A Finite-State Turn-Taking Model for Spoken Dialog Systems

Antoine Raux*
Honda Research Institute
800 California Street
Mountain View, CA 94041, USA
araux@hra.com

Maxine Eskenazi
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
max@cs.cmu.edu

Abstract

This paper introduces the Finite-State Turn-Taking Machine (FSTTM), a new model to control the turn-taking behavior of conversational agents. Based on a non-deterministic finite-state machine, the FSTTM uses a cost matrix and decision theoretic principles to select a turn-taking action at any time. We show how the model can be applied to the problem of end-of-turn detection. Evaluation results on a deployed spoken dialog system show that the FSTTM provides significantly higher responsiveness than previous approaches.

1 Introduction

Turn-taking, the process by which participants in a conversation alternate speech and silence, is an essential component of spoken interaction. In order to lead productive conversations, people need not only know *what* to say but also *when* to say it. Decades of research on Conversation Analysis and psycholinguistics (Duncan, 1972; Sacks et al., 1974; Oreström, 1983; Schegloff, 2000; Wesseling and van Son, 2005) have shown that human turn-taking behavior relies on a wide range of rules and signals at many different levels of language, from prosody to syntax, semantics, and discourse structure. In contrast, turn-taking in spoken dialog systems is often reduced to ad hoc rules only based on very low level features. This simplistic approach leads to inefficient, unnatural, and possibly confusing behavior (Porzel and Baudis, 2004; Ward et al., 2005).

Recently, more complex models of turn-taking have been proposed (Cassell et al., 2001; Thorisson, 2002; Kronild, 2006). Yet, these models still rely extensively on hand-coded expert knowledge and do not lend themselves to data-driven optimization. Furthermore, to our knowledge, no such model has been deployed in a widely used system outside of the laboratory. In this paper, we propose a flexible, practical model of turn-taking behavior that builds upon previous work on finite-state models of the conversational floor. Because of its simplicity and generality, this model can be applied to many turn-taking phenomena. At the same time, being grounded in decision theory, it lends itself well to data-driven optimization. We illustrate our approach by applying the model to a specific turn-taking task: end-of-turn detection.

2 Conversational Floor as a Finite-State Machine

2.1 6-state finite state models of turn-taking

In the 1960's and early 1970's, several researchers proposed models to explain the rhythmic turn-taking patterns in human conversation. In particular, Jaffe and Feldstein (1970) studied the mean duration of pauses, switching pauses (when a different speaker takes the floor), simultaneous speech, and (single-speaker) vocalizations in recorded dyadic conversations. Based on their observation that these durations follow exponential distributions, they proposed first-order Markov models to capture the alternation of speech and silence in dialog. Their initial model had four states: only participant A is speak-

*This research was conducted when the first author was a student at the Language Technologies Institute.

Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL, pages 629–637,
Boulder, Colorado, June 2009.

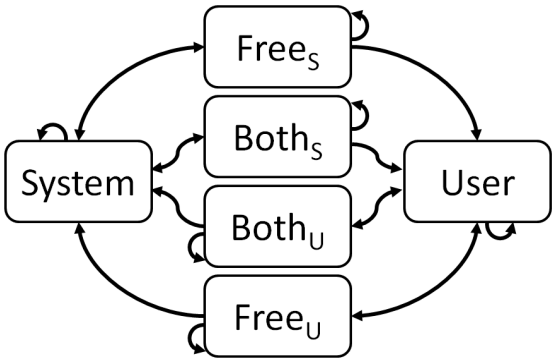


Figure 1: Our six-state model of turn-taking, inspired by Jaffe and Feldstein (1970) and Brady (1969). See section 3.1 for a description of the states.

ing; only participant B is speaking; both participants are speaking; and neither participant is speaking. However, such a model fails to distinguish switching pauses from A to B from switching pauses from B to A. Based on this observation, they extend their model to a six-state model which they found to better fit their data than the four-state model. Around the same time, Brady (1969) developed a very similar six-state model. He trained the parameters on a recorded conversation and compared the generated conversations to the original real one along several dimensions (pause and speech segment durations, overlaps, etc), finding that his model generally produced a good fit of the data.

2.2 Finite-State Models for Control

While Jaffe, Feldstein and Brady were primarily concerned with the analysis of human-human conversations, more recently, several researchers have proposed finite-state machines to control conversational agents. For instance, Cassell et al. (2001) model the conversational state of an embodied real estate agent as a 5-state machine. Two states indicate whether a user is present or not, whereas the other three indicate who holds the floor between the user and the agent, or whether the floor is open. Floor conflicts are not captured by this machine and are presumably resolved through simple rules (e.g. when the user speaks, the agent yields the floor).

Kronild (2006) proposes a much more complex model, based on Harel statecharts, which are an extension of finite-state machines for modeling and visualizing abstract control (Harel, 1987).

Thorisson's Ymir architecture (Thorisson, 2002) is an attempt to model the cognitive processes involved in conversation. It features dialog states, capturing, for example, who has the floor, and rules that govern the transition from one state to another based on "boolean conditions of perceptual features".

All these models are deterministic. At any point in time, the agent *knows* who owns the floor and uses fixed rules to take appropriate actions. These approaches assume 1) that the system can obtain perfectly reliable information on the state of the world, and 2) that the state itself is unambiguous.

3 The Finite-State Turn-Taking Machine

3.1 Extending the 6-state model for control

Our model, the Finite-State Turn-Taking Machine (FSTTM), uses the same six states as Jaffe and Feldstein: $USER$ and $SYSTEM$ represent states where one and only one of the participants claims the floor, $FREE_S$ and $FREE_U$ states where no participant claims the floor (following, resp., a $SYSTEM$ and $USER$ state), and $BOTH_S$ and $BOTH_U$ states where both participants claim the floor (following, resp. a $SYSTEM$ and $USER$ state). However, we apply this model to the control of a conversational agent, with a goal similar to that of Cassel, Thorisson, and Kronild. One important distinction is that we define the states in terms of the participants' *intentions* and *obligations* (in the sense of Traum and Allen (1994)) rather than the surface level observation of speech vs silence. For example, the state is $USER$ when the user has the obligation to speak (to respond to a system question) or the intention to speak, while at the same time, the system does not hold the floor. This does not necessarily mean that the user is speaking, for example at pauses during a user utterance.

As can be seen in Figure 1, not all transitions are valid. First, there is no direct transition between any of the intermediate states (the two $FREE$ states and two $BOTH$ states). The assumption is that to go from any of these state to another, the model will first go to either $SYSTEM$ or $USER$. This is an

approximation as there might be cases where, for example, both the system and user start speaking at the exact same time, going from a $FREE$ state to a $BOTH$ state. However these cases are rare enough that they can be approximated using a transition through either $SYSTEM$ or $USER$. Second, because intermediate states are conditioned on who had the floor previously, not all valid transitions are bidirectional. For example, there is no transition from $SYSTEM$ to $BOTH_U$. We associate pairs of user/system actions to each transition. The four possible actions are Grab the floor, Release the floor, Wait while not claiming the floor, and Keep the floor. For example, transition from $SYSTEM$ to $FREE_S$ corresponds to the user waiting silently and the system releasing the floor at the end of a prompt, noted (R, W) (we always note the system action first and user action second).

This representation allows us to formalize a wide variety of turn-taking phenomena in a unified framework. Specifically, there are 4 types of 2-step transitions from a single-floor-holder state ($SYSTEM$ or $USER$) to another (or the same) single-floor-holder state, which represent typical turn-taking phenomena:

Turn transitions with gap are the most common way the floor goes from one participant to the other. For example, at the end of a user utterance, once the user finishes speaking, the floor becomes free, after which the system starts responding, thus grabbing the floor. The resulting state sequence is:

$$SYSTEM \overset{(R,W)}{\rightarrow} FREE_S \overset{(W,G)}{\rightarrow} USER$$

Conversely, the transition with gap following a system prompt corresponds to:

$$USER \overset{(R,W)}{\rightarrow} FREE_S \overset{(W,G}{\rightarrow} USER$$

Turn transitions with overlap happen when a participant grabs the floor while it still belongs to the other. For example, when a user barges in on a system prompt, both participants hold the floor. Then, the system recognizes the barge-in attempt and relinquishes the floor, which becomes user's.

$$SYSTEM \overset{(K,G)}{\rightarrow} BOTH_S \overset{(R,K}{\rightarrow} USER$$

And conversely, when the system interrupts the user mid-utterance (which in dialog systems is more often the result of an intentional cut-in, rather than intentional interruption), the state sequence is:

$$USER \overset{(G,K)}{\rightarrow} BOTH_U \overset{(K,R)}{\rightarrow} SYSTEM$$

Failed interruptions happen when a participant barges in on the other and then withdraws before the original floor holder releases the floor. For example, when the system interrupts the user (often by mistake) but detects it and interrupts itself:

$$USER \overset{(G,K)}{\rightarrow} BOTH_U \overset{(R,K}{\rightarrow} USER$$

The converse is usually the result of the system failing to react fast enough to a user barge-in:

$$SYSTEM \overset{(K,G)}{\rightarrow} BOTH_S \overset{(K,R)}{\rightarrow} SYSTEM$$

Note that backchannels seem to fit in this category too. However, since backchannels, by definition, do not represent an attempt to grab the floor, they are not captured by the model as it is (for example, the floor should remain $SYSTEM$ when a user backchannels a system utterance).

Time outs start like transitions with gap but the intended next speaker (e.g. the user after a system prompt) does not take the floor and the original floor holder grabs it back. For instance, after a system prompt, if the floor remains free for a certain amount of time, the system attempts to re-establish the communication with the user, as follows:

$$SYSTEM \overset{(R,W)}{\rightarrow} FREE_S \overset{(G,W}{\rightarrow} SYSTEM$$

The opposite also happens when the system is to slow to respond to the user:

$$USER \overset{(W,R)}{\rightarrow} FREE_U \overset{(W,G}{\rightarrow} USER$$

While all the transitions above were described as deterministic, the actual state of the model is not fully observable. Specifically, while the system

knows whether its claiming the floor or not, it can only *believe* with some degree of uncertainty that the user does so. The system's knowledge of its own claim to the floor splits the state space into two disjoint subsets. When the system claims the floor, the state can be $SYSTEM$, $BOTH_S$, or $BOTH_U$). When the system does not claim the floor, the state can be $USER$, $FREE_U$, or $FREE_S$). In either case, the system needs to recognize the user's intention (i.e. whether the user claims to the floor or not) to maintain a probability distribution over the three states. Since the distinction between the two $BOTH$ states (resp. the two $FREE$ states) is based on past history that can be known with a high level of certainty, the uncertainty in state distribution is fully characterized by the probability that the user is claiming the floor, which will have to be estimated from observations, as we will see below.

3.2 Cost of Turn-Taking Actions

The problem we are facing is that of choosing the best system action given the system's belief about the current state of the model. That is achieved by applying the probabilistic decision theory principle of selecting the action with lowest expected cost. The actions available to the system are the four described above (G,R,K,W), although not all actions are available in all states. In fact, as can be seen in Table 1, there are always only two actions available in each state, depending on whether the system is claiming the floor or not.

Each action in each state has a particular cost. While there are many possible ways of defining these costs, we propose a simple cost structure that derives from the principles laid out in Sacks et al. (1974):

> Participants in a conversation attempt to minimize gaps and overlaps.

From this general principle, we derive three rules to drive the design of a cost matrix:

1. The cost of an action that resolves either a gap or an overlap is zero
2. The cost of an action that creates unwanted gap or overlap is equal to a constant parameter (potentially different for each action/state pair)
3. The cost of an action that maintains a gap or overlap is either a constant or an increasing function of the total time spent in that state

The resulting cost matrix is shown in Table 1, where

- $\boldsymbol{C_S}$ is the cost of interrupting a system prompt before its end when the user is not claiming the floor (false interruption)
- $\boldsymbol{C_O(\tau)}$ is the cost of remaining in an overlap that is already τ ms long
- $\boldsymbol{C_U}$ is the cost of grabbing the floor when the user is holding it (cut-in)
- $\boldsymbol{C_G(\tau)}$ is the cost of remaining in a gap that is already τ ms long

This cost structure makes a number of simplifying assumptions and there are many other possible cost matrices. For example, the cost of interrupting the user might vary depending on what has already been said in the utterance, so does the cost of interrupting a system prompt. A more principled approach to setting the costs would be to estimate from perceptual experiments or user studies what the impact of remaining in gap or overlap is compared to that of a cut-in or false interruption. However, as a first approximation, the proposed cost structure offers a simple way to take into account some of the constraints of interaction.

3.3 Decision Theoretic Action Selection

Given the state space and the cost matrix given above, the optimal decision at any point in time is the one that yields the lowest expected cost, where the expected cost of action A is:

$$C(A) = \sum_{S \in \Sigma} P(s = S|O) \cdot C(A, S)$$

where Σ is the set of states, O are the observable features of the world, and $C(A, S)$ is the cost of action A in state S, from the cost matrix in Table 1. In addition to the cost matrix' four constants, which we will consider as parameters of the model, it is thus necessary to estimate $P(s = S|O)$, which as seen above amounts to estimate the probability that the user is claiming the floor. Key to applying the FSTTM to a practical turn-taking problem is thus the construction of accurate estimates of the probabilities $P(s = S|O)$.

State \ Action	K	R	W	G
$SYSTEM$	0	C_S	-	-
$BOTH_S$	$C_O(\tau)$	0	-	-
$BOTH_U$	$C_O(\tau)$	0	-	-
$USER$	-	-	0	C_U
$FREE_U$	-	-	$C_G(\tau)$	0
$FREE_S$	-	-	$C_G(\tau)$	0

Table 1: Cost of system actions in each state (K: keep the floor, R: release the floor, W: wait without the floor, G: grab the floor, τ: time spent in current state, -: action unavailable).

4 Endpointing with the FSTTM

4.1 Problem formalization

In our FSTTM formalism, endpointing is the problem of selecting between the Wait and the Grab actions during a user utterance. We make the simplifying assumption that, once a user utterance has been detected, the only states with non-zero probability are $USER$ and $FREE_U$. While this does not capture cases where the system erroneously detects user speech (because there is, for example, background noise), it represents a working first approximation of the problem.

The main issue is to estimate the probability $P(s = FREE_U|O_t)$ (hereafter abbreviated as $P(F|O_t)$, $P(s = USER|O_t)$ being abbreviated as $P(U|O_t)$) where O_t represents all observable features at time t. Given that probability, the expected cost of grabbing the floor is:

$$\begin{aligned} C(G|O_t) &= P(U|O_t) \cdot C_U + P(F|O_t) \cdot 0 \\ &= (1 - P(F|O_t)) \cdot C_U \end{aligned}$$

Similarly, the expected cost of waiting is:

$$C(W|O_t) = P(F|O_t) \cdot C_G(\tau)$$

The system endpoints whenever the expected cost of grabbing the floor becomes higher than that of waiting.

We consider two separate cases for computing both $P(F|O_t)$ and $C_G(\tau)$: when a pause has been detected by the voice activity detector (VAD), and when no pause has been detected (yet). In the following sections, we provide details on the approximations and estimation methods for these two cases.

4.2 At pauses

If a pause has been detected by the VAD, we set the cost of waiting in the $FREE_U$ state to be proportional to the duration of the pause so far. If the user has released the floor, the duration of the current pause corresponds to the time spent in the $FREE_U$ state, i.e. τ in the cost matrix of Table 1. In this case, we set $C_G(\tau) = C_G^p \cdot \tau$ as a simple application of rule 3 from section 3.2.

We decompose the observations at time t, O_t, into observations available at the start of the pause (O), and observations made *during* the pause. With only audio information available, the only information available during the pause is its duration so far, i.e. τ. Specifically, we know that $d \geq \tau$, where d is the total duration of the pause (with $d = \infty$ at the end of a turn[1]). Consequently, $P(F|O_t)$ can be rewritten using Bayes rule as

$$\begin{aligned} P(F|O_t) &= \frac{P(d \geq \tau|O, F) \cdot P(F|O)}{P(d \geq \tau|O)} \\ &= \frac{P(F|O)}{P(d \geq \tau|O)} \end{aligned}$$

where $P(F|O)$ is the probability that the user released the floor without any knowledge of the duration of the pause, and $P(d \geq \tau|O)$ is the probability that the pause will last at least τ ms. We further decompose $P(d \geq \tau|O)$ into

$$P(d \geq \tau|O) = P(d \geq \tau, U|O) + P(d \geq \tau, F|O)$$

[1]Note that this is an approximation since the user could start speaking again after releasing the floor to reestablish the channel (e.g. by saying "Hello?"). However, in the vast majority of cases, the time after which the user resumes speaking is significantly longer than the time the system takes to endpoint.

$$
\begin{aligned}
&= P(d \geq \tau|O,U) \cdot P(U|O) + \\
&\quad P(d \geq \tau|O,F) \cdot P(F|O) \\
&= P(d \geq \tau|O,U) \cdot (1 - P(F|O)) \\
&\quad + P(F|O)
\end{aligned}
$$

Consequently, $P(F|O_t)$ is a function of $P(F|O)$ and $P(d \geq \tau|O,U)$. We estimate $P(F|O)$ by stepwise logistic regression on a training set of pauses labeled for finality (whether the pause is turn-final or turn-internal), using a wide range of features available from various components of the dialog system. Based on the well established observation that pause durations follow an exponential distribution (Jaffe and Feldstein, 1970; Lennes and Anttila, 2002; Raux et al., 2008), $P(d \geq \tau|O,U)$ is a function of mean pause duration, computed on the training set.

4.3 In speech

In some cases, it is not necessary to wait for the VAD to detect a pause to know with high confidence that the user has released the floor. For example, after a simple yes/no question, if the user says "YES", they are very likely to have released the floor, regardless of how long they remain silent afterwards. In order to exploit this fact and improve the responsiveness of the system in these highly predictable cases, we use a separate model to compute the expected costs of waiting and grabbing the floor before any pause is detected by the VAD (specifically, whenever the duration of the current pause is between 0 and 200 ms). In this case, we set the cost of waiting to a constant C_G^s. We train a logistic regression model to estimate $P(F|O_t)$ each time a new partial hypothesis is produced by the ASR during a user utterance. We use the same set of features as above.

5 Evaluation

5.1 Corpus and Features

We evaluated the effectiveness of the FSTTM on an actual deployed spoken dialog system. The system provides bus schedule information for a mid-size North American city. It is actually used by the general public and therefore constantly operates and collects data. In order to train the various probability estimation models and evaluate the approach in batch, we first collected a corpus of 586 dialogs between May 4, and May 14, 2008 (the "2008 corpus").

All of the features we used can be automatically extracted at runtime, and most of them were readily available in the system. They include dialog state information, turn-taking features, such as whether the current user utterance is a barge-in, and semantic information derived from the dialog state and partial recognition hypotheses provided by the speech recognizer. Dialog state is abstracted to three high-level states, which correspond to the type of system prompt directly preceding the user utterance: Open question ("What can I do for you?"); Closed question (e.g. "Where do you want to go?"); and Confirmation (e.g. "Going to the airport. Is this correct?").

To capture lexical cues correlated with the end of turns, we created a new feature called the boundary LM score. To compute it, we used previously collected data to train dialog-state-dependent statistical language models to estimate the probability that the hypothesis is complete. Boundary LM score is defined as the ratio of the log likelihood of the hypothesis being complete by that of the hypothesis being incomplete.

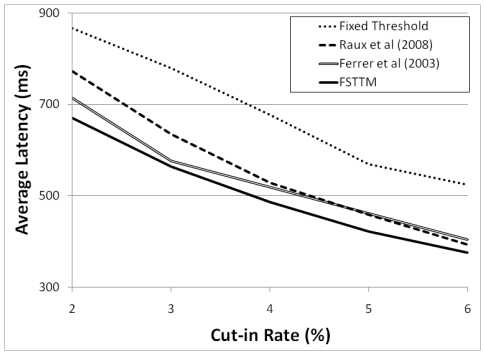
5.2 Estimating $P(F|O_t)$

We trained two logistic regression models using stepwise regression and 10-fold cross-validation for evaluation. The first model, whose performance is given in Table 2, estimates $P(F|O)$ at pauses. The model is unable to improve classification accuracy over the majority baseline for each state, however, the statistically significant improvement in average log likelihood indicates that the probability estimates are improved by using the features. The most informative feature in all three states was the boundary LM score introduced in section 5.1. Other selected features included the average number of words per user utterance so far and whether the current utterance is a barge-in (for the Open and Closed question states), as well as whether the partial hypothesis contained a confirmation marker such as "YES" or "SURE" (for the Confirmation state).

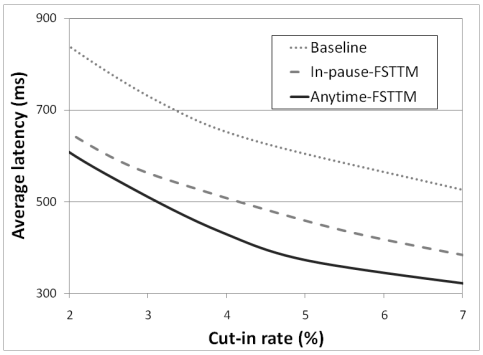
The second model performs the same regression, this time on all partial hypotheses received during speech segments. As seen in the "S" columns in Table 2, classification error was significantly reduced and the gain in average log likelihood were larger

	Open question		Closed question		Confirmation	
	P	S	P	S	P	S
Majority Baseline	38%	20%	25%	32%	12%	36%
Classification Error	35%	17%	26%	22%	12%	17%
Baseline log likelihood	-0.66	-0.50	-0.56	-0.63	-0.36	-0.65
Log likelihood	-0.61	-0.40	-0.50	-0.49	-0.30	-0.40

Table 2: Performance of state-specific logistic regression for estimating $P(F|O)$ at pauses (P) and in speech (S).



(a) In-pause evaluation on the 2007 corpus.



(a) Anytime evaluation on the 2008 corpus.

Figure 2: Batch evaluation of FSTTM endpointing.

than at pauses, particularly for the "Closed question" and "Confirmation" states. Again, boundary LM score was the most informative feature. The duration of the pause at the end of the partial hypothesis (between 0 and 200 ms) also proved well correlated with finality.

5.3 Batch Evaluation of the FSTTM

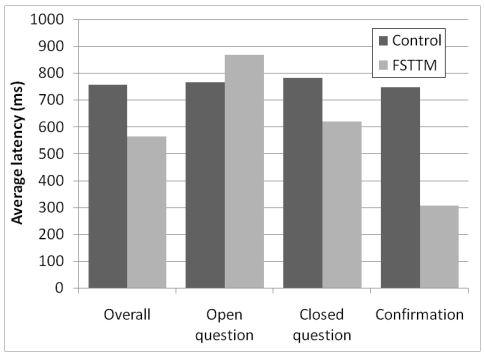
We performed two batch evaluations of the FSTTM. The first one aims at comparing in-pause-FSTTM with a fixed-threshold baseline as well as previous data-driven endpointing methods proposed in Ferrer et al. (2003) (reimplemented by us) and Raux et al. (2008). This evaluation was done on the corpus used in Raux et al. (2008) (the "2007 corpus"). As seen in Figure 2 (a), the FSTTM outperforms all other approaches (albeit only slightly compared to Ferrer et al.), improving over the fixed threshold baseline by up to 29.5%.

Second, we compared the anytime-FSTTM with in-pause-FSTTM and a fixed-threshold baseline (for reference) on the more recent 2008 corpus (since the 2007 corpus did not contain all necessary features for anytime-FSTTM). We set $C_G^p = 1$ and set C_G^s to either 0, leading to an endpointer that never endpoints during speech (in-pause-FSTTM), or 1000 (anytime-FSTTM). In both cases, we vary C_U to compute the latency / cut-in rate trade-off curve. The results are shown in Figure 2 (b). Anytime-FSTTM endpointing is consistently better than in-pause-FSTTM. For example, at a cut-in rate of 5%, anytime-FSTTM yields latencies that are on average 17% shorter than in-pause-FSTTM, and 40% shorter than the baseline. Additionally, we found that, in anytime-FSTTM, 30 to 40% of the turns are endpointed before the pause is detected by the VAD.

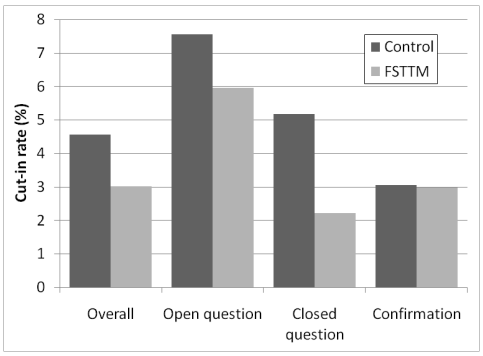
5.4 Live Evaluation

To confirm the results of the batch evaluation, we implemented our FSTTM model in the deployed system a let it run for ten days using either FSTTM or a fixed threshold for endpointing, resulting in a corpus of 171 FSTTM and 148 control dialogs. For FSTTM, we set $C_G^p = 1$, $C_G^s = 500$, and $C_U = 5000$. In the batch evaluation, these values correspond to a cut-in rate of 6.3% and an average latency of 320 ms. For the control condition, we set the fixed endpointing threshold to 555 ms, which also corresponded to about 6.3% cut-ins.

Figure 3 shows the average latency and cut-in rate



(a) Latency



(b) Cut-in rates

Figure 3: Live evaluation results. All confidence intervals for latency (not shown on the figure) fall within $+/-4ms$.

for both conditions. The FSTTM improves over the baseline on all metrics, reducing average latency by 193 ms ($p < 0.05$), cut-in rate by 1.5% (although this result is not statistically significant).

6 Discussion

Both batch and live evaluation results confirm the effectiveness of the FSTTM approach in improving system responsiveness. This approach significantly reduced endpointing latency over previous approaches. Boundary LM score got the highest weight in the regression, indicating that in a domain such as telephone-based information access, lexical cues are very informative for endpointing. The fact that boundary LMs can be computed without any human transcription effort (since they are trained on ASR output) makes them all the more appealing.

Essentially, the FSTTM provides a simple, unified model of turn-taking that lends itself to data-driven optimization. While we discussed specific cost structures and probability estimation techniques, the framework's flexibility opens it to other choices at many levels. By formalizing the overall turn-taking process in a probabilistic, decision-theoretic framework, the FSTTM extends and generalizes previous classification-based approaches to endpointing such as those proposed by Sato et al. (2002), Ferrer et al. (2003), Takeuchi et al. (2004), and our previous work (Raux et al., 2008).

Possible extensions of the approach include data-driven cost matrices to relax some of the assumptions introduced in section 3.2, as well as more complex state structures to handle, for example, multi-party conversations.

Finally, we plan to investigate more principled approaches, such as Partially Observable Markov Decision Processes or Dynamic Bayesian Networks, to model the different sources of uncertainty (detection errors and inherent ambiguity) and track the state distribution over time. Raux (2009) provides more details on all aspects of the approach and its possible extensions.

7 Conclusion

In this paper, motivated by existing finite-state models of turn-taking in dyadic conversations, we propose the Finite-State Turn-Taking Machine, an approach to turn-taking that relies on three core elements: a non-deterministic finite-state machine that captures the conversational floor; a cost matrix that models the impact of different system actions in different states; and a decision-theoretic action selection mechanism. We describe the application of the FSTTM to the key turn-taking phenomenon of end-of-turn detection. Evaluation both offline and by applying the FSTTM to a deployed spoken dialog system system showed that it performs significantly better than a fixed-threshold baseline.

Acknowledgments

This work is supported by the US National Science Foundation under grant number 0208835. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF. We would like to thank Alan Black for his many comments and advice.

References

P. T. Brady. 1969. A model for generating on-off speech patterns in two-way conversation. *The Bell System Technical Journal*, 48:2445–2472.

J. Cassell, T. Bickmore, L. Campbell, H. Vilhjalmsson, and H. Yan. 2001. More than just a pretty face: conversational protocols and the affordances of embodiment. *Knowledge-Based Systems*, 14:55–64.

S. Duncan. 1972. Some signals and rules for taking speaking turns in conversations. *Journal of Personality and Social Psychology*, 23(2):283–292.

L. Ferrer, E. Shriberg, and A. Stolcke. 2003. A prosody-based approach to end-of-utterance detection that does not require speech recognition. In *ICASSP*, Hong Kong.

D. Harel. 1987. Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, 8:231–274.

J. Jaffe and S. Feldstein. 1970. *Rhythms of Dialogue*. Academic Press.

F. Kronild. 2006. Turn taking for artificial conversational agents. In *Cooperative Information Agents X*, Edinburgh, UK.

Mietta Lennes and Hanna Anttila. 2002. Prosodic features associated with the distribution of turns in finnish informal dialogues. In Petri Korhonen, editor, *The Phonetics Symposium 2002*, volume Report 67, pages 149–158. Laboratory of Acoustics and Audio Signal Processing, Helsinki University of Technology.

B. Oreström. 1983. *Turn-Taking in English Conversation*. CWK Gleerup, Lund.

R. Porzel and M. Baudis. 2004. The tao of chi: Towards effective human-computer interaction. In *HLT/NAACL 2004*, Boston, MA.

A. Raux, , and M. Eskenazi. 2008. Optimizing endpointing thresholds using dialogue features in a spoken dialogue system. In *Proc. SIGdial 2008*, Columbus, OH, USA.

A. Raux. 2009. *Flexible Turn-Taking for Spoken Dialog Systems*. Ph.D. thesis, Carnegie Mellon University.

H. Sacks, E. A. Schegloff, and G. Jefferson. 1974. A simplest systematics for the organization of turn-taking for conversation. *Language*, 50(4):696–735.

R. Sato, R. Higashinaka, M. Tamoto, M. Nakano, and K. Aikawa. 2002. Learning decision trees to determine turn-taking by spoken dialogue systems. In *ICSLP 2002*, Denver, CO.

E.A. Schegloff. 2000. Overlapping talk and the organization of turn-taking for conversation. *Language in Society*, 29:1–63.

M. Takeuchi, N. Kitaoka, and S. Nakagawa. 2004. Timing detection for realtime dialog systems using prosodic and linguistic information. In *Proc. Speech Prosody 04*, Nara, Japan.

K. R. Thorisson, 2002. *Multimodality in Language and Speech Systems*, chapter Natural Turn-Taking Needs No Manual: Computational Theory and Model, From Perception to Action, pages 173–207. Kluwer Academic Publishers.

D. R. Traum and J. F. Allen. 1994. Discourse obligations in dialogue. In *Proc. ACL-94*, pages 1–8.

N. Ward, A. Rivera, K. Ward, and D. Novick. 2005. Root causes of lost time and user stress in a simple dialog system. In *Interspeech 2005*, Lisbon, Portugal.

W. Wesseling and R.J.J.H. van Son. 2005. Timing of experimentally elicited minimal responses as quantitative evidence for the use of intonation in projecting TRPs. In *Interspeech 2005*, pages 3389–3392, Lisbon, Portugal.