Learning Smooth, Human-Like Turntaking in Realtime Dialogue

Gudny Ragna Jonsdottir, Kristinn R. Thorisson, and Eric Nivel

Center for Analysis & Design of Intelligent Agents & School of Computer Science
Reykjavik University
Ofanleiti 2, IS-103 Reykjavik, Iceland
{gudny04,thorisson,eric}@ru.is

Abstract. Giving synthetic agents human-like realtime turntaking skills is a challenging task. Attempts have been made to manually construct such skills, with systematic categorization of silences, prosody and other candidate turn-giving signals, and to use analysis of corpora to produce static decision trees for this purpose. However, for general-purpose turntaking skills which vary between individuals and cultures, a system that can learn them on-the-job would be best. We are exploring ways to use machine learning to have an agent learn proper turntaking during interaction. We have implemented a talking agent that continuously adjusts its turntaking behavior to its interlocutors based on realtime analysis of the other party's prosody. Initial results from experiments on collaborative, content-free dialogue show that, for a given subset of turntaking conditions, our modular reinforcement learning techniques allow the system to learn to take turns in an efficient, human-like manner.

Keywords: Turntaking, Machine Learning, Prosody, End-of-utterance detection.

1 Introduction

Fluid turntaking is a dialogue skill that most people handle with ease. To signal that they have finished speaking and are expecting a reply, for example, people use various multimodal behaviors including intonation and gaze [1]. Most of us pick up on such signals without problems, automatically producing information based on data from our sensory organs to infer what the other participants intend. In amicable, native circumstances conversations usually go smoothly enough for people to not even realize the degree of complexity inherent in the process responsible for dynamically deciding how each person gets to speak and for how long.

Giving synthetic agents similar skills has not been an easy task. The challenge lies not only in the integration of perception and action in sensible planning schemes but especially in the fact that these have to be coordinated while marching to a real-world clock. Efficient handling of time is one of a few key components that sets current dialogue systems clearly apart from humans; for

H. Prendinger, J. Lester, and M. Ishizuka (Eds.): IVA 2008, LNAI 5208, pp. 162–175, 2008.
© Springer-Verlag Berlin Heidelberg 2008

example, speech recognition systems that have been in development for over a decade are still far from addressing the needs of realtime dynamic dialogue [2]. In spite of moderate progress in speech recognition technologies most systems still rely on silence duration as the main method for detection of end-of-utterance. However, as is well known and discussed by e.g. Edlund et al. [3], natural speech contains a lot of silences that do not indicate end-of-speech or end-of-turn, that is, silences where the speaker nonetheless does not want to be interrupted.

Although syntax, semantics and pragmatics indisputably can play a large role in the dynamics of turntaking, we have argued elsewhere that natural turntaking is partially driven by a content-free planning[1] system [4]. For this, people rely on relatively primitive signals such as multimodal coordination, prosody and facial expressions. In humans, recognition of prosodical patterns, based on the timing of speech loudness, silences and intonation, is a more light-weight process than word recognition, syntactic and semantic processing of speech [5]. This processing speed difference is even more pronounced in artificial perception, and such cues can aid in the process of recognizing turn signals in artificial dialogue systems.

J.Jr. was an agent that could interject back-channel feedback and take turns in a human-like manner without understanding the content of dialogue [6]; the subsequent Gandalf agent [7] adopted the key findings from J.Jr. in the Ymir architecture, an expandable granular architecture for cognition. We build directly on this work, introducing schemes for the automatic, realtime learning of a key turntaking decisions, that had to be built by hand in these prior systems. The system learns on-line to become better at taking turns in realtime dialogue, specifically improving its own ability to take turns correctly and quickly, with minimal speech overlap.

In the present work turntaking is viewed as a negotiation process between the parties involved and the particular observed patterns produced in the process are considered emergent [8], on many levels of detail [9], based on an interaction between many perception, decision and social processes, and their expression in a humanoid body. In our task two talking agents, each equipped with a dialogue model and dynamic speech planning capabilities, speak to each other. One of them listens to the prosody of the other (intonation and speech-on/off) and uses machine learning to best determine - as the speaker falls silent - how long to wait until starting to speak, that is, to take the turn. One way to think of this task is as a bet against time: The longer the duration of the silence the more willing we are to bet that the speaker expects us to start talking; the challenge is to reliably bet on this in realtime, during the silence, as soon as possible after the silence starts, with whatever information (prosody etc.) has been processed. In the present work we leave aside issues (perception, interpretation and actions) related to switching between different conversational topics. We also limit ourselves to detecting turn-giving indicators in deliberately-generated prosody, leaving out the topic of *turn-opportunity* detection (i.e. turn transition without prior indication from the speaker that she's giving the turn), which would e.g. be

[1] We use the term "planning" in the most general sense, referring to any system that makes a priori decisions about what should happen before they are put in action.

necessary for producing (human-like) interruptions. Our experiments show that our architecture and learning methodology allow the system to learn to take turns with human-like speed and reliability in about 5 minutes of interaction in which turn is successfully exchanged 7 times per minute, on average.

The rest of this paper is organized as follows. In section 2 we discuss related work in the field; we describe the system we have built in section 3, with section 4 detailing the learning mechanism. In section 5 we present the evaluation setup; sections 6 and 7 show results from experiments with the system interacting with itself humans. Section 8 is conclusion and future work.

2 Related Work

The problem of utterance segmenting has been addressed to some extent in prior work. Sato et. al [10] use a decision tree to learn when a silence signals to take turn. They annotated various features in a large corpus of human-human conversation to train and test the tree. Their results show that semantic and syntactic categories, as well as understanding, are the most important features. Their experiments have currently been limited to single domain, task oriented scenarios with annotated data. Applying this to a casual conversation scenario would inevitably increase the recognition time - as the speech recognizers vocabulary is enlarged - to the extent that content interpretation results are already obsolete for turn taking decisions by the time they are produced [2].

Traum et al. [11] and others have also addressed the problem of utterance segmenting, showing that prosodic features such as boundary tones do play a role in turntaking and Schlangen [12] has successfully used machine learning to categorize prosodic features from corpus, showing that acoustic features can be learnt. As far as we know, neither of these attempts have been applied to a real-time scenario.

Raux and Eskenazi [13] presented data from a corpus analysis of an online bus scheduling/information system, showing that a number of dialogue features, including type of speech act, can be used to improve the identification of speech endpoint, given a silence. They reported no benefits from prosody for this purpose, which is surprising given the many studies showing the opposite (cf. [1,7,11,12,14,15]). We suspect one reason could be that the pitch and intensity extraction methods they used did not work very well on the data selected for analysis. The authors tested their findings in a realtime system: Using information about dialogue structure - speech act classes, a measure of semantic completeness, and probability distribution of how long utterances go (but not prosody) - the system improved turntaking latency by as much as 50% in some cases, but significantly less in others. The Gandalf system [7] also used a measures of semantic, syntactic (and even pragmatic) completeness to determine turntaking behaviors, but data about its benefit for the turntaking per se is not available. The major lessons that can be learned from Raux and Eskenazi [raux08], echoing the work on Gandalf, is that turntaking can be improved through an integrated, coordinated use of various features in context.

Prosodic information has successfully been used to determine back-channel feedback. The Rapport Agent [14] uses gaze, posture and prosodic perception to among other things detect backchannel opportunities. The J.Jr. system, a communicative agent that could take turns in realtime dialogue with a human without understanding the content of the speech, used only prosodical information to make decisions about when to ask questions and when to interject back-channel feedback. The system was based on a finite state-machine formalism that was difficult to expand into a larger intelligent architecture [7]. Subsequent work on Gandalf [7] incorporated mechanisms from J.Jr. into the Ymir architecture, which was built as a highly expandable, modular system of preceptors, deciders and action modules; this architecture has recently been used in building an advanced vision and planning system for the Honda ASIMO robot [16].

3 System Architecture

We have built a multi-module dialogue system using the methodology described in [9,17]. The gross architecture of the system will be detailed elsewhere; here we focus on parts of the turntaking - the preceptors, deciders and action modules - needed to support learning for efficient turntaking. Following the Ymir architecture [7], our systems modules are categorized based on their functionality; perception modules, decider modules and action modules (see Figure 1). We will now describe these.

3.1 Perception

There are two main perceptors (perception modules) in the system, the Prosody-Tracker and the Prosody-Analyzer. The Prosody-Tracker is a low-level perceptor whose input is raw audio signal [18]. It computes speech signal levels and compares this to a set of thresholds to determine information about speech activity, producing timestamped Speech-On and Speech-Off messages. It also analyzes the speech pitch incrementally (in steps of 16 msec) and produces pitch values, in the form of a continuous stream of pitch message updates.

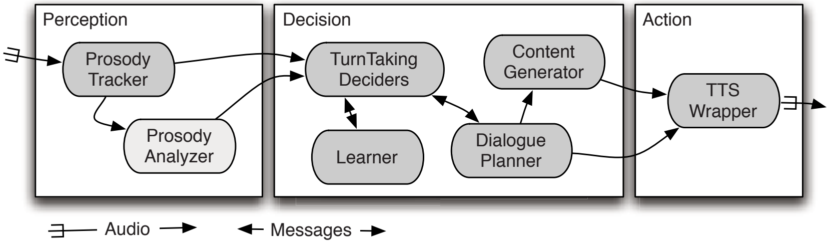


Fig. 1. System components, each component consists of one or more modules

Similar to [19], pitch is analyzed further by a Prosody-Analyzer perceptor to compute a more compact representation of the pitch pattern in a discrete state space, to support the learning: The most recent tail of speech right before a silence, currently the last 300 msec, are searched for minimum and maximum values to calculate a tail-slope value of the pitch. Slope is then split into semantic categories, currently we are using 3 categories for slope: Up, Straight and Down and 3 for relative value of pitch right before silence: Above, At, Below, as compared to the average pitch. Among the output of the Prosody-Analyzer is a symbolic representation of the particular prosody pattern identified in this tail period (see Figure 2). More features could be added into the symbolic representation, with the obvious side effect of increasing the state space. Figure 2 shows a 9 second long frame with speech periods, silences and categories. As soon as a silence is encountered (indicated by gray area) the slope of the most significant continuous pitch direction of the tail is computed.

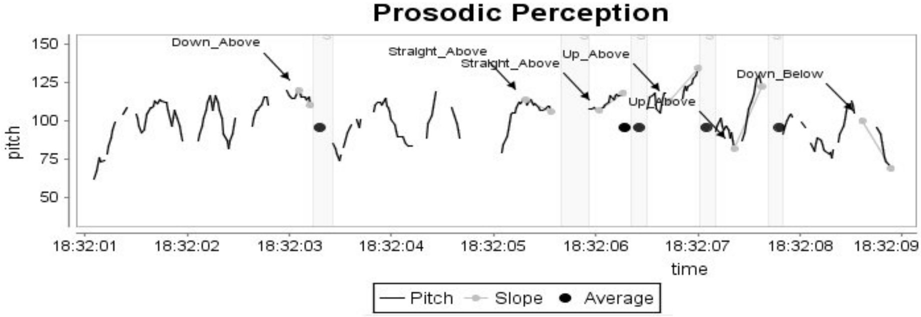


Fig. 2. A window of 9 seconds of speech, containing 6 consecutive utterances, categorized into descriptive groups. Slope is only categorized before silences.

3.2 Deciders

The dialogue state (I-have-turn, Other-has-turn etc.) is modeled with a distributed context system, implementing what can approximately be described as a distributed finite state machine. Context transition control in this system is managed by a set of deciders [9]. There is no limit on how many deciders can be active in a single system-wide context. Likewise, there is no limit to how many deciders can manage identical or non-identical transitions. Reactive deciders (IGTD,OWTD,...) are the simplest, with one decider per transition. Each contains at least one rule about when to transition, based on both temporal and other information. Transitions are made in pull manner; the Other-Accepts-Turn-Decider transits to context Others-Accepts-Turn (see Figure 3).

The Dialogue Planner and Learning modules can influence the dialogue state directly by sending context messages I-Want-Turn, I-Accept-Turn and I-Give-Turn. These decisions are under the supervisory control of the Dialogue Planner: If the Content Planner has some content ready to be communicated, the agent might want to signal that it wants turn and it may want to signal I-Give-Turn when content queue is empty (i.e. have-nothing-to-say). Decisions made by these

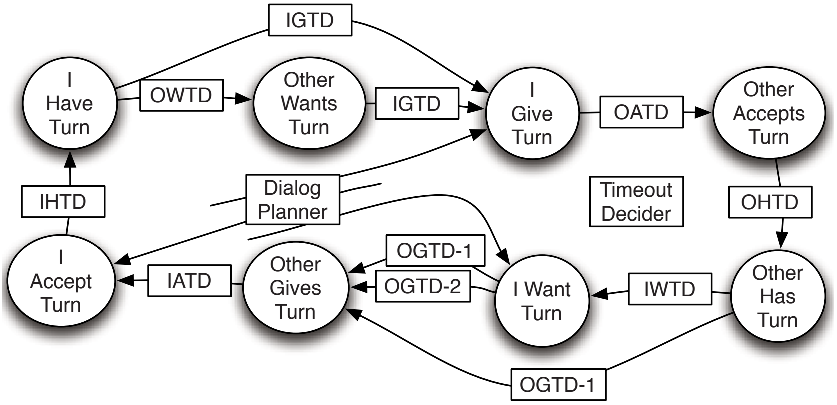


Fig. 3. The turntaking system can be viewed as a set of 8 context-states and 11 deciders. Each context-state has at least one associated decider for determining transition to it but each decider is only active in a limited set of contexts. In context-state I-Have-Turn, both I-Give-Turn-Decider (IGTD) and Other-Wants-Turn-Decider (OWTD) are active. Unlike other modules, the Dialog Planner can transition independent of the system's current context-state and override the decisions from the reactive deciders. A Timeout-Decider handles transitions if one of the negotiating context-states is being held unacceptably long but it's transitions are not included in the diagram.

modules override decisions made by other turntaking modules. The DP also manages the content delivery, that is, when to start speaking, withdraw or raise one's voice. The Content-Generator is responsible for creating utterances incrementally, in "thought chunks", typically of shorter duration than 1 second. While we are developing a dynamic content generation system based on these principles the CG simulates this activity by selecting thought units to speak from a predefined list; it signals when content is available to be communicated and when content has been delivered.

In the present system the module Other-Gives-Turn-Decider-2 (OGTD-2) uses the data produced by the Learner module to change the behavior of the system. At the point where the speaker stops speaking the challenge for the listening agent is to decide how long to wait before starting to speak. If the agent waits too long, and the speaker does not continue, there will be an unwanted silence; if he starts too soon and the speaker continues speaking, overlapping speech will result. We solve this by having OGTD-2 use information about prosody before prior silences to select an optimal wait-and-see time. This will be described in the next section.

4 The Learner

The learning mechanism is implemented as a highly isolated component in the modular architecture described above. It is based on the Actor-Critic distribution of functionality [20], where one or more actors make decisions about which actions to perform and a critic evaluates the effect of these actions on the environment; the separation between decision and action is important because in our system a decision can be made to act in the future. In the highly general and distributed mechanism we have implemented, any module in the system can take the role of an actor by sending out decisions and receiving, in return, an updated decision policy from an associated Learner module. A decision consists of a state-action pair: the action being selected and the evidence used in making that action represents the state. Each actor follows its own action selection policy, which controls how he explores his actions; various methods such as e-greedy exploration, guided exploration, or confidence value thresholds, can be used [20]. The Learner module takes the role of a critic. It consists of the learning method, reward functions, and the decision policy being learnt. A Learner monitors decisions being made in the system and calculates rewards based on the reward function, a list of decision/event pairs, and signals from the environment - in our case overlapping speech and too long silences - and publishes updated decision policy (the environment consists of the relevant modules in the system).

We use a delayed one-step Q-Learning method according to the formula:

$$Q(s, a) = Q(s, a) + \alpha[reward - Q(s, a)] \tag{1}$$

Where *Q(s,a)* is the learnt estimated return for picking action *a* in state *s*, and *alpha* is the learning rate. The reward functions - what events following what actions lead to what reward - need to be pre-determined in the Learner's configuration in the form of rules: A *reward* of x if *event* y succeeds at *action* z. Each decision has a lifetime in which system events can determine a reward, but reward can also be calculated in the case of an absence of an event, after the given lifetime has passed (e.g. no overlapping speech). Each time an action gets reward the return value is recalculated according to the formula above and the Learner broadcasts the new value.

In the current setup, Other-Gives-Turn-Decider-2 (OGTD-2) is an actor, in Sutton's [20] sense, that decides essentially what its name implies. This decider is only active in state I-Want-Turn. It learns an "optimal" pause duration so as not to speak on top of the other, while minimizing the lag in starting to speak. Each time a Speech-Off signal is detected, OGTD-2 receives analysis of the pitch in the last part of the utterance preceding the silence, from the Prosody-Analyzer. The prosody information is used to represent the state for the decision, a predicted safe pause duration is selected as the *action* and the Decision is posted. This pause duration determines when, in the future, the listener will start speaking/take the turn. In the case where the interlocutor starts speaking again before this pause duration has passed, two things can happen: (1) If he starts speaking before the predicted pause duration passes, the decider doesn't signal Other-Giving-Turn, essentially canceling the plan to start speaking. This leads to a better reward, since no overlapping speech occurred. (2) If he starts talking just after the pause duration has passed, after the decider signals Other-Gives-Turn, overlapping speech will likely occur, leading to negative reinforcement for this pause duration, based on the prosodic information. This learning strategy is based on the assumption that both agents want to take turns politely and efficiently. We have already begun expanding the system to be able to interrupt dynamically and deliberately - i.e. be "rude" - and the ability to switch back to being polite at any time, without destroying the learned data.

5 Evaluation Setup

We are aiming at an agent that can adapt its turntaking behavior to dialogue in a short amount of time. In this initial evaluation we focus exclusively on detecting turn-giving indicators in deliberately-generated prosody, leaving out the topic of turn-opportunity detection (i.e. turn transition without prior indication from the speaker that she's giving the turn), which would e.g. be necessary for producing (human-like) interruptions. The goal of the learning system is to learn to take turns with (ideally) no speech overlap, yet achieving the shortest possible silence duration between speaker turns. In this setup both agents always have something to say, leaving out issues such as turntaking perception and actions related to goal-directed switching between different conversational topics (think stream-of-consciousness conversations). First we describe an experiment where the system learns to take turns while interacting with itself. We then compare this to a pilot study of turntaking in human-human dialogue, using the same analysis mechanisms as in the autonomus system.

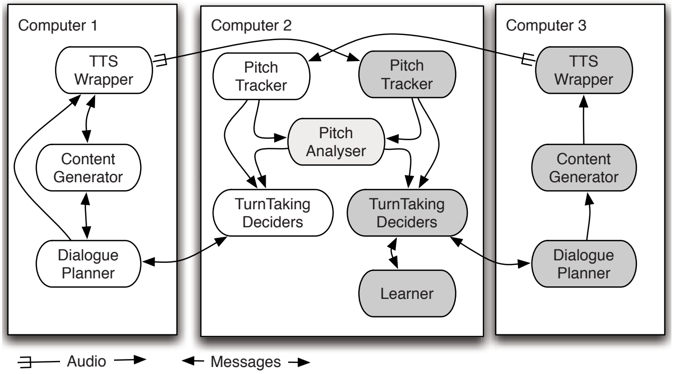


Fig. 4. Two agents distributed over a set of 3 computers. Time syncing and time drift pose a problem when measuring small time units, so all modules that we are specifically measuring time for are located on the same computer.

We have set up two instances of the system (agents) talking to each other (see Figure 4). One agent, Simon, is learning, the other, Kate, is not learning. Kate will only start speaking when she detects a 2-second pause; pause duration = 2 sec (controlled by her Other-Gives-Turn-Decider-1). Simon has the same 2 sec default behavior, but in addition he learns a variable pause duration (controlled by OGTD-2); its goal is to make this duration as small as possible, as described above. Content is produced in groups of between 1 and 5 randomly-selected sentence fragments, using the Loquendo speech synthesizer, which has been enabled

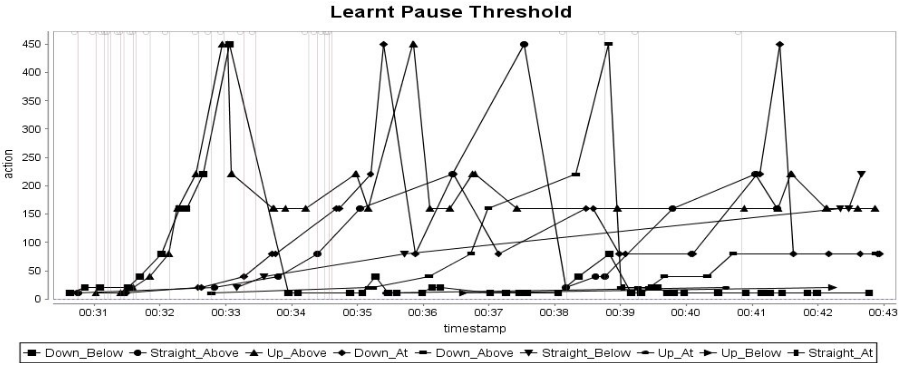


Fig. 5. The Learner's view of what is the best action (predicted "safe" pause duration) for each state, over a period of 13 minutes of learning, starting with no knowledge. Examining the actions, from smallest to largest pause duration, the system finds the optimal pause duration in as little as 3 minutes for the quickest learnt state (Down_Below), which is also the state with the shortest pause duration, indicating that this prosody pattern is a good sign of turn giving.

to start and stop synthesis at a moment's notice (around 100 msec). Loquendo uses markup to control the prosody; by adding a comma (,) to each fragment except the last one we can suppress a final fall [15] and keep the intonation up; by appending a period (.) we get a typical finall fall, e.g. "I went to my summer house," - "this weekend," - "the weather was nice," - "the whole time.". This way the intonation approximates typical spontaneous speech patterns. We selected sentences and fragment combinations to sound as natural as possible. (However, the fragments are selected randomly and assigned a role as either an intermediate fragment or last fragment.) Speech is thus produced incrementally by the combined speech synthesizer/speech planner system (speech synthesizer wrapper) as it receives each fragment, and the planner never commits to more than two fragments to the speech synthesizer at a time. As Loquendo introduces a short pause whenever there is a comma, and because of fluctuations in the transmission and execution time, pauses between fragments range from 31 to 4296 msecs, with 355 msecs being the average (see Figure 8).

5.1 Parameter Settings

Formulating the task as a reinforcement learning problem, the latest pitch tail represents the *state* and the pause duration is the *action* being selected. We have split the continuous action space into discrete logarithmic values starting with 10 msec and doubling the value up to 2 sec (the maximum pause duration where the system takes the turn by default). The action selection policy for OGTD-2 is e-greedy with 10% exploration and always selecting the shortest wait time if two or more actions share the top spot.

The reward given for decisions that do not lead to overlapping speech is the milliseconds in the selected pause; a decision to wait 100 msec from a Speech-Off signal until the I-Take-Turn decision is taken, receives a reward of -100 and -10 for deciding on a pause threshold of 10 msec. If, however, overlapping speech results from the decision, a reward of -2000 (same as waiting the maximum amount of time) is given. All rewards in the system are thus negative, resulting in unexplored actions being the best option at each time since return starts at 0.0 and once a reward has been giving the return goes down (exploration is guided towards getting faster time than the currently best action, so a pause duration larger than optimal is not explored). This can be seen in Figure 5 where each action is tried at least once before the system settles down at a tighter time range.

6 Results from System Interacting with Self

Looking at Figure 6, we clearly see that the learning agent starts by selecting pause durations that are too short and numerous overlaps are detected. The agent quickly learns the "safe" amount of time to wait before deciding turn was given, based on the prosody pattern. After 5 minutes of interaction interruptions have dropped below one per minute and occurrences after the 11th minute are solely caused by continued exploration. Each agent gets turn on average 4 times per minute (i.e. 8 successful turn exchanges). Average silence between turns is always larger than pause duration since it also includes delays resulting from processing the pitch and the time from deciding to speak until speech is delivered. These will be improved as we continue to tune the architecture's runtime performance. Interestingly, although pause duration is increased considerably for some prosody patterns, the influence on silence is minimal.

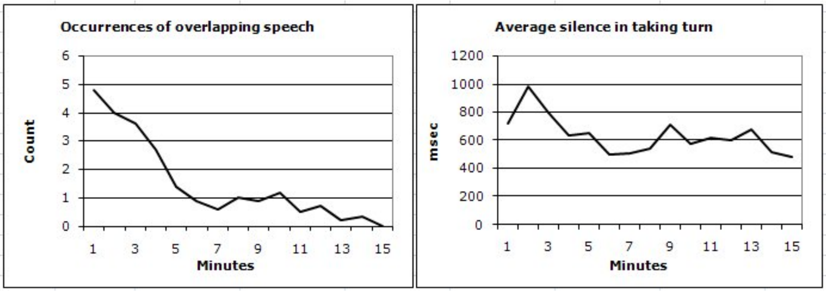


Fig. 6. When the system learns correct relations between prosody and pause duration, overlaps cease to occur (except as a function of exploring). Initial exploring of the action space is responsible for the average silence observed between turns increasing in the 2nd and the 3rd minute.

Based on our model of turntaking as a loosely-coupled system, it makes sense to compare the synchronization of "beliefs" in the agents about what turntaking context they are in (see Figure 7). At the beginning of the run there is not much consensus on who has the turn - the learning agent is constantly erroneously interrupting the other. After a few minutes of conversation the learning agent has adjusted his policy and turntaking goes smoothly.

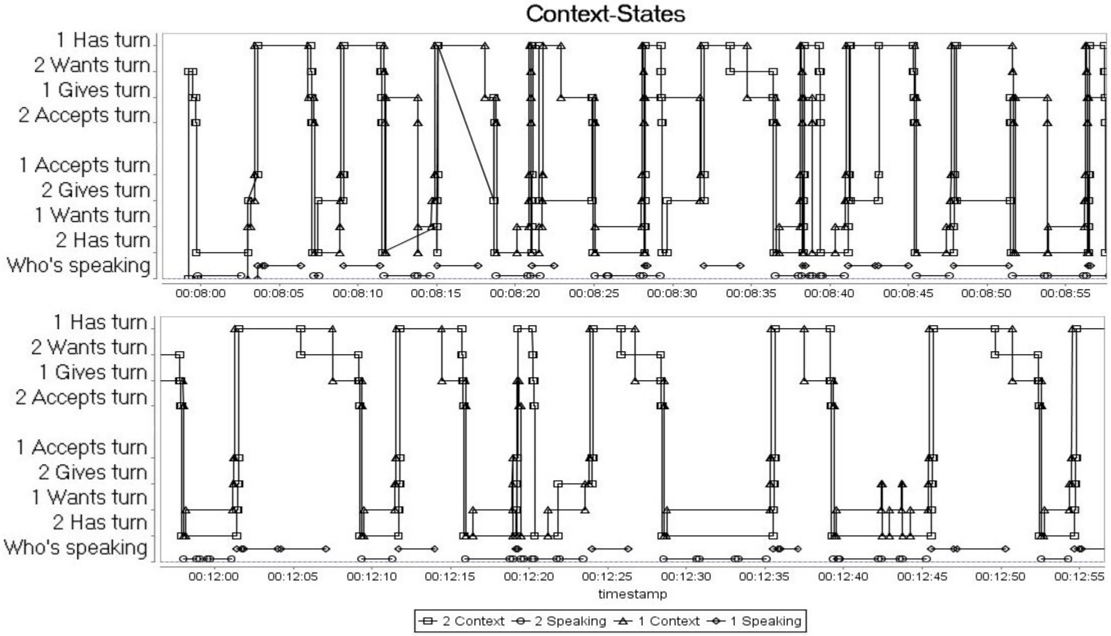
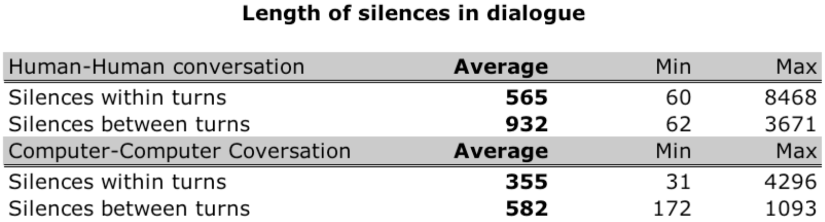


Fig. 7. Difference in dialogue context-states in the beginning of learning (upper graph) and after 5 minutes of learning (lower graph). At the bottom of each graph we plot who is speaking. The lines in each graph (1 Has Turn, 2 Wants turn, etc.) represent context-states in the two agents (see Figure 3). Each agent tries, using its perceptions, to align its current context-state with the other's. As one learns to do this, overlaps almost disappear, - a less dense graph means agent 1 has learned to couple his context-states to agent 2.

7 Comparison to Human-Human Interaction Study

We conducted a human-subject study to compare this system to. A priori we had reason to believe that people have more variable prosody generation patterns than the Loquendo speech synthesizer, with more variable silences within turn. In our human-subject study 8 telephone conversations were recorded in dual channel mode, and analyzed using the same mechanisms that the agents in our system use for deciding their turn behavior; each phone conversation was fed through our system just as if the sound was coming from two agents. The results show that silences within turn are considerably longer than expected (see Figure 8). Participants are also on average longer to take the turn than our agents, but minimum *successful* lag between turns is considerably shorter for humans (31 msec vs. 60 msec).



Length of silences in dialogue

Human-Human conversation	**Average**	Min	Max
Silences within turns	**565**	60	8468
Silences between turns	**932**	62	3671
Computer-Computer Coversation	**Average**	Min	Max
Silences within turns	**355**	31	4296
Silences between turns	**582**	172	1093

Fig. 8. Comparison of silence duration in human dialogue and between our agents. Within-turn silences: silences where the person is not giving turn; between-turn silences: successful turntaking opportunities.

For a more accurate comparison of human-human and computer-computer data, certain types of turntakings may need to be eliminated from the corpus, namely those that clearly involve a switch in topic (as these were casual conversations they include some long silences where neither party has anything to say). Another source of bias may be the fact that these dialogues were collected over Skype, which typically contains somewhat larger lag times in audio transmission than landline telephone, and certainly more lag than face-to-face conversation.

8 Conclusions and Future Work

We have built a system that uses prosody to learn optimal pause durations for taking turns, minimizing speech overlaps. The system learns this on the fly, in a full-duplex "open-mic" (dynamic interaction) scenario, and can take turns very efficiently in dialogues with itself, in human-like ways. The system uses prosodic information for finding categories of pitch that can serve a predictor of turn-giving behavior of interlocutors. As the system learns on-line it will be able to adjust to the particulars of individual speaking styles; while this remains to be tested our preliminary results indicate that this is indeed possible. At present the system is limited to a small set of turntaking circumstances where content does not play a role, assuming "friendly" conversation where both parties want to minimize overlaps in speech. Silences caused by outside interruptions - extended durations caused by searching for "the right words", and deliberate interruption techniques - are all topics for future study. The system is highly expandable, however, as it was built as part of a much larger system architecture that addresses multiple topic- and task-oriented dialogue, as well as multiple modes. In the near future we expect to expand the system to more advanced interaction types and situations, and to start using it in dynamic human-agent interactions. The learning mechanism described here will be expanded to learn not just the shortest durations but also the most efficient turntaking techniques in multimodal interaction under many different conditions. The turntaking system is architected in such a way as to allow a mixed-control relationship with outside processes. This means that we can expand it to handle situations where the goals of the dialogue may be very different from "being polite", even adversarial, as for example in on-air open-mic political debates. How easy this is remains to be seen; the main question revolves around the learning systems - how to manage learning in multiple circumstances without negatively affecting prior training.

Acknowledgments. This work was supported in part by a research grant from RANNIS, Iceland, and by a Marie Curie European Reintegration Grant within the 6th European Community Framework Programme. The authors wish to thank Yngvi Bjornsson for his contributions to the development of the reinforcement mechanisms.

References

1. Goodwin, C.: Conversational organization: Interaction between speakers and hearers. Academic Press, New York (1981)
2. Jonsdottir, G.R., Gratch, J., Fast, E., Thórisson, K.R.: Fluid semantic back-channel feedback in dialogue: Challenges and progress. In: Pélachaud, C., Martin, J.-C., André, E., Chollet, G., Karpouzis, K., Pelé, D. (eds.) IVA 2007. LNCS (LNAI), vol. 4722, pp. 154–160. Springer, Heidelberg (2007)
3. Edlund, J., Heldner, M., Gustafson, J.: Utterance segmentation and turn-taking in spoken dialogue systems (2005)
4. Thórisson, K.R.: Natural turn-taking needs no manual: Computational theory and model, from perception to action. In: Granström, B., House, D.I.K. (eds.) Multimodality in Language and Speech Systems, pp. 173–207. Kluwer Academic Publishers, Dordrecht (2002)
5. Card, S.K., Moran, T.P., Newell, A.: The model human processor: An engineering model of human performance. In: Handbook of Human Perception, vol. II. John Wiley and Sons, Chichester (1986)
6. Thórisson, K.R.: Dialogue control in social interface agents. In: INTERCHI Adjunct Proceedings, 139–140 (1993)
7. Thórisson, K.R.: Communicative Humanoids: A Computational Model of Psycho-Social Dialogue Skills. PhD thesis, Massachusetts Institute of Technology (1996)
8. Sacks, H., Schegloff, E.A., Jefferson, G.A.: A simplest systematics for the organization of turn-taking in conversation. Language 50, 696–735 (1974)

9. Thórisson, K.R.: Modeling multimodal communication as a complex system. In: Wachsmuth, I., Knoblich, G. (eds.) ZiF Research Group International Workshop. LNCS (LNAI), vol. 4930, pp. 143–168. Springer, Heidelberg (2008)
10. Sato, R., Higashinaka, R., Tamoto, M., Nakano, M., Aikawa, K.: Learning decision trees to determine turn-taking by spoken dialogue systems. In: ICSLP 2002, pp. 861–864 (2002)
11. Traum, D.R., Heeman, P.A.: Utterance units and grounding in spoken dialogue. In: Proc. ICSLP 1996., Philadelphia, PA, vol. 3, pp. 1884–1887 (1996)
12. Schlangen, D.: From reaction to prediction: Experiments with computational models of turn-taking. In: Proceedings of Interspeech 2006, Panel on Prosody of Dialogue Acts and Turn-Taking, Pittsburgh, USA (September 2006)
13. Raux, A., Eskenazi, M.: Optimizing endpointing thresholds using dialogue features in a spoken dialogue system. In: Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue, Columbus, Ohio, Association for Computational Linguistics, pp. 1–10 (June 2008)
14. Gratch, J., Okhmatovskaia, A., Lamothe, F., Marsella, S., Morales, M., van der Werf, R.J., Morency, L.P.: Virtual rapport. In: Gratch, J., Young, M., Aylett, R.S., Ballin, D., Olivier, P. (eds.) IVA 2006. LNCS (LNAI), vol. 4133, pp. 14–27. Springer, Heidelberg (2006)
15. Pierrehumbert, J., Hirschberg, J.: The meaning of intonational contours in the interpretation of discourse. In: Cohen, P.R., Morgan, J., Pollack, M. (eds.) Intentions in Communication, pp. 271–311. MIT Press, Cambridge (1990)
16. Ng-Thow-Hing, V., List, T., Thórisson, K.R., Lim, J., Wormer, J.: Design and evaluation of communication middleware in a distributed humanoid robot architecture. In: Prassler, E., Nilsson, K., Shakhimardanov, A. (eds.) IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2007) Workshop on Measures and Procedures for the Evaluation of Robot Architectures and Middleware (2007)
17. Thorisson, K.R., Benko, H., Arnold, A., Abramov, D., Maskey, S., Vaseekaran, A.: Constructionist design methodology for interactive intelligences. A.I. Magazine 25(4), 77–90 (2004)
18. Nivel, E., Thórisson, K.R.: Prosodica: A realtime prosody tracker for dynamic dialogue. Technical report, Reykjavik University Department of Computer Science, Technical Report RUTR-CS08001 (2004)
19. Thórisson, K.R.: Machine perception of multimodal natural dialogue. In: McKevitt, P., Nulláin, S.Ó., Mulvihill, C. (eds.) Language, Vision & Music, pp. 97–115. John Benjamins, Amsterdam (2002)
20. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. The MIT Press, Cambridge (1998)