It's About Time: Turn-Entry Timing For Situated Human-Robot Dialogue

Felix Gervits, Ravenna Thielstrom, Antonio Roque, Matthias Scheutz
Human-Robot Interaction Laboratory
Tufts University, Medford MA 02155
{felix.gervits, ravenna.thielstrom,
antonio.roque, matthias.scheutz}@tufts.edu

Abstract

Turn-entry timing is an important requirement for conversation, and one that few spoken dialogue systems consider. In this paper we introduce a computational framework, based on work from psycholinguistics, which is aimed at achieving proper turn-entry timing for situated agents. Our approach involves incremental processing and lexical prediction of the turn in progress, which allows a situated dialogue agent to start its turn and initiate actions earlier than would otherwise be possible. We evaluate the framework by integrating it within a cognitive robotic architecture and testing performance on a corpus of situated, task-oriented human-robot directives. We demonstrate that: 1) the system is superior to a non-incremental system in terms of faster responses, reduced gap between turns, and the ability to perform actions early, 2) the system can time its turn to come in immediately at a turn transition, or earlier to produce several types of overlap, and 3) the system is robust to various forms of disfluency in the input. Overall, this domain-independent framework can be integrated into existing dialogue systems to improve responsiveness, and is another step toward more natural and fluid turn-taking behavior.

1 Introduction

Behavioral evidence shows that humans are able to exchange turns extremely quickly in conversation – within a few hundred milliseconds on average (Levinson and Torreira, 2015). This is a universal human characteristic, though the nature of the timings varies slightly across languages (Stivers et al., 2009). There is some debate about exactly how humans achieve this performance, but evidence from psycholinguistic studies suggests that it is likely done by processing ongoing utterances incrementally and making lexicosyntactic predictions about the turn in progress (de Ruiter et al., 2006; Magyari and de Ruiter, 2012). This allows a listener to plan what to say and to anticipate the end of the speaker's turn accurately so that turn-transitions are seamless, and gaps between turns are minimized. It also allows for the production of speech overlap, to produce conversational behaviors such as backchanneling and repair. Such human behaviors are desirable for spoken dialogue systems (SDSs) where *naturalness* is a priority (Edlund et al., 2008).

SDS research has produced an impressive body of work on turn-taking (e.g. Bohus and Horvitz (2011); Kronlid (2006); Raux and Eskenazi (2009, 2012); Skantze and Schlangen (2009); Zhao et al. (2015)), and some early work on overlap and completions (Baumann and Schlangen, 2011; DeVault et al., 2011b; Gervits and Scheutz, 2018a). However, relatively little focus has been placed on using turn-taking capabilities for **responsive turn-entry timing**, especially for situated agents. One exception is the approach by Baumann and Schlangen (2011) which involves estimating word duration to produce collaborative completions.

We build on this prior work through the development of a framework for achieving responsive turn-entry timing, as well as a full set of adaptive human-like overlap and completion behaviors. Our approach involves using utterance-level predictions from partial input and information from a world modeler to determine *when* to enter the turn (including producing overlap at any of the entry points shown in Figure 1), and whether to initiate actions early. Such capabilities are particularly important for situated dialogue agents, as responses, and especially actions, often involve lengthy processing delays, which can be mitigated by preparing or initiating them during an ongoing turn. Section 2 describes how this framework builds on existing research, including our novel *Turn-Entry Manager (TEM)* described in Section 2.4. In Section 3 we describe implementation details related to integrating

Proceedings of the SIGdial 2020 Conference, pages 86–96
1st virtual meeting, 01-03 July 2020.

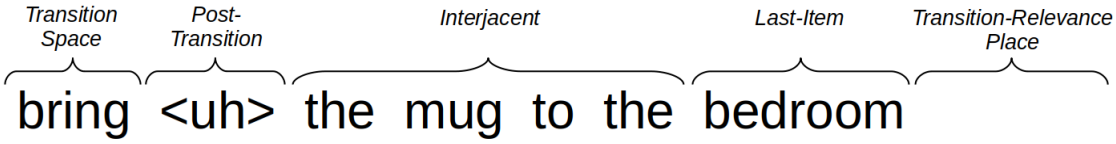

Figure 1: Entry points in a sample utterance based on (Jefferson, 2004). A fluid turn exchange starts at the transition point between turns (transition-relevance place or TRP), whereas earlier entry points indicate various types of overlap.

the framework in a cognitive robotic architecture. Then in Section 4 we evaluate our implementation on a corpus of situated human-robot dialogue utterances. Finally, we close with a discussion of the contributions and directions for future work.

2 A Framework for Turn-Entry Timing

Here we discuss the framework needed to manage turn-entry timing for situated dialogue agents, and the related work that the framework builds on.

2.1 Incremental Processing with Prediction

Obtaining an early understanding of the meaning of an utterance allows for faster feedback, supportive overlap, and faster actions. To achieve this, the SDS needs prediction, which is enabled by incremental processing.

Extensive prior work has supported fast and effective incremental processing with prediction (Paetzel et al., 2015; Skantze, 2017). For example, Schlangen and Skantze (2011) developed the Incremental Unit (IU) framework which supports incrementalaity with prediction, revision, and management of alternative hypotheses. This and other related approaches (e.g., Heintze et al. (2010); Skantze and Schlangen (2009)) involve interpreting meaning from each partial input rather than trying to predict the complete utterance. Other work has focused on predicting a full utterance (or semantic frame) from partial input using a maximum entropy classification approach (DeVault et al., 2011a; Sagae et al., 2009). These approaches attempt to find the point of maximal understanding at which a response can be initiated, and have been demonstrated to support the production of collaborative completions (DeVault et al., 2009). While these approaches use lexical cues in the input to generate predictions, other cues can also be used for situated interaction, including gesture and gaze (Kennington et al., 2013), and acoustic features (Maier et al., 2017; Ward et al., 2010). Our approach builds on this prior work in incremental processing, using it as a component in our overall framework.

2.2 Speech Overlap Production

Speech overlap has been shown to serve many useful functions in conversation, including responsiveness and repair (Jefferson, 2004), but historically the SDS community has viewed it as an intrusive property and used the term *barge-in*. Some SDS work exists on the topic of intentional overlap production, including a body of work aimed at producing appropriate backchannel feedback (Lala et al., 2017; Truong et al., 2010). Another example comes from DeVault et al. (2011b), who designed a prototype system using predictive capabilities to perform collaborative completions and backchannel feedback. This work provides a necessary first step, but it only covers a subset of the different types of overlap possible, leaving out those that occur at the transition space, post-transition, and interjacent positions (Drew, 2009). Moreover, this work does not deal with situated dialogue or issues of timing in speech synthesis. Situated dialogue presents additional opportunities for overlap which have yet to be explored, such as coming in mid-utterance to clarify an un-actionable command. Finally, if a system will be producing overlap, then mechanisms to manage and recover from overlap are also needed. A preliminary approach was demonstrated in Gervits and Scheutz (2018a) based on a corpus analysis in Gervits and Scheutz (2018b), but otherwise there is limited work in this area.

2.3 Preemptive Action Execution

For dialogue in real-world or virtual environments with humans, situated agents can use predictive language capabilities to perform actions early or at least begin some processing during an ongoing utterance. This has been explored by Hough and Schlangen (2016), who developed a real-time incremental grounding framework that supports "fluid" interaction using the IU framework. While the system performance is impressive, this work only

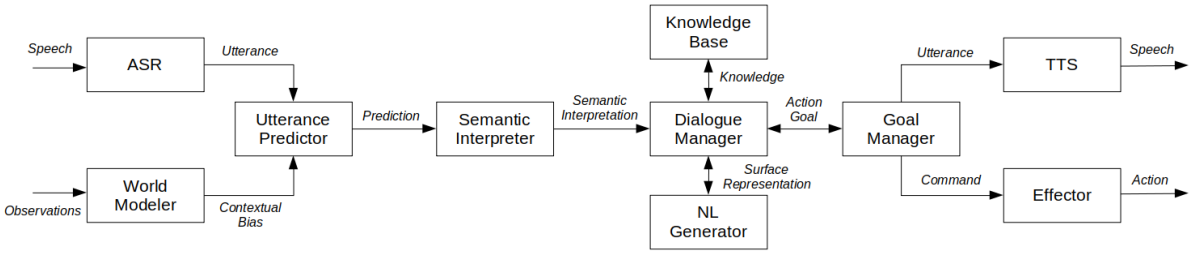

Figure 2: Component diagram of the turn-taking framework as implemented in the DIARC cognitive robotic architecture. Boxes represent architectural components and arrows represent the flow of information.

focused on action and did not involve timing dialogue responses. Moreover, the only actions considered were those that the robot *could* carry out. In human-robot interaction, a robot might be instructed to perform an action that it does not know how to do or that it cannot currently do. In order to respond early, the robot will need to simulate the action to determine if it will be successful. This simulation may involve a cognitive architecture carrying out an actual "mental" simulation of the action, or simply checking if the preconditions for the action are met. This is the type of processing that can be done during an ongoing utterance.

2.4 Turn-Entry Manager

Given the multitude of points for which a system may need to enter a turn (as shown in Figure 1), some process is needed to manage turn entry. We propose a *Turn-Entry Manager (TEM)* component that carries out these tasks. The TEM works as follows: it receives full utterance predictions from partial automatic speech recognition (ASR) results and determines *when* to initiate a follow-up utterance and action based on the confidence in the prediction as well as task context and agent goals. The most intuitive location for the TEM is in the Dialogue Manager (DM), as it uses information only available further along in the pipeline. The TEM will store the following information about its *prediction* of an ongoing utterance: semantics and text of the utterance, remaining words and expected duration of the utterance, response and action associated with the utterance, confidence in the prediction, cost of the action, entry time for the *transition-relevance place (TRP)* (Sacks et al., 1974) and several overlap positions, and latency of various components. Most of this information is updated with each increment received by the parser. Using this information, the TEM determines the timing of when to take a turn so as to achieve fluid turn transitions. Depending on its policy, it can also come in early to produce various kinds of overlap. While most SDSs have some process that manages turn entry, none, to our knowledge, possess the capabilities described here.

3 Implementation in a Cognitive Robotic Architecture

To effectively interact in a situated environment, robots need to react to and affect the environment, as well as to reason about the task and user; this requires a *cognitive robotic architecture*. We integrated our turn-entry timing framework into the DIARC architecture (Scheutz et al., 2019). We used DIARC due to its emphasis on situated robot dialogue (highlighted in Gervits et al. (2020)), although in principle our framework is general enough to be used with any architecture of its type. Below we discuss each of the key components in our architectural configuration.

3.1 Situated Natural Language Processing

Our work is mostly performed in the language-processing components of DIARC, shown in Figure 2. First, speech is received by the *ASR component*, which converts it into text. For ASR, we use the Sphinx4 recognizer, modified to output incremental results. A text interface can also be used to simulate incremental speech input. The word-by-word results are sent to the *Utterance Prediction component* (described further in Section 3.2), which generates a prediction using a bigram language model and sends the prediction to the *Semantic Interpreter component*. We use a rule-based parser that performs syntactic and semantic parsing, and converts the text of an utterance into a logical predicate form. The predicate is then sent to the *DM component*, which is a goal-based dialogue manager that uses a Prolog knowledge base for storing declarative knowledge, and for performing logical inference over that knowledge to engage

in mixed-initiative dialogue. The DM implements a version of the *TEM* described in Sec. 2.4. The DM also interacts with the *Goal Manager (GM)* component, which contains a database of actions that the robot can perform (including dialogue actions) and facilitates action execution. Actions in DIARC are defined by their pre-, post-, and operating conditions. The post-conditions of an action are goal predicates that describe a state of the world that an agent is trying to achieve, e.g., *did(self,moveTo(self,bookshelf))* for an action goal, and *did(self,spoke(okay))* for a dialogue goal. For dialogue actions, the DM obtains the surface form of the response utterance from the *NL Generator* component, and then submits the goal associated with the action to the GM. The GM then calls the *text-to-speech (TTS) component* (which is a wrapper for MaryTTs) to produce speech output. Physical actions are handled in a similar way, except that the *Effector* component corresponding to the action handles the execution.

3.2 Utterance Prediction with Contextual Bias

For utterance prediction, we implemented a bigram language model trained on the frequency distribution of bigrams in the HuRIC corpus (see Sec. 4). More sophisticated prediction algorithms are possible, but given the importance of speed, we chose a simple and effective approach. The prediction is computed as follows: given an initial word as input, the model generates a set of complete utterances based on the most probable follow-up words along with their associated probability. A cumulative probability threshold is used to determine when a prediction is sufficient, at which point the full utterance prediction with the highest probability is sent to the parser. If the threshold is not reached, then the algorithm waits for the next input word and repeats the same process.

A contextual bias is included to represent the influence of the situated environment as observed by the robot and included in a world model. This context influences the utterance predictor by increasing the probability of specific bigrams by a set amount, causing the model to favor those words. In our preliminary implementation, the context is hand-tuned for each utterance in the corpus[1], but situated agents would be able to determine this context by perceiving the environment, through task knowledge, or through the dialogue history.

3.3 An Algorithm for Turn-Entry Management

The *TEM* algorithm works as follows (see Algorithm 1): First, an utterance is received incrementally from the ASR component. In parallel, each word is sent to the Utterance Predictor component, where the bigram language model described in Sec. 3.2 generates predictions based on the frequency distribution of the training corpus and any contextual bias (Algorithm 1, line 3).

If a prediction clears a set threshold, then it is sent to the DM component. The DM first computes a score for the prediction based on the cost of the associated action and the confidence in the probability (line 5). If the score is above a set threshold then it continues. The score threshold can be set to minimize early execution for costly actions (e.g., actions that can cause delay to repair, such as movement) in the case of a wrong prediction. If the score threshold is exceeded, the DM next computes the TRP and last-item entry points based on the utterance start time and expected duration, accounting for the known TTS delay, which was about 40 ms in our system (lines 7-8).

Next, the preconditions for the action associated with the predicted utterance are checked (line 9). If the action exists and the preconditions are met, then a response is set (but not yet generated; line 13); otherwise, a failure explanation is generated and immediately produced (line 11). In the case that the preconditions are met, the DM sets the overlap type (TRP, last-item, or collaborative completion) based on a simple policy (line 14).[2] The action corresponding to the prediction is then performed (line 15). Finally, once the overlap type is set, a separate thread running every 1 ms waits until the current system time reaches the designated entry point and then produces the associated response (lines 22-26).

4 Evaluation

To evaluate the efficacy of our framework, we used a corpus of directives to a household robot from the S4R dataset of the HuRIC corpus (Bastianelli et al., 2014). The dataset consisted of 96 imperative utterances from a task in which people were

[1] For example, the context for the utterance "Grab the bottle on the kitchen table" may be 'kitchen' (describing the environment) and 'bottle' (describing an item in the environment)

[2] The current policy is that if the response utterance is an acknowledgment then the system will produce a last-item overlap, otherwise it will aim for the TRP with no overlap.

Algorithm 1 Turn-Entry Manager Algorithm

```
1:  procedure TEM(Utterance u)
2:      for all word ∈ u do
3:          Prediction p = generatePrediction(word)
4:          if p.probability > probThreshold then
5:              p.score = p.cost * p.confidence
6:              if p.score >= scoreThreshold then
7:                  p.TRP_entry = p.startTime + p.duration - TTS_delay
8:                  p.LI_entry = p.startTime + p.duration - TTS_delay - p.lastWord.duration
9:                  actionStatus = simulateAction(p.action)
10:                 if actionStatus == fail then
11:                     generateResponse(failure)
12:                 else
13:                     p.response = setResponse()
14:                     p.setOverlapType(p.response)
15:                     performAction(p.action)
16:                 end if
17:             end if
18:         end if
19:     end for
20: end procedure
21:
22: procedure WAITTOSPEAK(Prediction p)
23:     if currentTime >= p.TRP_entry then
24:         generateResponse(p.response)
25:     end if
26: end procedure
```

asked to give commands to a physical robot operating in a household environment. The language was unscripted and had few constraints, though people were told about the robot's capabilities and the locations and objects that it could recognize. While the evaluation corpus contains only directives (no dialogue), it includes the kinds of utterances commonly seen in situated task-based dialogues, to which a robot would need to promptly respond (and potentially initiate early), and serves as a useful benchmark to test our framework.

The central aim of the evaluation is to show how a situated agent given these instructions can make predictions and respond at the TRP compared to a non-incremental baseline system. We also seek to demonstrate the potential for overlap production and preemptive action execution. In addition to the standard directives in the corpus, we also evaluate several variants of them which contain disfluency. It is important that SDSs are resilient to disfluency, as it is common in team communication channels (particularly in remote communication) and has been implicated in effective team performance (Gervits et al., 2016a,b). Including disfluent utterances in the evaluation was done to show that the algorithm can handle variations in the input and still produce timely responses. Table 1 lists the utterance subsets that were constructed from the original corpus data. These include: 1) the original utterance, 2) utterance-initial non-lexical filler, 3) non-lexical filler after the first word, 4) 200 ms pause before the final word, and 5) repetition self-repair of the first word.

Subset	Example Utterance
1	go to the kitchen
2	<um>go to the kitchen
3	go <uh>to the kitchen
4	go to the <200 ms pause>kitchen
5	go- go to the kitchen

Table 1: Utterance subsets used in the evaluation

4.1 Approach

First, the text strings from the HuRIC corpus were extracted, along with the frequency distribution of bigrams. Parse rules (linking the text string to a semantic form) and actions (specifying the preconditions and effect) were defined for each utterance, and we generated the 5 subsets (see Table 1) for each utterance in the corpus.

While the system is capable of processing speech incrementally, we used incremental text input for the evaluation in order to abstract away some of the ASR noise (e.g., latency, errors, etc.)[3]. To simulate the timing associated with real speech, we added a delay before each word corresponding to 180 ms x the number of syllables in the word. This decision is based on the upper bound of the estimated duration of a syllable from Wilson and Wilson (2005), and is roughly in line with data from the *Switchboard* corpus, in which the mean syllable duration was 200 ms (SD: 103) (Greenberg, 1999). To handle the disfluency in Subsets 2-5, we used a simple keyword-spotting approach to detect fillers and pauses in the input, like most ASRs can do. These fillers were excluded from the recognizer result, but importantly their duration was added to the timing. We assume that fillers such as *uh* and *um* are one syllable in length, and so have a duration of 180 ms. While not all types of disfluencies are handled with these subsets, we leave prolongations and more complex self-repairs for future work.

The turn-taking policy used in the evaluation is that the robot will attempt to come in at the TRP if it made an early prediction and the action status of the prediction was successful. If the action status was a failure then the robot will overlap with the failure explanation immediately. The *score* threshold was set to 0 to maximize data collection. Other policies are, however, possible such as never overlapping, or using a higher score threshold to minimize wrong predictions for costly actions.

4.1.1 Measures and Hypotheses

Our primary measure was the Floor Transfer Offset (FTO), a term introduced by de Ruiter et al. (2006). FTO is defined as the time difference in ms between the start of a turn and the end of the previous turn. Positive values indicate gaps whereas negative values indicate overlap. We also computed the accuracy of the prediction model, the timing of when a prediction was made, and the point at which an action was initiated.

Overall, we expected the algorithm to perform well for the majority of examples in Subset 1, leading to smaller FTOs compared to a non-incremental system. This gives us:

***H1**: Incremental utterance prediction would lead to smaller FTOs and earlier actions than non-incremental processing without online prediction.*

The non-incremental baseline system we used is a similar DIARC configuration, with the main difference being that input is non-incremental and the Utterance Predictor component is bypassed. We ran utterances from Subset 1 in which a correct prediction was made through this non-incremental configuration to compare performance. Next, we expected that the timing in the system would work out such that it can time its turn to come in at or near the TRP for actionable predictions, and much earlier for un-actionable ones. Thus we have:

***H2**: Incremental utterance prediction would enable the system to 1) hit the TRP entry point for responses to actionable predictions, 2) initiate those actions early, 3) and produce interjacent (mid-speech) overlap for un-actionable predictions.*

If the system makes an early prediction, subsequent processing takes minimal time, so it should be able to hit the TRP for all but very late predictions. It would also be able to initiate the action shortly after the DM receives the prediction. For early predictions that are not actionable, it should produce an interjacent overlap well before the utterance is finished. Finally, we expect performance on Subsets 2-5 to be dependent on whether a prediction was made before or after the disfluency was detected. This is because the TRP entry point is computed from the expected duration of the predicted utterance, and this duration may be incorrect if the prediction did not incorporate the additional timing of the disfluency. This leads to:

***H3:** The approach would be robust to disfluency in the input, but only if the disfluency was detected before a prediction was made.*

Given *H3*, we expect the FTO for Subsets 1 and 2 to be close to 0 for correct predictions, since these involve either no filler or an utterance-initial filler (which will always be detected before a prediction is made). Subset 4 will likely have a negative FTO, as predictions will usually be made before the final word, and so the 200 ms pause will not be added to

[3] In particular, we experienced significant delays with incremental speech input. This is likely due to our Sphinx4 configuration, as others have reported much faster performance with the same ASR (Baumann and Schlangen, 2012).

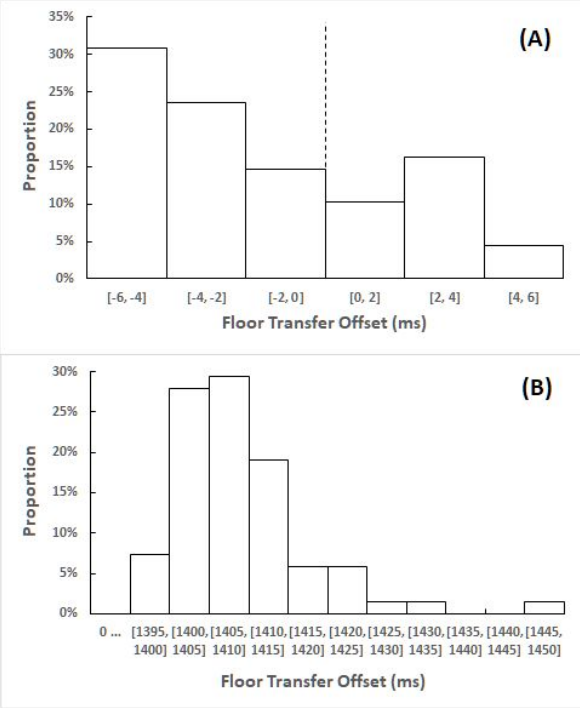

Figure 3: Histograms showing the floor transfer offset for A) predictive system and B) non-incremental (baseline) system for cases in which a correct early prediction was made. N = 68.

the utterance duration, leading to earlier turn entry.

4.2 Results

Below we present the results of the evaluation described in Sec. 4. In general, prediction accuracy of our bigram model was 70.8% with 340 of 480 test utterances predicted correctly. On average, a prediction was made 50.8 $\pm$ 17.7% of the way into an utterance, duration-wise.

4.2.1 H1: Incremental vs Non-Incremental Processing

H1 dealt with the difference in FTOs between our framework implementation and a non-incremental baseline configuration of the same architecture. We compared the correctly-predicted utterances from Subset 1 (N=68) and the same utterances tested on the baseline system. A Welch's independent-samples t-test showed a significant difference between FTOs for the incremental prediction cases (M = -1.1 $\pm$ 3.2 ms) compared to the baseline cases (M = 1409.5 $\pm$ 8.6 ms), $t(85) = 1259.2$, $p < .001$ (see Figure 3). These results support *H1* in that a system running our framework was able to take a turn significantly earlier than a non-incremental one that did not use the framework.

4.2.2 H2: Timing Turn-Entry

H2 stated that our framework implementation would allow the system to reliably come in at the TRP for actionable predictions, and produce early failure explanations in the form of interjacent overlap for un-actionable (i.e., incorrect) predictions. For Subset 1 (fluent) utterances, we found a mean FTO of -1.1 $\pm$ 3.2 ms. Since an FTO of 0 means a seamless transition, these results support *H2* in that the system was able to time its turn to hit the TRP very accurately for actionable predictions. For those predictions that were un-actionable in Subset 1, the system produced a failure explanation with a mean FTO of -683.2 $\pm$ 713.7 ms (see Figure 4). The earliest FTO was -2780 ms and the latest was -8 ms. These results provide further support for *H2* in that the system was able to provide early failure explanations (i.e., interjacent overlap) when a predicted action could not be performed. See Table 2 for an overview of the results.

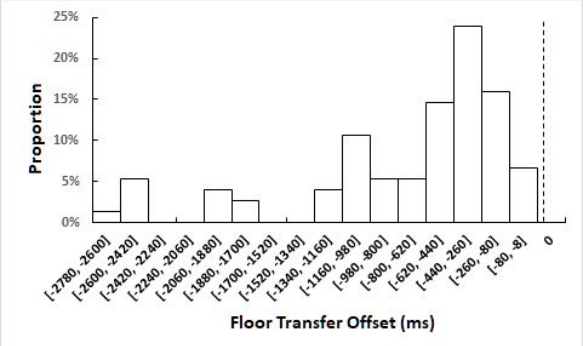

Figure 4: Floor Transfer Offset for cases in which the predicted utterance could not be performed. The system produced an interjacent overlap at the earliest opportunity. N = 15

To demonstrate preemptive action execution, we calculated (for Subset 1) the earliest point at which an action can be initiated. This is the point at which a prediction was sent to the DM and the preconditions for the corresponding action were checked. The difference between the end of the utterance and this point was 635 $\pm$ 197 ms, meaning that on average, an action could be initiated 635 ms before the end of an utterance.

As a supplementary analysis and to evaluate performance with varying syllable duration, we tested 10 random utterances from Subset 1 in which each syllable in the input was assigned a random duration between 100 and 200 ms (following Greenberg (1999)). The average syllable count for these utterances was 5.7 $\pm$ 1.6 and the average FTO was -16.5

	TRP Entry				Interjacent Entry	
	N	FTO (early prediction)	N	FTO (no prediction)	N	FTO
All	340	-55.7 ± 88.0	65	157.9 ± 47.5	75	-709.3 ± 714.2
Subset 1	68	-1.1 ± 3.2	13	164.8 ± 21.3	15	-683.2 ± 713.7
Subset 2	68	-5.5 ± 31.8	13	148.3 ± 19.4	15	-710.9 ± 703.5
Subset 3	68	-40.7 ± 75.9	13	147.8 ± 15.4	15	-746.4 ± 717.6
Subset 4	68	-191.2 ± 46.1	13	149.9 ± 13.9	15	-687.6 ± 780.1
Subset 5	68	-41.0 ± 75.6	13	176.0 ± 101.4	15	-630.6 ± 661.8

Table 2: Table of evaluation results. Mean values for Floor-Transfer Offset (FTO) are displayed for all evaluation cases (N = 480). For a given case, either an early prediction was made, or no prediction was made. If the prediction was correct and actionable, then a TRP entry was selected and an acknowledgment was produced. If the prediction was un-actionable (i.e., incorrect), then an interjacent overlap was selected and a failure explanation was produced.

± 87.9 ms, with a range of -155 to 152 ms. The difference between these results and the original set was that the predicted duration could now be wrong, and this was reflected in the slightly early entry times. Still, the mean FTO was close to 0, suggesting that the model still performs well with variable input.

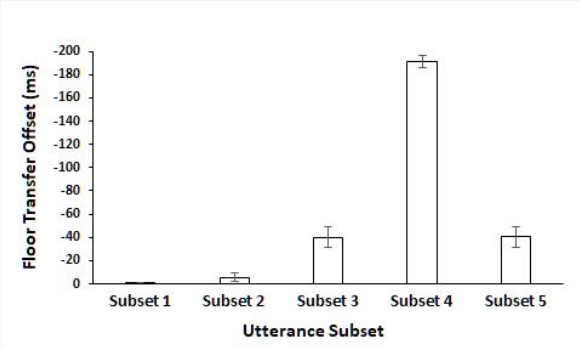

Figure 5: Floor Transfer Offset for correct predictions in each utterance subset. N = 340

4.2.3 H3: Robustness to Disfluency

To evaluate *H3*, which involved the robustness of the algorithm to disfluency in the input, we analyzed all of the disfluency cases in which a correct prediction was made (Subsets 2-5; N = 272). As expected, a key factor in correct timing here had to do with whether the prediction was made before or after the filler. This was confirmed with an independent-samples t-test, which found a significant difference between FTOs for predictions made after the filler (M = -2.4 ± 0.12 ms) compared to those made before the filler (M = -188.1 ± 0.17 ms), $t(127) = 44.6$, $p < .001$. Predictions made before the filler were most common in Subset 4 (making up 69% of the examples) and predictions made after the filler were made up entirely of Subsets 2, 3, and 5. In Figure 5, we show the mean FTO for each of the utterance subsets.

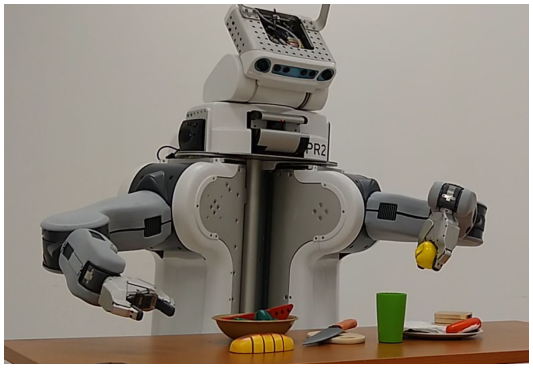

Figure 6: Robot performing a situated interactive task involving dialogue.

4.3 Demonstration

To supplement the evaluation and show a real-world use-case, we ran the framework on a PR2 robot using real speech input (see Figure 6). A video of the interaction is available at https://vimeo.com/410675260. This video compares our baseline (non-incremental) system to the system running our turn-entry timing framework, and demonstrates that a robot can reliably make predictions about ongoing utterances using speech input, and that it can initiate actions and responses early.

5 Discussion

5.1 Contributions

Overall, we found support for *H1*, *H2*, and partial support for *H3*.

For *H1*, we demonstrated that our system was able to take a turn significantly faster than a non-incremental version of the same architecture. This

is not surprising, as the advantages of incremental processing are well known (Baumann et al., 2017). However, the comparison quantifies the amount of time that our approach saves.

H2 was also supported in that the system was able to hit the TRP very accurately for correct and actionable predictions (see Figure 3 A). Moreover, those actions were initiated on average 635 ms before the TRP, providing further support for *H2*. For un-actionable predictions, interjacent overlap was produced on average 709 ms before the TRP, suggesting very early turn entry (see Figure 4).

Finally, *H3* was partially supported in that fillers that were processed before a prediction (i.e., utterance-initial fillers) had their duration added to the overall utterance duration, but fillers towards the end of an utterance (after the prediction) were not detected in time. In these latter cases, the system came in earlier than expected (40 - 191 ms early), which would be a last-item overlap, and would likely not require repair (see Figure 5).

Overall, our domain-independent framework can be integrated into various SDSs in order to support responsive dialogue behavior and early actions, as well as to enable certain kinds of overlap that would not be achievable in other approaches.

5.2 Limitations and Future Work

One limitation is that the evaluation involved text rather than real speech and only considered simple directives. More work is clearly needed to evaluate the accuracy of the proposed approach with respect to variable speech input. Nevertheless, state-of-the-art ASRs can display very low recognition latency (e.g., Baumann et al. (2009)), suggesting that this would not significantly change our results.

Another limitation is that a fixed syllable duration was used to estimate timing, which was the same duration used in the input text. Since syllable length is a parameter in the model, this can be adjusted as needed to better estimate spoken syllable length. We have shown in a supplementary analysis on 10 utterances that the approach works reasonably well with variable syllable length. Future work will test other methods of estimating utterance length, including the clever duration modeling technique used in Baumann and Schlangen (2011) involving the ASR and TTS modules. The current results can be thought of as a best case scenario, and we expect that with more accurate duration estimates of real speech, system performance will approach this upper bound.

Recovering from incorrect predictions is an important area for future work. Currently, when the system makes a prediction it cannot change it, even if new input comes in that contradicts the prediction (this is because the timing is very tight). In future work, it should be possible for the *TEM* to be updated if the prediction changes. This will support the handling of utterances such as those in Subset 4 which were characterized by late pauses.

Finally, the prediction model itself can be improved, perhaps through the use of a neural approach (Maier et al., 2017) or one that incorporates syntactic or prosodic features (Ward et al., 2010). Though we focus on lexico-syntactic cues for prediction, future work could leverage recent findings suggesting that prosody is more important to end-of-turn projection than previously thought (Barthel et al., 2016; Bögels and Torreira, 2015).

6 Conclusion

We have introduced a framework for turn-entry timing in human-robot dialogue which enables a situated agent to make incremental predictions about an ongoing utterance and time its turn to hit a variety of entry points. We implemented the framework in a robotic architecture and evaluated it on a corpus of human-robot directives from a situated interactive task. The system integrating our framework is significantly faster than a non-incremental system, and can produce fluid responses and various types of overlap, as well as execute actions preemptively. Moreover, the approach is robust to several forms of disfluency in the input. This framework offers a number of benefits for situated dialogue agents, including better responsiveness, the ability to produce various types of overlap (interjacent, last-item, backchannels, and collaborative completions), and preemptive action execution. These interactive capabilities are a step toward more natural and flexible turn-taking for situated dialogue agents.

Acknowledgments

This work was partly funded by a NASA Space Technology Research Fellowship under award 80NSSC17K018 and AFOSR Grant FA9550-18-1-0465. We are especially grateful to Brad Oosterveld, Evan Krause, and Zach Munro for their technical assistance.

References

Mathias Barthel, Sebastian Sauppe, Stephen C Levinson, and Antje S Meyer. 2016. The timing of utterance planning in task-oriented dialogue: Evidence from a novel list-completion paradigm. *Frontiers in Psychology*, 7:1858.

Emanuele Bastianelli, Giuseppe Castellucci, Danilo Croce, Luca Iocchi, Roberto Basili, and Daniele Nardi. 2014. Huric: a human robot interaction corpus. In *Proceedings of LREC 2014*, pages 4519–4526.

Timo Baumann, Michaela Atterer, and David Schlangen. 2009. Assessing and improving the performance of speech recognition for incremental systems. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 380–388.

Timo Baumann, Casey Kennington, Julian Hough, and David Schlangen. 2017. Recognising conversational speech: What an incremental asr should do for a dialogue system and how to get there. In *Dialogues with Social Robots*, pages 421–432. Springer.

Timo Baumann and David Schlangen. 2011. Predicting the micro-timing of user input for an incremental spoken dialogue system that completes a user's ongoing turn. In *Proceedings of the SIGDIAL 2011 Conference*, pages 120–129. Association for Computational Linguistics.

Timo Baumann and David Schlangen. 2012. Inpro_iss: A component for just-in-time incremental speech synthesis. In *Proceedings of the ACL 2012 System Demonstrations*, pages 103–108.

Sara Bögels and Francisco Torreira. 2015. Listeners use intonational phrase boundaries to project turn ends in spoken interaction. *Journal of Phonetics*, 52:46–57.

Dan Bohus and Eric Horvitz. 2011. Decisions about turns in multiparty conversation: from perception to action. In *Proceedings of the 13th international conference on multimodal interfaces*, pages 153–160.

David DeVault, Kenji Sagae, and David Traum. 2009. Can i finish?: learning when to respond to incremental interpretation results in interactive dialogue. In *Proceedings of the SIGDIAL 2009 Conference: The 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 11–20. Association for Computational Linguistics.

David DeVault, Kenji Sagae, and David Traum. 2011a. Detecting the status of a predictive incremental speech understanding model for real-time decision-making in a spoken dialogue system. In *Twelfth Annual Conference of the International Speech Communication Association*.

David DeVault, Kenji Sagae, and David Traum. 2011b. Incremental interpretation and prediction of utterance meaning for interactive dialogue. *Dialogue & Discourse*, 2(1):143–170.

Paul Drew. 2009. Quit talking while I'm interrupting: a comparison between positions of overlap onset in conversation. In *Talk in Interaction: Comparative Dimensions*, pages 70–93.

Jens Edlund, Joakim Gustafson, Mattias Heldner, and Anna Hjalmarsson. 2008. Towards human-like spoken dialogue systems. *Speech communication*, 50(8-9):630–645.

Felix Gervits, Kathleen Eberhard, and Matthias Scheutz. 2016a. Disfluent but effective? a quantitative study of disfluencies and conversational moves in team discourse. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3359–3369.

Felix Gervits, Kathleen Eberhard, and Matthias Scheutz. 2016b. Team communication as a collaborative process. *Frontiers in Robotics and AI*, 3:62.

Felix Gervits and Matthias Scheutz. 2018a. Pardon the interruption: Managing turn-taking through overlap resolution in embodied artificial agents. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 99–109.

Felix Gervits and Matthias Scheutz. 2018b. Towards a conversation-analytic taxonomy of speech overlap. In *Proceedings of the 11th edition of the Language Resources and Evaluation Conference (LREC)*.

Felix Gervits, Dean Thurston, Ravenna Thielstrom, Terry Fong, Quinn Pham, and Matthias Scheutz. 2020. Toward genuine robot teammates: Improving human-robot team performance using robot shared mental models. In *19th International Conference on Autonomous Agents and Multi-Agent Systems*.

Steven Greenberg. 1999. Speaking in shorthand–a syllable-centric perspective for understanding pronunciation variation. *Speech Communication*, 29(2-4):159–176.

Silvan Heintze, Timo Baumann, and David Schlangen. 2010. Comparing local and sequential models for statistical incremental natural language understanding. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 9–16. Association for Computational Linguistics.

Julian Hough and David Schlangen. 2016. Investigating fluidity for human-robot interaction with real-time, real-world grounding strategies. In *Proceedings of the 17th Annual SIGdial Meeting on Discourse and Dialogue*.

Gail Jefferson. 2004. A sketch of some orderly aspects of overlap in natural conversation. *Pragmatics and Beyond New Series*, 125:43–62.

Casey Kennington, Spyridon Kousidis, and David Schlangen. 2013. Interpreting situated dialogue utterances: an update model that uses speech, gaze, and gesture information. *Proceedings of SigDial 2013*.

Fredrik Kronlid. 2006. Turn taking for artificial conversational agents. In *International Workshop on Cooperative Information Agents*, pages 81–95. Springer.

Divesh Lala, Pierrick Milhorat, Koji Inoue, Masanari Ishida, Katsuya Takanashi, and Tatsuya Kawahara. 2017. Attentive listening system with backchanneling, response generation and flexible turn-taking. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 127–136.

Stephen C Levinson and Francisco Torreira. 2015. Timing in turn-taking and its implications for processing models of language. *Frontiers in psychology*, 6:731.

Lilla Magyari and Jan-Peter de Ruiter. 2012. Prediction of turn-ends based on anticipation of upcoming words. *Frontiers in psychology*, 3:376.

Angelika Maier, Julian Hough, and David Schlangen. 2017. Towards deep end-of-turn prediction for situated spoken dialogue systems. *Proceedings of INTERSPEECH 2017*.

Maike Paetzel, Ramesh Manuvinakurike, and David DeVault. 2015. so, which one is it? the effect of alternative incremental architectures in a high-performance game-playing agent. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 77–86.

Antoine Raux and Maxine Eskenazi. 2009. A finite-state turn-taking model for spoken dialog systems. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 629–637.

Antoine Raux and Maxine Eskenazi. 2012. Optimizing the turn-taking behavior of task-oriented spoken dialog systems. *ACM Transactions on Speech and Language Processing (TSLP)*, 9(1):1–23.

Jan-Peter de Ruiter, Holger Mitterer, and Nick J Enfield. 2006. Projecting the end of a speaker's turn: A cognitive cornerstone of conversation. *Language*, 82(3):515–535.

Harvey Sacks, Emanuel A Schegloff, and Gail Jefferson. 1974. A simplest systematics for the organization of turn-taking for conversation. *Language*, pages 696–735.

Kenji Sagae, Gwen Christian, David DeVault, and David Traum. 2009. Towards natural language understanding of partial speech recognition results in dialogue systems. In *Proceedings of The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 53–56.

Matthias Scheutz, Thomas Williams, Evan Krause, Bradley Oosterveld, Vasanth Sarathy, and Tyler Frasca. 2019. An overview of the distributed integrated cognition affect and reflection diarc architecture. In *Cognitive Architectures*, pages 165–193. Springer.

David Schlangen and Gabriel Skantze. 2011. A general, abstract model of incremental dialogue processing. *Dialogue & Discourse*, 2(1):83–111.

Gabriel Skantze. 2017. Towards a general, continuous model of turn-taking in spoken dialogue using lstm recurrent neural networks. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 220–230, Saarbrcken, Germany. Association for Computational Linguistics.

Gabriel Skantze and David Schlangen. 2009. Incremental dialogue processing in a micro-domain. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 745–753. Association for Computational Linguistics.

Tanya Stivers, Nicholas J Enfield, Penelope Brown, Christina Englert, Makoto Hayashi, Trine Heinemann, Gertie Hoymann, Federico Rossano, Jan Peter de Ruiter, Kyung-Eun Yoon, et al. 2009. Universals and cultural variation in turn-taking in conversation. *Proceedings of the National Academy of Sciences*, 106(26):10587–10592.

Khiet P Truong, Ronald Poppe, and Dirk Heylen. 2010. A rule-based backchannel prediction model using pitch and pause information. In *Eleventh Annual Conference of the International Speech Communication Association*.

Nigel G Ward, Olac Fuentes, and Alejandro Vega. 2010. Dialog prediction for a general model of turn-taking. In *Eleventh Annual Conference of the International Speech Communication Association*.

Margaret Wilson and Thomas P Wilson. 2005. An oscillator model of the timing of turn-taking. *Psychonomic bulletin & review*, 12(6):957–968.

Tiancheng Zhao, Alan W Black, and Maxine Eskenazi. 2015. An incremental turn-taking model with active system barge-in for spoken dialog systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 42–50.