure 1b does not suffer from these drawbacks. It consists of concatenating beliefs on dialogue history b_d, user intentions b_i and context b_c. The resulting belief vector is then used as input to a single policy $\pi_p(b_d, b_i, b_c)$ for the entire user population. An algorithm that optimizes π_p now jointly learns DM and the usage of context therein. This allows for the learner to only use context when it is beneficial and liberates us from defining segmentation or similarity criteria. The composed learning task, however, may be significantly more challenging as users from different segments may have conflicting desires. This might lead to a form of negative transfer that the algorithm optimizing π_p has to be robust to which may require more training data.

Domain	# Items	Group 1 & 2	Group 2 only
CR	110	price range	area, food
SFR	271	price range, allowed for kids, good for meal	area, near, food
LAP	123	utility, price range, weight range, warranty, is for business computing	family, processor class, sys memory, platform, drive range, battery rating
FIN	14	minimum age, purpose, account	name, insurance, max. duration, min. duration, max. principal, min. principal

Table 1: Usage of slots for constraints for the two user groups. Group 1 denotes users unfamiliar with the domain or 'laypersons' while Group 2 denotes users experienced in the domain or 'experts'. Expert users always use three constraints, whereas layperson users have between one and three constraints.

5 EXPERIMENTAL SETUP

The goal of this paper is to evaluate the proposed approaches for personalized dialogue management. We split this goal into the following research questions. In a personalized DM task,

Q1 when do learning-based algorithms outperform handcrafted algorithms?

Q2 when do belief state-based approaches outperform segmentation-based approaches?

Q3 how well do existing approaches generalize to the novel domain of financial product recommendation?

Regarding these research questions, we hypothesize:

H1 learning-based approach only outperform handcrafted approaches in the presence of preprocessing errors.

H2 belief state-based approaches perform comparable to or better than segmentation-based approaches.

H3a in the new domain, learning-based approaches perform comparable to existing domains.

H3b in the new domain, handcrafted approaches perform worse than in existing domains.

The experimental setup is based on a benchmark suite for task-oriented dialog management [5]. The suite includes a user simulator, a dialog management module and DM algorithms. The benchmark further consists of recommendation tasks in three domains: recommendation of restaurants in Cambridge (CR), of restaurants in San Francisco (SFR) and laptops (LAP), we refer to [5] for details. We extend this benchmark in three ways. Firstly, we add a new domain of recommending financial products. Secondly, we extend the user simulator to include context. Finally, we add our proposed algorithms and additional non-POMDP-based algorithms to the benchmark.

5.1 Recommending Financial Products

The financial domain is an interesting addition as it is different from domains currently in the benchmark: the number of interactions with a single user is typically limited, there may be large gaps in between interactions and user intentions are typically not constant over interactions. It is, for example, unlikely that a single customer needs multiple recommendations based on an intention to finance a car purchase. This renders approaches that require multiple interactions with a single user or that rely on direct estimation of user preferences inapplicable.

A second particularity of this domain is that different users have different familiarity with products. As a result, users in this domain have differing preferences and ability to express them. For example, customers that have a car loan will be more familiar with technicalities of secured loans and therefore be more capable of expressing their preferences for similar loans in detail. Such differences are common in domains with complex products, such as the financial, technology and automotive domains. Although the exact formulation of context is not the focus of this work and may vary per domain, we consulted with domain experts in the financial domain on contextual factors currently used in determining how to communicate with users across various channels. These domain experts indicate that one of the major factors in communicating about a product is whether the user consumes a product from the same product category.

Differences with other domains are not limited to typical interaction patterns, however: the item set X is distinctive in this novel setting as well. This item set was developed using using well-known ontology engineering practices and evaluated with domain experts [21] [2]. The resulting item set consists of 14 products and 13 features. Nine out of these can be used as a constraint by the user, see Table 1 for an overview. All other slots are only used to

| | RQ | Entropy | | POMDP | | | |
		EMDB	EMDM	HDC	RL_v	RL_s	RL_{bs}
Task-specific		v					
NLU/DST-error aware				v	v	v	v
Adaptive			v		v	v	v
Uses context						fixed	adaptive

Table 2: Overview of qualities of approaches. RL_v, RL_s and RL_{bs} describe the vanilla, segmentation-based and belief-state based versions of *GP*, *A2C*, *DQN* and *eNAC*.

inform the user about the product and not relevant to the recommendation task. The number of values for all constraint features is 64. When compared to the existing domains in literature, the novel FIN domain has a relatively small item set and relatively large number of constraint-slots. We add this item set as an 'ontology' to the Pydial benchmark for DM systems [5] which is described in the next Sections in more detail.

5.2 User Simulator

We adapt the user simulator in the benchmark as described in [26] to reflect the scenario from the previous section. A full description of this simulator is out of scope and we limit ourselves to the main concepts before moving on to the extensions. In the simulator, actions by the simulated user are conditioned on the dialogue so far and on behavior parameters and includes an error model for ASR and NLU modules. Parameters for all of these have been tuned using data from experiments with real users, for details see [26]. Behavior parameters are sampled at the start of each dialogue and according to distributions that have been set in user profiles so that each dialogue is with a user with individual behavior characteristics. Similarly, up to three constraints c_j are sampled randomly for each new simulated user. Additionally, heuristics to constrain the action space can be enabled or disabled. These *action masks* make part of the action space unavailable and ease the learning task. A combination of user model, error model and availability of action mask is denoted as an 'environment'. In total, the benchmark we use includes six different environments [5].

We extend the tuned simulator with user context to reflect the scenario from the previous section. Two user groups are modelled. The first group represents 'laypersons' that express constraints for specific slots only; the second group represents knowledgeable users that express constraints for all slots. All slots and their usage per group are listed in Table 1. The usage of slots between groups for the FIN domain has been set after consultation with domain experts. For the CR, SFR and LAP domains, these are set to allow for a comparison of approaches across settings.

We add a b_c to describe the user context and add per-slot constraint usage parameters to the simulator. Specifically, b_c is a vector of two values, describing the belief on the user having experience in the domain or not. Although our approach facilitates a wide range of values, we here limit ourselves to the case of fully certain upfront knowledge, i.e. $b_c \in \{0, 1\}^2$. We assume that interactions with both types of users are equally likely.

5.3 Algorithms

We evaluate our approach using all algorithms in the benchmark presented in [5] and measure per-dialogue rewards according to equation 1 in Section 4.1 across 10 random seeds with 4000 training and 500 test dialogues each. The benchmark contains one hand-crafted policy, *HDC*, and four RL-based algorithms: *GP* for GP-SARSA, *A2C*, *eNAC* and *DQN*. All of these algorithms are based on the POMDP formalism introduced in Section 4.1. *GP* is a data-efficient nonparametric value-based approach that uses Gaussian Processes to estimate $Q^\pi(b, a)$ from equation 3 [8]. *DQN* similarly estimates these Q values using a neural network, i.e. it is a parametric approach [28] [31]. *A2C* and *eNAC* are parametric algorithms that estimate the policy $\pi(b)$ as defined in equation 3 directly, where *A2C* estimates $Q(b, a)$ additionally [7]. We refer to [5] for more detail on these algorithms. We include vanilla versions of the learning algorithms, versions based on segmentation and versions based on an altered belief-state and denote these by $_v$, $_s$ and $_{bs}$ subscripts respectively.

We further extend the benchmark with three non-RL-based algorithms.[1] The algorithms were selected based on the task formalization of Section 2 and to enable a comparison of learning algorithms versus handcrafted algorithms. Specifically, we add a randomized baseline, an algorithm with a search heuristic and a state-of-art learning method from [34]. This last method keeps a history of successful dialogues as trajectories of user utterances u and system actions $\langle u^1, a^1, ..., u^\ell, a^\ell \rangle$ up to a successful recommendation a^ℓ. During a dialogue $\langle u^1, a^1, ..., u^t \rangle$, the system selects the action a^t that minimizes the entropy of all past successful recommendations a^ℓ, breaking ties with a random selection. We denote this approach with *EMDM* for 'Entropy Minimization Dialog Management'.

The two remaining non-POMDP-based algorithms are a randomized baseline and a baseline that uses information about the product database. The randomized baseline randomly asks for constraints on feature f_j until there are no differentiating features in X_{C_t} and then recommends some item $x_i \in X_{C_t}$ randomly. We denote this baseline with *RQ* for 'Random Question'. The second baseline has the same strategy for recommending an item, but differs in selecting f_j. Given the current X_{C_t}, it selects the f_j with the highest entropy in the candidate item set X_{C_t} and requests the user preference for it. This is a task-specific approach that uses a entropy as a heuristic to search the item set X_{C_t} efficiently. We denote this benchmark as *EMDB* for 'Entropy Minimization DataBase'. All non-POMDP-based approaches, i.e. *RQ*, *EMDM* and *EMDB*, have no way of dealing with errors from the ASR and NLU modules in Figure 1. The output of these modules with the highest confidence score is simply assumed as correct and used as input to these algorithms.

5.4 Environment and Hyperparameters

All experiments were run on Intel Xeon Silver 4110 Processors using Python version 2.7.9, TensorFlow version 1.12.0, NumPy version 1.15.4 and SciPy version 1.2.0. Ten different random seeds ranging from zero to ten were used. Hyperparameters were set as in [5], we repeat them here. For the *GP* algorithm, a linear kernel was used on the state space and a Kronecker delta kernel was used on the

---

[1]Code: https://bitbucket.org/florisdenhengst/pydial/commits/tag/web-intelligence-19

action space. The 'scale' variable of these determines the rate of exploration and was set to 3.

DQN, *A2C* and *eNAC* use an ϵ-greedy exploration strategy during training where ϵ is linearly scaled between ϵ_s and 0.05 in training, i.e. for the 4,000 dialogues. Exploration was turned off during evaluation. See Table 3 for values of ϵ_s and network architecture for the neural network based approaches. For these, the architecture consisted of three layers of fully connected feedforward of varying sizes. The Adam optimizer was used for training with an initial learning rate of 0.001. We refer to the code repository for further details on the hyperparameters.

	# Nodes		
Model	Hidden Layer 1	Hidden layer 2	ϵ_s
DQN	300	100	.5
A2C	200	75	.5
eNAC	130	50	.3

Table 3: Hyperparameters for neural network based approaches.

6 RESULTS

In this section, we describe the results with respect to the research questions from Section 5. Table 4 lists all results.

Q1 Figure 2a shows the performance of the best algorithms in an environment where ASR/NLU errors are absent. According to hypothesis H1, we expected the *HDC* and *EMDB* algorithms to outperform learning algorithms. We analyse the performance of these algorithms per domain. The CR domain contains relatively little slots and groups are similar. The task-specific *EMDB* algorithm moderately outperforms learning-based approaches GP_s and DQN_s which in turn outperform the *HDC* algorithm. Moving to the FIN domain, DQN_s and GP_s outperform *HDC* due to the large difference between groups. We analyze the poor results of *EMDB* in this novel domain below (Q3). In the LAP domain, the *EMDB* algorithm performs the worst out of the selected algorithms. This domain has a large number of slots hence there is likely to be a differentiating feature f_j that will be selected according to *EMDB*. The *EMDB* algorithm thus keeps on asking for new f_j, even when the user has already listed all of their requirements. Comparing *HDC* with learning-based approaches in this domain, it performs comparable to DQN_s and GP_s. The reason for this may be that this is a relatively challenging learning task which limits the benefits of personalization. The SFR domain has a relatively large item set X and a moderate number of slots. The search heuristic of *EMDB* works as expected here and GP_s and DQN_s moderately outperform handcrafted approaches. Overall, we find that –in contrast to H1– learning-based approaches perform comparable or better than both handcrafted approaches, *even in the absence of ASR/NLU errors.*

We now compare these families of approaches in an environment with ASR/NLU errors in Figure 2b. In this setting, the gold standard *HDC* algorithm degrades more than learning approaches, further supporting the benefits of learning approaches in a scenario with different user groups. The difference can be explained by *HDC*'s response to an unclear answer for some slot: it requests the user to

confirm the most likely value as recorded by the ASR/NLU modules. Such a request will not further the dialogue if that particular slot does not contain a constraint for the user. The *HDC* algorithm does not take this into account, whereas learning approaches can adapt to the laypersons' inability to informatively respond after such a confirmation request and ask for other constraints first. The *EMDB* algorithm cannot handle uncertainty from ASR/NLU outputs. It assumes the most likely preference as indicated by ASR/NLU modules. This assumption is occasionally incorrect and generally ruins *EMDB*'s performance.

Q2 In contrast to hypothesis H2, performance of belief state- and segmentation-based personalization approaches vary across domains, environments and used learning algorithms. For the *GP* algorithm, segmentation generally outperforms vanilla and belief-state based approaches in both environments. This suggests that *GP* suffers less from lack of training data as a result of segmentation, which is in line with earlier findings that *GP* is a data efficient algorithm [8]. The performance of this algorithm relies on the chosen kernel. In the benchmark, a linear kernel is used. This kernel assumes a linear relation between $Q^\pi(b, a)$ and the belief state b. We briefly analyze this linearity assumption by considering two similar belief states b that only differ in the belief on user group membership for the current user b^c. The linearity assumption implies that some favorable action for the first group is unfavorable for the other group. This assumption clearly does not hold for some actions, e.g. requesting some f_j that is used by both groups.

For *DQN*, some negative effects of segmentation can be seen in cases with a complex learning problem, i.e. in environments with ASR/NLU errors and in domains with a large state space. These negative effects can be mainly seen in domains with larger state spaces LAP and SFR. Regarding the belief state-based approach, results indicate that it performs comparable or slightly better than the vanilla approaches in most configurations. We hypothesized that this approach would learn to exploit differences in user population without suffering from the drawback of limited training data as in the segmentation-based approach. Although our findings indicate that the latter is generally the case, the benefits of personalization diminish for more complex learning problems in environments 4-6. A possible explanation for this is that the algorithms' hyperparameters, specifically the neural network architecture for *DQN* and kernel for *GP*, were not optimized to the personalization setting.

Q3 Figure 3 shows how POMDP-based approaches hold over various domains in all included environments. We omit non-POMDP-based approaches here due to their poor performance in environments 3-6. When comparing the novel FIN domain, the gold standard *HDC* is outperformed by all considered learning algorithms. The learning algorithms generalize to the new domain. The *HDC* policy was handcrafted for the other four domains and does not transfer well to a novel domain with different characteristics. To analyze the results of *EMDB* in the FIN domain, we consider again Figure 2. In the FIN domain, the item set X is small which makes the search heuristic on which *EMDB* relies inapplicable. These results are in line with hypotheses H3a and H3b.
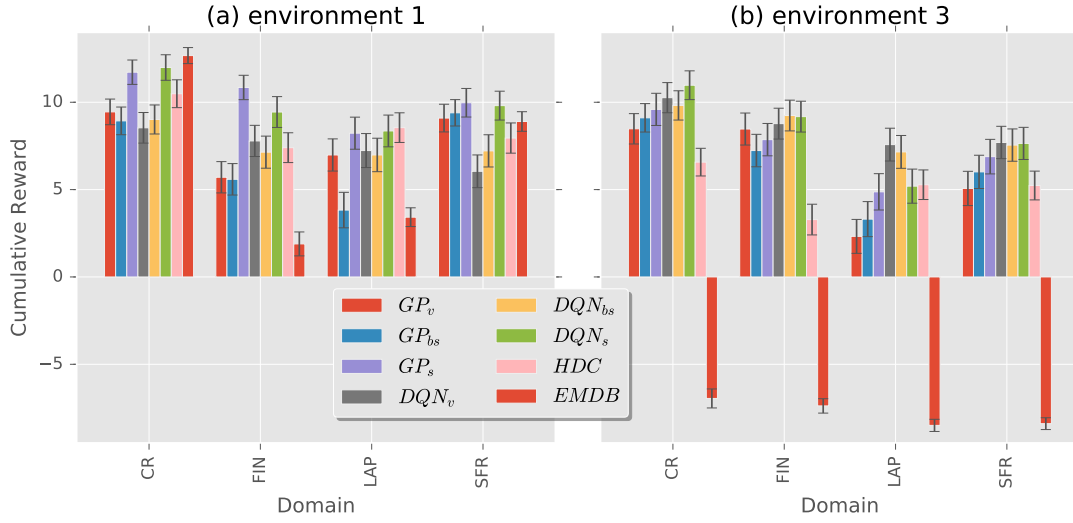
Figure 2: Average reward per dialogue in test set for environments without (a) and with (b) ASR/NLU errors.

env	Error Model	Action Masks	User Model	domain	$A2C_v$	$A2C_{bs}$	$A2C_s$	DQN_v	DQN_{bs}	DQN_s	$eNAC_v$	$eNAC_{bs}$	$eNAC_s$	GP_v	GP_{bs}	GP_s	HDC	RQ	$EMDB$	$EMDM$
1	0%	on	normal	CR	12.2	11.6	10.6	8.5	9.0	12.0	1.4	6.5	10.6	9.4	8.9	11.7	10.5	**12.7**	**12.7**	-4.7
				FIN	**10.9**	8.5	7.2	7.8	7.1	9.4	4.1	0.8	8.0	5.7	5.6	10.8	7.4	2.3	1.9	-12.3
				LAP	5.9	4.1	0.6	7.2	7.0	8.4	7.5	7.9	5.9	7.0	3.8	8.2	**8.5**	4.2	3.4	-14.0
				SFR	6.4	6.4	5.1	6.0	7.2	9.8	5.2	5.0	8.3	9.1	9.4	**10.0**	8.0	9.2	8.9	-8.8
2	0%	off	normal	CR	2.8	2.3	2.2	11.8	11.2	11.3	-4.4	-3.8	3.2	11.7	11.6	11.7	11.9	**12.7**	**12.7**	-4.7
				FIN	2.8	3.2	3.4	10.7	9.8	5.7	-3.2	-2.2	3.8	8.1	5.4	6.7	8.5	2.3	-10.0	-12.3
				LAP	-2.7	-2.4	-2.5	6.3	5.7	1.8	-3.3	-3.7	-0.1	-1.0	-0.9	-0.9	10.3	4.2	3.4	-14.0
				SFR	-0.8	0.1	-1.6	9.4	7.4	7.4	5.0	5.1	0.2	8.8	8.6	5.4	10.3	9.2	8.9	-8.8
3	15%	on	normal	CR	8.2	8.1	7.8	10.3	9.8	**11.0**	7.0	8.0	10.0	8.5	9.1	9.6	6.6	-7.4	-7.0	-5.3
				FIN	6.2	5.4	3.2	8.8	**9.2**	**9.2**	4.6	7.0	6.8	8.5	7.2	7.9	3.3	-7.8	-7.4	-12.5
				LAP	-1.3	-0.9	-2.2	**7.6**	7.2	5.2	5.7	5.5	4.6	2.3	3.3	4.9	5.3	-8.7	-8.5	-14.3
				SFR	0.8	1.1	0.1	**7.7**	7.5	7.6	6.3	7.1	4.2	5.1	6.0	6.9	5.2	-8.4	-8.4	-9.7
4	15%	off	normal	CR	2.4	2.6	1.4	**10.2**	9.5	7.1	0.9	1.7	2.9	9.6	9.7	8.9	6.6	-7.4	-7.0	-5.3
				FIN	3.3	4.2	1.3	**9.6**	7.1	4.6	-1.0	-1.0	4.3	6.4	5.2	5.4	3.3	-7.8	-7.4	-12.5
				LAP	-3.0	-3.1	-2.7	4.6	3.4	-0.1	-3.8	-0.3	-2.5	-1.1	-1.0	-1.0	**5.3**	-8.7	-8.5	-14.3
				SFR	-1.0	0.2	-1.8	5.2	**6.7**	4.3	-1.1	2.0	0.9	4.6	4.7	2.5	5.2	-8.4	-8.4	-9.7
5	15%	off	unfriendly	CR	6.6	4.6	4.8	7.0	**9.7**	8.3	4.9	7.7	7.6	7.5	8.3	8.9	6.7	-7.5	-7.5	-5.5
				FIN	2.2	2.1	1.6	6.2	**7.2**	4.1	4.4	5.5	5.1	5.3	4.6	5.6	2.5	-7.8	-7.5	-12.8
				LAP	-3.3	-2.0	-3.1	3.7	**4.1**	1.9	1.8	1.8	0.5	-0.0	-0.1	1.7	3.0	-8.6	-8.4	-14.6
				SFR	-2.1	-0.1	-1.1	**5.3**	4.6	4.6	2.3	3.3	4.1	3.8	3.6	3.5	3.7	-8.4	-8.4	-10.3
6	30%	on	normal	CR	4.2	4.2	4.8	6.4	**7.8**	7.2	6.2	7.1	7.2	6.8	7.1	7.3	5.6	-4.7	-4.7	-5.8
				FIN	0.6	0.2	0.5	3.7	3.8	3.8	3.8	4.8	**5.6**	5.2	3.5	4.9	2.5	-7.6	-7.0	-12.6
				LAP	-2.8	-2.6	-2.3	**4.9**	3.4	3.1	3.2	3.3	2.0	-2.0	-1.2	0.4	3.2	-9.3	-8.8	-14.5
				SFR	1.6	-1.8	-0.5	**5.7**	4.6	4.6	4.2	4.7	4.9	3.6	2.4	3.5	3.5	-8.3	-8.0	-9.7
		mean			2.51	2.34	1.54	7.28	7.09	6.35	2.57	3.49	4.52	5.54	5.38	6.03	6.12	-2.92	-3.37	-10.38

Table 4: Average reward per dialogue for test set across environments, domains and algorithms in the benchmark.
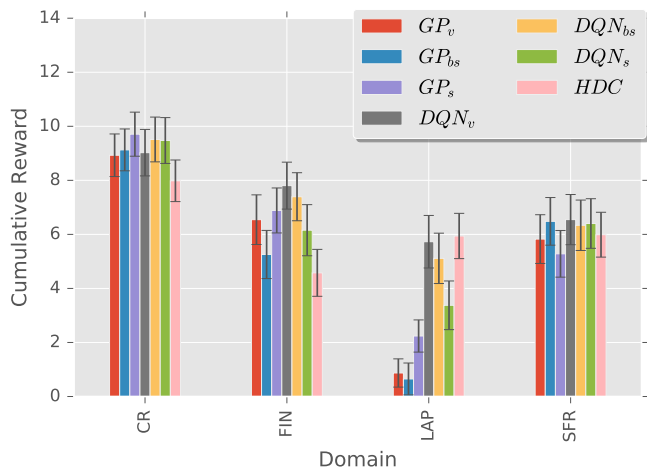
Figure 3: Per-dialogue reward of selected algorithms in test set, averaged over all environments.

7 DISCUSSION

In this work, we have proposed two approaches to DM using personal context and evaluated them on various environments, in various domains and using various algorithms. The approaches leverage existing contextual information about a particular user and can offer personalized DM even in the absence of previous interactions with a particular user.

In order to evaluate our approaches, we have extended an existing benchmark for conversational item recommendation with two user contexts and associated behavior patterns. The behavior patterns reflect those found in domains where 'expert' and 'layperson' users have differing knowledge about the available items. Results indicate that learning a dialogue policy is beneficial in settings with differing user behaviors. Notably, the addition of context boosts performance of learned dialogue managers to comparable or higher levels than a handcrafted gold standard and task-specific approaches, even in an environment without noise from preprocessing modules.

We find that performance of learning approaches varies with environment, domain, and algorithm. Specifically, data efficiency could be investigated by increasing the number of training dialogues. Similarly, the applicability of the approaches could be investigated by varying the difference between user groups. Furthermore, varying hyperparameter settings such as neural network architecture and learning rate and more powerful and stable RL algorithms may lead to more the complex behaviors in the new setting such as those in [11]. More experiments are necessary to further investigate performance characteristics for the proposed approaches.

With regards to methodology, we have introduced a case validated by domain experts in the financial domain and added it to an existing benchmark of item recommendation. We have extended a realistic user simulator with additional behavior parameters for all domains in the benchmark to comprehensively test our approaches. Although these additional parameters are suitable to test our approaches technically, they were not sampled from real-world data. Comparing the approaches in real-world settings, such as an evaluation with real users or an evaluation in a configuration where

behavior parameters are based on real-world differences between experts and laypersons would be interesting next steps.

Finally, we tested our approaches to the usage of context in a specific case with different user groups with static context information and a constant action space. Our approaches, however, are general and could be applied to various other usages of context to dialogue policy optimization. Especially interesting would be the inclusion of sentiment estimates as in [24]. Together with an extension of the action space, these could aid in making the conversation more natural by conditioning e.g. trust-building system responses on conversation content and context at the same time.

REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. In *Recommender systems handbook*, pages 217–253. Springer, 2011.
[2] Grigoris Antoniou and Frank Van Harmelen. *A semantic web primer*. MIT press, 2004.
[3] Jeesoo Bang, Hyungjong Noh, Yonghee Kim, and Gary Geunbae Lee. Example-based chat-oriented dialogue system with personalized long-term memory. In *2015 International Conference on Big Data and Smart Computing (BigComp)*, pages 238–243. IEEE, 2015.
[4] Anouschka Bergmann, Kathleen Currie Hall, and Sharon Miriam Ross. *Language files: Materials for an introduction to language and linguistics*. Ohio State University Press, 2007.
[5] Iñigo Casanueva, Paweł Budzianowski, Pei-Hao Su, Nikola Mrkšić, Tsung-Hsien Wen, Stefan Ultes, Lina Rojas-Barahona, Steve Young, and Milica Gašić. A benchmarking environment for reinforcement learning based task oriented dialogue management. In *Deep Reinforcement Learning Symposium, 31st Conference on Neural Information Processing Systems*, 2017.
[6] Inigo Casanueva, Thomas Hain, Heidi Christensen, Ricard Marxer, and Phil Green. Knowledge transfer between speakers for personalised dialogue management. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 12–21, 2015.
[7] Mehdi Fatemi, Layla El Asri, Hannes Schulz, Jing He, and Kaheer Suleman. Policy networks with two-stage training for dialogue systems. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 101–110, 2016.
[8] Milica Gašić, Filip Jurčíček, Simon Keizer, François Mairesse, Blaise Thomson, Kai Yu, and Steve Young. Gaussian processes for fast policy optimisation of POMDP-based dialogue managers. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 201–204. Association for Computational Linguistics, 2010.
[9] Aude Genevay and Romain Laroche. Transfer learning for user adaptation in spoken dialogue systems. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 975–983, 2016.
[10] Mehmet H Göker and Cynthia A Thompson. Personalized conversational case-based recommendation. In *European Workshop on Advances in Case-Based Reasoning*, pages 99–111. Springer, 2000.
[11] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1856–1865, 2018.
[12] Matthew Henderson, Blaise Thomson, and Jason D Williams. The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 263–272, 2014.
[13] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 79–86. ACM, 2010.
[14] Yonghee Kim, Jeesoo Bang, Junhwi Choi, Seonghan Ryu, Sangjun Koo, and Gary Geunbae Lee. Acquisition and use of long-term memory for personalized dialog systems. In *International Workshop on Multimodal Analyses Enabling Artificial Agents in Human-Machine Interaction*, pages 78–87. Springer, 2014.
[15] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.
[16] Diane J Litman, Michael S Kearns, Satinder Singh, and Marilyn A Walker. Automatic optimization of dialogue management. In *COLING 2000 Volume 1: The 18th International Conference on Computational Linguistics*, pages 502–508. Association for Computational Linguistics, 2000.
[17] Omid Madani and Dennis DeCoste. Contextual recommender problems. In *Proceedings of the 1st international workshop on Utility-based data mining*, pages 86–89. ACM, 2005.

[18] Tariq Mahmood, Ghulam Mujtaba, and Adriano Venturini. Dynamic personalization in conversational recommender systems. *Information Systems and e-Business Management*, 12(2):213–238, 2014.

[19] John McCarthy, Marvin L Minsky, Nathaniel Rochester, and Claude E Shannon. A proposal for the Dartmouth summer research project on artificial intelligence, august 31, 1955. *AI magazine*, 27(4):12, 2006.

[20] Kaixiang Mo, Yu Zhang, Shuangyin Li, Jiajun Li, and Qiang Yang. Personalizing a dialogue system with transfer reinforcement learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[21] Natalya F Noy, Deborah L McGuinness, et al. Ontology development 101: A guide to creating your first ontology. Technical Report SMI-2001-0880, Stanford Medical Informatics, 2001.

[22] Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007.

[23] Andrzej Pelc. Searching games with errors - fifty years of coping with liars. *Theoretical Computer Science*, 270(1-2):71–109, 2002.

[24] Soujanya Poria, Erik Cambria, Newton Howard, Guang-Bin Huang, and Amir Hussain. Fusing audio, visual and textual clues for sentiment analysis from multimodal content. *Neurocomputing*, 174:50–59, 2016.

[25] Nicholas Roy, Joelle Pineau, and Sebastian Thrun. Spoken dialogue management using probabilistic reasoning. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 93–100, 2000.

[26] Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. Agenda-based user simulation for bootstrapping a POMDP dialogue system. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 149–152, 2007.

[27] Heung-Yeung Shum, Xiao-dong He, and Di Li. From Eliza to XiaoIce: challenges and opportunities with social chatbots. *Frontiers of Information Technology & Electronic Engineering*, 19(1):10–26, 2018.

[28] Pei-Hao Su, Paweł Budzianowski, Stefan Ultes, Milica Gasic, and Steve Young. Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 147–157, 2017.

[29] Cynthia A Thompson, Mehmet H Goker, and Pat Langley. A personalized system for conversational recommendations. *Journal of Artificial Intelligence Research*, 21:393–428, 2004.

[30] A. M. Turing. Computing machinery and intelligence. *Mind*, LIX(236):433–460, 10 1950.

[31] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double Q-learning. In *Thirtieth AAAI conference on artificial intelligence*, volume 2, page 5, 2016.

[32] Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. The dialog state tracking challenge. In *Proceedings of the 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 404–413, 2013.

[33] Jason D Williams and Steve Young. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422, 2007.

[34] Ji Wu, Miao Li, and Chin-Hui Lee. An entropy minimization framework for goal-driven dialogue management. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[35] Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. POMDP-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179, 2013.