This is a post-peer-review, pre-copyedit version of an article published in **Pattern Analysis and Applications**. The final authenticated version is available online at: http://dx.doi.org/10.1007/s10044-018-0743-y".

User-Aware Dialogue Management Policies over Attributed Bi-Automata

Abstract Designing dialogue policies that take user behavior into account is complicated due to user variability and behavioral uncertainty. Attributed Probabilistic Finite State Bi-Automata (A-PFSBA) have proven to be a promising framework to develop dialogue managers that capture the users' actions in its structure and adapt to them online, yet developing policies robust to high user uncertainty is still challenging. In this paper, the theoretical A-PFSBA dialogue management framework is augmented by formally defining the notation of exploitation policies over its structure. Under such definition, multiple path based policies are implemented, those that take into account external information and those which do not. These policies are evaluated on the Let's Go corpus, before and after an online learning process whose goal is to update the initial model through the interaction with end-users. In these experiments the impact of user uncertainty and the model structural learning is thoroughly analyzed.

Keywords Dialogue Systems · User Adaptation · Attributed Bi-Automata · Dialogue Management · Path Based Policies

1 Introduction

Spoken Dialogue Systems (SDS) enable human-machine interaction using spoken language in a natural way [5]. A key task that every SDS has to carry out is controlling the logic structure of the interaction, also known as dialogue management. The Dialogue Manager (DM) is the module responsible for controlling the dialogue flow, using decision making strategies or policies. Several approaches have been proposed to model the DM statistically: Partially Observable Markov Decision Processes (POMDP) [28], Deep Learning [25, 27, 29] and Stochastic Finite-State models [8, 23]. When it comes to decision making, POMDP approaches commonly use Reinforcement Learning, applying both Monte Carlo Q-Learning [28] or Gaussian Processes [3]. Deep learning approaches, usually learn the exploitation policy with respect to a loss function [20, 25, 18] while encoding the dialogue interaction structure in a sequential fashion. Recent proposals combine Reinforcement Learning and Deep Learning, interacting with simulated users to optimize the network policy [2, 10]. The Stochastic Finite-State approach presented in [23] uses Probabilistic Finite State Bi-Automata (PFSBA) to jointly model the dialogue interaction between user and system actions. In order to encode the dialogue history through the interaction, the PFSBA states can be augmented with a discrete alphabet. This augmented model is also known as Attributed PFSBA or A-PFSBA. The A-PFSBA paradigm separates the structural learning of the dialogue interaction and its exploitation, rendering flexibility when it comes to decision making [4].

An initial exploration of the potential and flexibility of A-PFSBA was done in [19], where the inclusion of attributes in the PFSBA structure showed to improve performance. In addition, an online learning method based on successfully completed dialogues demonstrated the capability of learning from user interactions, overcoming the limitations of previously explored turn-by-turn online learning procedures [13]. Although promising results were obtained, a local Maximum Likelihood exploitation policy which did not take user behavior into account was used and path based exploitation policies were not explored. Some policies that take user behavior into account were presented in [4], but they achieved lower results than local user agnostic policies. The uncertainty of user behavior was hypothesized as the main reason for these results. Although this hypoth-

esis is rationale, it was not evaluated as inference and decision making under uncertainty is a laborious task [12, 11]. In addition, the lack of evaluation after user adaptation renders it untested.
Extending the online learning approach of the A-PFSBA framework presented in [19] to research in decision making strategies started in [4], this paper presents the following theoretical and experimental contributions:

- A formal definition of dialogue management policies over the A-PFSBA structure, extending the original definition of the framework given in [23]. This policy definition is flexible enough to encode decision making strategies both for system and user turns, as well as allowing the encoding of domain knowledge.
- Implementation of multiple path based policies that take user behavior into account. These policies are based on: (1) the Maximum Probability path, (2) a path that searches for new information slots to complete, and (3) a path based policy that exploits domain information in order to find the path that maximizes task completion.
- A thorough analysis of the implemented path based policies and user adaptation through the online learning method presented in [19]. Each policy is evaluated performing a grid search over the path length and the user-awareness ratio before and after the online learning phase. This experimentation is carried out in order to evaluate the hypothesis that policies that take user behavior into account perform worse due to user uncertainty [4].

Experiments are carried out on the Let's Go corpus [14], allowing direct comparison with previous work by [13, 19].
The paper is structured as follows: Section 2 explains spoken dialogue interaction as an stochastic process and describes the A-PFSBA formulation to model these interactions. A formal definition of exploitation policies over the A-PFSBA formulation is presented in Section 3. Section 4 introduces the experimental setup and the implemented exploitation policies and metrics. Section 5 presents the results of the experiments and their analysis. Finally, the main conclusions are summarized in Section 6, where future guidelines are also set.

2 Attributed Probabilistic Finite State Bi-Automata for Dialogue Management

This section describes spoken dialogue interaction in terms of a stochastic process that can be modeled by a Probabilistic Finite State Bi-Automata. Under such framework, a dialogue $\mathbf{z}$ can be viewed as a sequence of system and user interactions $\mathbf{z} = (a_0, f_1, \cdots, a_t, f_{t+1})$ where a are the system actions and f the user responses. As depicted in Fig. 2, each user response can be corrupted due to Automatic Speech Recognition (ASR) errors, so it is common for DMs to work on a decoded space Σ extracted from a Spoken Language Understanding (SLU) component, where each f_t is mapped to its corresponding decoding $\hat{d}_t \in \Sigma$:

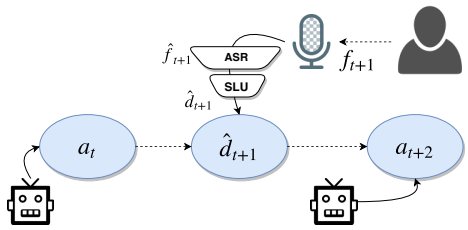

Fig. 1 Dialogue interaction as a Stochastic Process where the user response f_{t+1} is corrupted by the ASR to $\hat{f}_{t+1}$ and estimated by the SLU in $\hat{d}_{t+1}$

$$\hat{d}_t = \arg\max_{d \in \Sigma} P(d \mid f_t)$$

Then, the probability of a system action a_t given by the DM can be defined as:

$$P(a_t \mid \hat{d}_{t-1},\ a_{t-2},\ \cdots,\ \hat{d}_1,\ a_0)$$

where $\hat{d}_{t-1}$ is the estimated decoding of the user response f_{t-1}. On the other hand, the probability of an user response in a the dialogue can be defined as follows:

$$P(f_t \mid d_t) P(d_t \mid a_{t-1},\ d_{t-2}, \cdots,\ d_1,\ a_0)$$

where a_{t-1} is the system action in the previous turn. Note that there is no need to estimate the user action f_t, as it is not corrupted by any ASR error.
Instead of maintaining the whole sequence of system/user interactions, it is usual to encode the history of the dialogue until time $t-1$ in a state q_{t-1}. This way, the previous notations can be shortened to $P(a_t|\hat{d}_{t-1}, q_{t-1})$ for the system action probabilities and $P(f_t|a_{t-1}, q_{t-1})$ for the user responses. Because the A-PFSBA framework considers dialogue interaction an stochastic process of *bi-strings*, it can model user-system action tuples $(a_t, \hat{d}_{t+1})$ using an alphabet of *bi-strings* [23]. Their structure is trained by maximizing the probability of model M to generate a given sample of dialogues Z, being $\mathbf{z}$ each one of the dialogues that compose the corpus Z.

$$\hat{M} = \arg\max_M\ P_M(Z) = \arg\max_M \prod_{\mathbf{z} \in Z} P_M(\mathbf{z})$$

As described by [24] the A-PSFBA model can then be defined as $\hat{M} = (\Sigma, \Delta, \Omega, \Gamma, \delta, q_0, P_f, P)$ where

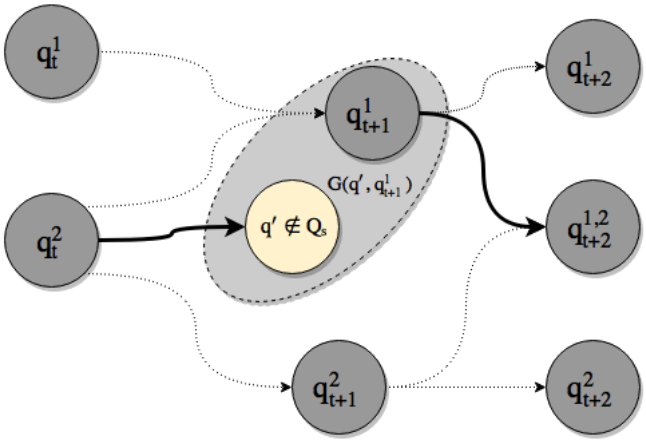

Fig. 2 Smoothing procedure where the unknown system state q' is approximated to the nearest state q'_{t+1} using the distance function G

- Σ is the alphabet of the user's decoded responses, $d \in \Sigma$.
- Δ is the alphabet of system actions, $a \in \Delta$.
- Ω is the alphabet of attributes, $\omega \in \Omega$.
- Γ is an extended alphabet $\Gamma \subseteq (\Sigma^{\geq m} \times \Delta^{\geq n})$ that contains the combinations of the user's decoded responses and system actions.
- $Q = Q_S \cup Q_U$ is the set of states labeled by *bi-strings* and attributes: $[(\tilde{d}_i : \tilde{a}_i), \Omega] \in \Gamma \times \Omega$.
  - Q_S are the system turn states.
  - Q_U are the user turn states.
- $\delta \subseteq Q \times \Gamma \times Q$ is the union of two sets of transitions $\delta = \delta_S \cup \delta_U$ as follows:
  - $\delta_S \subseteq Q_S \times \Gamma \times Q_U$ is a set of system transitions of the form $(q, (\epsilon : \tilde{a}_i), q')$ where $q \in Q_S,\ q' \in Q_U$ and $(\epsilon : \tilde{a}_i) \in \Gamma$.
  - $\delta_U \subseteq Q_U \times \Gamma \times Q_S$ is a set of user transitions of the form $(q, (\tilde{d}_i : \epsilon), q')$ where $q \in Q_U,\ q' \in Q_S$ and $(\tilde{d}_i : \epsilon) \in \Gamma$.
- $q_0 \in Q_S$ is the unique initial state: $(\epsilon : \epsilon)$ where ϵ is the empty symbol.
- $P_f : Q \rightarrow [0, 1]$ is the final-state probability distribution.
- $P : \delta \rightarrow [0, 1]$ defines the transition probability distributions $P(q, b, q') \equiv P(q',\ b \mid q)\ \forall b \in \Gamma$ and $q, q' \in Q$ such that:

$$P_f(q) + \sum_{b \in \Gamma, q' \in Q} P(q, b, q') = 1\ \forall q \in Q$$

where transition (q, b, q') is completely defined by the initial state q and the transition state b. Thus, $\forall q \in Q,\ \forall b \in \Gamma,\ |\{q' : \{(q, b, q')\}| \leq 1$

2.1 Generalization to Unseen States

As field-deployed SDS have to deal with unseen situations, it is advisable to endow the dialogue system with a backoff smoothing strategy [24], so that the system is capable of continuing with the interaction each time the user leads the dialogue to an unknown state, $q' \notin Q_S$. A common method is to use the nearest system state $q \in Q_S$ according to some distance function:

$$q = \begin{cases} q', & \text{if } q' \in Q_S \\ \min_{q \in Q_S} G(q', q), & \text{otherwise} \end{cases} \quad (1)$$

where G is the distance function that defines the relationship between the A-PFSBA states. Fig. 2 shows the previously described scenario: the user gives some unknown response in the state q_t^2 and the system is driven into an unknown state $q' \notin Q_S$. In this situation, the system searches for the closest state q_{t+1}^1 according to the distance function G and uses it to continue with the dialogue.

2.2 Dialogue Manager

Given the A-PFSBA model $\hat{M}$, a DM can be defined as a function whose goal is to return the best system action given an user response decoding, the state at the current turn under a policy Π_{DM} and a smoothing strategy with a distance function G:

$$DM_\Pi : Q \times \Sigma \rightarrow \Delta \times Q$$

$$\Pi_{DM}(q_t, d_t, \hat{M}, G) \rightarrow a_{t+1},\ q_{t+1}$$

Note that within the A-PFSBA paradigm, the structural learning of the model $\hat{M}$ is independent of its exploitation policy definition Π_{DM} or the smoothing strategies defined.

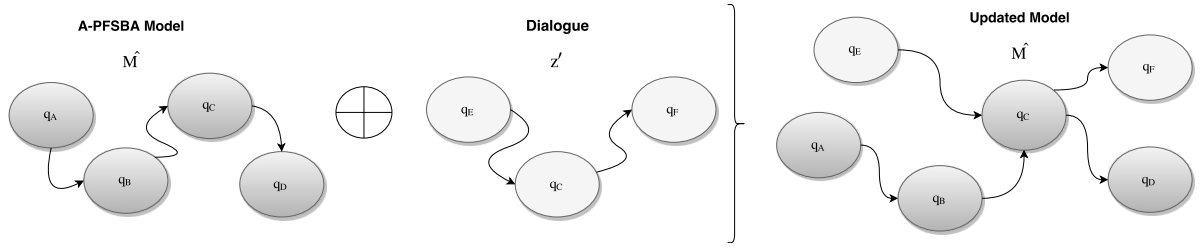

Fig. 3 Online Learning procedure where the initial model $\hat{M}$ is augmented with the A-PFSBA inferred from the correct dialogue z'

2.3 User Model

Simulating the behavior of the final users to augment the dialogues available for training and evaluation of Stochastic and Deep Learning-based DM is a common practice. User Models (UM) interact with the DM, generating synthetic dialogues [15, 16]. Several statistical and machine learning approaches have been proposed to model the user [20, 17, 7] but, in this paper, the same A-PFSBA paradigm is used to model the UM stochastically, mirroring the structure of the DM.
The goal of the UM is to return some user feedback given a system hypothesis and the current state under a certain policy Π_{UM}:

$$UM_{\Pi} : Q \times \Delta \rightarrow F \times Q$$

$$\Pi_{UM}(q_t, a_t, \hat{U}, G) \rightarrow f_{t+1},\ q_{t+1}$$

Where $\hat{U}$ is the A-PFSBA structure of the UM and a chosen policy Π_{UM}. When it comes to designing this policy, stochastic policies are generally chosen over those that maximize the likelihood/expected path-value in order to generate synthetic dialogues with more variability.

2.4 Online Learning

The ability to adapt and learn from unseen situations on the run is a powerful property of the A-PFSBA formulation. The online learning algorithm presented in [19] employs a Quality Metric QM to determine whether a new dialogue is suitable for learning or not. Using this metric, the A-PFSBA model learns from those dialogues rendered successful by the QM, augmenting the initial model by learning the new states and transitions of the new dialogues. This approach overcomes the drawbacks of previous turn-by-turn learning algorithms [13], that learned from both correct and incorrect dialogues.
Figure 2 shows the previous scenario where an unseen dialogue $\mathbf{z}'$ is rendered valid by a given QM, so the initial A-PFSBA model of the DM is augmented with the A-PFSBA model corresponding to z'.
Formally, let $\hat{M}$ be the A-PFSBA model inferred from Z dialogue samples, let $\mathbf{z}' \notin Z$ be an unseen dialogue sample and $\hat{M}_{\mathbf{z}'}$ the A-PFSBA inferred from $\mathbf{z}'$. If the QM renders $\mathbf{z}'$ valid for the learning process, $\hat{M}$ is expanded by merging it with $\hat{M}_{\mathbf{z}'}$. By doing so, the states q_x and the corresponding set of transitions $\delta[q_x] = \{(q, (\tilde{d}_i : \tilde{a}_i), q') | q = q_x\}$ of $\hat{M}_{\mathbf{z}'}$ are added to $\hat{M}$. The online learning pseudo-algorithm is defined as follows:

Algorithm 1 Online Learning

```
procedure A-PFSBAUpdate
    M̂ ← A-PFSBA from samples Z
    M̂_z' ← A-PFSBA from z'
    if QM(z') is True then
        for q_z ∈ Q_z' do:
            M̂ ← merge(M̂, q_z, δ[q_z])
            M̂ ← update_edge_count(M̂)
    return M̂
```

3 Exploitation Policies in the A-PFSBA Framework

The following section formally defines exploitation policies over the A-PFSBA framework and formulates both local and path based policies. This formulation also complies with the original PFSBA definition.

3.1 Policy Definition

One key component in every DM is the policy Π, that Sutton & Barto defined [21] as:

> A policy defines the learning agent's way of behaving at a given time. Roughly speaking, a policy is a mapping from perceived states of the environment to action to be taken when in those states.

Table 1 Main features of the Let's Go Corpus

Let's Go Corpus Statistics					
Dialogues	1840	System Turns	28141	System Dialogue Acts	49
Attributes	14	User Turns	28071	User Dialogue Acts	138

When it comes to spoken dialogue interaction, the agent would be the DM, the perceived states would be the dialogue states Q and the actions would be the system actions $a \in \Delta$. Then, the policy Π corresponds to a mapping from each system dialogue state $q \in Q_S$ to the set of system actions Δ.

The policy Π can be represented in multiple forms, either deterministically from the current state [1] or stochastically over the set of the possible actions [6, 28, 22]. More generally, it can be seen as a ranking problem, where the policy Π associates a score to each action $a \in \Delta$ given the current dialogue state, as reinforcement-learning methodologies do [9].

In the A-PFSBA framework, the policy corresponds to a decision/ranking function that maps the current system dialogue state $q \in Q_S$ and the set of possible transitions $\Delta_q = \{a_j \mid \exists\ (q,\ (\epsilon\ :\ a_j),\ q')\} \subseteq Q_S \times \Gamma \times Q_U$, i.e. the alphabet of actions associated to the state q.

Because the A-PFSBA formulation captures the transitions of both system and user actions, user information can be exploited in an straightforward way to determine path based policies when defining decisional strategies.

3.2 Path Based Policies

Path based policies can be defined as a scoring function over an A-PFSBA path of states with depth D $\theta = (q_s, q_1, \cdots, q_D)$ where $q_i \in Q$. The score associated to a given path or path-value $V(\theta)$ needs to take into account every taken step, the differences between the departure and the final states (q_s and q_D), the length of the path and the distance in time (as more distant actions should have lesser impact). These properties can be summarized in the following path-value function:

$$V(\theta) \ = \ \psi(q_s, q_D) \cdot \frac{\lambda}{|\theta|} \prod_{i=s}^{D} \gamma^i \ \cdot \phi(q_i, q_{i+1}) \qquad (2)$$

where the function $\psi(q_s, q_D)$ is the endpoint-value function that evaluates the differences between the departure state q_s and the final state q_D of the path θ, λ is the length normalization factor that determines the penalization of the dialogue length, $\gamma \leq$ is the discount factor that controls the temporal decay and ϕ is the step-value function that associates the reward for transitioning from the state q_i to q_{i+1}. The step-value function can be defined separately for user-taken steps ϕ_U or system-taken steps ϕ_S:

$$\phi(q_i, q_{i+1}) = \begin{cases} \phi_U(q_i, q_{i+1}), & if\ q_i\ \in Q_U\ and\ q_{i+1}\ \in Q_S \\ \phi_S(q_i, q_{i+1}), & if\ q_i\ \in Q_S\ and\ q_{i+1}\ \in Q_U \end{cases}$$

Then, the system action a to perform in a departure system state q_s is the one that maximizes the expected path-value of all the possible paths θ that depart from q_s and perform system action a.

$$a = \texttt{argmax}_{a \in \Delta_{q_s}} \frac{1}{|\Theta_{q_s,a}|} \sum_{\theta \in \Theta_{q_s,a}} V(\theta)$$

where $\Theta_{q_s,a}$ is the set of paths θ that start in state q_s and perform system action a as the first action. The search space is restricted by Δ_{q_s}, which corresponds to the alphabet of system actions associated to the departure state:

$$\Delta_{q_s} = \{a_j \mid \exists\ (q_s,\ (\epsilon\ :\ a_j),\ q')\} \subseteq Q_S \times \Gamma \times Q_U$$

3.3 Local Policies

Previous experiments in [13, 19] employed local decisional strategies over the bi-automata structure (i.e. taking into account only the current state q_s). Local policies can be represented as a subset of path based policies, i.e. those that are constrained to paths θ that contain only the departure and final state.

$$V_{local}(\theta) \ = \ \psi(q_t, q_{t+1}) \cdot \phi(q_t, q_{t+1}) \qquad (3)$$

4 Experimental Setup

The following section presents the experiments carried out to validate and evaluate the implemented path based policies on the Let's Go Corpus [14]. The results of [19] are replicated as a baseline and the online learning procedure is also replicated in order to measure the impact of user uncertainty.

Table 2 Let's Go Dialogue Formatting Example in terms of A-PFSBA alphabets

$q = [(\tilde{d_i} : \tilde{a_i}), \tilde{\omega_i}]$	System Actions and User Feedbacks
$q_0 = [(\epsilon : \epsilon), \epsilon] \in Q_S$	S: Welcome to the CMU Let's Go bus information system. To get help... $\tilde{a_1}$ =inform_welcome,inform_get_help,request_query_departure_place
$q_1 = [(\tilde{a_1} : \epsilon), \epsilon] \in Q_U$	U: I'm leaving from CMU. $\tilde{d_1}$ =inform_departure_place, PlaceInformation_registered_stop $\tilde{\omega_0} = \{\}$
$q_2 = [(\tilde{a_1} : \tilde{d_1}), \tilde{\omega_0}] \in Q_S$	S: Departing from <query.departureplace CMU>. Did I get that right? $\tilde{a_2}$ =Explicit_confirm, request_query_departure_place $\tilde{\omega_0} = \{\}$
$q_3 = [(\tilde{a_2} : \tilde{d_1}), \tilde{\omega_0}] \in Q_U$	U: Yes. $\tilde{d_2}$ = Generic_yes $\tilde{\omega_1} = \{< query.departure.place >\}$

4.1 Corpus Description

The Let's Go SDS developed by Carnegie Mellon University (CMU) exploits the Olympus architecture using RavenClaw [1] as DM to provide schedule and route information about the city of Pittsburgh bus service to the general public. The corpus linked to such SDS was collected from real user interactions during 2005, so events such as unexpected dialogue closing, spontaneous talking, sudden noise, etc. are observed. Some of the corpus statistics are shown in Table 1. In the corpus, feedback decoding is done using the the CMU Phoenix Parser [26], so each user state Q_U and system state Q_S is represented by a string. The attributes are discrete values related to bus schedule information. Table 2 shows a dialogue example of the corpus, where each state q is composed of a system action $a_t \Delta$, user decoded feedback $d_t \in \Gamma$ and its attributes $\omega \in \Omega$ that encode the relevant information of the dialogue history (e.g. that the user has already determined the place of departure). The corpus was split in half to build two A-PFSBA models, $\hat{M}$ to be used as the DM and $\hat{U}$ as the UM.

4.2 Smoothing Distance and Evaluation Metrics

In this section, the smoothing distance used to generalise to unseen states as described in Section 2.1 and the metrics used to evaluate the success of a dialogue are described in detail.

4.2.1 Smoothing Distance

The distance function (G) used in this work is the attribute-weighted Levenshtein distance employed in [19] and defined as follows:

$$G(q, q') = \text{dist}((\tilde{d_q} : \tilde{a_q}), (\tilde{d_{q'}} : \tilde{a_{q'}})) + \lambda(|\tilde{\omega_q} \cap \tilde{\omega_{q'}}| - |\tilde{\omega_q} \cup \tilde{\omega_{q'}}|)$$

where *dist* corresponds to the Levenshtein distance between the bi-string of system action and user action decoding and λ is a parameter which penalizes the distance depending on the amount of attributes in which the states differ. This distance is used for the smoothing process of both the DM and the UM.

4.2.2 Evaluation Metrics

The evaluation metrics employed correspond to the Task Completion rate (TC) and the Average Dialogue Length (ADL). In the Let's Go domain, the task is rendered complete when the DM carries out a coherent query to the database and retrieves the information asked by the user. A query is determined coherent when the user has given enough information to do a complete query to the database, i.e. the departure place, arrival place and time must be determined. The pseudo-code presented in Algorithm 2 describes the Task Completion metric adapted for the Let's Go scenario in the following experiments, which returns a boolean value that determines the success of the dialogue. The Average Dialogue Length measures the number of turns that the dialogue lasts on average, where each user/system interaction counts as a turn.

Algorithm 2 Task Completion

```
procedure TASK COMPLETION(dialogue)
    Departure_info = check_departure(dialogue)
    Arrival_info = check_arrival(dialogue)
    When_info = check_when_time(dialogue)
    Request_Next_Bus = check_next(dialogue)
    Is_query_to_db = check_query(dialogue)
    Is_Info = Departure_info and Arrival_info and When_info
    if Is_query_to_db is False then
        return False
    if (Is_Info or Request_Next_Bus) is True then
        return True
    return False
```

4.3 Local Policies

Two local policies have been implemented, a deterministic one based on the maximum transition probability between the A-PFSBA states and a stochastic one which samples the next action from the transition probability distribution.

- **Maximum Probability (MP):** the DM chooses the action $\hat{a}$ from state q_j that maximizes the transition probability

$$\hat{a} = \texttt{argmax}_{a \in \Delta_{q_s}} P(q_s\ , (\epsilon :\ a\ ),\ q'\ )$$

  This policy is exclusive of the DM.

- **Random Sampling (RS):** as the policy of the UM Π_{UM} has to be non-deterministic in order to achieve variance in the generated dialogues, the user action to perform is randomly sampled from the distribution of user actions seen in the current state.
  This policy is mainly used by the UM, the DM only uses it during the online learning phase to create dialogues with more variability in order to learn new strategies.

4.4 Path Based Policies

Three different path based policies exclusive of the DM have been implemented following the path-value function of Equation 2.[1] All the policies use the same step-value function defined in the following equation:

$$\phi(q_i, q_{i+1}) = \begin{cases} \phi_U(q_i, q_{i+1}) = \frac{P((q_i, (\epsilon:\ a_k),\ q_{i+1})\ )}{P((q_i, (\epsilon:\ a_k),\ q_{i+1})\ )^{1-\beta}} \\ \phi_S(q_i, q_{i+1}) = P((q_i, (\epsilon :\ a_k),\ q_{i+1})\ ) \end{cases} \tag{4}$$

where the $\beta \in [0, 1]$ parameter is the user-awareness rate. This parameter weights the user transition probability in the scoring function $\phi(q_i, q_{i+1})$. When $\beta = 0$, the user is ignored: every transition probability is equal to 1 and the user transition probabilities are not taken into account for the final score. On the other hand, when $\beta = 1$ the user transition probability is taken into account in the scoring function and more probable user-actions achieve a higher score.

- **Maximum Probability Path (MPP):** chooses the path of system actions with maximum probability. The endpoint-value function used in the MPP policy is:

$$\psi(q_0, q_D) = 1$$

- **Attributed Path (AP):** chooses the path with highest probability that also searches to complete as many dialogue attributes as possible. The endpoint-value function is changed to:

$$\psi(q_0, q_D) = \frac{1}{1 + (|\omega_{q_D}| - |\omega_{q_0}|)}$$

  where ω_{q_0} and ω_{q_D} are the attributes of the initial and the final state.

- **Task Completion Path (TCP):** chooses the path with highest score according to the Task Completion rate, i.e. the path that satisfies most constraints to consider a dialogue successful. The endpoint-value function is modified to:

$$\psi(q_0, q_D) = \frac{1}{1 + (TCS(q_0, q_D))}$$

  where $TCS(q_0, q_D)$ is a scoring version of the Task Completion metric shown in Algorithm 2. Instead of using the boolean output of the TC rate, a constant value λ is added [2] for each constraint satisfied (*Departure_info, Arrival_info, ...*) through the path. In this policy, instead of guiding the dialogue to simply fulfill attributes, the dialogue manager selects those actions that guide the interaction to satisfy the constraints needed to complete the task.

As it is intractable to calculate every possible path in the set of dialogue paths that start in the state q_s and perform a as the first action $\Theta_{q_s,a}$ to estimate the best action $\hat{a}$ for each system state, Monte Carlo sampling is used to generate multiple paths from their transition probabilities.

5 Policy Evaluation

In the following section, the implemented local and path based policies are tested before and after an user adaptation phase carried out using the online learning procedure of Section 2.4 that updates the A-PFSBA structure in a dialogue-by-dialogue basis. In addition, each path based policy is evaluated performing a grid search over the path length or depth D and the user-awareness rate $\beta \in [0, 1]$ in order to evaluate the impact of the user uncertainty and the structural learning of the A-PFSBA. As this grid search is performed before and after the online learning phase, the adaptation capacity of the A-PFSBA is also evaluated.
Results achieved using the Maximum Probability local policy explored in [19] are summarized in Table 3 and

[1] In order to avoid numerical underflow, the logarithm is applied to the product

[2] $\lambda = 0.25$

Table 3 Structural and Maximum Probability policy evaluation before and after online learning.

	States-DM	Transitions-DM	States-UM	Transitions-UM	TC (%)	ADL
CMU RavenClaw	—	—	—	—	54.0	32.33 ± 1.2
A-PFSBA MP	11005	14737	11058	14988	**60.02 ± 1.36**	30.98 ± 0.94
A-PFSBA After OL MP	14700	21952	11058	14988	**69.39 ± 1.34**	31.46 ± 0.69

set as baselines. The first row of Table 3 shows results achieved by the RavenClaw DM. The second and third rows show information regarding the structure of both the DM and the UM together with the performance of the DM in terms of TC and ADL metrics, before and after user adaptation through online learning. The learning curve shown in Figure 5 shows that the DM reaches its saturation point at 50.000 dialogues.

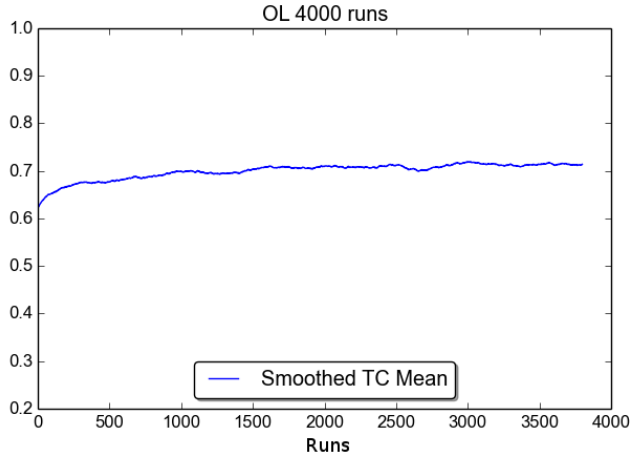

Fig. 4 Smoothed learning curve of the TC rate during the online learning procedure for user adaptation

5.1 Path Based Policy Behavior Before User Adaptation

This section evaluates the performance of the implemented path based policies and the impact of the path length or depth D and the user-awareness rate β in the TC and ADL metrics, before the user adaptation phase.

Table 4 Best path based policy results before user adaptation

	TC (%)	ADL
MP Local	60.02 ± 1.4	30.98 ± 0.9
MPP	59.3 ± 0.6	32.2 ± 0.3
AP	59.5 ± 0.6	32.8 ± 0.3
TCP	61.2 ± 0.6	32.5 ± 0.3

Table 4 shows the best results obtained for each path based policy compared to the local MP policy set as baseline. Overall differences in TC rate are not significant, with **TSP** performing slightly better than **MPP** and **AP** but without statistical significance with respect to the local MP policy, as there is an overlap in their confidence intervals[3]. The slight improvement over the TC rate of **TCP** can be attributed to the inclusion of external information in the dialogue policy.

Regarding the ADL metric, the local MP policy tends to generate slightly shorter dialogues. This is usually better than long dialogues[4] in task-oriented dialogue systems as is the case of Let's Go scenario. Nevertheless, a difference of 1 turn can be considered negligible from the point of view of the end users.

Path based policies depend on the path length or depth D and the user-awareness rate β parameters, where the depth determines how much future steps the policies take into account and the user-awareness rate represents the relevance given to the user transition probabilities in the scoring function of the policy. These parameters have a direct impact on the performance of path based policies. The user-awareness rate β measures the uncertainty of the user behavior in the modeled scenario. Policies that perform worse when β is set to 1 than when $\beta = 0$ indicate that the user transition probabilities are not correctly estimated. Also, the variability of the Task Completion rate conditioned over the path length or depth D indicates how well the A-PFSBA model is fitted to the user. Long paths performing worse than short paths signal that the model is not taking into account paths that the user employs commonly. To evaluate the impact of these parameters in the implemented path based policies, Figure 5 shows a spline-smoothed graph for each of the analysed policies.

Previous to the online learning phase, the relationship between the path length and the user-awareness rate is clear for the three policies: long and user-aware paths perform worse. This conclusion validates the hypothesis of [4] that path based policies perform worse than local policies overall due to user uncertainty. In addition, it is clear that the initial models are not fitted to the

[3] 95% confidence interval.

[4] In social dialogue systems the longer the dialogue the better, as their goal is to maximize the user engagement with the system.

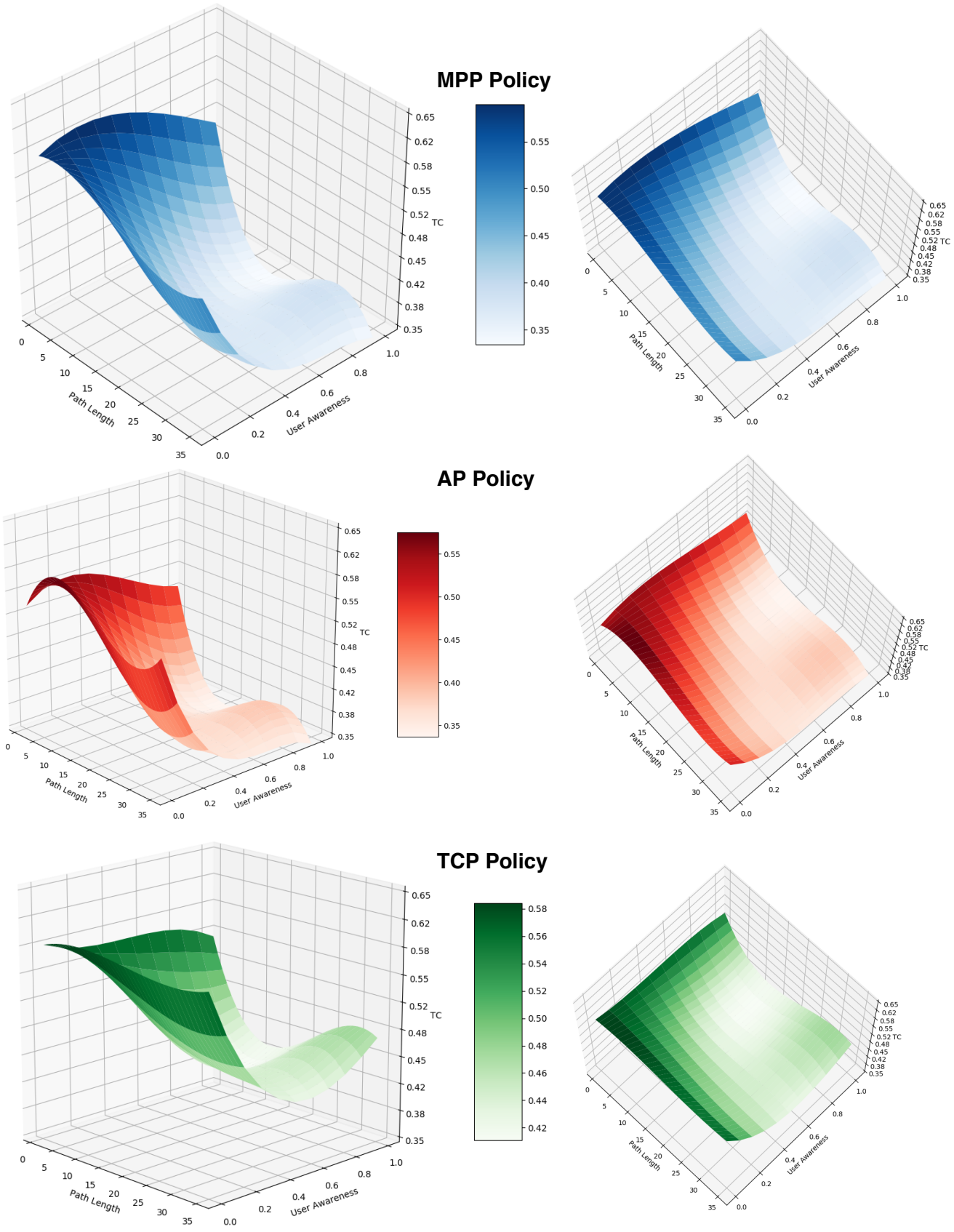

Fig. 5 Spline-smoothed plots of the TC rate of the path based policies before online learning with different perspectives of the same plot. Left: front view, right: top view.

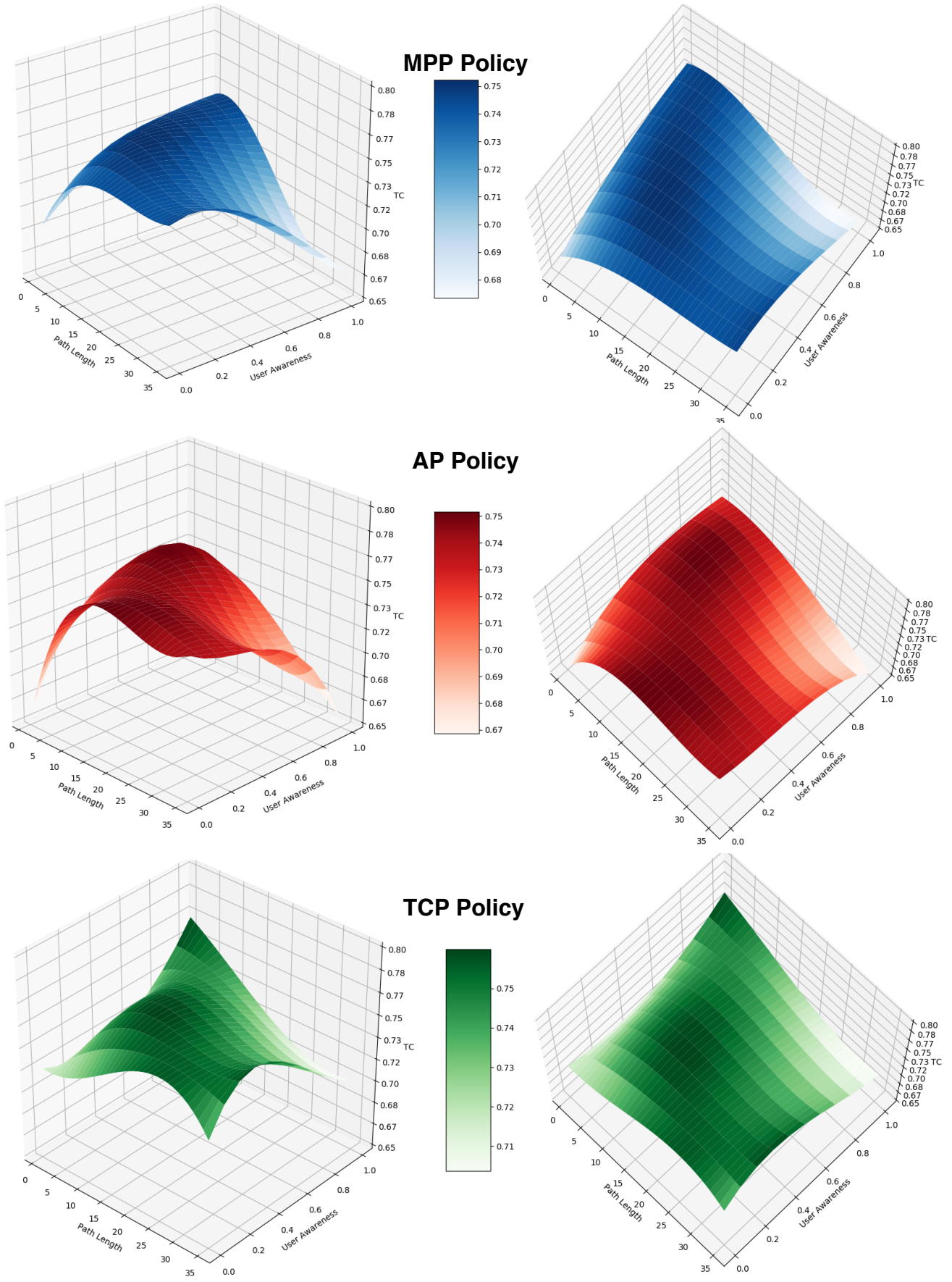

Fig. 6 Spline-smoothed plots of the TC rate of the path based policies after online learning with different perspectives of the same plot. Left: front view, right: top view.

user, as the TC rate gets worse when the path length is increased.
Also, it is interesting to note that the TCP policy has a higher low-boundary. This can be attributed to the inclusion of external information such as the Task Completion score, which reduces the amount of decay introduced by the path length and the user uncertainty.

5.2 Path Based Policy Behavior After User Adaptation

The performance of the implemented path based policies and the impact of the path depth and the user-awareness rate parameters on the TC rate and ADL metrics is once again evaluated in this section, but after a user adaptation phase.

Table 5 Best path based policy results after user adaptation

	TC (%)	ADL
MP Local	69.4 ± 1.4	31.5 ± 1.0
MPP	73.8 ± 0.5	**30.5** ± 0.3
AP	**74.9** ± 0.5	**29.9** ± 0.3
TCP	**75.0** ± 0.5	31.6 ± 0.3

Table 5 shows the best results for each implemented policy after the online learning phase. These results demonstrate that once user behavior is fitted to the A-PFSBA structure of the DM, path based policies perform better than local policies and are the ones that generate shorter dialogues. Nevertheless, dialogue length differences are still negligible from the perspective of the end users. **AP** and **TCP** policies equally outperform the other policies, since both use information additional to the transition probabilities. Taking into account the close performance of the **MPP** policy, one might wonder whether including external information such as dialogue attributes or the task completion score is necessary. However, this is arguable because task completion information is implicitly codified in the online learning process: only those dialogues that qualify according to the TC rate are included in the DM A-PFSBA model and dialogues with missing attributes will not be successful, since attributes are required to query the database and render the task complete. Additionally, the inclusion of external information makes the **TCP** policy consistent in both scenarios, before and after online learning.
The relationship between the path length or depth D and the user-awareness rate β changes drastically in every path based policy after online learning as shown in Figure 5. The penalization that both the path length or depth D and the user-awareness rate β introduced before online learning is drastically diminished, demonstrating that the online learning algorithm proposed in [19] is suitable to fit the A-PFSBA DM to the user. It is interesting to note that the degradation of the TC rate due to path length and user-awareness is higher in the MPP policy, as it does not include neither dialogue attributes nor task completion information when making decisions. Another pattern that repeats across the three policies is that the equilibrium between both the path length and the user-awareness rate parameters yields consistent results, i.e. if one wants to use a longer path, the user-awareness rate should be lower to compensate. This trade-off is clearer in the **TCP** policy, where the highest results can be observed in the center of the plot. Once again, the **TCP** policy is the one that has the highest low-boundary. This is another clear indicator that the inclusion of external information can improve the robustness and consistency of the exploitation policies. The results shown in Table 5 and Figure 5 confirm the hypothesis raised in [4] that path based policies perform worse when the DM is not adapted to the user, as they also take into account user behavior. In addition, the results obtained also demonstrate that the A-PFSBA framework is capable of adapting to user behavior on the run applying the online learning algorithm proposed in [19]. Regarding the average dialogue length of the generated interactions, there is no significant difference between path based and local policies.

6 Conclusions and Future Work

In this paper, the Attributed Stochastic Finite State Bi-Automata (A-PFSBA) paradigm is used to model dialogues as a stochastic process of user/system *bi-string* interaction. This approach has the advantage that the structural learning of the dialogues with the A-PFSBA framework and its exploitation policy for dialogue management are independent of each other. In the paper, the theoretical A-PFSBA framework is augmented by introducing a formal definition of exploitation policies. Under such definition, three path based policies are implemented: (i) the classical Maximum Probability Path policy; (ii) an Attributed Path policy, which searches to complete dialogue attributes; and (iii) a Task Completion Path policy, which searches for those dialogue interactions that maximize the chance of success using external information. These policies are tested before and after an online learning phase and are evaluated in terms of Task Completion rate and Average Dialogue Length, conditioned over the parameters of path length or depth and user-awareness rate.

Results empirically demonstrate that when external information such as the task completion is included in the path based policies, these are able to achieve slightly better results than local policies without user adaptation. In addition, the inclusion of external information results in more robust policies after user adaptation. The impact of the path length and user-awareness rate parameters before and after online learning demonstrates that the learning algorithm is valid when it comes to fit the A-PFSBA DM model to new users on the run.

After online user adaptation, the performance of path based policies increases significantly in comparison to the local policies. This demonstrates that once the uncertainty of user behavior is reduced, path based exploitation policies can model the possible user actions sensibly.

The paper consolidates the A-PFSBA framework for dialogue management, demonstrating its flexibility to adopt different exploitation policies. As future work, we plan to research alternative ways to exploit external information in dialogue policies and to develop methods for inferring the optimal parameters to tackle user uncertainty on the run. In addition, testing the A-PFSBA framework on other dialogue corpora and tasks is also intended.

7 Acknowledgement

This work has been partially founded by the Spanish Minister of Science under grants TIN2014-54288-C4-4-R and TIN2017-85854-C4-3-R and by the European Commission H2020 SC1-PM15 EMPATHIC project, RIA grant 69872.

References

1. Bohus D, Rudnicky AI (2009) The ravenclaw dialog management framework: Architecture and systems. Computer Speech & Language 23(3):332–361
2. Cuayáhuitl H (2017) Simpleds: A simple deep reinforcement learning dialogue system. In: Dialogues with Social Robots, Springer, pp 109–118
3. Gašić M, Jurčíček F, Keizer S, Mairesse F, Thomson B, Yu K, Young S (2010) 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, Association for Computational Linguistics, pp 201–204
4. Ghigi F, Torres MI (2015) Decision making strategies for finite-state bi-automaton in dialog management. In: Natural Language Dialog Systems and Intelligent Assistants, Springer, pp 209–221
5. Gorin AL, Riccardi G, Wright JH (1997) How may i help you? Speech communication 23(1-2):113–127
6. Griol D, Callejas Z, López-Cózar R (2010) Statistical dialog management methodologies for real applications. In: Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, Association for Computational Linguistics, pp 269–272
7. Hurtado LF, Griol D, Sanchis E, Segarra E (2007) A statistical user simulation technique for the improvement of a spoken dialog system. In: Iberoamerican Congress on Pattern Recognition, Springer, pp 743–752
8. Hurtado LF, Planells J, Segarra E, Sanchis E, Griol D (2010) A stochastic finite-state transducer approach to spoken dialog management. In: Eleventh Annual Conference of the International Speech Communication Association
9. Jurčíček F, Thomson B, Young S (2012) Reinforcement learning for parameter estimation in statistical spoken dialogue systems. Computer Speech & Language 26(3):168–192
10. Li J, Monroe W, Ritter A, Galley M, Gao J, Jurafsky D (2016) Deep reinforcement learning for dialogue generation. arXiv preprint arXiv:160601541
11. Li Z, Tang J (2017) Weakly supervised deep matrix factorization for social image understanding. IEEE Transactions on Image Processing 26(1):276–288
12. Li Z, Liu J, Tang J, Lu H (2015) Robust structured subspace learning for data representation. IEEE transactions on pattern analysis and machine intelligence 37(10):2085–2098
13. Orozko OR, Torres MI (2015) Online learning of stocastic bi-automaton to model dialogues. In: Iberian Conference on Pattern Recognition and Image Analysis, Springer, pp 441–451
14. Raux A, Langner B, Bohus D, Black AW, Eskenazi M (2005) Let's go public! taking a spoken dialog system to the real world. In: Ninth European Conference on Speech Communication and Technology
15. Schatzmann J, Young S (2009) The hidden agenda user simulation model. IEEE transactions on audio, speech, and language processing 17(4):733–747
16. Schatzmann J, Georgila K, Young S (2005) Quantitative evaluation of user simulation techniques for spoken dialogue systems. In: 6th SIGdial Workshop on DISCOURSE and DIALOGUE
17. Schatzmann J, Weilhammer K, Stuttle M, Young S (2006) A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. The knowledge engineering re-

view 21(2):97–126
18. Serban IV, Sordoni A, Bengio Y, Courville AC, Pineau J (2016) Building end-to-end dialogue systems using generative hierarchical neural network models. In: AAAI, vol 16, pp 3776–3784
19. Serras M, Torres MI, Del Pozo A (2017) Online learning of attributed bi-automata for dialogue management in spoken dialogue systems. In: Iberian Conference on Pattern Recognition and Image Analysis, Springer, pp 22–31
20. Serras M, Torres MI, del Pozo A (2017) Regularized neural user model for goal oriented spoken dialogue systems. In: International Workshop on Spoken Dialogue Systems (IWSDS)
21. Sutton RS, Barto AG (1998) Reinforcement learning: An introduction, vol 1. MIT press Cambridge
22. Thomson B, Yu K, Keizer S, Gašić M, Jurčíček F, Mairesse F, Young S (2010) Bayesian dialogue system for the let's go spoken dialogue challenge. In: Spoken Language Technology Workshop (SLT), 2010 IEEE, IEEE, pp 460–465
23. Torres MI (2013) Stochastic bi-languages to model dialogs. In: Proceedings of the 11th International Conference on Finite State Methods and Natural Language Processing, pp 9–17
24. Torres MI, Benedí JM, Justo R, Ghigi F (2012) Modeling spoken dialog systems under the interactive pattern recognition framework. In: Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR), Springer, pp 519–528
25. Vinyals O, Le Q (2015) A neural conversational model. arXiv preprint arXiv:150605869
26. Ward W (1995) The cmu atis system. In: Proc. of ARPA Spoken Language Systems Technology Workshop
27. Williams JD, Zweig G (2016) End-to-end lstm-based dialog control optimized with supervised and reinforcement learning. arXiv preprint arXiv:160601269
28. Young S, Gašić M, Thomson B, Williams JD (2013) Pomdp-based statistical spoken dialog systems: A review. Proceedings of the IEEE 101(5):1160–1179
29. Zhao T, Eskenazi M (2016) Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning. In: 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, p 1