SIMULATING SPOKEN DIALOGUE WITH A FOCUS ON REALISTIC TURN-TAKING

Timo Baumann
University of Potsdam
timo@ling.uni-potsdam.de

Abstract. We present a system for testing turn-taking strategies in a simulation environment, in which artificial dialogue participants exchange *audio* streams in real time – unlike earlier turn-taking simulations, which interchanged unambiguous symbolic messages. Dialogue participants autonomously determine their turn-taking behaviour, based on their analysis of the incoming audio. We use machine-learning methods to classifiy the continuous audio signal into symbolic turn-taking states. We experiment with various rule sets and show how simple, local management rules can create realistic behavioural patterns.

1. Introduction

Turn-taking management, i. e. deciding who may speak when in a dialogue, is an important subtask of interaction management. The classical model of turn-taking (Sacks, et al. 1974) describes turn-taking as *locally managed* (depending only on a local context) and *predictive* (upcoming turn endings are signalled in advance by the interplay of syntax, semantics and prosody). Current spoken dialogue systems (SDSes) on the other hand, use reactive turn-taking schemes, with the turn being taken after a silence of fixed length or of contextually determined length (Ferrer, et al. 2002). This limits the interactivity of SDSes, as turns have to be separated by intervening silence.

The prediction of turn endings (EoT prediction) has been investigated by a number of authors. Schlangen (2006) trains classifiers to predict the end of turn (*EoT*) but uses features that are not calculated strictly incrementally. Turn-management has also been studied before, but typically in simulation systems that interchange symbolic messages and work in a centrally managed environment (Padilha 2006). In the present paper, we combine the efforts for EoT-prediction and turn-taking simulation. We propose an incremental classification of speech into speech states that control the system's turn-taking. We first evaluate the classification itself and then combined with different turn-management strategies in a dialogue simulation environment.

Dialogue simulation itself has a long standing tradition in the development of SDSes, but the main focus seems to be on the improvement of dialogue strategies (Schatzmann, et al. 2006) and audio is usually just used to trigger realistic ASR errors (López-Cózar, et al. 2003), which contrasts with the focus of the present paper: Our goal is to show how realistic turn-taking behaviour can be simulated using only local context for the classification of speech into classes relevant to turn-taking management combined with simple,

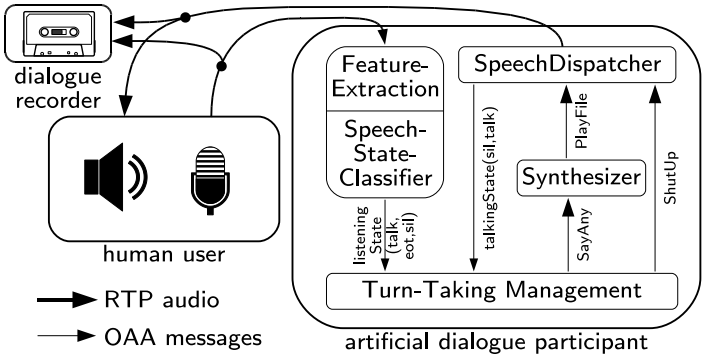

Figure 1: A human user conversing with an artificial DP in our interaction environment (structured as in section 2). A dialogue recorder wiretaps their conversation.

locally managed rules. Dialogue strategies in general are *not* locally managed and thus learning dialogue strategies seems to require the more complex reinforcement learning instead of simple classifier training which we use.

We do not (and do not need to) take into account the content of the dialogues and in fact we limit our speech analysis to simple prosodic features for the EoT prediction. Thus, for this work, we abstract away from all questions of content management and let our dialogue participants speak randomly selected pre-recorded utterances – though with proper turn-taking.

The remainder of the paper is structured as follows: Section 2 describes the system architecture and Section 3 the corpora we use. Section 4 evaluates the speech state classification and Section 5 demonstrates and evaluates some simple turn-management strategies. We close with conclusions and ideas for further work.

2. Architecture of the Interaction Environment

Our architecture defines an *interaction environment* in which *dialogue participants* (DPs) communicate with each other. Interaction is purely non-symbolic, using asynchronous audio streams over RTP (Schulzrinne, et al. 2003). There is no common clock, or other synchronisation required between DPs. The architecture provides a headset tool for human DPs, and monitoring tools to listen to ongoing dialogues and to record them to disk.

Figure 1 shows two dialogue participants – one human, one artificial – conversing in the environment described above. The artificial DP on the right of figure 1 is structured as described below.

Artificial DPs are realized as modular and extensible collections of event-driven software agents in the open agent architecture, OAA (Martin, et al. 1999). In the OAA each software agent advertises its own abilities to solve problems (such as generating utterances) and may itself request other agents to solve sub-problems (e. g. sending data over RTP). For audio processing inside the DP we rely on the Sphinx-4 framework (Walker, et al. 2004) which we extended for our audio-processing pipeline. In the current system, we do not yet use Sphinx' abilities as a speech recognizer and most other modules that would be needed for a real dialogue system are missing. These are obvious enhancements for later versions.

2.1. Speech Generation

Speech generation consists of a *synthesizer* and a *dispatcher*. The synthesizer currently selects from a corpus of pre-recorded utterances and will be extended to include text-to-speech. To make turn-taking management harder and the system more realistic a fixed delay of 100 ms between signal to the module and onset of the recorded utterance is introduced at this point.[1] This delay is realized by sending 100 ms of recorded silence before the utterance and utterances are also followed by 100 ms of *recorded* silence. (If DPs were to send digital zeros directly before and after their utterances, speech state classification, as described below, would become trivial.)

The *speech dispatcher* continuously sends an RTP stream in packets of 10 ms, either audio from a file or sine waves if so instructed by the synthesizer, or silence (digital zero). It can also be ordered to interrupt the audio and to revert to silence. The dispatcher also publishes its current *speech state* which may be one of **sil**, start of turn (**SoT**), **talk**, or end of turn (**EoT**) to the DP it is part of.

2.2. Speech Analysis

Speech analysis focuses solely on local prosodic analysis for the classification of the *listening state* (which should reflect the *interlocutor's* speech state, as described above). In order to be effective, classification must happen with as short a lag as possible. While short lags would allow for *reactive* behaviour, we aim to *predict* when the interlocutor's end of turn is approaching in order to achieve smooth turn changes and counter-balance the 100 ms lag before a response can be uttered by the speech generation.

We use machine learning to classify each received frame (10 ms) of audio as silence (**sil**), ongoing talk (**talk**) or end of turn (**EoT**). Classification is based exclusively on signal power, pitch and derived features. Our pitch extraction is modelled after the first three steps of the YIN algorithm (de Cheveigné & Kawahara 2002). As no smoothing or dynamic programming is applied to the pitch extraction, results are computed incrementally in real-time and become available instantaneously. The algorithm runs at several times real-time on average hardware. On the corpora described below, the gross error rate is 1.6 % compared to the well known ESPS algorithm (Talkin 1995).

In order to track changes over time, we derive features by windowing over past values of pitch and power with sizes ranging from 20 to 500 ms. While the features calculated on smaller windows help to smooth and to remove outliers due to failures of the pitch extraction, the larger windows are expected to capture long-term trends. We calculate the arithmetic *mean* and the *range* of the values, the mean difference between values within the window and the relative position of the minimum and maximum. We also perform a linear regression and use its *slope*, the MSE of the regression and the *error* of the regression for the last value in the window.

2.3. Turn-Taking Management

The *turn-taking management agent* determines whether to start or stop emitting utterances on the basis of the states of the generation and analysis modules. An important aspect in turn-taking management is *robustness*. To be robust, the turn-taking strategy must not

[1] In a dialogue system NLG and TTS would require processing time; for humans there is a delay between starting to plan an utterance and the start of the articulation (Levinson 1983).

depend on its interlocutor acting and reacting in certain ways. Naturally, "good" dialogue will only evolve from friendly dialogue partners, but the turn-management strategy must prevent dead-locks due to the interlocutor's behaviour.

Upon the reception of dialogue state change notifications from the analysis module, the agent decides about emitting messages to the generation module, ordering it to talk or to hush, according to a defined turn-taking strategy. Messages are only emitted with certain probabilities. The probabilities to start or hush were determined empirically to lead to natural performance. If no action is taken, the agent sleeps for a short while (currently, 50 ms) being awakened if another message is received (for example **EoT** changing to **sil**). Thus, exact timings are non-deterministic and randomly differ between agents. The probability to start an utterance is set to 0.1, and to hush during an utterance to 0.3.

3. Corpora

We perform our experiments with two different corpora, one of simple pseudo-speech, one of read speech. Each corpus contains material from two different speakers (one female, one male) for which we train separate speech analyzers, in order to be able to simulate dialogues with one male and one female each.

For pseudo-speech our speakers repeatedly uttered the syllable /ba/ instead of the actually occuring syllables in a script of 50 utterances (questions, informative sentences, confirmations, etc). By always uttering the same syllable, we remove segment-inherent influences on power and pitch variation, while at the same time retaining sentence intonation. For read speech we relied on the two major speakers of the Kiel Corpus of Read Speech, KCoRS (IPDS 1994). That corpus contains some 600 utterances for each speaker.

The two corpora differ in size and complexity. Our controlled pseudo-speech poses hardly any problem for pitch-extraction and does not contain voiceless speech, silence during the occlusion of voiceless plosives or other potentially "difficult" audio. The KCoRS on the other hand contains far more training material. Also, as the pseudo-speech does not convey any semantic meaning, subjects in a listening test for the evaluation of generated turn-taking patterns would not be distracted by nonsense dialogue.

The performance of a speech state classifier on both of our corpora is likely to be better than on a corpus of real dialogue speech as it is more homogenous (especially compared to speaker-independent speech state classification). Thus, our results should be considered an upper bound on realistic results.

The start and end of each utterance were hand-annotated and each 10 ms of audio was assigned to one of the listening states as described above with **EoT** being assigned to frames in the vicinity of ± 50 ms of the utterance end. For the turn-taking management experiments, we crop the audio files so that each utterance is preceeded and succeeded by 100 ms of silence.

4. Speech Analysis Evaluation

We used the machine learning toolkit Weka (Witten & Frank 2000) to train various speaker-dependent classifiers. For the evaluation 80 % of each corpus were used as training- and 20 % as test-set. Tables 1 and 2 show the results of the OneR-, J48 and JRip-algorithms for each corpus. OneR finds the most predictive feature to be the dy-

classifier	female speaker					male speaker				
	$Acc.$	F_{sil}	F_{talk}	F_{EoT}	FAR	$Acc.$	F_{sil}	F_{talk}	F_{EoT}	FAR
OneR	96.1	0.98	0.96	0.00	21.4	92.8	0.96	0.93	0.13	65.5
J48	94.8	0.98	0.95	0.50	68.9	96.3	0.97	0.97	0.71	64.3
JRip	95.3	0.98	0.95	0.55	68.3	96.2	0.97	0.97	0.80	59.2
Stateful JRip	95.9	0.98	0.95	0.59	48.4	95.5	0.97	0.96	0.72	50.0
Stateful JRip, shifted	96.2	0.98	0.96	0.59	48.4	96.4	0.97	0.97	0.80	47.5

Table 1: Accuracy, per-class f-measures and false alarm rate for various speech state classifiers for the *pseudo-speech* corpus.

classifier	female speaker					male speaker				
	$Acc.$	F_{sil}	F_{talk}	F_{EoT}	FAR	$Acc.$	F_{sil}	F_{talk}	F_{EoT}	FAR
OneR	94.5	0.97	0.96	0.03	65.4	93.7	0.92	0.96	0.10	80.7
J48	97.3	0.98	0.98	0.61	71.1	96.1	0.96	0.98	0.42	84.1
JRip	96.6	0.97	0.98	0.73	61.1	95.9	0.97	0.96	0.61	65.7
Stateful JRip	96.4	0.96	0.98	0.70	31.9	94.9	0.97	0.96	0.58	50.0
Stateful JRip, shifted	96.9	0.97	0.98	0.74	31.6	95.5	0.97	0.96	0.64	48.9

Table 2: Accuracy, per-class f-measures and false alarm rate for various speech state classifiers for the *KCoRS* speakers.

namic range of frame energy over the last 100 or 200 ms. JRip outperforms J48, but has far worse training complexity. Separation of speech and silence (which here is the recorded silence in the corpus, *not* digital zero) is done with high accuracy. Recognition of **EoT** regions is of lower quality, but still surpasses results in (Schlangen 2006).[2]

While the data and their states are sequential in nature, the classifiers as described above evaluate each frame independently. At the same time, recognizing the other speaker's start or end of turn a little too late or too early hardly matters, while frequently changing the listening state may lead to bad dialogue behaviour. This is measured in the *false alarm rate* (FAR), defined as the proportion of over-generated state changes.

The analysis of classification output showed that wrong classifications would often last for only one frame. We implemented a *stateful classifier* that only changes state after two consecutive classifications of the underlying classifier. This strongly decreases FAR but introduces systematic errors in the classification (every actual state change will be registered one frame too late) and reduces precision/recall measures. When this is accounted for in the evaluation, the stateful classifier outperforms the base classifier also in these measures.

The results show, that the complexity of the KCoRS is counterbalanced by its 10 times larger size. This may indicate, that speech state classification for real dialogue speech would be feasible with a sufficiently large corpus and speaker-normalized prosodic features.

5. Simple Strategies for Turn-Taking

We outline some simple strategies to turn-control. Their purpose is to exemplify how very restricted locally managed behaviour with some simple rules can already lead to acceptable turn-taking behaviour as postulated by the local management model of Sacks et al. (1974), without the need for a dialogue history, or complex temporal reasoning.

[2]Results cannot be easily compared, as Schlangen (2006) recognizes turn-final *words* using prosodic *and syntactic* features on a more complex corpus, reaching an f-measure of 0.36.

measure	strategy 1		strategy 2		strategy 3	
gap	14.0 %	351 ms	18.7 %	358 ms	17.4 %	362 ms
speaker a	31.4 %	1259 ms	35.9 %	1009 ms	36.5 %	1079 ms
speaker b	39.3 %	1415 ms	39.8 %	1165 ms	40.8 %	1225 ms
clash	15.4 %	1184 ms	5.6 %	317 ms	5.3 %	278 ms

Table 3: Distribution and mean duration of dialogue states for three turn-taking strategies with *pseudo-speech*.

measure	strategy 1		strategy 2		strategy 3	
gap	14.1 %	528 ms	20.7 %	477 ms	18.9 %	454 ms
speaker a	36.2 %	1764 ms	40.5 %	1456 ms	34.7 %	1232 ms
speaker b	26.2 %	1437 ms	24.8 %	1307 ms	42.0 %	1540 ms
clash	23.5 %	1915 ms	4.0 %	253 ms	4.4 %	243 ms

Table 4: Distribution and mean duration of dialogue states for three turn-taking strategies with *KCoRS* speakers.

5.1. Measuring Turn-Management Success

The *dialogue state* can be described by the current speech state of each of the dialogue participants, with each speech state being either **talk** or **sil**. For two-party dialogue, this results in four states: two "good" states where either one of the dialogue participants is talking and two "bad" states: *Clashes* when both participants talk simultaneously, and *gaps* with neither of them talking.

According to Sacks et al. (1974), speakers try to optimize their behaviour so as to minimize the occurence of both clashes and gaps. That is why we choose clashes and gaps as basic measures for turn-taking success. Slight gaps and clashes occur all the time, but they are not always perceptually relevant. We thus decided to calculate the proportion of clashes and gaps over the course of the dialogue as well as their mean duration.

For evaluation purposes, we set up two artificial dialogue participants and let them talk with each other for about 10 minutes for each of the following strategies. We recorded the internal states and calculated the described measures. The audio itself was recorded but not further analyzed in the evaluation. The results of the strategies described below are shown in tables tables 3 and 4.

5.2. Strategy 1: Talk When Nobody Talks

Rule: *Start an utterance when neither you nor your interlocutor is talking.* (Implicitly: *Continue talking until your utterance is finished.*)

The performance with this strategy strongly depends on the round-trip time from one agent's decision to take the turn until the other agent notices the turn being taken. The shorter the lags introduced by the talking agent's internal communication, audio transmission, prosodic processing and classification, and the listening agent's internal communication, the more likely it is for a dialogue participant to notice its interlocutor talking (and then listen until he has finished) *before* she has started talking herself. For longer lags, the DP will decide to talk even though its interlocutor may already have started talking himself. As can be seen, this strategy leads to a large amount of clahes.

5.3. Strategy 2: Hush When Both Talk

Rule as above, plus: *Stop your utterance when both you and your interlocutor are talking.*

The rule proves effective in reducing simultaneous talk as clashes are reduced by 65 % (pseudo-speech) and over 80 % (KCors) respectively. At the same time, this strategy leads to the introduction of utterance truncations, when an utterance was stopped prematurely. (Actually, the majority of utterances (71 % for pseudo-speech) was truncated, but many of these truncations occur in the silent phases before or after the actual talk and do not have any deteriorating effect on the perceived turn-taking performance.) Truncations could be reduced with a higher probability to hush during **SoT**.

5.4. Strategy 3: Start Talking Early

The previous strategies only react after turns have started or ended. In order to initiate actions early and *anticipates* turn changes, this strategy exploits the **EoT** class of the speech analysis (which was ignored before) in the first rule: *Start an utterance, when you are not talking and your interlocutor is ending their turn* or *has already finished.*

By starting utterance planning before the interlocutor's preceding utterance is finished, the dialogue participant can hide some of the lag introduced by its speech generation module. The duration of both gaps and clashes is reduced compared to strategy 2, for gaps because turns will be taken over more quickly and for clashes due to the original talk-owner noticing the turn-change earlier, avoiding the start of a new utterance.

The durations for gaps and clashes with this strategy are similar to those reported for parts of the Verbmobil corpus by Weilhammer & Rabold (2003), with 363 ms and 331 ms respectively.[3] Performance could be further improved by using a lower probability to hush during **EoT**.

6. Conclusion and Future Directions

We have presented a flexible, modular architecture for dialogue strategy evaluation where arbitrary pairings of human users and artificial dialogue participants can be created. We have discussed a case-study in this environment, where pairs of artificial DPs converse in real time via audio. Each DP autonomously decides on their turn-taking behaviour (start or stop talking) based on a local analysis of the audio signal and using machine-learned classifiers. We tested these with corpora of simplified speech and achieve good recognition performance. Three implemented turn-management rulesets, all of them *locally-managed* in the sense of Sacks et al. (1974), i. e. not requiring dialogue memory, were shown to create increasingly realistic behavioural patterns.

We plan to use the components developed for this system in an interactive spoken dialogue system. For the speech state classification, we will need normalized prosodic features that allow for speaker independent speech state classification. At the same time, ASR will make features relative to syllable information (stress patterns, speech rate, ...) accessible, as well as word hypotheses. We may also want to look into classifier confidence scores, only emitting speech state changes if the classifier is reasonably certain.

In real dialogue, the problem of hesitations arises. Our classification will have to be extended to distinguish hesitational interruptions from normal EoT. We would also like to identify positions in a turn where a back-channelling utterance might be appropriate.

[3]Note, that their numbers are for turn changes only, while we do not distinguish between gaps at turn changes and at turn continuations.

Acknowledgements

I would like to thank my supervisor David Schlangen for his constant guidance and support and the anonymous reviewers for their insightful comments and suggestions.

References

A. de Cheveigné & H. Kawahara (2002). 'YIN, a fundamental frequency estimator for speech and music'. *The Journal of the Acoustical Society of America* **111**(4):1917–1930.

L. Ferrer, et al. (2002). 'Is the Speaker Done Yet? Faster and More Accurate End-Of-Utterance Detection Using Prosody'. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP2002)*, Denver, USA.

IPDS (1994). 'The Kiel Corpus of Read Speech'. CD-ROM.

S. C. Levinson (1983). *Pragmatics*. Cambridge Textbooks in Linguistics. Cambridge University Press.

R. López-Cózar, et al. (2003). 'Assessment of dialogue systems by means of a new simulation technique'. *Speech Communication* **40**(3):387–407.

D. Martin, et al. (1999). 'The Open Agent Architecture: a framework for building distributed software systems'. *Applied Artificial Intelligence* **13**(1/2):91–128.

E. G. Padilha (2006). *Modelling Turn-taking in a Simulation of Small Group Discussion*. Ph.D. thesis, School of Informatics, University of Edinburgh, Edinburgh, UK.

H. Sacks, et al. (1974). 'A Simplest Systematic for the Organization of Turn-Taking in Conversation'. *Language* **50**:735–996.

J. Schatzmann, et al. (2006). 'A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies'. *The Knowledge Engineering Review* **21**(02):97–126.

D. Schlangen (2006). 'From Reaction to Prediction: Experiments with Computational Models of Turn-Taking'. In *Interspeech 2006*, Pittsburgh, USA.

H. Schulzrinne, et al. (2003). 'RTP: A Transport Protocol for Real-Time Applications'. RFC 3550 (Standard).

D. Talkin (1995). 'A Robust Algorithm for Pitch Tracking (RAPT)'. In W. B. Kleijn & K. K. Paliwal (eds.), *Speech Coding and Synthesis*, chap. 14, pp. 495–518. Elsevier.

W. Walker, et al. (2004). 'Sphinx-4: A Flexible Open Source Framework for Speech Recognition'. Tech. Rep. SMLI TR2004-0811, Sun Microsystems Inc.

K. Weilhammer & S. Rabold (2003). 'Durational Aspects in Turn Taking'. In *Proc. of the ICPhS*, Barcelona, Spain.

I. H. Witten & E. Frank (2000). *Data Mining. Practical Machine Learning Tools and Techniques with Java Implementations.* Morgan Kaufmann.