Dialogue Systems Domain Interaction Using Reinforcement Learning

Porfírio Filipe[1,2,3], Paulo Araújo[3], and Nuno Mamede[1,4]

[1] L[2]F INESC-ID – Spoken Language Systems Laboratory, Lisbon, Portugal
{porfirio.filipe, nuno.mamede}@l2f.inesc-id.pt
[2] GuIAA – Grupo de Investigação em Ambientes Autónomos, Lisbon, Portugal
[3] ISEL – Instituto Superior de Engenharia de Lisboa, Lisbon, Portugal
paraujo@deetc.isel.ipl.pt
[4] IST – Instituto Superior Técnico, Lisbon, Portugal

Abstract. This paper describes research about using a reinforcement learning approach to optimize our Domain Knowledge Manager (DKM) that is part of a mixed-initiative task based Spoken Dialogue System (SDS) architecture, namely to access an Ambient Intelligence (AmI) scenario. Assuming that practical dialogue and domain-independent hypothesis are true, we have considered a clear separation between discourse dependent and domain dependent knowledge, which allows reducing the complexity of SDS typical components, specially the Dialoguer Manager (DM). In this context, we believe that is possible to get better DM strategies optimizing the interaction between DM and DKM. For this, we propose a new feature, for the DKM, based on learning and suggest the best task-artifact pairs to satisfy a DM query using the DM feedback as reward. The proposed DKM feature has been tested in our simulator based on Portuguese language.

Keywords: Spoken Dialogue System, Reinforcement Learning, Domain Knowledge Management, Dialogue Management.

1 Introduction

Research in Spoken Dialogue System (SDS) emerged around the late-1980s because of two major government funded projects: the DARPA Spoken Language Systems in the United States and the Esprit SUNDIAL in Europe [1, 2]. The DARPA project was concerned with the domain of Air Travel Information Services (ATIS). The Esprit SUNDIAL project, funded by the European Community, was concerned with flight and train schedules in English, French, German and Italian [3].

Speech-based human-computer interaction and particularly SDS development face several challenges in order to be more widely accepted. One of the challenges in dialogue management is to control the dialogue flow (dialogue strategy) in an efficient and natural way. Dialogue strategies designed by humans are prone to errors, labour-intensive and non-portable, making automatic design an attractive alternative.

Learning dialogue strategies from real users is a very expensive and time-consuming process, making automatic learning an attractive alternative [4].

After more than ten years, automatic learning of optimal dialogue strategies is now a leading domain of research, with several recent advances [5, 6]. In this approach, the population of users defines the stochastic environment, the dialogue system's actions are its utterances and the state is represented by the entire dialogue so far. The goal is to design a SDS that takes actions to maximize some measure of reward. In this context, it becomes possible, at least in principle, to apply a Reinforcement Learning (RL) [7] approach to find a good action-selection (i.e., dialogue) strategy. However, the practical application of RL to SDSs faces a number of severe technical challenges. First, representing the dialogue state by the entire dialogue so far is often neither feasible nor conceptually useful, and the so-called belief state approach is not possible, since we do not even know what features are required to represent the belief state. Second, there are many different choices for the reward function, even among systems providing very similar services to users [8].

Summarizing, our contribution tries to improve DM strategies optimizing the domain interaction. Section 2 gives an overview of the proposed adaptive approach. Section 3 describes our reinforcement learning based proposal showing an application example. Finally, in Section 4, we present concluding remarks and future work.

2 Adaptive Approach

In this paper, we propose a divide to conquer approach assuming that practical dialogue and domain-independent hypothesis are true [9]. In order to introduce our approach, we considered, within an Ambient Intelligent (AmI) scenario [10], the simplified SDS architecture presented in **Fig. 1**.

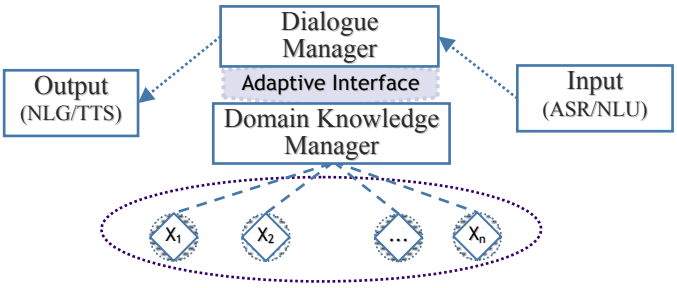

Fig. 1. SDS architecture for adaptation to a dynamic domain.

In Fig. 1, X_1, X_2, …, X_n represent a generic AmI artifact (or device), such as a lamp or a typical household appliance according to our previous work in [11, 12]. This figure shows the Domain Knowledge Manager (DKM) interacting, via an adaptive interface, with the Dialogue Manager (DM) and simultaneously working as an artifact broker presenting a dynamic integrated view of the domain. In this architecture, we assume a clear separation between discourse dependent (DM job) and domain dependent (DKM job) knowledge allows reducing the complexity of the SDS. DM interacts with the DKM, via a concept query, in order to solve possible user's

request ambiguities. When the user's request is unambiguous, the DM calls the DKM that invokes the selected task supported by the selected artifact.

The design of a DM that can be easily customized to new domains and in which different dialogue strategies can be explored, should only concern phenomena related to the dialogue with the user, focusing on dialogue and on discourse strategies. The DM component should not be involved in the process of accessing a background system or performing domain reasoning. For this, should be considered another component of SDS architecture, the DKM, which handles these features [13].

The DKM is in charge for retrieving and coordinating knowledge from the different domain knowledge sources and application systems, traditionally named background system. The DM can deliver a request to the DKM and in return expects an answer retrieved from the background system. When a request is under-specified or contains inconsistencies from the DKM's point of view, a specification of the problem will be returned to the DM that will start a clarification dialogue. In these circumstances, the DKM allows the customization of the DM enabling dynamic domain adaptation providing an easy to use small interface, instead of a conventional service interface with several remote procedures/methods.

In order to achieve its goal, the DKM includes a domain model with three independent knowledge components: DISCOURSE model, WORLD model, and TASK model. This domain model architecture was adapted from the Unified Problem solving Method Development Language (UPML) [14]. For the sake of space, a detailed description of the domain model components will be omitted. For a more complete description see our previous work in [15, 16].

Our aim is the optimization of the DM strategies taking advantage from the proposed DKM feature, which allows the DKM to learn (and suggest) the best task-artifact pairs to satisfy a DM query, using the DM confirmation feedback as reward. This proposal contributes to optimize the DM strategy allowing, for instance, to start a clarification dialogue with the best task-artifact pairs suggested by the DKM. Inevitably, this approach also reduces the amount of user's interactions needed to solve dialogue ambiguities (clarifications).

3 Proposal

Reinforcement Learning (RL) is a promising option to use when the application domain is complex and uncertain. The goal of RL is to maximize the cumulative reward [6]. We propose the use of unsupervised learning in opposition to supervised learning because we do not want to consider any kind of initial training set. Using our RL approach, the DKM can operate with no previous training data about DM interaction and can learn form the experience, using a general method of problem solving that is trial and error.

Each DM query processing is considered an RL episode. The DKM answers the DM query suggesting a rank of the best task-artifact pairs. The DM returns feedback indicating the selected task-artifact pair, which is the best.

The main idea of this proposal is to use a simple and efficient RL algorithm to apply in a scenario that involves huge state space and dynamic change of domain artifacts.

The DKM domain model has represented a set of heterogeneous artifacts belonging to the domain. We propose a learning approach based in three different contexts levels. First context is about the selection of the best artifact. Second context is about the selection of the best task name support by the artifact. Finally, third context is about the task role.

This context levels are necessary to solve ambiguous discourse and guarantees that when user talks about artifacts or tasks, the artifacts have priority over single tasks. On first level, domain artifacts are suggested according the learned data. On the second level, tasks are suggested with priority over task roles. On third, level the task roles are simply suggested.

This adaptive approach minimizes the learning information and optimizes the learning process. The learned dada is maintained in the context of each artifact represented in the DKM domain model.

3.1 Applying RL

The DM queries are modeled as states and DKM answers are modeled as possible actions. The reward function depends on DM feedback based on the SDS user choices. This approach allows system to evolutes with the uncertain environment and collects rewards.

We propose the use of the Q-Learning off-policy Temporal Difference (TD) control [7] to learn each Q matrix representing artifacts, tasks, roles and values. The advantage of temporal difference algorithms is that they update Q matrix based on passed experience without waiting for a final reward. This kind of method allows bootstrapping.

In each Q matrix, the first column represents possible states. First line represents possible actions. The $Q(s_t, a_t)$ represents the action-value function learned where s_t represents state and a_t represents action. For each iteration, action-values are updated according to (1). The presented formula has two configuration parameters α and γ.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right] \quad (1)$$

The α parameter configures values between]0, 1] for constant step size. This parameter allows configuring the weight for past Q matrix values.

The γ parameter configures how immediate is the reward. It is possible to configure values between [0, 1[. When values are near zero only immediate reward is considered. When values are near one, future reward is considered with greater weight.

Each step of Q-learning approximates to Q* (the optimal action-value function) independent of policy being followed.

On first context level, artifacts as considered as states and all possible tasks for that specific artifact act as possible actions. In this particular case, all the positions of the Q matrix table are possible and have values that represent the learned rewards.

3.2 Application Example

The experimental setup is based on our AmI simulator, originally developed for Portuguese users. This domain simulator incorporates a basic dialogue manager and several artifact simulators, such as a microwave oven, a fryer, freezer, a lamp and a window. The debug of an invoked task can be made analyzing the interaction with the target artifact. We can attach and detach artifacts, execute tasks, obtain the answers and observe the subjacent artifact behaviors. We can also consult and print several data about the several knowledge representations.

Fig. 2 contains a screenshot of the domain simulator that is showing a kitchen lamp.

Fig. 2. Screenshot of the kitchen lamp simulation.

In order to illustrate the application of our proposal we describe the learning process applied to a lamp (Lâmpada) that is a common artifact in an AmI environment, particularly in home domain.

Next XML representation is a partial knowledge model of a kitchen lamp where is represented a task that modifies the lamp intensity.

```
<DomainModel>
<DiscourseModel>
  <Concept Identifier="221" Type="Action">
<!-- CHANGING -->
    <LinguisticDescriptor Type="Synonym">
      <WordDescriptor Language="pt-PT" PartOfSpeech="Verb"
Word="MODIFICAR"/>
    </LinguisticDescriptor>
  </Concept>
  <Concept Identifier="114" Type="Attribute">
<!-- INTENSITY -->
```

```
    <LinguisticDescriptor Type="Synonym">
      <WordDescriptor Language="pt-PT" PartOfSpeech="Noun"
Word="INTENSIDADE"/>
    </LinguisticDescriptor>
  </Concept>
  <Concept Identifier="186" Type="Unit">
<!-- PERCENTAGE -->
    <LinguisticDescriptor Type="Synonym">
      <WordDescriptor Language="pt-PT" PartOfSpeech="Noun"
Word="PERCENTAGEM"/>
    </LinguisticDescriptor>
  </Concept>
  <Concept Identifier="-5" Type="Name">
<!-- KITCHEN LAMP -->
    <LinguisticDescriptor Type="Synonym">
      <WordDescriptor Language="pt-PT" PartOfSpeech="Noun" Word="LUZ"/>
      <WordDescriptor Language="pt-PT" PartOfSpeech="Preposition"
Word="DA"/>
      <WordDescriptor Language="pt-PT" PartOfSpeech="Noun"
Word="COZINHA"/>
    </LinguisticDescriptor>
  </Concept>
  <Concept Identifier="129" Type="Active">
<!-- LAMP -->
    <LinguisticDescriptor Type="Synonym">
      <WordDescriptor Language="pt-PT" PartOfSpeech="Noun" Word="LUZ"
PhoneticTranscription='l"uS'/>
    </LinguisticDescriptor>
    <SemanticDescriptor Source="WordNet" Meaning="light source" Label="n"
Position="1"/>
  </Concept>    <.../>
</DiscourseModel>
<TaskModel>
  <Task Identifier="-5" Name="221">
    <Role Name="114" Range="186" Restrition='&amp;"114"&gt;=0 &amp;&amp;
100 &lt;= &amp;"114"' Type="IN"/>
  </Task>              <Task .../>
</TaskModel>
<WorldModel>
  <TypeHierarchy>
    <Class Identifier="-4" Name="129" Class="..."/>
    <Class .../>
  </TypeHierarchy>
  <Mediator>
    <Artifact Identifier="-1" Name="-5"/>
  </Mediator>
</WorldModel>
<Bridge>
  <ArtifactClass Artifact="-1" Class="-4"/>
  <ArtifactTask Artifact="-1" Task="-5"/>
  <ArtifactTask .../>  <ClassTask .../>
</Bridge>
</DomainModel>
```

Some of the data about the artifact kitchen lamp is used to demonstrate the application of our proposal.

Table 1. Artifact task.

Task / Artifact	T001	T002	T003	T004	T005	T006	T007
Lâmpada	22.0	22.0	0	14.0	0	16.0	0

Table 1 represents the artifact lamp (Lâmpada) and respective possible tasks: T001-Ligar, turn on; T002-Desligar, turn off; T003-Reduzir, to reduce; T004-Aumentar, to increase; T005-Dizer, to say (state); T006-Dizer, to say (color); T007-

Dizer, to say (room). Initially this table is filled with zero. According to the reward this tables is updated. The tasks (actions) are suggested according to a descend order.

Table 2. Task role.

Task Role / Artifact task	R001	R002	R003	R004	R005
T001	-	-	-	-	-
T002	-	-	-	-	-
T003	0	-	-	-	-
T004	-	20	-	-	-
T005	-	-	0	-	-
T006	-	-	-	18	-
T007	-	-	-	-	0

Table 2 represents the task roles of artifact lamp (Lâmpada). The task roles are: R001-Intensidade (Fracção), intensity (sub multiple); R002-Intensidade (Múltiplo), intensity (multiple); R003-Estado, state; R004-Cor, color; R005-Sala, room. When task role not belongs in the tasks, a character '-' is present. When a task role is related with the task the Q matrix table is filled with zero. This table is updated in respective position according to obtained reward.

Table 3. Task role values.

Task Role Value / Task role	V001	V002	V003	V004	V005	V006	V007	V008	V009	V010	V011	V012	V013	V014
R001	15	10	8	-	-	-	-	-	-	-	-	-	-	-
R002	-	-	-	16	9	7	-	-	-	-	-	-	-	-
R003	-	-	-	-	-	-	18	17	-	-	-	-	-	-
R004	-	-	-	-	-	-	-	-	3	0	0	-	-	-
R005	-	-	-	-	-	-	-	-	-	-	-	0	0	15

Table 3 represents the task roles values of artifact lamp (Lâmpada). The task roles values are: V001-Metade, half; V002-Um terço, third part; V003-Quarto, quarter; V004-Duplicar, double; V005-triplicar, triple; V006-Quadruplicar, quadruplicate; V007-Ligado, on; V008-Desligado, off; V009-Branco, white; V010-Amarelo, yellow; V011-Vermelho, red; V012-Quarto, bed room; V013-Sala, living-room; V014-Cozinha, kitchen. When task role value cannot be used to fill a task role, a character '-' is present. When a task role value is related with the task role the Q matrix table is filled with zero. This table is updated in respective position according to obtained reward.

4 Concluding Remarks and Future Work

Machine learning applied to SDS dialogue management strategy design is a growing research area. Training of such strategies can be done using human users or using corpora of human computer dialogue. However, the size of the state space grows exponentially according to the state variables taken into account, making the task of learning dialogue strategies for large-scale SDS very difficult.

Our contribution tries to optimize DM strategies within an AmI scenario, not learning dialogue strategies, but adding a new feature in the DKM. This feature allows the DKM to learn and suggest the best task-artifact pairs that satisfies a DM query. This is a significant contribution to optimize DM strategies, and simultaneously the portability, of the SDS multi propose architecture being developed in our lab.

The presented ideas have been applied in a domain materialized as a set of heterogeneous artifacts that represents a home environment. The proposed DKM feature has been tested, with success, in our domain simulator based on Portuguese language.

As future work, we expect to address a larger set of artifacts in a multi user interaction scenario extending the learning mechanism to deal with different user's profiles. We also expect to explore the presented ideas, more deeply, applying to levels of hierarchical reinforcement learning first in DKM, as we are doing now, and the second to learn about each one of the domain's artifacts by itself, combining a global versus local learning process.

References

1. McTear, M.F.: Spoken Dialogue Technology: Enabling the Conversational User Interface. ACM Computer Survey, 34(1):90-169 (2002)
2. McTear, M.F.: Spoken Dialogue Technology: Toward the Conversational User Interface. Springer Verlag, London (2004)
3. Peckham, J.: A New Generation of Spoken Dialogue Systems: Results and Lessons from the SUNDIAL Project. In Proceedings of the 3rd European Conference on Speech Communication and Technology, 33 – 40 (1993)
4. Cuayahuitl, H., Renals, S., Lemon, O., Shimodaira, H.: Reinforcement Learning of Dialogue Strategies with Hierarchical Abstract Machines. IEEE/ACL Spoken Language Technology (2006)
5. Levin, E., Pieraccini, R., Eckert, W.: Learning Dialogue Strategies within the Markov Decision Process Framework. In Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (1997)
6. Lemon, O., Pietquin, O.: Machine Learning for Spoken Dialogue Systems. Tutorial paper in Proceedings of the Interspeech (2007)
7. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, ISBN: 0-262-19398-1 (1998)
8. Singh, W S., Litman, D., Walker, M.:Reinforcement Learning for Spoken Dialogue Systems. Advances in Neural Information Processing Systems 12, MIT Press (2000)
9. Allen, J.F., Byron, D.K., Dzikovska, M., Ferguson, G., Galescu, L., Stent, A.: Towards Conversational Human Computer Interaction. AI Magazine, 22(4):27–37 (2001).

10. Ducatel, K., Bogdanowicz, M., Scapolo, F., Leijten, J., Burgelman, J-C.: Scenarios for Ambient Intelligence in 2010. IST Advisory Group Report. IPTSSeville (Institute for Prospective Technological Studies) (2001)
11. Filipe, P., Mamede, N.: A Framework to Integrate Ubiquitous Knowledge Modeling. In Proceedings of the 5th International Conference on Language Resources and Evaluation (2006)
12. Filipe, P.: Dynamic Integration of Artifacts in Dialogue Systems. PhD Thesis, IST - Technical University of Lisbon (2007)
13. Flycht-Eriksson, A., Jönsson, A.: Dialogue and Domain Knowledge Management in Dialogue Systems. In Proceedings of the 1st SIGdial Workshop on Discourse and Dialogue, Hong Kong (2000)
14. Fensel, D., Benjamins, V., Motta, E., Wielinga, B.: UPML: A Framework for Knowledge System Reuse. In Proceedings of the 16th International Joint Conference on Artificial Intelligence (1999)
15. Filipe, P., Morgado, L., Mamede, N.: An Adaptive Domain Knowledge Manager for Dialogue Systems. In 9th International Conference on Enterprise Information Systems (2007)
16. Filipe, P., Mamede, N.: Hybrid Knowledge Modeling for Ambient Intelligence. In Proceedings of the 9th Workshop User Interfaces for All (ERCIM-UI4ALL) Special Theme: "Universal Access in Ambient Intelligence Environments" (2006)