Pardon the Interruption: Managing Turn-Taking through Overlap Resolution in Embodied Artificial Agents

Felix Gervits and Matthias Scheutz
Human-Robot Interaction Laboratory
Tufts University
Medford, MA 02155
{felix.gervits,matthias.scheutz}@tufts.edu

Abstract

Speech overlap is a common phenomenon in natural conversation and in task-oriented interactions. As human-robot interaction (HRI) becomes more sophisticated, the need to effectively manage turn-taking and resolve overlap becomes more important. In this paper, we introduce a computational model for speech overlap resolution in embodied artificial agents. The model identifies when overlap has occurred and uses timing information, dialogue history, and the agent's goals to generate context-appropriate behavior. We implement this model in a Nao robot using the DIARC cognitive robotic architecture. The model is evaluated on a corpus of task-oriented human dialogue, and we find that the robot can replicate many of the most common overlap resolution behaviors found in the human data.

1 Introduction

Efficient turn-taking is at the heart of human social interaction. The need to fluidly and quickly manage turns-at-talk is essential not only for task-oriented dialogues but also in everyday conversation. Speech overlap is also a ubiquitous feature of natural language dialogue, and serves various supportive functions that people utilize to manage turn-taking (Jefferson, 2004). As spoken dialogue systems continue to advance, it is important that they support increasingly natural interactions with human interlocuters involving both turn-taking and overlap resolution.

Research in the field of HRI has generally overlooked the supportive role of overlap and the ways in which it affects coordination. However, robots are envisioned to serve as teammates in complex domains that involve a great deal of communication with humans (Fong et al., 2003). This requires nuanced methods to handle fluid turn-taking and overlap, especially because the frequency of overlap is higher in task-oriented settings involving remote communication (Heldner and Edlund, 2010).

In this work, we present a formal framework and computational model for overlap identification and resolution behavior in embodied, artificial agents. The present focus is on mechanisms to allow an agent to handle being overlapped on its turn. The model is based on empirical work in a search and rescue domain, and utilizes a variety of features including overlap timing and dialogue context to resolve overlap in real-time in a human-like manner. We implement the model in the DIARC cognitive robotic architecture (Scheutz et al., 2007) and demonstrate its performance on various overlap classes from the behavioral data.

2 Related Work

Below we present some of the relevant theoretical and computational background literature that has informed our work.

2.1 Turn-Taking and Speech Overlap

There has been a great deal of empirical work on both turn-taking and overlap phenomena (De Ruiter et al., 2006; Jefferson, 1982, 2004; Levinson and Torreira, 2015; Magyari and de Ruiter, 2012). Many of these approaches lend support to the model of turn-taking organization proposed by Sacks et al. (1974). On this view, turns-at-talk are separated by a transition-relevance place (TRP), which is located after a complete[1] segment of speech, and represents a point at which a speaker change can "legally" occur. The claim is that people can readily predict

[1] "Complete" in this sense refers to syntactic, pragmatic, and prosodic features of the turn in progress.

Proceedings of the SIGDIAL 2018 Conference, pages 99–109,
Melbourne, Australia, 12-14 July 2018.

the location of a TRP and thus aim to start their turn around that point. However, since natural language is fast-paced and complex, sometimes people miss the TRP, resulting in overlap..

Using this model, Jefferson (1986) identified several types of overlap based on their location relative to the TRP (before, during, slightly after, and much after; see Fig. 1). These overlap types have been systematically examined over the years and have been shown to capture a large range of human overlap phenomena (Jefferson, 2004). Importantly, such an account suggests that overlap is not to be confused with *interruption* (Drew, 2009). While interruption implies a kind of intrusion into the turn, overlap is oftentimes affiliative in nature. For example, people may start their turn slightly before their interlocuter has reached a TRP in order to minimize the gap between turns. This is known as *Last-Item overlap*, and can be accomplished by projecting the end of the first starter's turn. The second starter can also come in slightly after the TRP in order to respond to the content of the first starter's prior turn; such late entry is known as *Post-Transition overlap*. Additionally, the second starter can come in mid-turn (far from the TRP) as a kind of "recognitional" overlap in order to repair, clarify, or otherwise respond to the content of the first starter's turn in progress - this is known as an *Interjacent overlap*. Overlap can also be unintentional, as in *Transition-Space overlap*. This type usually involves simultaneous turn start-up wherein two people both take the turn at the TRP. In sum, because overlap is classified into these functional categories (largely based on timing), it is possible to identify the function of an overlap in a particular context as well as the behaviors that people use to manage and resolve overlap (see Gervits and Scheutz (2018)). These properties make overlap identification and resolution appealing targets for the design of more natural spoken dialogue systems.

2.2 Speech Overlap in Dialogue Systems

While overlap resolution is important in human conversation, it has not historically received the same treatment in dialogue systems. One reason for this may be that it is seen as interruption, and thus not worthy of additional study. Many systems actually ignore overlap altogether, and simply continue speaking throughout the overlapping segment (e.g., Allen et al. (1996)). While such systems may be effective for certain applications (e.g., train booking), they are not sufficient for dialogue with social agents in collaborative task environments. On top of being less fluid and natural, these systems also present problems for grounding. If the system produces an utterance in overlap, it may not be clear that a person understood or even heard what was said.

An alternative approach, and a popular one used by some commercial dialogue systems that handle overlap, is one wherein the agent responds to overlap by simply dropping out (see e.g., Raux et al. (2006)). Apart from the fact that such a system may drop its turn when detecting ambient microphone noise, another problem is that it ignores the supportive benefit that overlap can provide. An example of this is a second starter coming in at the Last-Item position in order to minimize inter-turn gaps (see Dialog *1* below[2]). Since these overlaps are among the most common, it is very inefficient for a system to abandon an utterance at the Last-Item point. Since neither of the above-mentioned approaches can address the challenges at hand, a more nuanced approach is clearly necessary.

Recently, there have been more advanced attempts at modeling overlap behavior (DeVault et al., 2009; Selfridge and Heeman, 2010; Zhao et al., 2015). Many of these approaches involve incremental parsing to build up a partial understanding of the utterance in progress and identify appropriate points to take the turn (e.g., Skantze and Hjalmarsson (2010)). Such incremental models have been used for the generation of collaborative completions (Baumann and Schlangen, 2011; DeVault et al., 2009) and feedback (DeVault et al., 2011; Skantze and Schlangen, 2009) during a human's turn. While these computational approaches tend to focus on overlapping the human, it is also important to handle overlap when the system/agent has been overlapped. Relatively little work has been done to this end, and there remain many open questions about how to interpret the function of overlap as well as how to respond. Moreover, overlap management for HRI is an under-explored area, and one which presents additional challenges for dealing with situated, embodied interaction. The present work attempts to tackle some of these challenges.

[2]All dialogs in the paper are from human interactions in the CReST corpus. *S* represents the Searcher role and *D* represents the Director. Overlap is shown in brackets.

3 Framework Description

As a framework for classifying overlap, we use the scheme from Gervits and Scheutz (2018) which includes categories from Eberhard et al. (2010), Jefferson (1986), and Schegloff (2000) as well as our own analyses. Included in this framework is a set of categories for identifying overlap (onset point, local dialogue history) and overlap management behavior. We provide formal definitions of the various categories of the scheme below, and in Section 5 we show how a model using this framework was integrated in a robotic architecture.

An utterance in our scheme is represented as follows: $U_{agent} = SpeechAct(\alpha, \beta, \sigma, \chi, \Omega, \pi)$, where agent can be the human or robot, α represents the speaker, β represents the recipient, σ represents the surface form of the utterance, χ represents the dialogue context, Ω represents a set of four time intervals corresponding to possible overlap onset points (see below), and π represents a boolean priority value (see Section 5.2). The surface form of an utterance, σ is an ordered set of lexical items in the utterance: $\sigma = \{item_{initial}, ..., item_{final}\}$. Dialogue context, χ, can be realized in various ways, but here we assume it to be a record with at least one field to represent the previous utterance and one field to represent the current dialogue sequence. Every utterance also has a speech act type associated with it to denote the underlying communicative intention. These include various types of questions, instructions, statements, acknowledgments, and others from Carletta et al. (1997).

We also include the following components (see Section 3.2 for more detail): 1) a set of competitive overlap resolution behaviors, **C**, which include {*Continue, Disfluency*, *Self-repair*}, and 2) a set of non-competitive overlap resolution behaviors, **NC**, which include {*Drop Turn, Single Item, Wrap Up*, *Finish Turn*}. Operational definitions for these behaviors can be found in Gervits and Scheutz (2018).

3.1 Overlap Onset Point

Onset point is the key feature for classifying the function of an overlap, and refers to the window of time in which the overlap occurred (see Jefferson (2004)). There are four types in the scheme (see Fig. 1), and these are represented as elements of Ω, where $\Omega = \{\Omega_{TS}, \Omega_{PT}, \Omega_{IJ}, \Omega_{LI}\}$, and each element is a bounded time interval specifying a lower and an upper bound. The first overlap interval, *Last-Item* (see Dialog *1*) refers to overlap occurring on the last word or lexical item[3] before a TRP. Last-Item overlap is defined in our scheme as an interval containing the range of time from the onset to the offset of the final lexical item in the utterance: $\Omega_{LI}(U_{agent}) = [|onset(item_{final}) + 1|, |offset(item_{final})|]$. These values can be obtained from the speech synthesizer or estimated from syllable count.

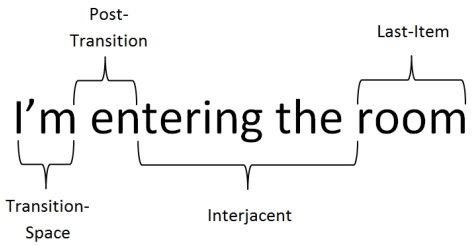

Figure 1: Key overlap onset points.

1) *D: ...one yellow block . per blue b[ox]*
S: [o k]ay

Two other overlap types in our scheme are the *Transition-Space* (see Dialog *2*) and *Post-Transition* (see Dialog *3*). *Transition-Space* overlaps are characterized by simultaneous turn startup, and occur when overlap is initiated within a conversational beat (roughly the length of a spoken syllable) after the first starter began their turn. While the length of a conversational beat varies depending on the rate of speech, it has been estimated to be around 180 ms so this is the value we have implemented (see Wilson and Wilson (2005)). Transition space is thus defined as the following interval: $\Omega_{TS}(U_{agent}) = [|onset(item_{initial})|, |len(beat)|]$, or [1, 180].

2) *S: Yes*
(0.5)
D: [So is]-
S: A[n d I] just leave that there correct?

The Post-Transition case is similar to Transition-Space except that here the timing window is offset by an additional conversational beat (see Dialog *3*. Note that the TRP here is between the words "sure" and "where"). The interval is defined in our scheme as: $\Omega_{PT}(U_{agent}) = [|len(beat) + 1|, |2(len(beat))|]$, or [181, 360] using 180 ms as the length of a beat.

3) *S: Is there a time limit?*

[3]Note that lexical items need not be single words, but may also be collocations such as "traffic light".

D: I'm- I'm not sure whe[re are you?]
S: [o k a y]

The final overlap type is the *Interjacent* (see Dialog *4*). This type of overlap occurs when the second starter comes in during the middle of the first starter's turn, i.e., not directly near a TRP. In our scheme, Interjacent overlap is defined as an interval specifying a range from the offset of the Post-Transition window (361 ms) to the onset of the Last-Item window: $\Omega_{IJ}(U_{agent}) = [|2(len(beat)) + 1|, |onset(item_{final})|]$.

4) *D: Okay maybe that was a-*
(0.5)
D: like they said th[e r e w a s]- [oka]y
S: [it was a pin]k b[o x]

3.2 Overlap Management Behaviors

The overlap management category describes various ways in which overlap can be resolved[4]. We distinguish between non-competitive behaviors, which do not involve an intent to take the turn, and competitive behaviors, which involve a "fight" for the turn. Non-competitive behaviors include simply dropping out, or uttering a single word or lexical item (e.g., "okay"). *Wrap Up* is a specific non-competitive behavior which involves briefly continuing one's turn ("wrapping up") after being overlapped and then stopping at the next TRP. Wrap Up is performed by a speaker when the overlap occurs near the end of their planned turn (within 4 beats, or 720 ms of the TRP). *Finish Turn* similarly involves reaching the TRP, but this behavior only involves a completion of the word or lexical item on which the overlap occurred (as in Last-Item). Both are considered non-competitive because the intent is to relinquish the turn.

In contrast, the competitive behaviors involve maintaining one's turn during overlap. One such behavior is *Continue*, in which the overlapped speaker simply continues their turn. This differs from Wrap Up in that the speaker continues beyond the next TRP, and so is not relinquishing the turn. Other competitive behaviors include disfluencies and self-repairs from Lickley (1998), which are only marked as competitive if they occurred within two conversational beats of the point of overlap (following Schegloff (2000)) and no other behavior was performed. These categories include silent/filled pauses, prolongations, various types of self-repairs, and combinations of all of these.

[4]We are not claiming that any of these behaviors are intentionally produced by speakers to manage overlap (though some may be), but rather that they result from the stochastic nature of fluid turn-taking.

4 Collaborative Remote Search Task

Our task domain is a search and rescue scenario in which human dyads perform a collaborative, remote search task (CReST) in a physical environment (Eberhard et al., 2010). In the task, one person is designated the *director*, and sits in front of a computer monitor that displays a map of the search environment (see Fig. 2). The other person is the *searcher* and is physically situated in the search environment. The two teammates communicate with a remote headset and must locate a variety of colored blocks scattered throughout the environment within an 8-minute time limit. We are interested in how people communicate in this domain so as to inform dialogue and coordination mechanisms for more natural and effective HRI.

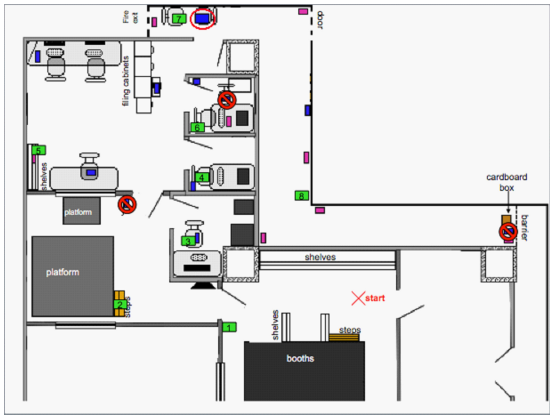

Figure 2: Map of environment from the Collaborative Remote Search Task (CReST).

Language data from 10 dyads performing this task (2712 utterances and 15194 words) was previously transcribed and annotated for a number of features, including: syntax, part-of-speech, utterances, words, disfluencies, conversational moves, and turns (Gervits et al., 2016a,b). Instances of overlap in the CReST corpus were also categorized according to their onset point and other features. (Gervits and Scheutz, 2018). There were a total of 541 overlaps in the 10 teams that we analyzed, with Transition-Space and Last-Item overlaps being the most frequent (see Table 1).

5 Model Implementation

To demonstrate our proposed model, we implemented it in the natural language pipeline of the

Table 1: Distribution of overlap onset points in the CReST corpus.

Overlap onset	Frequency
Transition-Space	35%
Post-Transition	15%
Interjacent	15%
Last-Item	35%

DIARC cognitive robotic architecture (Scheutz et al., 2007). The architecture was integrated in a SoftBank Robotics Nao robot and evaluated on the CReST corpus data. Although the CReST task was intended for a robot to fill the role of the searcher, we provide examples in which the robot can fill either role. Currently, we have implemented all of the non-competitive behaviors from the scheme, and two of the competitive behaviors (Continue and Repetition). A full implementation of all the behaviors is ongoing work.

5.1 Dialogue Management in the DIARC Architecture

The DM in DIARC is a plan-based system that allows the agent to reason over the effects of utterances and actions based on its goals. Such a system is capable of not just responding to human-initiated dialogue, but also initiating its own speech actions to accomplish goals. The DM receives utterances from the Natural Language Understanding (NLU) component that are represented using the formalism described above: $U_{agent} = SpeechAct(\alpha, \beta, \sigma, \chi, \Omega, \pi)$. Utterances of this form are also generated by the DM, and sent to the Natural Language Generation (NLG) component as output. The flow of dialogue is handled in our system through explicit exchange sequences which are stored in the dialogue context, χ. An example of such a sequence is: $AskYN(A, B) \Rightarrow ReplyY(B, A) \Rightarrow Ack(A, B)$. This represents a sequence involving a yes-no question, followed by a reply-yes, followed by an acknowledgment. A list of known sequences is provided to the system, and the current sequence is represented in a stack called Exchanges. The system always prioritizes the latest exchange added, which becomes important for managing several cases of overlapping speech (see Section 5.3 for more details).

5.2 Model Configuration

Several additional components are needed to implement the model described above. First, we require a mechanism to determine whether to compete for the turn or not. This decision is partly determined by dialogue history (e.g., previous speaker in the Post-Transition case) but also by utterance priority. As a result, a boolean priority value, π, is assigned to every utterance that a system running the model produces in a given context, χ: $\pi(U_{agent})$. This represents the urgency of that utterance at that point in the dialogue, and is used as a tiebreaker in several of the cases to determine whether to hold the turn or not.

We also need specific behaviors for managing turn-taking and dialogue context in the face of overlap. Since the DM in our architecture is a plan-based system, utterances can be thought of as (speech) actions performed to achieve a goal of the agent. As a result, dropping out of a turn (even when appropriate) should not result in the utterance being indefinitely abandoned. Thus, we need a mechanism whereby the system can store a dropped utterance and produce it later. A question then arises about exactly when is appropriate to produce the stored utterance. Our method for addressing these problems involves storing a dropped utterance in a priority queue called NLGrequests, and removing it from the current Exchanges stack. With this method, the system responds to the exchange that the human's overlapped utterance produces until it is resolved. At this point, the system will initiate utterances stored in NLGrequests, in order of priority.

One remaining topic to discuss is how to handle different kinds of feedback in overlap. Given that acknowledgments come in many varieties depending on context (Allwood et al., 1992), we distinguish between several different functions of acknowledgments in our system. Specifically, *continuers*, sometimes known as backchannel feedback, are distinguished from affirmations related to perception or understanding. This is accomplished using the onset point at which these acknowledgments occur. Acknowledgments during the Interjacent position are treated as continuers so that the agent does not attempt to drop out, compete for the turn, or add this feedback to the exchange. On the other hand, acknowledgments occurring at the Last-Item position are treated differently, and are included in the current exchange.

For identifying acknowledgments, we use a simple filter that includes several of the most common feedback words, including "okay", "yeah", "right", and "mhm".

5.3 An Algorithm for Overlap Resolution

We now turn to the task of selecting the appropriate behavior for detecting and resolving speech overlap (see Algorithm 1). A key design goal for the algorithm was speed. It is important that overlap is detected, identified, and resolved within a few hundred milliseconds in order to accommodate human expectations.

The algorithm described here operates during the robot's turn, checking for an overlapping utterance by the human. Since we are modeling remote communication, the robot transmits its speech directly to a headset worn by the human (i.e., it does not hear its own voice). In this way, we avoid the problem of disambiguating multiple simultaneous speech streams, and allow the robot to parse the human's utterance during overlap. For the algorithm, both overlapped utterances, U_{human} and U_{robot}, as well as the overlap onset point, are taken as input. The main flow of the algorithm involves using this onset point in a switch statement to decide which case to enter, and consequently, which resolution behavior to perform. The algorithm output is a behavior that corresponds to the function of the overlap.

The first step in the procedure, before considering the various cases, is to check if U_{robot} is a Single Item or Wrap Up (see Alg. 1, line 3). We have found that people do not typically compete for such utterances, so the robot's behavior here is to just finish its turn. Both utterances are then added to the Exchanges stack in the local dialogue context, χ.

If U_{robot} is not a Single Item or Wrap Up, then the algorithm checks the onset point and goes into the respective case for each type. Each case is handled in a unique way in order to select the proper competitive or non-competitive behavior based on the "function" of that overlap type. For example, because Transition-Space overlap is characterized by simultaneous startup, it uses the priority of the robot's utterance, $\pi(U_{robot})$, to determine whether to hold the turn or not (see Alg. 1, line 7). If priority is low, then it drops the turn; otherwise it competes for the turn. Post-transition overlap uses a similar mechanism, but first checks the previous speaker (see Alg. 1, line 16). This is done to give the human a chance to respond if the robot had the prior turn. Likewise, if the human had the prior turn, the robot is given a chance to respond, but only if $\pi(U_{robot})$ is high. Interjacent overlap also uses the priority mechanism, but first checks if U_{human} is a backchannel (see Alg. 1, line 31); if so, it will continue the turn. Finally, Last-Item overlap involves finishing the current turn and adding both overlapping utterances to the Exchanges stack. This means that if an acknowledgment occurs in this position, it is treated as part of the exchange rather than as backchannel feedback.

In all cases in which a turn is dropped (see e.g., Alg. 1, line 8), this involves not just abandoning U_{robot} immediately, but also storing it for later in the NLGrequests priority queue. The system simultaneously parses the ongoing U_{human} and adds this to the top of the Exchanges stack.

Competing for the turn (e.g., Alg. 1, line 12) involves producing one of the competitive behaviors from $\mathbf{C}$, including *Continue*, *Disfluency*, and *Self-Repair*. Selecting which behavior to employ is a challenging problem due to its stochastic nature, and one which remains elusive even in the empirical literature (but see Schegloff (2000) for some ideas). Our approach is based largely on our analysis of the CreST corpus, specifically on the frequency of the various overlap management behaviors for each overlap type. We use a proportion-based selection method[5] which assigns a probability for a behavior to be selected, p_b, based on its frequency (in the corpus) over the sum of the frequency of all behaviors, f_i, where $|\mathbf{C}|$ is the number of competitive behaviors:

$$p_b = \frac{f_b}{\sum_{i=1}^{|\mathbf{C}|} f_i}$$

As an example, we found that for Transition-Space overlaps, Continues were used 24% of the time in resolution, and Repetition were used 3% of the time. Since we only have these two competitive behaviors currently implemented ($|\mathbf{C}| = 2$), the algorithm will produce a Continue about 89% of the time and a Repetition about 11% of the time for Transition-Space overlaps in which it is competing for the turn. These probabilities vary depending on the overlap type.

[5] This is analogous to the fitness proportionate selection operator for genetic algorithms - see Back (1996)

6 Evaluation

Below we present the results of a qualitative evaluation on the CReST corpus data.

6.1 Results

To evaluate our algorithm, we demonstrate that it can handle the main classes of overlap observed in the corpus data[6]. These include the four main overlap types (see Fig. 1), the resolution behaviors, and the additional features from Section 5.2, including handling feedback and restarting abandoned utterances.

Transition-Space overlap (simultaneous startup) is handled by using the priority of the robot's utterance to modulate behavior. If we set $\pi(U_{robot})$ = low, then it will drop the turn, as the director does in Dialog *2*. On the other hand, if priority is high, then it will maintain the turn as the searcher does in the same example with a Continue. We have also implemented the Repetition behavior, which the director performs to maintain the turn in Dialog *5*. The Repetition is maintained until the other speaker stops talking. Note that, as in the corpus, these competitive behaviors are not invoked during the production of a single word or lexical item. See Dialog *3* for an example where the searcher produces "okay" in overlap.

5) *D: Can you hold on a second?*
D: They'[re- they're] giving me instructions
S: [y e a h]

Post-transition overlap is characterized by a late entry by the second starter. The algorithm handles this case by checking the previous speaker and dropping out if the robot had the prior turn. Otherwise, it uses priority as a tiebreaker as in the Transition-Space case. Dialog *6* below shows an example of prior speaker being used to resolve overlap. The behavior of the director in this example is demonstrative of the algorithm's performance. On the third line, the director says "I'm not sure" which ends in a TRP. They then continued their turn with "I - I don't..." at which point the searcher overlaps to respond to the previous utterance and the director drops out mid-turn.

6) *S: Do I just take-*
D: There's other things in the box too um .
D: I'm not sure I- [I don't know what they]-
S: [o k a y . I'm just tak]ing everything in the box

Interjacent overlap is handled solely through the use of the priority mechanism to determine turn-holding or turn-yielding behavior. As demonstrated above, both of these cases are readily handled by the algorithm, and only require that $\pi(U_{robot})$ be reasonably set.

Last-Item overlap is handled by finishing the turn, and adding U_{human} to the current exchange, as in Dialog *1*. Here, the algorithm replicates the director's behavior of finishing the turn and treating the searcher's feedback as an acknowledgment in the current exchange.

Handling different kinds of feedback is another important component of our approach. In Section 5.2 we showed that continuers at the Interjacent point are handled differently than those at the Last-Item point. In Dialog *7* below, the director produces a continuer ("yeah") at the Interjacent point, followed by a "got that" at the last item position. The continuer is identified by the algorithm as such (and effectively ignored), whereas the Last-Item acknowledgment is added to the current exchange: $Stmt(A, B) \Rightarrow Ack(B, A)$.

7) *S: like . um . there's a green box number t[wo o]n the st[a i r]s*
D: [yeah] [got that]

Wrap Up is another class of overlap behavior that was observed in the corpus. We handle these cases by checking the remaining length of U_{robot} after the overlap onset. If the utterance is within 4 conversational beats (720 ms) of completion then the robot will simply finish it, as seen in Dialog *8*. Otherwise, resolution is handled based on the time window in which the overlap occurred.

8) *D: ... but was there? O[r was there not?]*
S: [n o:::]

Finally, resolving the effect of overlap on the current dialogue sequence represents a common pattern seen in the corpus. The algorithm handles this differently depending on whether the robot held the turn or dropped out. If the robot held the turn, then U_{robot} is used as the next element in the exchange. Otherwise, the robot drops the turn, and stores U_{robot} in NLGrequests to be uttered after the current exchange is complete. An example of this behavior can be seen in Dialog *9* from the corpus. Our algorithm behaves as the director in this case. It drops the "go down" utterance to quickly handle the new $Stmt(A, B) \Rightarrow Ack(B, A)$ exchange introduced by the searcher in the second line. The

[6]There is an accompanying video showing some of the algorithm behaviors. It can be found at: https://vimeo.com/260654351

abandoned utterance is now at the top of the NL-Grequests stack, so it is restarted once the prior exchange is complete.

9) *D: G[o d o w n]-* *[yeah yeah ok]ay*
S: [there's lik]e boxes all ov[er the place]
D: Go down
S: Okay
D: And turn- turn right

6.2 Discussion

We have show that the categories of our formal framework are robust and can account for much of human overlap behavior in task-oriented remote dialogue. This model represents a step towards the goal of more natural and effective turn-taking for HRI. A main advantage of our approach is that it enables robots running the model to manage overlap in human-like ways, at human-like timescales, and at minimal computational cost. By handling the different kinds of overlap, robots can produce a wide range of supportive behaviors, including: maintaining dialogue flow during overlap, allowing people to start their turn early for more efficient turn transitions, supporting recognitional overlap during the robot's turn, dropping out to allow a human to clarify or respond, prioritizing urgent messages by holding the turn, and handling simultaneous startup.

One potential issue is that, with only two of the competitive turn-holding behaviors implemented, the current system will tend to produce continues most of the time when competing for the turn. As mentioned previously, this can be problematic because continues present ambiguity in grounding. We will need to conduct empirical studies using our model to explore the grounding cost of different competitive turn-holding behaviors and establish which are the most effective. It is likely that trade offs between model accuracy and usability will be necessary moving forward. For example, in order to maintain grounding, the system may need to prolong its turn-holding behavior until the human stops talking. This is not necessarily what we find in the human data, but nevertheless it may be crucial for a dialogue system.

7 Future Work and Conclusion

7.1 Future Work

While we have demonstrated that our model can handle various classes of behaviors found in the corpus, other components of the system still need to be considered for future evaluation. The components described in Section 5.2 such as priority modulation, feedback handling, delaying abandoned utterances, sequence organization (using the Exchange stack), and behavior selection will need to be separately evaluated in future work. Moreover, a comparison of this system with "non-humanlike" dialogue systems (e.g., Funakoshi et al. (2010) and Shiwa et al. (2009)) will inform whether naturalness and responsiveness are desirable components in a dialogue system.

The other main direction of future work is extending the model to produce overlap on a human's turn. This will require a fully incremental system to predict potential turn completion points. By building up a partial prediction of the utterance in progress, the system will be able to generate backchannel feedback, recognitional overlap, collaborative completions, and other instances of intentional overlap. It will also be able to engage in fluid turn-taking to avoid accidental overlap altogether, and to recover quickly when it happens.

7.2 Conclusion

We have introduced a formal framework and computational model for embodied artificial agents to recover from being overlapped while speaking. The model is informed by extensive empirical work both from the literature as well as from our own analyses. We have integrated the model in the DIARC cognitive robotic architecture and demonstrated how an agent running this model recovers from common overlap patterns found in a human search and rescue domain. The utility of the model is that it can quickly identify and resolve overlap in natural and effective ways, and at minimal computational cost. This project is a step in a larger effort to model various aspects of human dialogue towards the goal of developing genuine robot teammates that can communicate and coordinate effectively in a variety of complex domains.

Acknowledgments

This work was funded by a NASA Space Technology Research Fellowship under award 80NSSC17K0184. We would like to thank Shereen Oraby and the anonymous reviewers for their helpful contributions.

References

James F Allen, Bradford W Miller, Eric K Ringger, and Teresa Sikorski. 1996. A robust system for natural spoken dialogue. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 62–70. Association for Computational Linguistics.

Jens Allwood, Joakim Nivre, and Elisabeth Ahlsén. 1992. On the semantics and pragmatics of linguistic feedback. *Journal of semantics*, 9(1):1–26.

Thomas Back. 1996. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press.

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.

Jean Carletta, Stephen Isard, Gwyneth Doherty-Sneddon, Amy Isard, Jacqueline C Kowtko, and Anne H Anderson. 1997. The reliability of a dialogue structure coding scheme. *Computational linguistics*, 23(1):13–31.

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.

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. 2011. 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*.

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.

Kathleen M Eberhard, Hannele Nicholson, Sandra Kübler, Susan Gundersen, and Matthias Scheutz. 2010. The indiana "cooperative remote search task"(crest) corpus. In *Proceedings of the 11th edition of the Language Resources and Evaluation Conference (LREC)*.

Terrence Fong, Charles Thorpe, and Charles Baur. 2003. Collaboration, dialogue, human-robot interaction. In *Robotics Research*, pages 255–266. Springer.

Kotaro Funakoshi, Mikio Nakano, Kazuki Kobayashi, Takanori Komatsu, and Seiji Yamada. 2010. Non-humanlike spoken dialogue: a design perspective. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 176–184. Association for Computational Linguistics.

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

Mattias Heldner and Jens Edlund. 2010. Pauses, gaps and overlaps in conversations. *Journal of Phonetics*, 38(4):555–568.

Gail Jefferson. 1982. Two explorations of the organization of overlapping talk in conversation. In *Tilburg Papers in Language and Literature 28*. University of Tilburg.

Gail Jefferson. 1986. Notes on 'latency' in overlap onset. *Human Studies*, 9(2-3):153–183.

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

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

Robin J Lickley. 1998. HCRC disfluency coding manual. In *Technical Report HCRC/TR-100*. Human Communication Research Centre, University of Edinburgh.

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

Antoine Raux, Dan Bohus, Brian Langner, Alan W Black, and Maxine Eskenazi. 2006. Doing research on a deployed spoken dialogue system: One year of let's go! experience. In *Ninth International Conference on Spoken Language Processing*.

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

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

Matthias Scheutz, Paul Schermerhorn, James Kramer, and David Anderson. 2007. First steps toward natural human-like hri. *Autonomous Robots*, 22(4):411–423.

Ethan O Selfridge and Peter A Heeman. 2010. Importance-driven turn-bidding for spoken dialogue systems. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 177–185. Association for Computational Linguistics.

Toshiyuki Shiwa, Takayuki Kanda, Michita Imai, Hiroshi Ishiguro, and Norihiro Hagita. 2009. How quickly should a communication robot respond? delaying strategies and habituation effects. *International Journal of Social Robotics*, 1(2):141–155.

Gabriel Skantze and Anna Hjalmarsson. 2010. Towards incremental speech generation in dialogue systems. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 1–8. 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.

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.

Algorithm 1 Speech Overlap Management

procedure MANAGEOVERLAP(U_{human},U_{robot}, onset point) ▷ Input: Human and robot utterances in context χ, where $U_{agent} = SpeechAct(\alpha, \beta, \sigma, \chi, \Omega, \pi)$; onset point is the timing of the overlap. Output: Resolution behavior
  while robot speaking **do**
    if $\sigma(U_{robot})$ = *Single Item* or length($\sigma(U_{robot})$) - onset point $<$ 720 **then**
      Finish Turn(U_{robot}) ▷ Single Item or Wrap Up. Stop at the next TRP
      Exchanges.push(U_{robot}, U_{human})
    else if onset point $\in \Omega_{TS}$ **then** ▷ Transition-space case. 180 ms of start of U_{robot}
      if $\pi(U_{robot})$ = low **then** ▷ Low priority utterance. Non-competitive resolution.
        Drop Turn(U_{robot})
        NLGrequests.push(U_{robot}) ▷ Store utterance for later
        Exchanges.push(U_{human}) ▷ Add human's utterance to current exchange
      else ▷ High priority utterance. Maintain turn.
        Compete(U_{robot}) ▷ Perform one of the competitive resolution behaviors
        Exchanges.push(U_{robot})
      end if
    else if onset point $\in \Omega_{PT}$ **then** ▷ Post-transition case. 180-360 ms of start of U_{robot}
      if $\chi(U_{robot})$.previous_speaker = robot **then** ▷ Drop turn to allow for response
        Drop Turn(U_{robot})
        NLGrequests.push(U_{robot})
        Exchanges.push(U_{human})
      else if $\chi(U_{robot})$.previous_speaker = human **then** ▷ Use priority to determine behavior
        if $\pi(U_{robot})$ = low **then**
          Drop Turn(U_{robot})
          NLGrequests.push(U_{robot})
          Exchanges.push(U_{human})
        else
          Compete(U_{robot})
          Exchanges.push(U_{robot})
        end if
      end if
    else if onset point $\in \Omega_{IJ}$ **then** ▷ Interjacent case. Mid-turn overlap.
      if $\sigma(U_{human}) \in$ {Backchannels} **then**
        Continue(U_{robot}) ▷ Allow backchannel feedback
        Exchanges.push(U_{robot})
      else if $\pi(U_{robot})$ = low **then**
        Drop Turn(U_{robot})
        NLGrequests.push(U_{robot})
        Exchanges.push(U_{human})
      else
        Compete(U_{robot})
        Exchanges.push(U_{robot})
      end if
    else if onset_point $\in \Omega_{LI}$ **then** ▷ Last-Item case. End of turn.
      Finish_Turn(U_{robot})
      Exchanges.push(U_{robot}, U_{human})
    end if
  end while
end procedure