DINASTI: Dialogues with a Negotiating Appointment Setting Interface

Layla El Asri[1,2], Romain Laroche[1], Olivier Pietquin[3]
[1]Orange Labs / Issy-les-Moulineaux, France
[2]UMI 2958 GeorgiaTech-CNRS / Metz, France
[3]University Lille 1, LIFL (UMR 8022 CNRS/Lille 1) - SequeL team / Lille, France
layla.elasri@orange.com romain.laroche@orange.com olivier.pietquin@univ-lille1.fr

Abstract

This paper describes the DINASTI (DIalogues with a Negotiating Appointment SeTting Interface) corpus, which is composed of 1734 dialogues with the French spoken dialogue system NASTIA (Negotiating Appointment SeTting InterfAce). NASTIA is a reinforcement learning-based system. The DINASTI corpus was collected while the system was following a uniform policy. Each entry of the corpus is a system-user exchange annotated with 120 automatically computable features.The corpus contains a total of 21587 entries, with 385 testers. Each tester performed at most five scenario-based interactions with NASTIA. The dialogues last an average of 10.82 dialogue turns, with 4.45 reinforcement learning decisions. The testers filled an evaluation questionnaire after each dialogue. The questionnaire includes three questions to measure task completion. In addition, it comprises 7 Likert-scaled items evaluating several aspects of the interaction, a numerical overall evaluation on a scale of 1 to 10, and a free text entry. Answers to this questionnaire are provided with DINASTI. This corpus is meant for research on reinforcement learning modelling for dialogue management.

Keywords: Spoken Dialogue Systems, Reinforcement Learning, Corpus

1. Introduction

In a Spoken Dialogue System (SDS), the dialogue manager controls the behaviour of the system by choosing which dialogue act to perform according to the current state of the dialogue. Adaptive SDS now integrate data-driven statistical methods to optimise dialogue management. Among these techniques, Reinforcement Learning (RL) (Sutton and Barto, 1998; Levin et al., 1997; Williams and Young, 2007; Pietquin and Dutoit, 2006) compares and assesses management strategies with a numerical reward function. Since this function serves as a dialogue quality evaluator, it must take into account all the different variables which come into play in dialogue success. SDS evaluation might be used to emphasise these variables (Lemon and Pietquin, 2012). Indeed, a promising way to design a reward function is to deduce it after having carried a user evaluation campaign on the SDS. Another fundamental issue in modelling RL-based systems is the definition of the state space because it implies selecting relevant dialogue features and dealing with the fact that some are continuous (Paek, 2006). This paper proposes a corpus to support research on reward function and state space modelling for adaptive SDS. The corpus, named DINASTI (DIalogues with a Negotiating Appointment SeTting Interface), includes many automatically computable features to test and compare models for representing the state space. The corpus is also accompanied by an evaluation for each dialogue to support research on reward function modelling.

NASTIA (Negotiating Appointment SeTting InterfAce) is a French[1] Spoken Dialogue System (SDS) for scheduling an appointment with an engineer in case of landline dysfunction. Appointment scheduling systems were previously designed and evaluated during the CLASSiC EU FP7 project[2] (Laroche et al., 2011) about machine learning optimisation for SDS. This evaluation served to design NASTIA so that dialogue management would be more suited to user demands concerning the appointment scheduling task. The SDS was tested on 1734 scenario-based dialogues with 385 volunteers who were asked to interact at most 5 times with the system. To allow comparison, the same experimental protocol as the one that served to evaluate the appointment scheduling systems designed during CLASSiC was followed. This experimental protocol is recalled in this paper.

NASTIA's dialogue manager is an RL agent implemented as a Module-Variable Decision Process (MVDP) (Laroche et al., 2009). The dialogue manager is an automaton. A node in the automaton is a dialogue phase. During a phase, the behaviour of the system is either hard-coded or RL-based. In the latter case, the system must choose between different actions according to dialogue history. For example, NASTIA can try different procedures to carry out the appointment scheduling task, apply different speech recognition rejection recovering strategies, *etc*. During corpus collection, system actions were picked randomly following a uniform probability law. Consequently, the possible actions at a given dialogue phase were visited approximately the same number of times.

After each dialogue, the tester was asked to fill in a questionnaire. This questionnaire measures subjective task completion. It also contains 7 Likert-scaled items to evaluate various aspects of the dialogue such as efficiency, fluidity or easiness to follow. Finally, it asks the user to give a numerical overall evaluation on a scale of 1 to 10 and express remarks about the dialogue or the system in general.

The DINASTI corpus contains 21587 user-system exchanges. Each exchange is annotated with 120 automatically computable features. Answers to the evaluation ques-

[1]In all that follows, system utterances and user answers are translated from French

[2]Computational Learning in Adaptive Systems for Spoken Conversation, http://www.classic-project.org/

Dialogue phase	System utterance	User utterance
Greetings	Welcome to the appointment scheduling service. An engineer must intervene to repair your line.	
Strategy choice Action: System Initiative		
Ask for day	On which day are you available ?	Friday, the 16th.
Unavailability	There are no available slots on Friday the 16th.	
Give Information Action: Yes	This week, only Thursday morning and Saturday afternoon are available.	
Strategy choice Action: User Initiative		
Open question	When would you like to book an appointment ?	Thursday the 15th during the morning.
Appointment confirmation	Please confirm the following appointment with yes or no: Thursday, the 15th of July, during the morning.	Yes
Goodbye	Your appointment has been recorded. Thank you for using the service. Goodbye.	

Table 1: Example of a dialogue with NASTIA. Dialogue phases containing a point of choice are in bold type. The action chosen by the point of choice is below the phase name.

tionnaire are also provided with the corpus.

This paper is organised as follows. First, Section 2. describes NASTIA. The experimental protocol applied to collect the dialogues in the corpus is then presented in Section 3.. Section 4. introduces the evaluation questionnaire. Then, the features selected for corpus annotation are listed in Section 5.. Finally, Section 6. discusses potential usage of DINASTI.

2. NASTIA

A said in the introduction, the behaviour of NASTIA's dialogue manager at certain phases of the dialogue is RL-based. An RL-based phase includes one or several points of choice. A point of choice chooses an action among a set of possibilities according to its current internal state (corresponding to its belief of the dialogue state).

First, NASTIA has a point of choice where it chooses between several negotiation strategies. Three approaches were implemented: User Initiative (UI), System Initiative (SI) and List of Availabilities (LA). UI gives the *dialogue initiative* to the user, asking her/him: "When would you like to book an appointment?". SI is a more conservative strategy where the user is asked to stipulate constraints (week, day, half-day) until only one available slot is identified or user constraints stop matching system availabilities. Finally, LA consists of proposing a list of four availabilities to the user. The user is asked to interrupt the listing once a suitable appointment has been proposed. An example of dialogue illustrating the SI and UI strategies is given in Figure 1. Dialogue phases which contain a point of choice are in bold type.

NASTIA has four other points of choice. The second point of choice decides the help message to play after a user has requested it. NASTIA has three possibilities: recall the dialogue context; give to the user the possibility to cancel the help command then recall the dialogue context then recall the available commands (repeat and help); give to the user the possibility to cancel the help command then recall the dialogue context.

The third point of choice deals with the confirmation strategy. After the user has proposed an appointment date, the system chooses between three confirmation strategies. First it can choose not to ask for a confirmation. Then, the implicit confirmation strategy simply consists of repeating what was understood. Finally, following the explicit strategy, NASTIA asks "I understood you were available on [understood date]. Is it correct?".

The fourth point of choice is visited after a speech recognition rejection or a user time out. The SDS may play a help message or inform the user that she/he were not understood/heard and wait for her/him to repeat/say something.

The fifth point of choice decides if the system should provide information about its calendar after an appointment setting failure or after the user has expressed some constraints. For instance, in the dialogue in Table 1, the user says he is available on Friday the 16th, which does not match system availabilities. NASTIA then informs the user that the week of the 16th, only Thursday morning and Saturday afternoon are available.

3. Corpus collection

3.1. Recruitment

All volunteers to the experiment were Orange employees recruited by Email. The first recruitment campaign received 627 answers. An Email was then sent to these subscribers with 5 hyperlinks. Each hyperlink was associated with a code to make sure each call was unique. A code was composed of the call identifier (5 digits) and the scenario number (2 digits). A last digit was added for Cyclic Redundancy Check (CRC).

A user guide was attached to the Email. It explained the scenario, how to make a call and then fill in the question-

naire. After clicking one of the links, the user was sent to a web page explaining the scenario which was the following: *Today is Monday, July 12th and your landline is non-functional. After it diagnosed that the intervention of an engineer on site was required, the technical service has redirected you to a spoken dialogue system to book an appointment. Your aim is to set an appointment at one of the available slots on the following calendar.*
Then the user was displayed a calendar as the one shown on Figure 1.
Each calendar corresponded to a scenario. To enable comparison, the dialogue scenarios were the same as the ones designed for the tests of CLASSiC Systems 2, 3 and 4 (Laroche et al., 2011). 12 scenarios were uniformly distributed among the participants. Each scenario was characterised by a system and a user difficulty (going from 1 to 4). These levels of difficulty were computed according to the first common availability. For example, if the first availability of the user was the second availability of the system, then the scenario was of difficulty 1 for the user. Under the calendar was indicated the phone number to call. This phone number connected to a DTMF front-end system that asked the user to enter the code. The front-end system then transmitted the call to NASTIA after having extracted from the code the following information: the call identifier to be written in the logs and the scenario number so that NASTIA could download the corresponding calendar.
After performing the call, users filled in the evaluation questionnaire on the same page where the calendar was displayed.
In total, 385 participants made 1 to 5 calls, with an average of 4.6 calls per participant. This resulted in 1734 dialogues and 21587 system-user exchanges, among which 7508 are decision turns, *i.e.* turns where the system needed to choose amongst several actions.

4. Evaluation

The evaluation questionnaire is translated in Appendix A. Questions 1 and 2 required a yes/no answer. For Question 3, the user had to select the appointment date if an appointment had been set. Questions 4 to 10 were evaluated

Figure 1: Example of user calendar for the scenario-based dialogues. The green slots are the available ones.

according to a six-point Likert scale: completely disagree, disagree, mostly disagree, mostly agree, agree, completely agree. Another option was added to Question 5 in case there had been no speech recognition mistakes. For Question 11, the users were asked to rate the dialogue on a scale of 1 to 10. Finally, Question 12 was free text, to report any problem or give a general opinion on the system.

5. Corpus annotation

Corpus annotation was performed on the basis of the parameters described by Schmitt et al. (Schmitt et al., 2008). This feature set is composed of features returned by the speech recognition, natural language understanding and dialogue management modules. The features were shown to be relevant to predict the interaction quality with an SDS (Schmitt et al., 2011; Ultes and Minker, 2013) and to identify problematic dialogues (Walker et al., 2002).
The DINASTI corpus only includes computable features because it is designed to enable online RL (Daubigney et al., 2011; Gašić et al., 2011). Indeed, a behaviour learnt on the scenario-based corpus will not be perfectly suited for real-life situations. There is a difference of commitment between a user who is pretending to book an appointment and one who is really facing problems with her/his landline (Laroche et al., 2011). Moreover, real users' availabilities are not likely to be distributed according the same patterns as the ones proposed in our scenarios.
The feature set is described in Table 2. System and user dialogue acts are described in Tables 3 and 4 in Appendix B.
The #RuleUsage and #TagUsage features are returned by the system's Natural Language Understanding (NLU) component. NLU in NASTIA is rule-based: 38 grammar rules can be triggered to understand the user's utterance. NLU only works with a limited number of concepts which are the tags of the #TagUsage feature.

6. Corpus usage

The corpus was collected for manifold purposes. First, DINASTI may be used for testing feature selection optimisation algorithms. In this line of research, Paek and Chickering (Paek and Chickering, 2005) modelled dialogue management as an influence diagram and used a Bayesian structure search algorithm to infer the relevant features for reward prediction. Another method was proposed by (Rieser and Lemon, 2011) who used correlation-based feature selection to model the state space of a car-embedded SDS. (Li et al., 2009) and (Chandramohan et al., 2010) also integrated feature selection in RL algorithms for dialogue management. We release the DINASTI corpus to encourage more research on the subject: a feature set may be inferred at each point of choice to optimally predict user satisfaction.
Secondly, another crucial parameter of RL is the reward function. Inverse Reinforcement Learning (IRL, (Russell, 1998; Ng and Russell, 2000; Klein et al., 2012)) learns a reward function from a set of examples where a learning agent follows an optimal policy. Paek and Pieraccini (Paek and Pieraccini, 2008) suggested to apply IRL on Human-Human dialogues to learn a reward function that would pro-

Feature name	Feature signification
#DecisionTurns	Number of decisions up to this turn. A decision turn is a dialogue turn where the system has to choose between different possible actions.
#SystemTurns	Number of system turns up to this turn.
#UserTurns	Number of user turns up to this turn.
#RePrompts	Number of re-prompts up to this turn.
#ASRRejections	Number of speech recognition rejections up to this turn.
#TimeOuts	Number of user time outs up to this turn.
ASRConfidence	Mean ASR confidence score up to this turn.
#SystemQuestions	Number of system questions up to this turn.
#HelpMessages	Number of prompted help messages up to this turn.
WPST	Mean number of words per system turn up to this turn.
WPUT	Mean number of words per user turn up to this turn.
DD	Dialogue duration in seconds.
#SystemDialogueActs (18 features)	Number of times each system dialogue act has been performed. System dialogue acts are: SDA_GREETING, SDA_GOODBYE, SDA_INFORM, SDA_REPAIR, SDA_ASK_DAY, SDA_ASK_DATE, SDA_ASK_OTHER_PERIOD, SDA_NOT_AVAILABLE, SDA_ASK_CONFIRMATION, SDA_DATE_PROPOSITION, SDA_ASK_WHICH, SDA_ASK_PERIOD, SDA_LIST, SDA_ASK_WEEK SDA_ASK_OTHER_WEEK, SDA_REPEAT, SDA_ERROR
#UserDialogueActs (9 features)	Number of times each user dialogue act has been performed. User dialogue acts are: UDA_NOINPUT, UDA_NOMATCH, UDA_CONFIRM, UDA_PROPOSE_DATE, UDA_ASK_HELP, UDA_CONTRADICT UDA_DO_NOT_KNOW, UDA_ASK_REPEAT, UDA_SAY_NONE
#SystemNegociationStrategies (4 features)	Number of times each negotiation strategy has been chosen: SNS_LIST, SNS_SYS_INIT, SNS_USER_INIT, SNS_SYS_PROPOSITION
#RuleUsage (38 features)	Number of times each grammar rule has been triggered
#TagUsage (39 features)	Number of times each word tag has been recognised

Table 2: Features in the DINASTI corpus.

vide the SDS with the ability to mimic human operators behaviour. Following this idea, (Boularias et al., 2010) learnt a reward function for a POMDP-based SDS in a Wizard-of-Oz (WOZ) setting, where a human expert takes the place of the dialogue manager. The expert is provided with the user utterance understood by the natural language processing module and, given this noisy written entry, s/he chooses the next action of the system. Another way to learn a reward function is to infer it from a set of evaluated dialogues (Walker, 2000; Sugiyama et al., 2012; El Asri et al., 2012). User overall evaluation might be used as a reward function to learn an optimal policy for the system but it was shown in (El Asri et al., 2013) that learning was accelerated by inferring from these scores a diffuse reward function. Such a function gives a reward after each system decision instead of waiting for the end of the dialogue. Besides, as said in Section 5., it is important to have a function that can be used online to adapt to real users behaviour.

Finally, research on user simulation (Schatzmann et al., 2006; Pietquin et al., 2009; Chandramohan et al., 2011) may also be carried on the corpus. The negotiation task implies unusual constraints on user simulation design. Indeed, it is not a slot-filling task with a static goal. In DINASTI, the user's goal might change during the dialogue, when an appointment is unavailable. Besides, the user may take over the task or dialogue initiative at any point of the dialogue, which is interesting for user adaptivity and expertise modelling research.

7. Conclusion

This document described a corpus of annotated and evaluated dialogues dedicated to research on reinforcement learning modelling. The potential applications of the corpus are state space and reward function modelling as well as user simulation design. This corpus is now under preparation for publication during the course of next year.

8. Acknowledgements

The authors would like to thank all the members of the NADIA team in Orange for their help during the conception of the test of the system. The authors would also like to thank the *Innovation primeur* team in Orange for their help with the experimentation and all the volunteers who interacted with the system. Finally, we also thank the anonymous reviewers for their valuable comments on the submitted abstract.

9. References

Boularias, A., Chinaei, H. R., and Chaib-draa, B. (2010). Learning the reward model of dialogue pomdps from data. In *Proc. of NIPS*.

Chandramohan, S., Geist, M., and Pietquin, O. (2010). Optimizing Spoken Dialogue Management with Fitted Value Iteration. In *Proc. of Interspeech*.

Chandramohan, S., Geist, M., Lefèvre, F., and Pietquin, O. (2011). User simulation in dialogue systems using inverse reinforcement learning. In *Proc. of Interspeech*.

Daubigney, L., Gašić, M., Chandramohan, S., Geist, M., Pietquin, O., and Young, S. (2011). Uncertainty management for online optimisation of a pomdp-based large-scale spoken dialogue system. In *Proc. of Interspeech*, pages 1301–1304.

El Asri, L., Laroche, R., and Pietquin, O. (2012). Reward function learning for dialogue management. In *Proc. of STAIRS*.

El Asri, L., Laroche, R., and Pietquin, O. (2013). Reward shaping for statistical optimisation of dialogue management. In *Proc. of SLSP*.

Gašić, M., Jurčíček, F., Thomson, B., Yu, K., and Young, S. (2011). On-line policy optimisation of spoken dialogue systems via live interaction with human subjects. In *Proc. of IEEE ASRU*.

Klein, E., Geist, M., Piot, B., and Pietquin, O. (2012). Inverse Reinforcement Learning through Structured Classification. In *Proc. of NIPS*.

Laroche, R., Putois, G., Bretier, P., and Bouchon-Meunier, B. (2009). Hybridisation of expertise and reinforcement learning in dialogue systems. In *Proc. of Interspeech*.

Laroche, R., Putois, G., Bretier, P., Aranguren, M., Velkovska, J., Hastie, H., Keizer, S., Yu, K., Jurčíček, F., Lemon, O., and Young, S. (2011). D6.4: Final evaluation of classic towninfo and appointment scheduling systems. Technical report, CLASSIC Project.

Lemon, O. and Pietquin, O., (2012). *Data-Driven Methods for Adaptive Spoken Dialogue Systems*. Springer.

Levin, E., Pieraccini, R., and Eckert, W. (1997). Learning dialogue strategies within the markov decision process framework. In *Proc. of IEEE ASRU*.

Li, L., Williams, J. D., and Balakrishnan, S. (2009). Reinforcement learning for dialog management using least-squares policy iteration and fast feature selection. In *Proc. of Interspeech*.

Ng, A. Y. and Russell, S. (2000). Algorithms for inverse reinforcement learning. In *Proc. of ICML*, pages 663–670.

Paek, T. and Chickering, D. M. (2005). The markov assumption in spoken dialogue management. In *Proc. of SIGdial Workshop on Discourse and Dialogue*, pages 35–44.

Paek, T. and Pieraccini, R. (2008). Automating spoken dialogue management design using machine learning : An industry perspective. *Speech Communication*, 50.

Paek, T. (2006). Reinforcement learning for spoken dialogue systems: Comparing strengths and weaknesses for practical deployment. In *Proc. of Interspeech, Dialog-on-Dialog Workshop*.

Pietquin, O. and Dutoit, T. (2006). A probabilistic framework for dialog simulation and optimal strategy learning. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(2):589 – 599.

Pietquin, O., Rossignol, S., and Ianotto, M. (2009). Training Bayesian networks for realistic man-machine spoken dialogue simulation. In *Proc. of IWSDS 2009*.

Rieser, V. and Lemon, O. (2011). Learning and evaluation of dialogue strategies for new applications: Empirical methods for optimization from small data sets. *Computational Linguistics*, 37.

Russell, S. (1998). Learning agents for uncertain environments (extended abstract). In *Proc. of COLT*.

Schatzmann, J., Weilhammer, K., Stuttle, M., and Young, S. (2006). A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *The Knowledge Engineering Review*, 21(2).

Schmitt, A., Hank, C., and Liscombe, J. (2008). Detecting problematic dialogs with automated agents. In *Perception in Multimodal Dialogue Systems*, Lecture Notes in Computer Science, pages 72–80.

Schmitt, A., Schatz, B., and Minker, W. (2011). Modeling and predicting quality in spoken human-computer interaction. In *Proc. of SIGDIAL*.

Sugiyama, H., Meguro, T., and Minami, Y. (2012). Preference-learning based Inverse Reinforcement Learning for Dialog Control. In *Proc. of Interspeech*.

Sutton, R. S. and Barto, A. G., (1998). *Reinforcement Learning. An introduction*. MIT Press.

Ultes, S. and Minker, W. (2013). Improving interaction quality recognition using error correction. In *Proc. of SIGDIAL*.

Walker, M. A., Langkilde-Geary, I., Hastie, H. W., Wright, J., and Gorin, A. (2002). Automatically training a problematic dialogue predictor for a spoken dialogue system. *Journal of Artificial Intelligence Research*, 16:293–319.

Walker, M. A. (2000). An application of reinforcement learning to dialogue strategy selection in a spoken dialogue system for email. *Journal of Artificial Intelligence Research*, 12:387–416.

Williams, J. D. and Young, S. (2007). Partially observable markov decision processes for spoken dialog systems. *Computer Speech and Language*, 21:231–422.

Appendix A: Evaluation questionnaire

1. Have you booked an appointment?
2. Was the appointment booked on one of your available slots?
3. When did you book the appointment?
4. During your dialogue with the system, you knew what to say.
5. You could easily recover from system misunderstandings.
6. Understanding the system was easy.
7. The system provided enough information for the dialogue to be easy to follow.
8. The dialogue with the system was efficient.
9. The dialogue with the system was fluid.
10. The system was concise.

11. Overall evaluation.
12. Do you have any remarks or comments?

Appendix B: System and user dialogue acts

Feature name	Feature signification
SDA_GREETING	Greet the user
SDA_GOODBYE	Say goodbye to the user
SDA_INFORM	Provide information about the system's calendar
SDA_REPAIR	Recover from ASR rejection or user time out
SDA_ASK_DAY	Ask the user on which day they are available
SDA_ASK_DATE	Ask the user when she/he is available
SDA_ASK_OTHER_PERIOD	Ask the user if she/he is available during the morning or the afternoon, after she/he has refused an appointment the same day respectively during the afternoon or the morning.
SDA_NOT_AVAILABLE	Inform the user that a slot in not available
SDA_ASK_CONFIRMATION	Ask the user for a confirmation
SDA_DATE_PROPOSITION	Propose a slot to the user
SDA_ASK_WHICH	After having proposed a list of 4 slots, ask the user if a slot is suitable
SDA_ASK_PERIOD	Ask the user if they are available during the morning or the afternoon
SDA_LIST	Propose a list of available slots
SDA_ASK_WEEK	Ask the user on which week they are available
SDA_ASK_OTHER_WEEK	Ask the user if they are available the other week
SDA_REPEAT	Repeat the last system prompt
SDA_ERROR	Inform the user an error has occurred

Table 3: System dialogue acts.

Feature name	Feature signification
UDA_NOINPUT	User time out
UDA_NOMATCH	ASR rejection
UDA_CONFIRM	The user confirms what the system has understood
UDA_PROPOSE_DATE	The user expresses constraints
UDA_ASK_HELP	The user requests the help section
UDA_CONTRADICT	The user contradicts what the system has understood
UDA_ASAP	The user says she/he wants to set an appointment as fast as possible
UDA_ASK_REPEAT	The user asks the system to repeat
UDA_SAY_NONE	The user answers "none" after the system has proposed four slots

Table 4: User dialogue acts recognised by NASTIA.