Optimizing Endpointing Thresholds using Dialogue Features in a Spoken Dialogue System

Antoine Raux and **Maxine Eskenazi**
{antoine,max}@cs.cmu.edu
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Abstract

This paper describes a novel algorithm to dynamically set endpointing thresholds based on a rich set of dialogue features to detect the end of user utterances in a dialogue system. By analyzing the relationship between silences in user's speech to a spoken dialogue system and a wide range of automatically extracted features from discourse, semantics, prosody, timing and speaker characteristics, we found that all features correlate with pause duration and with whether a silence indicates the end of the turn, with semantics and timing being the most informative. Based on these features, the proposed method reduces latency by up to 24% over a fixed threshold baseline. Offline evaluation results were confirmed by implementing the proposed algorithm in the Let's Go system.

1 Introduction

1.1 Responsiveness in Dialogue

Although the quality of speech technologies has improved drastically and spoken interaction with machines is becoming a part of the everyday life of many people, dialogues with artificial agents still fall far short of their human counterpart in terms of both comfort and efficiency. Besides lingering problems in speech recognition and understanding, Ward et al (Ward et al., 2005) identified turn-taking issues, specifically responsiveness, as important shortcomings. Dialogues with artificial agents are typically rigid, following a strict one-speaker-at-a-time structure with significant latencies between turns. In a previous paper, we concurred with these findings when analyzing issues with the Let's Go system (Raux et al., 2006). In contrast, empirical studies of conversation have shown that human-human dialogues commonly feature swift exchanges with little or no gap between turns, or even non-disruptive overlap (Jaffe and Feldstein, 1970; Sacks et al., 1974). According to Conversation Analysis and psycholinguistic studies, responsiveness in human conversations is possible because participants in the conversation exchange cues indicating when a turn might end, and are able to anticipate points at which they can take over the floor smoothly. Much research has been devoted to finding these cues, leading to the identification of many aspects of language and dialogue that relate to turn-taking behavior, including syntax (Sacks et al., 1974; Ford and Thompson, 1996; Furo, 2001), prosody (Duncan, 1972; Oreström, 1983; Chafe, 1992; Ford and Thompson, 1996; Koiso et al., 1998; Furo, 2001), and semantics (Oreström, 1983; Furo, 2001). However, regarding this last aspect, Orestrom notes about his corpus that ”there is no simple way to formalizing a semantic analysis of this conversational material”. This difficulty in formalizing higher levels of conversation might explain the relatively low interest that conversational analysts have had in semantics and discourse. Yet, as conversational analysts focused on micro-levels of dialogue such as turn-taking, computational linguists uncovered and formalized macro-level dialogue structure and devised well-defined representations of semantics for at least some forms of dialogues (Allen and Perrault, 1980; Grosz and Sidner, 1986; Clark, 1996), which have in turn been implemented in spoken dialogue systems (Rich and Sidner, 1998; Allen et al., 2005).

Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue, pages 1–10,
Columbus, June 2008.

1.2 Current Approaches to Turn-Taking in Spoken Dialogue Systems

Unfortunately, while socio- and psycho-linguists revealed the complexity of conversational turn-taking behavior, designers of practical spoken dialogue systems have stuck to a simplistic approach to end-of-turn detection (hereafter *endpointing*). Typically, silences in user speech are detected using a low-level Voice Activity Detector (VAD) and a turn is considered finished once a silence lasts longer than a fixed threshold. This approach has the advantage of being simple, only relying on easily computable low-level features. However, it leads to suboptimal behavior in many instances. First, False Alarms (FA) happen when a pause lasts longer than the threshold and gets wrongly classified as a gap[1]. Second, latency occurs at every gap, because the system must wait for the duration of the threshold before classifying a silence as gap. When setting the threshold, system designers must consider the trade-off between these two issues: setting a low threshold reduces latency but increases FA rate, while setting a high threshold reduces FA rate but increases latency.

To help overcome the shortcomings of the single-threshold approach, several researchers have proposed to exploit various features. Sato et al (Sato et al., 2002) used decision trees to classify pauses longer than 750 ms as gap or pause. By using features from semantics, syntax, dialogue state, and prosody, they were able to improve the classification accuracy from a baseline of 76.2% to 83.9%. While this important study shows encouraging results on the value of using various sources of information in a dialogue system, the proposed approach (classifying long silences) is not completely realistic (what happens when a gap is misclassified as a pause?) and does not attempt to optimize latency. An extension to this approach was proposed in (Takeuchi et al., 2004), in which a turn-taking decision is made every 100 ms during pauses. However, in this latter work the features are limited to timing, prosody, and syntax (part-of-speech). Also the reported classification results, with F-measures around 50% or below do not seem to be sufficient for practical use.

[1]We use the terminology from (Sacks et al., 1974) where a *pause* is a silence within a turn while a *gap* is a silence between turns. We use the term *silence* to encompass both types.

Similarly, Ferrer and her colleagues (Ferrer et al., 2003) proposed the use of multiple decision trees, each triggered at a specific time in the pause, to decide to either endpoint or defer the decision to the next tree, unless the user resumes speaking. Using features like vowel duration or pitch for the region immediately preceding the silence, combined with a language model that predicts gaps based on the preceding words, Ferrer et al are able shorten latency while keeping the FA rate constant. On a corpus of recorded spoken dialogue-like utterances (ATIS), they report reductions of up to 81% for some FA rates. While very promising, this approach has several disadvantages. First it relies on a small set of possible decision points for each pause, preventing fine optimization between them. Second, the trees are trained on increasingly smaller datasets requiring smoothing of the tree scores to compensate for poor training of the later trees (which are trained on increasingly small subsets of pauses from the training set). Finally, and perhaps most importantly, these authors have investigated prosodic and lexical features, but not other aspects of dialogue, such as discourse structure, timing, and semantics.

In this paper, we propose a new approach to endpointing that directly optimizes thresholds using automatically extracted dialogue features ranging from discourse to timing and prosody. Section 2 outlines the proposed algorithm. Section 3 describes the analysis of the relationship between silences and a wide range of features available to a standard spoken dialogue system (hereafter *dialogue features*). Evaluation results, both offline and in the deployed Let's Go system are given in Section 4.

2 Dynamic Endpointing Threshold Decision Trees

2.1 Overview

One issue with current approaches to endpointing is that they rely on binary gap/pause classifiers and the relationship between optimizing for classification accuracy vs optmizing to minimize latency is unclear. Also, the performance we obtained when applying classification-based approaches to the Let's Go data was disappointing. The accuracy of the classifiers was not sufficient for practical purposes, even with the improvements proposed by (Ferrer et al., 2003). We hypothesize that the discrepancy between these results and the good performances reported by others is due to the noisiness of the Let's Go data (see Section 3.1.1). To overcome these issues, we propose a method that directly optimizes endpointing thresholds using a two-stage process. First, silences are clustered based on dialogue features so as to create groups of silences with similar properties. Second, a single threshold is set for each cluster, so as to minimize the overall latency at a given false alarm rate. The result of the training process is thus a decision tree on dialogue features that contains thresholds at its leaves. At runtime, every time a silence is detected, the dialogue system runs the decision tree and sets its endpointing threshold accordingly. The following sections describe the two training stages.

2.2 Feature-based Silence Clustering

The goal of the first stage of training is to cluster silences with a similar FA rate/latency tradeoff. The intuition is that we would like to generate low-threshold clusters, which contain mostly gaps and short pauses, and clusters where long pauses would be concentrated with no or very few gaps, allowing to set high thresholds that reduce cut-in rate without hurting overall latency. We used a standard top-down clustering algorithm that exhaustively searches binary splits of the data based on feature values. The split that yields the minimal overall cost is kept, where the cost C_n of cluster K_n is defined by the following function:

$$C_n = G_n \times \sqrt{\frac{1}{|K|} \sum_{p \in K} Duration(p)^2} \quad (1)$$

where G_n the number of gaps in K_n and $Duration(p)$ the duration of a pause p, set to zero for gaps. While other cost functions are possible, the intuition behind this formula is that it captures both the cluster's gap ratio (first factor) and its pause duration distribution (second factor: root mean square of pause duration). The splitting process is repeated recursively until the reduction in cost between the original cost and the sum of the costs of the two split clusters falls below a certain threshold. By minimizing $C(K)$, the clustering algorithm will find questions that yield clusters with either a small G_n, i.e. mostly pauses, or a small root mean square pause duration. Ultimately, at the leaves of the tree are sets of silences that will share the same threshold.

2.3 Cluster Threshold Optimization

Given the clusters generated by the first phase, the goal of the second phase is to find a threshold for each cluster so that the overall latency is minimized at a given FA rate. Under the assumption that pause durations follow an exponential distribution, which is supported by previous work and our own data (see Section 3.2), we show in Figure 3 in appendix that there is a unique set of thresholds that minimizes latency and that the threshold for any cluster n is given by:

$$\theta_n = \frac{\mu_n \times log(\beta_n \times \frac{E \times \mu_n}{\sum \mu_n})}{G_n} \quad (2)$$

where μ_n and β_n can be estimated from the data.

3 Silences and Dialogue Features

3.1 Overview of the Data

3.1.1 The Let's Go Corpus

Let's Go is a telephone-based spoken dialogue system that provides bus schedule information for the Pittsburgh metropolitan area. It is built on the Olympus architecture (Bohus et al., 2007), using the RavenClaw dialogue management framework, and the Apollo interaction manager (Raux et al., 2007) as core components. Outside of business hours callers to the bus company's customer service are offered the option to use Let's Go. All calls are recorded and extensively logged for further analysis. The corpus used for this study was collected between December 26, 2007 and January 25, 2008, with a total of 1326 dialogues, and 18013 user turns. Of the calls that had at least 4 user turns, 73% were complete, meaning that the system provided some schedule information to the user.

While working on real user data has its advantages (large amounts of data, increased validity of the results), it also has its challenges. In the case of Let's Go, users call from phones of varying quality (cell phones and landlines), often with background noises such as cars, infant cries, loud television sets, etc. The wide variability of the acoustic conditions makes any sound processing more prone to error than on carefully recorded corpora. For example, as reported in (Raux et al., 2005), the original speech recognizer had been found to yield a 17% word error rate on a corpus of dialogues collected by recruiting subjects to call the system from an office. On the live Let's Go data, that same recognizer had a 68% WER. After acoustic and language model retraining/adaptation, that number was brought down to about 30% but it is still a testimony to the difficulty of obtaining robust features, particularly from acoustics.

3.1.2 Correcting Runtime Endpointing Errors

Let's Go uses a GMM-based VAD trained on previously transcribed dialogues. Endpointing decisions are based on a fixed 700 ms threshold on the duration of the detected silences. One issue when analyzing pause distributions from the corpus is that observed user behavior was affected by system's behavior at runtime. Most notably, because of the fixed threshold, no recorded pause lasts more than 700 ms. To compensate for that, we used a simple heuristic to rule some online endpointing decisions as erroneous. If a user turn is followed within 1200 ms by another user turn, we consider these two turns to be in fact a single turn, unless the first turn was a user barge-in. This heuristic was established by hand-labeling 200 dialogues from a previous corpus with endpointing errors (i.e. each turn was annotated as correctly or incorrectly endpointed). On this dataset, the heuristic has a precision of 70.6% and a recall of 75.5% for endpointing errors. Unless specified, all subsequent results are based on this modified corpus.

3.2 Turn-Internal Pause Duration Distribution

Overall there were 9563 pauses in the corpus, which amounts to 0.53 pauses per turn. The latency / FA rate trade-off for the corpus is plotted in Figure 1. This curve follows an exponential function (the R^2 on the linear regression of latency on $Log(FA)$ is 0.99). This stems from the fact that pause duration approximately follows an exponential distribution, which has been observed by others in the past (Jaffe and Feldstein, 1970; Lennes and Anttila, 2002).

One consequence of the exponential-like distribution is that short pauses strongly dominate the distribution. We decided to exclude silences shorter than

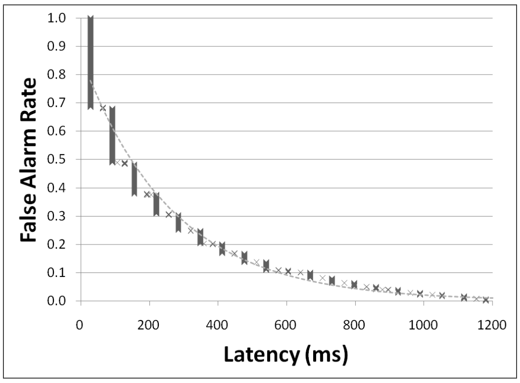

Figure 1: Overall False Alarm / Latency trade-off in the Let's Go corpus. The dashed line represents a fitted curve of the form $FA = e^{\beta + \alpha \cdot Latency}$.

200 ms from most of the following analysis for two reasons: 1) they are more prone to voice activity detection errors or short non-pause silences within speech (e.g. unvoiced stop closure), and 2) in order to apply the results found here to online endpointing by the system, some amount of time is required to detect the silence and compute necessary features, making endpointing decisions on such very short silences impractical. Once short silences have been excluded, there are 3083 pauses in the corpus, 0.17 per turn.

3.3 Relationship Between Dialogue Features and Silence Distributions

3.3.1 Statistical Analysis

In order to get some insight into the interaction of the various aspects of dialogue and silence characteristics, we investigated a number of features automatically extracted from the dialogue recordings and system logs. Each feature is used to split the set of silences into two subsets. For nominal features, all possible splits of one value vs all the others are tested, while for continuous and ordinal features, we tried a number of thresholds and report the one that yielded the strongest results. In order to avoid extreme cases that split the data into one very large and one very small set, we excluded all splits where either of the two sets had fewer than 1000 silences. All the investigated splits are reported in Appendix, in Table 1 and 2. We compare the two subsets generated by each possible split in terms of two metrics:

- Gap Ratio (GR), defined as the proportion of gaps among all silences of a given set. We report the absolute difference in GR between the two sets, and use chi-square in a 2x2 design (pause vs gap and one subset vs the other) to test for statistical significance at the 0.01 level, using Bonferroni correction to compensate for multiple testings.

- Mean pause duration. The strength of the interaction is shown by the difference in mean pause duration, and we use Mann Whitney's Rank Sum test for statistical significance, again at the 0.01 level, using Bonferroni correction.

We group features into five categories: discourse, semantics, prosody, turn-taking, and speaker characteristics, described in the following sections.

3.3.2 Discourse Structure

Discourse structure is captured by the system's dialogue act immediately preceding the current user turn. In the Let's Go dialogues, 97.9% of system dialogue acts directly preceding user turns are questions[2]. Of these, 13% are open questions (e.g. "What can I do for you?"), 39% are closed questions (e.g. "Where are you leaving from?") and 46% are confirmation requests (e.g. "Leaving from the airport. Is this correct?")[3]. There are many more pauses in user responses to open questions than to the other types (cf Table 1). One explanation is that user answers to open questions tend to be longer (2046 ms on average, to be contrasted with 1268 ms for turns following closed questions and 819 ms for responses to confirmation questions). Conversely, confirmation questions lead to responses with significantly fewer pauses. 78% of such turns contained only one word, single YES and NO answers accounting for 81% of these one-word responses, which obviously do not lend themselves to pauses. Discourse context also has an effect on pause durations, albeit a weak one, with open questions leading to turns with shorter pauses. One possible explanation for this is that pauses after closed and confirmation questions tend to reflect more hesitations and/or confusion on the user's side, whereas responses to open questions also have pauses in the normal flow of speech.

3.3.3 Semantics

Semantic features are based on partial speech recognition results and on their interpretation in the current dialogue context. We use the most recent recognition hypothesis available at the time when the silence starts, parse it using the system's standard parser and grammar, and match the parse against the "expectation agenda" that RavenClaw (Bohus and Rudnicky, 2003) maintains. The expectation level of a partial utterance indicates how well it fits in the current dialogue context. A level of 0 means that the utterance can be interpreted as a direct answer to the last system prompt (e.g. a "PLACE" concept as an answer to "Where are you leaving from?", a "YES" or a "NO" after a confirmation question). Higher levels correspond to utterances that fit in a broader dialogue context (e.g. a place name after the system asks "Leaving from the airport. Is this correct?", or "HELP" in any context). Finally, non-understandings, which do not match any expectation, are given a matching level of $+\infty$.

Expectation level is strongly related to both finality and pause duration. Pauses following partial utterances of expectation level 0 are significantly more likely to be gaps than those matching any higher level. Also, very unexpected partial utterances (and non-understandings) contain shorter pauses than more expected ones. Another indicative feature for finality is the presence of a positive marker (i.e. a word like "YES" or "SURE") in the partial utterance. Utterances that contain such a marker are more likely to be finished than others. In contrast, the effect of negative markers is not significant. This can be explained by the fact that negative responses to confirmation often lead to longer corrective utterances more prone to pauses. Indeed, 91% of complete utterances that contain a positive marker are single-word, against 67% for negative markers.

3.3.4 Prosody

We extracted three types of prosodic features: acoustic energy of the last vowel, pitch of the last voiced region, and duration of the last vowel. Vowel

[2] The remaining 2.1% belong to other cases such as the user barging in right after the system utters a statement.

[3] The high number of confirmations comes from the fact that Let's Go is designed to ask the user to explicitly confirm every concept.

location and duration were estimated by performing phoneme alignment with the speech recognizer. Duration was normalized to account for both vowel and speaker identity. Energy was computed as the log-transformed signal intensity on 10ms frames. Pitch was extracted using the Snack toolkit (Sjolander, 2004), also at 10ms intervals. For both energy and pitch, the slope of the contour was computed by linear regression, and the mean value was normalized by Z-transformation using statistics of the dialogue-so-far. As a consequence, all threshold values for means are expressed in terms of standard deviations from the current speaker's mean value.

Vowel energy, both slope and mean, yielded the highest correlation with silence finality, although it did not rank as high as features from other categories. As expected, vowels immediately preceding gaps tend to have lower and falling intensity, whereas rising intensity makes it more likely that the turn is not finished. On the other hand, extremely high pitch is a strong cue to longer pauses, but only happen in 5.6% of the pauses.

3.3.5 Timing

Timing features, available from the Interaction Manager, provide the strongest cue to finality. The longer the on-going turn has been, the less likely it is that the current silence is a gap. This is true both in terms of time elapsed since the beginning of the utterance and number of pauses observed so far. This latter feature also correlates well with mean pause duration, earlier pauses of a turn tending to be longer than later ones.

3.3.6 Speaker Characteristics

These features correspond to the observed pausal behavior so far in the dialogue. The idea is that different speakers follow different patterns in the way they speak (and pause), and that the system should be able to learn these patterns to anticipate future behavior. Specifically, we look at the mean number of pauses per utterance observed so far, and the mean pause duration observed so far for the current dialogue. Both features correlate reasonably well with silence finality: a higher mean duration indicates that upcoming silences are also less likely to be final, so does a higher mean number of pauses per turn.

3.4 Discussion

What emerges from the analysis above is that features from all aspects of dialogue provide information on silence characteristics. While most previous research has focused on prosody as a cue to detect the end of utterances, timing, discourse, semantic and previously observed silences appear to correlate more strongly with silence finality in our corpus. This can be partly explained by the fact that prosodic features are harder to reliably estimate on noisy data and that prosodic features are in fact correlated to higher levels of dialogue such as discourse and semantics. However, we believe our results make a strong case in favor of a broader approach to turn-taking for conversational agents, making the most of all the features that are readily available to such systems. Indeed, particularly in constrained systems like Let's Go, higher level features like discourse and semantics might be more robust to poor acoustic conditions than prosodic features. Still, our findings on mean pause durations suggest that prosodic features might be best put to use when trying to predict pause duration, or whether a pause will occur or not. The key to more natural and responsive dialogue systems lies in their ability to combine all these features in order to make prompt and robust turn-taking decisions.

4 Evaluation of Threshold Decision Trees

4.1 Offline Evaluation Set-Up

We evaluated the approach introduced in Section 2 on the Let's Go corpus. The set of features was extended to contain a total of 4 discourse features, 6 semantic features, 5 timing/turn-taking features, 43 prosodic features, and 6 speaker characteristic features. All evaluations were performed by 10-fold cross-validation on the corpus. Based on the proposed algorithm, we built a decision tree and computed optimal cluster thresholds for different overall FA rates. We report average latency as a function of the proportion of turns for which any pause was erroneously endpointed, which is closer to real performance than silence FA rate since, once a turn has been endpointed, all subsequent silences are irrelevant.

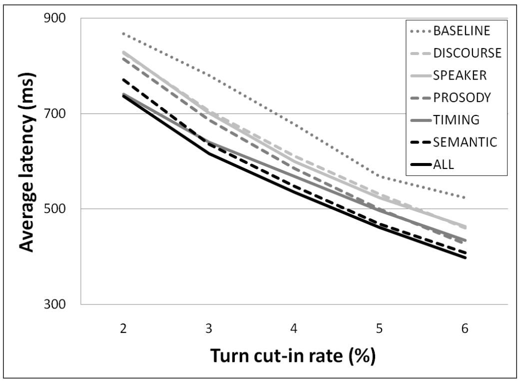

Figure 2: Performance of the proposed approach using different feature sets.

4.2 Performance of Different Feature Sets

First we evaluated each feature set individually. The results are shown in Figure 2. We concentrate on the 2-6% range of turn cut-in rate where any reasonable operational value is likely to lie (the 700 ms threshold of the baseline Let's Go system yields about 4% cut-in rate). All feature sets improve over the baseline. Statistical significance of the result was tested by performing a paired sign test on latencies for the whole dataset, comparing, for each FA rate the proportion of gaps for which the proposed approach gives a shorter threshold than the single-threshold baseline. Latencies produced by the decision tree for all feature sets were all found to be significantly shorter ($p < 0.0001$) than the corresponding baseline threshold.

The best performing feature set is semantics, followed by timing, prosody, speaker, and discourse. The maximum relative latency reductions for each feature set range from 12% to 22%. When using all features, the performance improves by a small but significant amount compared to any single set, up to a maximum latency reduction of 24%. This confirms that the algorithm is able to combine features effectively, and that the features themselves are not completely redundant. However, while removing semantic or timing features from the complete set degrades the performance, this is not the case for discourse, speaker, nor prosodic features. This result, similar to what (Sato et al., 2002) reported in their own experiment, indicates that prosodic features might be redundant with semantic and timing features.

4.3 Live Evaluation

We confirmed the offline evaluation's findings by implementing the proposed approach in Let's Go's Interaction Manager. Since prosodic features were not found to be helpful and since their online extraction is costly and error-prone, we did not include them. At the beginning of each dialogue, the system was randomly set as a baseline version, using a 700 ms fixed threshold, or as an experimental version using the tree learned from the offline corpus. Results show that median latency (which includes both the endpointing threshold and the time to produce the system's response) is significantly shorter in the experimental version (561 ms) than in the baseline (957 ms). Overall, the proposed approach reduced latency by 50% or more in about 48% of the turns. However, global results like these might not reflect the actual improvement in user experience. Indeed, we know from human-human dialogues that relatively long latencies are normal in some circumstances while very short or no latency is expected in others. The proposed algorithm reproduces some of these aspects. For example, after open questions, where more uncertainty and variability is expected, the experimental version is in fact slightly slower (1047 ms vs 993 ms). On the other hand, it is faster after closed question (800 ms vs 965 ms) and particularly after confirmation requests (324 ms vs 965 ms), which are more predictable parts of the dialogue where high responsiveness is both achievable and natural. This latter result indicates that our approach has the potential to improve explicit confirmations, which are often thought to be tedious and irritating to the user.

5 Conclusion

In this paper, we described an algorithm to dynamically set endpointing threshold for each silence. We analyzed the relationship between silence distribution and a wide range of automatically extracted features from discourse, semantics, prosody, timing and speaker characteristics. When all features are used, the proposed method reduced latency by up to 24% for reasonable false alarm rates. Prosodic features did not help threshold optimization once other feature were included. The practicality of the approach and the offline evaluation results were confirmed by implementing the proposed algorithm in the Let's Go system.

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

J. F. Allen and C. R. Perrault. 1980. Analyzing intention in utterances. *Artificial Intelligence*, 15:143–178.

J. F. Allen, G. Ferguson, A. Stent, S. Stoness, M. Swift, L. Galescu, N. Chambers, E. Campana, and G. S. Aist. 2005. Two diverse systems built using generic components for spoken dialogue (recent progress on trips). In *Interactive Demonstration Track, Association of Computational Linguistics Annual Meeting*, Ann Arbor, MI.

D. Bohus and A. Rudnicky. 2003. RavenClaw: Dialog management using hierarchical task decomposition and an expectation agenda. In *Eurospeech03*, Geneva, Switzerland.

D. Bohus, A. Raux, T. Harris, M. Eskenazi, and A. Rudnicky. 2007. Olympus: an open-source framework for conversational spoken language interface research. In *HLT-NAACL 2007 workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technology*, Rochester, NY, USA.

W. L. Chafe, 1992. *Talking Data: Transcription and Coding Methods for Language Research*, chapter Prosodic and Functional Units of Language, pages 33–43. Lawrence Erlbaum.

H.H. Clark. 1996. *Using language*. Cambridge University Press.

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.

C. E. Ford and S. A. Thompson, 1996. *Interaction and Grammar*, chapter Interactional Units in Conversation: Syntactic, Intonational, and Pragmatic Resources for the Management of Turns, pages 134–184. Cambridge University Press.

H. Furo. 2001. *Turn-Taking in English and Japanese. Projectability in Grammar, Intonation, and Semantics.* Routeledge.

B. J. Grosz and C. Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204.

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

H. Koiso, Y. Horiuchi, S. Tutiya, A. Ichikawa, and Y. Den. 1998. An analysis of turn-taking and backchannels based on prosodic and syntactic features in japanese map task dialogs. *Language and Speech*, 41(3-4):295–321.

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.

A. Raux, B. Langner, D. Bohus, A. W. Black, and M. Eskenazi. 2005. Let's Go Public! taking a spoken dialog system to the real world. In *Proc. Interspeech 2005*, Lisbon, Portugal.

A. Raux, D. Bohus, B. Langner, A. W. Black, and M. Eskenazi. 2006. Doing research on a deployed spoken dialogue system: One year of Let's Go! experience. In *Proc. Interspeech 2006*, Pittsburgh, PA, USA.

A. Raux, , and M. Eskenazi. 2007. A multi-layer architecture for semi-synchronous event-driven dialogue management. In *Proc. ASRU 2007*, Kyoto, Japan.

C. Rich and C.L. Sidner. 1998. Collagen: A collaboration manager for software interface agents. *An International Journal: User Modeling and User-Adapted Interaction*, 8(3-4):315–350.

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.

Kare Sjolander. 2004. The snack sound toolkit. http://www.speech.kth.se/snack/.

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.

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.

Category	Feature test	Number of Silences	Gap Ratio	Difference
Timing	Pause start time $\geq$ 3000 ms	1836 / 19260	65% / 87%	-23%
Timing	Pause number $\geq$ 2	3379 / 17717	69% / 88%	-19%
Discourse	Previous question is open	3376 / 17720	70% / 88%	-18%
Semantics	Utterance expectation level $\geq$ 1	10025 / 11071	78% / 92%	-14%
Individual	Mean pause duration $\geq$ 500 ms	1336 / 19760	72% / 86%	-14%
Semantics	Utterance contains a positive marker	4690 / 16406	96% / 82%	13%
Prosody	Mean energy of last vowel $\geq$ 5	1528 / 19568	74% / 86%	-12%
Prosody	Slope of energy on last vowel $\geq$ 0	6922 / 14174	78% / 89%	-10%
Individual	Mean number of pauses per utterance $\geq$ 3	1929 / 19267	76% / 86%	-10%
Semantic	Utterance is a non-understanding	6023/15073	79% / 88%	-9%
Discourse	Previous question is a confirmation	8893 / 12203	90% / 82%	8%
Prosody	Duration of last vowel $\geq$ 1	1319 / 19777	78% / 86%	-8%
Prosody	Mean pitch on last voiced region $\geq$ 5	1136 / 19960	92% / 85%	7%
Prosody	Slope of pitch on last voiced region $\geq$ 0	6617 / 14479	82% / 87%	-4%
Semantics	Utterance contains a negative marker	2667 / 18429	87% / 85%	2%*
Discourse	Previous question is closed	8451 / 12645	86% / 85%	1%*

Table 1: Effect of Dialogue Features on Pause Finality. In columns 3 and 4, the first number is for silences for which the condition in column 2 is true, while the second number is for those silences where the condition is false. * indicates that the results are not statistically significant at the 0.01 level.

Category	Feature test	Number of Pauses	Mean pause Duration (ms)	Difference (ms)
Prosody	Mean pitch on last voiced region $\geq$ 4	172 / 2911	608 / 482	126
Semantics	Utterance Expectation Level $\geq$ 4	2202 / 881	475 / 526	-51
Prosody	Slope of energy on last vowel $\geq$ 1	382 / 2701	446 / 495	-39
Timing	Pause number $\geq$ 2	1031 / 2052	459 / 504	-45
Discourse	Previous question is open	1015 / 2068	460 / 504	-43
Individual	Mean pause duration $\geq$ 500 ms	370 / 2713	455 / 494	-39*
Prosody	Mean energy of last vowel $\geq$ 4.5	404 / 2679	456 / 494	-38*
Semantics	Utterance contains a positive marker	211 / 2872	522 / 487	35*
Discourse	Previous question is closed	1178 / 1905	510 / 477	33*
Timing	Pause start time $\geq$ 3000 ms	650 / 2433	465 / 496	-31*
Semantic	Utterance is a non-understanding	1247 / 1836	472 / 502	-30*
Prosody	Duration of last vowel $\geq$ 0.4	1194 / 1889	507 / 478	29*
Individual	Mean number of pauses per utterance $\geq$ 2	461 / 2622	474 / 492	-19*
Semantics	Utterance contains a negative marker	344 / 2739	504 / 488	16*
Prosody	Slope of pitch on last voiced segment $\geq$ 0	1158 / 1925	482 / 494	-12*
Discourse	Previous question is a confirmation	867 / 2216	496 / 487	9*

Table 2: Effect of Dialogue Features on Pause Duration. In columns 3 and 4, the first number is for silences for which the condition in column 2 is true, while the second number is for those silences where the condition is false. * indicates that the results are not statistically significant at the 0.01 level.

Let (K_n) be a set of n silence clusters, the goal is to set the thresholds (θ_n) that minimize overall mean latency, while yielding a fixed, given number of false alarms E. let us define G_n the number of gaps among the silences of K_n. For each cluster, let us define $E_n(\theta_n)$ the number of false alarms yielded by threshold θ_n in cluster n, and the total latency L_n by:

$$L_n(\theta_n) = G_n \times \theta_n \tag{3}$$

Assuming pause durations follow an exponential distribution, as shown in Section 3, the following relation holds between L_n and E_n:

$$e^{\frac{L_n(\theta_n)}{\mu_n}} = \beta_n \times E_n(\theta_n) \tag{4}$$

where μ_K and β_K are cluster-specific coefficients estimated by linear regression in the log domain. If we take the log of both sides, we obtain:

$$L_n(\theta_n) = \mu_n \times log(\beta_n \times E_n(\theta_n)) \tag{5}$$

Theorem 1. *If (θ_n) is a set of thresholds that minimizes $\sum_n L_n$ such that $\sum_n E_n(\theta_n) = E$, then $\exists A s.t. \forall n, \frac{dL_n}{dE_n}(\theta_n) = A$*

Informal proof. The proof can be done by contradiction. Let us assume (θ_n) is a set of thresholds that minimizes $\sum_n L_n$, and $\exists (p,q) s.t. \frac{dL_p}{dE_p}(\theta_p) > \frac{dL_q}{dE_q}(\theta_q)$. Then, there exists small neighborhoods of θ_p and θ_q where $L_p(E_p)$ and $L_q(E_q)$ can be approximated by their tangents. Since their slopes differ, it is possible to find a small ϵ such that the decrease in FA yielded by $\theta_p + \epsilon$ is exactly compensated by the increase yielded by $\theta_q - \epsilon$, but the reduction in latency in K_q is bigger than the increase in K_p, which contradicts the fact that (θ_n) minimizes L. □

From Theorem 1, we get $\exists A s.t. \forall n \frac{dL_n}{dE_n} = A$. Thus, by deriving Equation 5, $\frac{\mu_n}{E_n} = A$ which gives $E_n = \frac{\mu_n}{A}$. Given that $\sum E_n = E$, $\frac{\sum \mu_n}{A} = E$. Hence, $A = \frac{\sum \mu_n}{E}$. From 5, we can infer the values of $L_n(\theta_n)$ and, using 3, the optimal threshold θ_n for each cluster:

$$\theta_n = \frac{\mu_n \times log(\beta_n \times \frac{E \times \mu_n}{\sum \mu_n})}{G_n} \tag{6}$$

where the values of μ_n and β_n can be estimated by linear regression from the data based on 5.

Figure 3: **Derivation of the formula for optimal thresholds**