

NASTIA: Negotiating Appointment Setting Interface

Layla El Asri^{1,2}, Rémi Lemonnier¹, Romain Laroche¹, Olivier Pietquin³, Hatim Khouzaimi^{1,4}

¹Orange Labs / Issy-les-Moulineaux, France

²UMI 2958 GeorgiaTech-CNRS / Metz, France

³University Lille 1, LIFL (UMR 8022 CNRS/Lille 1) - SequeL team / Lille, France

⁴Laboratoire d'Informatique d'Avignon-CERI / Avignon, France

{layla.elasri,romain.laroche,hاتم.khouzaimi}@orange.com remi.lemonnier@ens-cachan.fr olivier.pietquin@univ-lille1.fr

Abstract

This paper describes a French Spoken Dialogue System (SDS) named NASTIA (Negotiating Appointment SeTting InterfAce). Appointment scheduling is a hybrid task halfway between slot-filling and negotiation. NASTIA implements three different negotiation strategies. These strategies were tested on 1734 dialogues with 385 users who interacted at most 5 times with the SDS and gave a rating on a scale of 1 to 10 for each dialogue. Previous appointment scheduling systems were evaluated with the same experimental protocol. NASTIA is different from these systems in that it can adapt its strategy during the dialogue. The highest system task completion rate with these systems was 81% whereas NASTIA had an 88% average and its best performing strategy even reached 92%. This strategy also significantly outperformed previous systems in terms of overall user rating with an average of 8.28 against 7.40. The experiment also enabled highlighting global recommendations for building spoken dialogue systems.

Keywords: Spoken Dialogue Systems, Reinforcement Learning, Evaluation

1. Introduction

NASTIA (Negotiating Appointment SeTting InterfAce) is a French Spoken Dialogue System (SDS). Its task is to schedule an appointment with a user who needs the intervention of an engineer on site. This SDS pursues the research on the appointment scheduling task, started by Lacson (2004) and continued during the CLASSiC EU FP7 project¹ (Laroche et al., 2011). Appointment scheduling is both a slot-filling and a negotiation task. NASTIA's contribution to it lays in the fact that it is able to adapt its negotiation strategy according to dialogue progress. This adaptation is made with Reinforcement Learning (RL) (Sutton and Barto, 1998), which was shown to be suited to negotiating SDS (Heeman, 2009; Georgila and Traum, 2011). NASTIA can choose between three negotiation strategies, more or less conservative, depending on the course of the dialogue. Other choices such as confirmation strategies are made with RL.

An RL-based dialogue manager chooses between different actions depending on its current state, which models the manager's beliefs about the dialogue (Lemon and Pietquin, 2012). These decisions are based on rewards distributed to the system after each action, or at the end of a dialogue. The manager seeks for the most rewarding sequence of decisions. Hand-crafting the state space and the reward function of an SDS raises several issues (Paek, 2006). The reward function defines the task of the system. Following (Walker et al., 1997), the reward here is defined so that the system learns to maximise user satisfaction. Hand-crafting a state space that enables learning to distinguish between different levels of user satisfaction is challenging since it requires selecting relevant features and finding an appropriate model to discretise the state space (El Asri et al., 2012).

NASTIA's development follows a novel approach to set up the RL parameters of an SDS. NASTIA's dialogue man-

ager is based on an automaton, modelling the dialogue as a sequence of phases. A phase might require one or several decisions be taken. Dialogue logics is ensured by the automaton-based structure. Therefore, each action can be tried by the system without the dialogue be irreversibly harmed. It is proposed here to collect a set of evaluated dialogues with a random, uniform choice of actions. Each dialogue is a path in the automaton. From this set of paths and the corresponding evaluations, it is possible to infer the relevant features for decision making and a reward function.

Such an evaluation campaign has been carried and this paper reports its results. 1734 dialogues were collected with 385 users interacting with the system at most five times (El Asri et al., 2014). After each dialogue, the user filled in an evaluation questionnaire about the dialogue. The evaluation protocol was the same as the one set up for the evaluation of the CLASSiC systems. The conclusion of this evaluation is twofold. First, with a uniformly random behaviour, NASTIA outperformed all the systems designed during CLASSiC in terms of user satisfaction. This means that all the strategies implemented in the system were relevant for the appointment scheduling task. Then, the comments left by the testers show a need for RL. Indeed, some users expressed a preference for one particular strategy over the other two and the system could learn to switch between strategies according to user behaviour.

The rest of this paper is organised as follows. First, Section 2. introduces the appointment scheduling task and the SDS previously built for this task. Then, Section 3. describes NASTIA. Section 4. presents the results of the experiment and compares these results to the ones of the systems designed during the CLASSiC project. Finally, Section 5. discusses these results and proposes directions for future work.

¹Computational Learning in Adaptive Systems for Spoken Conversation, <http://www.classic-project.org/>

2. Related work

2.1. The SCHEDULER

The SCHEDULER (Lacson, 2004) is an SDS dedicated to the management of medical appointments. The first action performed by the SCHEDULER is to ask for the patient's name to check s/he is already in hospital records. Then the user may choose to create, cancel or check an appointment. For the creation task, the user has to indicate the practitioner's name and a day. If some information are missing and the system cannot query the database, the user is asked for the missing items. Users may also specify a time preference and if they do not, the system simply proposes its first availability for the given day. The SCHEDULER was evaluated according to the three following criteria: task success, task ease and difficulties encountered during the dialogue. Nevertheless, the evaluation only concerned 15 calls which were not scenario-based so user constraints were not modelled in this experiment.

2.2. Systems designed during CLASSiC

The CLASSiC project gave birth to three systems enabling users to schedule an appointment with an engineer in the case of a dysfunction of their landline. These three systems are referred to as Systems 2, 3 and 4 in (Laroche et al., 2011).

System 2 (Jurčiček et al., 2010) was a state-of-the-art POMDP-based (Partially Observable Markov Decision Processes) SDS (Sondik, 1971; Williams and Young, 2007). After each user input, the dialogue manager received the N-best list of semantic hypotheses, updated its dialogue state and chose its next dialogue act accordingly. With this system, the user could provide one or several constraints such as day of week, day of month *etc.* Then the system asked the user to refine her/his constraints until it could identify a unique available slot or it determined that there is no available slot matching these constraints. The system could also provide information about its available slots given the user's constraints or offer an alternative if user's constraints did not correspond to any available appointment.

System 3 was also Reinforcement Learning-based (RL). In this case, RL was cast as a Module-Variable Decision Process (MVDP, (Laroche et al., 2009)). This system was designed to assess the influence of Text-To-Speech (TTS) prosody on users' behaviour. Each system utterance could be synthesised with one of these intonations: calm, neutral or dynamic. The negotiation strategy of System 3 was hand-coded. The system started each dialogue proposing to the user its first availability. Then, if the user rejected the proposition, the system asked for her/his first availability. If this was not a free slot in the system's calendar, the system proposed its next availabilities and so on until an appointment was booked or the system had no more propositions to make.

System 4, on the other hand, was not RL-based. The system either proposed a time slot or asked for different constraints such as week, day, half-day until it was able to make a proposition matching the constraints or reject the constraints. The system chose between the two strategies on the basis of the number of remaining slots. If there were

2 or fewer time slots, it proposed a time slot. Otherwise, it asked the user for her/his constraints. If so, the system asked the user to specify turn by turn a day, a week and a half-day. First, the system asked for the most restricting parameter, *i.e.* the one that minimised the number of questions to ask to the user.

These three systems were tested and compared on scenario-based dialogues. The main results of this study will be discussed in Section 4..

These experiments enabled to point out the parts of the appointment scheduling process that needed to be improved. The conception of NASTIA resulted from the analysis provided in (Laroche et al., 2011).

3. NASTIA

3.1. Issues previously identified

Many of the problems detailed in the CLASSiC evaluation (Laroche et al., 2011) could be explained in terms of uncooperative behaviour of the system. For instance, it was noticed that users were sometimes confused by system feedbacks. Let us take the example of a user saying s/he would like to book an appointment on Friday afternoon. In this case, most of the time, the user meant the upcoming Friday afternoon.

Yet, in accordance with Grice's quality maxim (Grice, 1989), systems 3 and 4 would not make any assumption on the desired week. Thus, if the first appointment available was Friday afternoon of the following week, both systems 3 and 4 would have directly proposed this appointment without stipulating that the upcoming Friday was not available. Users tended to distrust speech recognition so, in this case, they often chose to refuse the proposition and repeat their request.

NASTIA disambiguates these cases prompting the user with an implicit confirmation. To such a user utterance, NASTIA would answer *Friday the 16th* to let the user know that it was supposing they were meaning the upcoming Friday. This new formulation respects Grice's quantity maxim as it provides to the user the necessary amount of information for them to understand the course of the dialogue.

Other modifications of the same nature were made resulting in many prompts being reformulated to move towards a better accordance with the Gricean cooperativity principles and make the system less of a black box to the user.

3.2. Dialogue modelling in NASTIA

Appointment scheduling is modelled as a slot-filling task with three parameters: day, week and half-day (morning or afternoon). NASTIA's dialogue manager is based on a finite state machine. Each node of the machine is a dialogue phase. Dialogue phases in NASTIA are for example: Welcome, Confirm, Ask_open.Question, Ask_For_Day, Recovery (from speech recognition rejection or user time out), *etc.*

RL was integrated into this automaton with the MVDP hybrid framework (Laroche et al., 2009). Following this framework, a dialogue phase may contain one or several point(s) of choice. NASTIA contains five points of choice in five different phases.

Initiative strategy	User Initiative (UI); System Initiative (SI); List of Availabilities (LA)
ASR rejections; User inactivity	Play a help message; Tell the user their utterance was not understood
Confirmation strategy	Explicit confirmation; Implicit confirmation; No confirmation
System calendar information	Give information; Do not give information
Help message	Recall dialogue context; Give the possibility to cancel + Recall dialogue context + Recall available commands; Give the possibility to cancel + Recall dialogue context

Table 1: Actions of the appointment setting system.

The first point of choice determines the negotiation strategy. The User Initiative (UI) strategy consists of asking the user: “When would you like to book an appointment?”. System Initiative (SI) asks the user which day, which week and which half-day in three different dialogue turns. The order of the questions is decided as for System 4 (see previous section). In addition to these classical strategies, a third option was implemented where the system directly proposes a List of Availabilities (LA) to the user, waiting for her/him to interrupt the listing after an adequate appointment has been proposed. This last option was inspired by recent work on incremental dialogue management (Schlangen and Skantze, 2011). Incremental dialogue management and barge-in in particular are at the heart of current research on SDS and more and more complex models can be found in the literature (Selfridge et al., 2013). Interruptions are common in human dialogue (Strombergsson et al., 2013), which makes it much more interactive than turn-taking human/machine conversation. Thus, including incrementality in a dialogue system is likely to make it more reactive and human-like.

Figure 1 describes the way NASTIA carries a negotiation to set an appointment. The system chooses which strategy to follow at the beginning of each dialogue and after each appointment setting failure. This leaves to NASTIA the opportunity to adapt its way of realising the task in function of the course of the dialogue as it was proposed for instance by (Chu-Carroll, 2000) and (Litman and Pan, 2002). If the system picks out the LA strategy, four available slots are

proposed to the user. If the user has not interrupted the system after the fourth proposition, the system asks the user to confirm that none of the slots is suitable and then the negotiation strategy is decided upon again, the system may either keep to this strategy or switch to UI or SI. While the system lists its availabilities, the user can also interrupt the system to propose some constraints. For instance, if the system starts listing slots for a week during which the user is not available, the user has the possibility to interrupt the system and say “next week”. If so, the system switches to SI and asks for the missing slots.

The second point of choice concerns contextual help generation. The user may express a help request at any moment of the dialogue. If so, the system may combine three components of help messages:

- (a) Tell the user: “You have required the help section” and let them the possibility to answer “no” in case the system misunderstood the user’s request.
- (b) Recall the current context of the dialogue (e.g. “You were asked when you would like to make an appointment”) and tell the user what they can say (e.g. “You can answer saying for instance *this Friday afternoon, this week in the morning or Monday the 19th.*”)
- (c) Recall the available commands (Repeat and Help)

NASTIA chooses amongst three combinations: (b); (a) + (b) + (c) or (a) + (b).

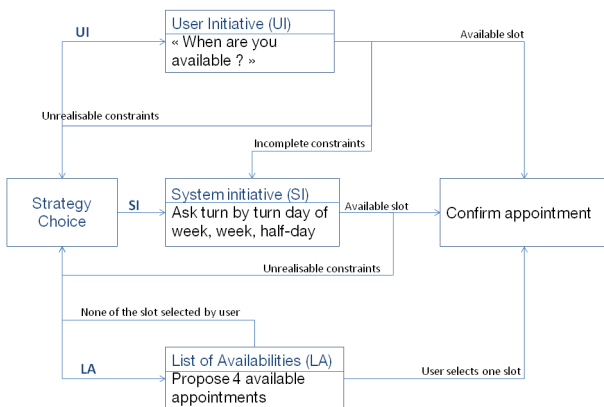


Figure 1: NASTIA’s appointment scheduling strategies.

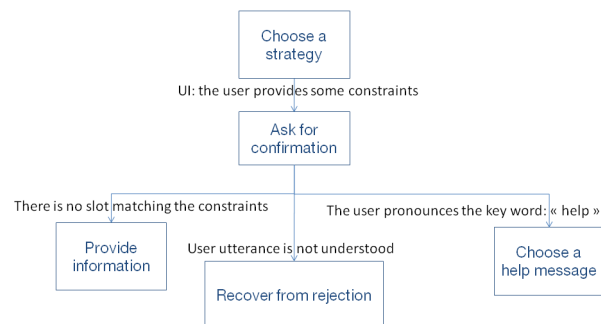


Figure 2: Example of dialogue involving all of NASTIA’s points of choice.

The third point of choice is visited after a user has proposed a time slot. NASTIA may follow three confirmation strategies. Following the first strategy, the system does not ask for any confirmation. The implicit confirmation strategy simply consists of repeating what was understood. In case the system misunderstood her/his utterance, the user can barge in to correct the system. The explicit strategy requires a yes/no answer. The system asks: "I understood you were available on [understood date]. Is it correct?"

The fourth point of choice has been implemented to compare two strategies for speech recognition rejections and user time outs recovery. The SDS may play the (b) help message or inform the user that s/he was not understood/heard so that the user repeats/says something.

Like CLASSiC System 2, NASTIA can provide information about its calendar after an appointment setting failure or after the user has expressed some constraints. This is decided by the fifth point of choice. System 2 could tell the user that there are no appointments except x and y given the constraints. During a dialogue, NASTIA keeps up to date the number of available and unavailable slots matching user constraints. If one of the two numbers goes below three, the point of choice can decide to list the available/unavailable slots. These slots might be completely defined but also days, half-days or weeks. For instance, if a user says s/he is available this week during the morning, the system may answer "This week, during the morning, Tuesday and Friday are not available". All of NASTIA's points of choice as well as their action sets are gathered in Table 1. Figure 2 illustrates the dialogue contexts in which the points of choice are visited.

4. Results

4.1. Comparison to CLASSiC Systems 2, 3 and 4

We compare the performance of NASTIA with the ones of CLASSiC's systems 2, 3 and 4 on the basis of System and User Task Completion (resp. STC and UTC), elapsed time (in seconds) and overall user evaluation on a scale of 1 to 10. These results are given in Table 2. NASTIA was evaluated on 1734 dialogues. The calls were performed by Orange collaborators. User task completion is derived from the answers to questions 1 and 2 in the questionnaire given in Appendix A. During the experiment, 12 user calendars were randomly assigned to each dialogue. For each calendar, there was only one common availability with the system. In Table 2, system task completion is equal to 1 if the right appointment has been booked and 0 otherwise.

There were 628 evaluated dialogues for System 2, 740 for System 3 and 709 for System 4. Systems 3 and 4 share the same automatic speech recognition, natural language understanding and text-to-speech components as NASTIA. During the CLASSiC experiment, these systems largely outdid System 2 concerning STC and UTC. System 3 was the one that did best in terms of overall evaluation and led to the shortest dialogues.

NASTIA performed similarly to System 3 in terms of overall evaluation. Although dialogues were in average 6 seconds longer with NASTIA, STC and UTC are significantly higher.

4.2. Strategies comparison

The system has to choose an initiative strategy at the beginning of every dialogue. In table 2, we have also computed 95% confidence intervals for the key performance indicators in function of the first decision made by the system. It shows that LA entailed significantly higher evaluations and shorter dialogues, no matter the policy followed by the system afterwards. Dialogues are shorter and the mean evaluation with LA is clearly higher than the mean rating of CLASSiC's System 3 (7.40).

As shown in Section 2., the main difference between NASTIA and Systems 3 and 4 concerns the negotiation strategy. NASTIA can try several strategies during the same dialogue. In addition to this flexibility, in case of speech recognition rejection or user time out, System 3 would tell the user that she/he had not been heard and then repeat its question. System 4 would ask the user to confirm its first then second hypothesis and if neither was accepted by the user, the system would repeat its initial question. It was observed during the CLASSiC experiments that users tended to try and barge in instead of waiting for the system to repeat its question. NASTIA leaves to the user this possibility. Finally, providing information to the user about the system's availabilities given the user's constraints is in better accordance with Grice's principles of cooperativity since the system contributes to the dialogue by providing as much information as it can according to its current beliefs.

5. Discussion

The answers to question 12 in the evaluation questionnaire in Appendix A shed light on the users' current perception of task-oriented automated spoken dialogue.

First, several users expressed the fact that they would have appreciated to be more guided during the dialogue. The approach in NASTIA is to let the user barge in at almost any moment of the dialogue. Thus, there are blanks of a few seconds after system utterances. For instance, after the system has told the user that no appointment matches her/his constraints, the system waits in case the user directly specifies new constraints. Users wrote that they did not know what to say during these blanks and they would have rather the SDS took over the dialogue more quickly. Testers are more accustomed² to hear the system say "your turn to talk" when they are supposed to interact. A similar remark was expressed by some of the users for whom the system chose the UI strategy. These users said this strategy was more comfortable to them than the LA strategy. Nevertheless, after the system has told that an appointment was not available, they would have wanted the system to switch strategy, to be more directive. This shows that users might not be ready for natural dialogue with a task-oriented SDS. Contrary to listening-oriented systems (Meguro et al., 2009), task-oriented systems are expected to be more directive and allow a narrower range of user utterances.

Another interesting point about the UI strategy is that users progressively learnt to use it. As said before, testers interacted at most five times with the system. Some users were confronted to UI more than once. They wrote that

²especially in a commercial context

System	STC	UTC	Time (sec)	Rating	Number of calls
System 2	79 ± 3%	68 ± 4%	97 ± 5	5.21 ± 0.23	628
System 3	81 ± 3%	83 ± 4%	69 ± 3	7.40 ± 0.17	740
System 4	83 ± 3%	85 ± 3%	98 ± 5	6.54 ± 0.18	709
NASTIA	88 ± 2%	92 ± 1%	75 ± 3	7.75 ± 0.09	1734
NASTIA UI	87 ± 3%	89 ± 3%	84 ± 6	7.57 ± 0.16	587
NASTIA LA	92 ± 3%	95 ± 2%	61 ± 5	8.28 ± 0.14	562
NASTIA SI	87 ± 3%	92 ± 2%	79 ± 5	7.43 ± 0.17	585

Table 2: Performance comparison between NASTIA and CLASSiC’s systems 2, 3 and 4. STC is System Task Completion and UTC, User Task Completion. Time is measured in seconds. We provide 95% confidence intervals for the mean of the binomial (STC and UTC) and the normal law (Time and Rating).

at first, they were not sure of the date format expected by the system but then they found out the day of week/day of month/half-day format was well understood by the system and enabled to fasten ruling out slots. Thus it seems important to keep the three negotiation strategies as, the more users call this system, the more comfortable they are with UI but the other two strategies are important to keep for less experimented users. The choice of strategy according to current dialogue context should be successfully learnt with reinforcement learning. If a strategy fails, dialogue history and what was observed during previous dialogues should inform NASTIA about what strategy to try next.

This shows that dialogue management requires a finer representation of the course of the dialogue than the one only relying on points of choice. To make an efficient decision, the system needs to know more than its current point of choice, it must also take into account dialogue history. Therefore, future work will consist of learning a state space representation from user ratings to apply RL and learn an optimal behaviour for the system at each point of choice, according to dialogue history. Another point encouraging the use of reinforcement learning is that, as noticed by Laroche et al. (2011), the overall rating for System 3 is higher than the one for System 4 even though the task completion is higher with System 4. Task completion is not the most important parameter to define a successful dialogue according to testers but it might not be the case for real users.

6. Conclusion

This paper discussed dialogue management for the appointment scheduling task. It described a spoken dialogue system designed for this task. The Negotiating Appointment Setting Interface (NASTIA) combines three negotiation strategies to perform appointment scheduling. This system was tested on 1734 scenario-based dialogues. This paper reports the results of this study and the general conclusions about dialogue management that were drawn from it. Future work will consist of applying reinforcement learning to find an optimal way to carry an appointment setting dialogue, according to user behaviour.

7. Acknowledgements

The authors would like to thank all the members of the NASTIA 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.

8. References

- Chu-Carroll, J. (2000). Mimic: An adaptive mixed initiative spoken dialogue system for information queries. In *Proc. of ANLP*, pages 97–104.
- 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. (2014). DINASTI: Dialogues with a Negotiating Appointment Setting Interface. In *Proc. of LREC (to be published)*.
- Georgila, K. and Traum, D. (2011). Reinforcement learning of argumentation dialogue policies in negotiation. In *Proc. of InterSpeech*.
- Grice, P., (1989). *Studies in the Way of Words*, chapter Logic and Conversation. Harvard University Press, Cambridge MA.
- Heeman, P. (2009). Representing the reinforcement learning state in a negotiation dialogue. In *Proc. of ASRU*.
- Jurčiček, F., Keizer, S., Mairesse, F., Yu, K., Young, S., Janarthanam, S., Hastie, H., Liu, X., and Lemon, O. (2010). D5.4: Proof-of-concept CLASSIC Appointment Scheduling system (“System 2”). Technical report, CLASSIC Project.
- Lacson, R. (2004). The medical appointment scheduler. In *Proc. of MEDINFO*.
- 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., Jurčiček, 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.
- Litman, D. J. and Pan, S. (2002). Designing and evaluating an adaptive spoken dialogue system. *User Modeling and User-Adapted Interaction*, 12:111–137.
- Meguro, T., Higashinaka, R., Dohsaka, K., Minami, Y., and Isozaki, H. (2009). Analysis of listening-oriented dialogue for building listening agents. In *Proc. of SIGDIAL*.
- Paek, T. (2006). Reinforcement learning for spoken dialogue systems: Comparing strengths and weaknesses for practical de-

- ployment. In *Proc. of Interspeech, Dialog-on-Dialog Workshop*.
- Schlangen, D. and Skantze, G. (2011). A general, abstract model of incremental dialogue processing. *Dialogue and Discourse*, 2:83–111.
- Selfridge, E., Arizmendi, I., Heeman, P., and Williams, J. D. (2013). Continuously predicting and processing barge-in during a live spoken dialogue task. In *Proc. of SIGDIAL*.
- Sondik, E. J. (1971). The optimal control of partially observable markov processes. Technical report, Stanford Electronics labs.
- Strombergsson, S., Hjalmarsson, A., Edlund, J., and House, D. (2013). Timing responses to questions in dialogue. In *Proc. of Interspeech*.
- Sutton, R. S. and Barto, A. G., (1998). *Reinforcement Learning. An introduction*. MIT Press.
- Walker, M., Hindle, D., Fromer, J., Fabbrizio, G., and Mestel, C. (1997). Evaluating competing agent strategies for a voice e-mail agent. In *Proc. of EuroSpeech*.
- 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?

Some appointment setting tips for how to ask over the phone. These sales appointment techniques will help you schedule calls. The main objection that an appointment-setting advisor will get is that the prospect customer has not really heard what has been said. So the customer will say to them that they are too busy, or that they have another solution in place, showing no interest in developing the conversation. To deal with objections like this, firstly ACKNOWLEDGE the customer's concerns. This article presents the design of a generic negotiation dialogue game between two or more players. The goal is to reach an agreement, each player having his own preferences over a shared set of... The goal is to reach an agreement, each player having his own preferences over a shared set of options. Several simulated users have been implemented. An MDP policy has been optimised individually with Fitted Q-Iteration for several user instances. Then, the learnt policies have been cross evaluated with other users. Results show strong disparity of inter-user performances. This illustrates the importance of user adaptation in negotiation-based dialogue systems. Keywords. Spoken dialogue systems Dialogue simulation Reinforcement learning Negotiation Game theory. Auto-negotiation can only be disabled when you issue the set port speed {mod_num/port_num {10 | 100}} command. This is the output when you issue the set port speed 1/1 10 command on switch B. Both ports negotiate full-duplex and 100 Mb. Auto-Negotiation on Catalyst Switches that Run Cisco IOS Software. The commands described in this section apply to these types of switch products: Catalyst 2900XL, 3500XL, 2950, 3550, 2948G-L3, 4908G-L3, Catalyst 4500/4000 that runs Cisco IOS System Software (Supervisor Engine III), and the Catalyst 6500/6000 that runs Cisco IOS System Software. Issue the show interface slot/port command (without the status keyword) to see the port speed and duplex. NativeIOS#show interfaces fastethernet 3/1 status.