"So, which one is it?" The effect of alternative incremental architectures in a high-performance game-playing agent

Maike Paetzel
University of Hamburg and
USC Institute for Creative Technologies
8paetzel@informatik.uni-hamburg.de

Ramesh Manuvinakurike and **David DeVault**
USC Institute for Creative Technologies
Playa Vista, CA, USA,
{manuvinakurike,devault}@ict.usc.edu

Abstract

This paper introduces Eve, a high-performance agent that plays a fast-paced image matching game in a spoken dialogue with a human partner. The agent can be optimized and operated in three different modes of incremental speech processing that optionally include incremental speech recognition, language understanding, and dialogue policies. We present our framework for training and evaluating the agent's dialogue policies. In a user study involving 125 human participants, we evaluate three incremental architectures against each other and also compare their performance to human-human gameplay. Our study reveals that the most fully incremental agent achieves game scores that are comparable to those achieved in human-human gameplay, are higher than those achieved by partially and non-incremental versions, and are accompanied by improved user perceptions of efficiency, understanding of speech, and naturalness of interaction.

1 Introduction

This paper presents and evaluates a game playing dialogue agent named Eve that relies on several forms of incremental language processing to achieve its best performance. In recent years, the development and adoption of incremental processing techniques in dialogue systems has continued to advance, and more-and-more research systems have included some form of incremental processing; see for example (Selfridge et al., 2013; Hastie et al., 2013; Baumann and Schlangen, 2013; Dethlefs et al., 2012; Selfridge et al., 2012; DeVault et al., 2011; Skantze and Schlangen, 2009; Schlangen et al., 2009). One compelling high-level motivation for systems builders to incorporate incremental processing into their systems is to reduce system response latency (Skantze and Schlangen, 2009). Recent studies have also demonstrated user preference of incremental systems over non-incremental counterparts (Skantze and Schlangen, 2009; Aist et al., 2007), shown positive effects of incrementality on user ratings of system efficiency and politeness (Skantze and Hjalmarsson, 2010), and even shown increases in the fluency of user speech when appropriate incremental feedback is provided (Gratch et al., 2006).

Despite this progress, there remain many open questions about the use of incremental processing in systems. One important research direction is to explore and clarify the implications and advantages of alternative incremental architectures. Using pervasive incremental processing in a dialogue system poses a fundamental challenge to traditional system architectures, which generally assume turn-level or dialogue act level units of processing rather than much smaller and higher frequency incremental units (Schlangen and Skantze, 2011). Rather than completely redesigning their architectures, system builders may be able to gain some of the advantages of incrementality, such as reduced response latencies, by incorporating incremental processing in select system modules such as automatic speech recognition or language understanding. The extent to which all modules of a dialogue system need to operate incrementally to achieve specific effects needs further exploration.

Another important research direction is to develop effective optimization techniques for dialogue policies in incremental systems. Incremental dialogue policies may need to make many fine-grained decisions per second, such as whether to initiate a backchannel or interruption of a user utterance in progress. Developing data-driven approaches to such decision-making may allow us to build more highly optimized, interactive, and ef-

fective systems than are currently possible (Ward and DeVault, 2015). Yet the computational techniques that can achieve this fine-grained optimization in practice are not yet clear. Approaches that use (Partially Observable) Markov Decision Processes and a reinforcement learning framework to optimize fine-grained turn-taking control may ultimately prove effective (see e.g. (Kim et al., 2014; Selfridge et al., 2012)), but optimizing live system interactions in this way remains a challenge.

In this paper, we present a case study of a high-performance incremental dialogue system that contributes to both of these research directions. First, our study investigates the effects of increasing levels of incremental processing on the performance and user perceptions of an agent that plays a fast-paced game where the value of rapid decision-making is emphasized. In a user study involving 125 human participants, we demonstrate a level of game performance that is broadly comparable to the performance of live human players. Only the version of our agent which makes maximal use of incremental processing achieves this level of performance, along with significantly higher user ratings of efficiency, understanding of speech, and naturalness of interaction.

Our study also provides a practical approach to the optimization of dialogue policies for incremental understanding of users' referential language in finite domains; see e.g. (Schlangen et al., 2009). Our optimization approach delivers a high level of performance for our agent, and offers insights into how the optimal decision-making policy can vary as the level of incrementality in system modules is changed. This supports a view of incremental policy optimization as a holistic process to be undertaken in conjunction with overall system design choices.

2 The RDG-Image Game

In the RDG-Image game (Paetzel et al., 2014; Manuvinakurike and DeVault, 2015), depicted in Figure 1, one person acts as a director and the other as a matcher. Players see a set of eight images on separate screens. The set of images is exactly the same for both players, but they are arranged in a different order on the screen. Image sets include pets (Figure 1), fruits, bicycles, road signs, and robots, among others.

One of the eight images is randomly selected as a target image (TI) and it is highlighted on the director's screen with a thick red border as shown in Figure 1. The goal of the director is to describe the TI so that the matcher is able to uniquely identify it from the distractors. The director and matcher are able to talk back-and-forth freely in order to identify the TI. When the matcher believes he has correctly identified the TI, he clicks on the image and communicates this to the director who has to press a button to continue with the next TI. The team scores a point for each correct guess, with a goal to complete as many images as possible.

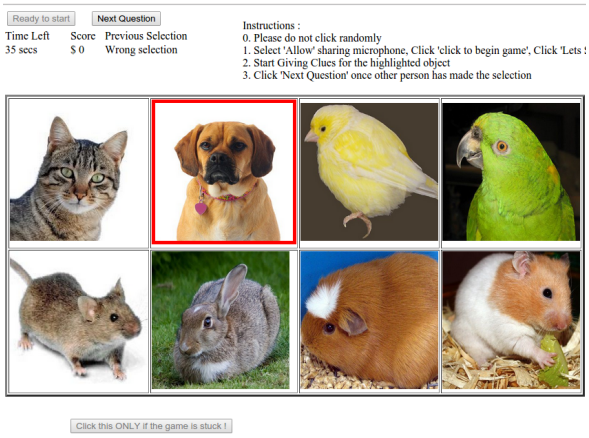


Figure 1: Browser interface for the director. The target image is highlighted by a red border. The *Next Question* button moves on to the next target.

Each team participates in 4 main game rounds. In this study, the roles remain the same for the players across all four rounds and our agent is always in the matcher role. The maximum number of TIs within each round is 12, and the rounds have a variable duration ranging from 45 to 60 seconds. The time limit for each round was chosen based on analysis of the subdialogues for that round's image sets in our earlier game corpora (Paetzel et al., 2014; Manuvinakurike and DeVault, 2015) and was set specifically to prevent participants in this study from exhausting the 12 images in a round before they run out of time. In this way, the speed and accuracy of communication are always the limiting factor to higher scores.

One game in this study consists of one training round, during which participants get comfortable with the interface and their partner, plus four main game rounds which are scored. The maximum game score is therefore 48 points (4*12). Following our approach in (Manuvinakurike and DeVault, 2015), participants are incentivized to score quickly with a bonus of $0.02 per point scored.

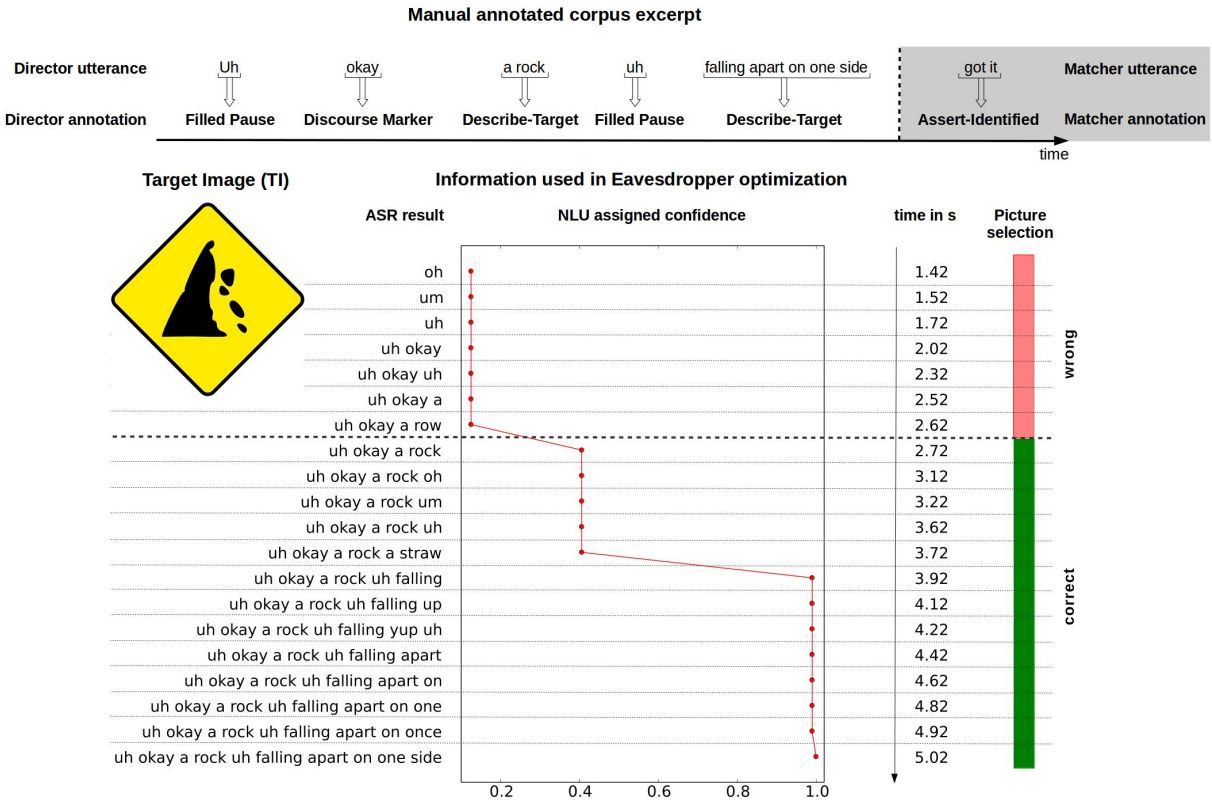


Figure 2: An image subdialogue from the RDG-Image lab corpus. The upper part shows the manual DA annotation. The lower part shows information used in the Eavesdropper policy optimization. For brevity, we include only partial ASR results that differ from the previous one. In the middle and at right are the NLU's evolving classification confidence, elapsed time, and correctness of the NLU's best guess image.

3 Observations of Human Matchers

Two corpora of human-human gameplay have previously been collected for the RDG-Image game, including the RDG-Image lab corpus (collected in our lab) (Paetzel et al., 2014) and the RDG-Image web corpus (collected on the web) (Manuvinakurike and DeVault, 2015). These corpora were used to design our automated agent.

A first step was to identify the most common types of matcher utterances and behaviour in our lab corpus. To support this analysis, 21 dialogue acts (DAs) were defined. The most important DAs for our automated matcher agents are *Assert-Identified*, used for utterances such as *Got it!* that assert the TI has been identified, and *Request-Skip*, used for utterances such as *Let's move on* that request the director to advance to the next TI.

34 human-human games were manually transcribed and annotated for dialogue acts (DAs) by a human annotator, resulting in 5415 annotated DAs. The inter-annotator agreement, measured by Krippendorf's alpha, is 0.83. 40.70% of all matcher DAs were *Assert-Identified*, and this is by far the most common DA by the matcher. For the matcher, this is followed by 15.83% of DAs which are annotated as *Out-of-domain* DAs such as laughter or meta-level discussion of the game. All other matcher DAs occur in less than 6.5% of DAs each.

Our analysis of these annotations revealed that, typically, the matcher simply listens to the director's continuing descriptions until they can perform an *Assert-Identified*, rather than taking the initiative to ask questions, for example. The top of Figure 2 shows a typical image subdialogue.

4 Design of the Agent Matcher

Based on our observations of human matchers, we focused our design of Eve on the *Assert-Identified* and *Request-Skip* acts. *Request-Skip* is a move not often used by matchers in human-human gameplay, where teams tend to take additional time as needed to agree on each image, and where teams eventually score a point for 92-98% of the TIs they encounter (depending on the image set). We anticipated that Eve might struggle with certain images

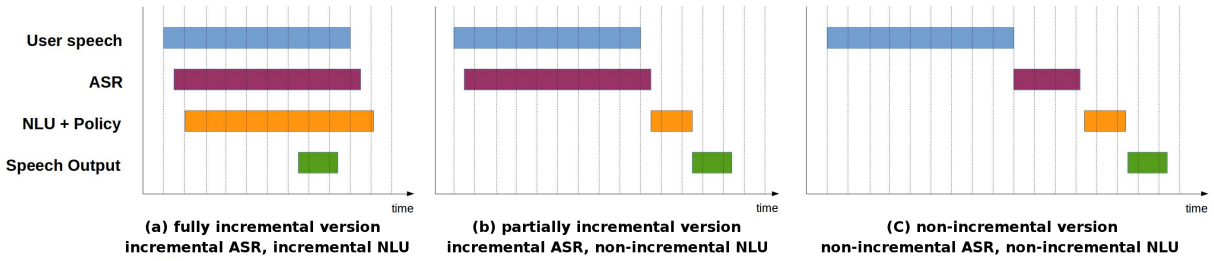


Figure 3: Timeline of the processing order of the modules in the three different versions of incrementality.

or image sets, because its NLU would be data-driven and its understanding limited to previously seen description types. Eve is therefore designed to use *Request-Skip* strategically if trying to score on the current TI appears not a good use of time.

To train our agent, the 16 image sets containing the most training examples per set were chosen from the RDG-Image lab and web corpora. Additionally, two sets of simple geometric shapes from the lab corpus were selected to serve as a training round in this study. The lab corpus includes 34 games with 68 unique participants and the web corpus includes 179 participants (some of them in multiple games). In our total training data, on average, there are 256.13 image subdialogues per image set.

4.1 Voice Activity Detection (VAD), Automatic Speech Recognition (ASR)

Audio is streamed from the user's browser to our voice activity detector, which uses the Adaptive Multi-Rate (AMR) codec (3rd Generation Partnership Project, 2008) to classify each incoming 20ms audio frame as containing voice activity or not. The VAD works incrementally in all versions of our agent. It emits voice activity events and delivers segments of detected speech (in units of 100ms) to the ASR.

Our ASR is based on Kaldi (Povey et al., 2011), and is specifically adapted from the work of (Plátek and Jurčíček, 2014), which provides support for online, incremental recognition using Kaldi. Discriminative acoustic models are trained using a combination of our in-domain audio data and out-of-domain audio using Boosted Maximum Mutual Information (BMMI) with LDA and MLLT feature transformations (Plátek and Jurčíček, 2014). Statistical language models are created using our transcribed data.

Incremental ASR. In versions of our agent with incremental ASR, detected user speech is streamed into the ASR every 100ms for online decoding, and incremental (partial) ASR results are immediately computed and sent to the NLU and policy modules. Incremental ASR is illustrated at the left of Figure 2. It is used in the fully incremental and partially incremental versions of our agent, which are illustrated in Figure 3(a) and (b).

Non-incremental ASR. In the non-incremental version of our agent (see Figure 3(c)), detected user speech is buffered until the VAD segment is concluded by the VAD. At that point, all speech is provided to the ASR and the final ASR result is computed and provided to the NLU and policy.

The non-incremental (NonInc) version serves as a performance baseline where none of ASR, NLU, or policy run incrementally. The partially incremental (PartInc) version helps quantify the benefits that come from reducing system latency through online decoding in the ASR. The fully incremental (FullInc) version explores the benefits of reacting more continuously during user speech.

4.2 Natural Language Understanding (NLU)

Our NLU operates on 1-best text outputs from the ASR. At each time t, all the 1-best texts for the current TI (i.e., spanning multiple VAD segments) are concatenated to form a combined text d_t which we call the image subdialogue text. For example, at time $t = 2.72$ in Figure 2, the NLU input is $d_t =$ *uh okay a rock*.

Prior to classification, stop-words are filtered out.[1] This process yields for example the filtered text $\text{filtered}(\textit{uh okay a rock}) = \textit{rock}$. From the filtered text, unigrams and bigrams are calculated. To reduce overfitting, only those unigrams and bigrams which occur more than three times in our training corpus are kept. The remaining unigrams and bigrams are used as input for the classifiers.

[1]The stop-word list is based on http://jmlr.org/papers/volume5/lewis04a/a11-smart-stop-list/english.stop and extended by domain-specific stop words.

A separate classifier is trained for each image set. The approach is broadly similar to (DeVault et al., 2011), and each partial ASR result is probabilistically classified as one of the eight TIs. The training data maps all the image subdialogue texts in our corpora for that image set to the correct TI. To select the classifier type, Weka (Hall et al., 2009) was used on manually transcribed data from the RDG-Image lab corpus. Multiple classifiers were tested with 10-fold cross validation. The best performance was achieved using a Naive Bayes classifier, which classified 69.15% of test instances correctly. Maximum Entropy classification performed second best with 61.37% accuracy.

4.3 General Form of Eve's Dialogue Policies

Eve's dialogue policies take the following form. Let the image set at time t be $\mathcal{I}_t = \{i_1, ..., i_8\}$, with the correct target image $T \in \mathcal{I}_t$ unknown to the agent. The maximum probability assigned to any image at time t is $P_t^* = \max_j P(T = i_j|d_t)$. Let $\mathrm{elapsed}(t)$ be the elapsed time spent on the current TI up to time t.

Eve's parameterized policy is to continue waiting for additional user speech until either her confidence P_t^* exceeds a threshold IT, or else the elapsed time on this TI exceeds a threshold GT. The *identification threshold (IT)* represents the minimal classifier confidence at which Eve performs an *Assert-Identified* (by saying *Got it!*). The *give-up threshold (GT)* is the time in seconds after which Eve performs a *Request-Skip*. Eve uses NeoSpeech[2] TTS to interact with the dialogue partner. All Eve utterances are pre-synthesized to minimize output latency.

Eve's policy is invoked by different trigger events depending on the incremental architecture. In the FullInc version (Figure 3(a)), the policy is invoked with each new partial and final ASR result (i.e. every 100ms during user speech). In the PartInc and NonInc versions (Figure 3(b) and (c)), the policy is invoked only after a new final ASR result becomes available.

Each time Eve's policy is invoked, Eve selects an action using Algorithm 1.[3] Eve's policy allows the agent to make trade-offs that incorporate both its confidence in its best guess and the opportunity cost of spending too much time on an image. In Section 5, we describe how we optimize the numeric parameters IT and GT in these policies.

Algorithm 1 Eve's dialogue policy

```
if P_t^* > IT & |filtered(d_t)| ≥ 1 then
    Assert-Identified
else if elapsed(t) < GT then
    continue listening
else
    Request-Skip
end if
```

Note that this policy learning problem could also be cast in a reinforcement learning (RL) framework. In theory, a RL model could learn when to Assert-Identified, continue listening, or Request-Skip based on the current dialogue state. One challenge in this approach would be encoding the state space in a compact way (while capturing aspects of history and temporal features relevant to action selection). A second challenge would be to use the modest amount of available data to build a user simulation that can generate incremental descriptions of objects by simulated users in a realistic way. It would be interesting to compare such an approach to our approach here in future work.

5 Policy Optimization

Optimization of the parameters IT and GT in Algorithm 1 is done using a metaphor of the agent as an *eavesdropper* on human-human gameplay. To train our agent, we start by imagining the agent as listening to the speech in human-human image subdialogues from our corpora. We imagine that as the human director describes an image to his partner, our eavesdropping agent simulates making its own independent decisions about when, if it were the matcher, it would commit to a specific TI (by saying "Got it!") or request an image skip.

For example, in Figure 2, we visualize the ASR results that would be arriving in the FullInc architecture, and the time at which they would be arriving, as this human director describes the TI as *uh okay a rock uh falling apart on one side*. In the middle and right, we visualize what the agent's NLU confidence would be in its best guess (P_t^*) as these ASR results arrive. At the right, we show that this best guess is incorrect until time 2.72.

In our optimizations in this study, we assume that the objective metric to be maximized is *points*

[2] http://www.neospeech.com/

[3] Requiring $|\mathrm{filtered}(d_t)| \geq 1$ prevents Eve from ever saying *Got it!* before any content words (non-stop words) have been received from the ASR. This could otherwise happen if the learned IT happens to be less than Eve's prior at the start of a new image.

per second (points/s). The key idea in this optimization is that each value of parameters IT and GT in Algorithm 1 translates into a specific simulatable agent response and outcome for each director description of a TI in our corpus. For example, if IT=0.3 and GT=5, then in the figure's example the agent would commit to its best interpretation at time 2.72 by performing Assert-Identified ("Got it!"). The agent would turn out to be correct and score a point. The time taken to score this point would be 2.72 seconds, plus some additional time for the matcher to say "Got it!" and for the director to click the Next Question button in the UI (see Figure 1). Our agent needs 0.5 seconds to say "Got it!", and we add an additional 0.25 seconds equal to the mean additional director click latency in our corpora. The total simulated time for this image is therefore 2.72+0.5+0.25 = 3.47 seconds.[4]

If one simulates this decision-making across an entire corpus, then for each value of IT and GT, one can calculate the total number of points hypothetically scored, total time hypothetically elapsed, and thus an estimated performance in points/s for the policy. As the parameter space is tractable here, we perform grid search across possible values of IT (step .01) and GT (step 1) and select values that maximize total points/s. We carried out this optimization *for each combination of image set and incrementality type*. Our optimization accounts for when ASR results would become available in a given incremental architecture.

Perhaps the biggest concern with this approach is that it assumes that human directors, when interacting with the agent, would produce similar utterances to what they produced when interacting with a human matcher. We have two reasons for believing this is true enough. First, as discussed in Section 3, the matcher's utterances in human-human gameplay typically play a limited role in changing the director's descriptions. Second, our results in live human-agent interactions, reported in Section 7, confirm that high performance can be achieved under this assumption.

In Table 1, the learned values for IT and GT are compared over four sample image sets (from among the 18 that are trained) in various incrementality conditions. An interesting observation is that *the optimized dialogue policy changes as the incrementality type changes*. For example, the FullInc policy for pet images (depicted in Figure 1) will wait up to 8 seconds (GT) for the confidence to reach 0.7 or higher (IT). The NonInc policy, on the other hand, will give up if confidence does not reach 0.89 within 2 seconds. Intuitively, one reason these policies can vary is that an ability to understand and respond incrementally can reduce the risk associated with waiting for additional user speech and ASR results. In the PartInc and NonInc versions, once the user begins to speak, the agent must wait for the user to complete their (possibly long) utterance before it can assess the (possibly unhelpful) new information and respond. The decision to let the user speak is therefore relatively heavyweight. This leads for example to an immediate skip for the Bikes in the NonInc version. In the FullInc version, the agent always has the option to listen to a little more speech and reconsider.

	Fully Incremental		Partially Incremental		Non-incremental	
Image set	IT	GT	IT	GT	IT	GT
Pets	0.7	8	0.52	8	0.89	2
Zoo	0.61	8	0.58	3	0.23	4
Cocktails	0.88	8	0.48	1	0.44	10
Bikes	0.80	18	0.49	7	0.0	0

Table 1: *Identification threshold* and *give-up threshold* in optimized policies for 4 image sets.

5.1 Offline Policy Evaluation Results

Our eavesdropper framework allows policies to not only be trained, but also evaluated in offline simulation, both in terms of total points scored and total points/s (which is the direct optimization metric). An excerpt from our offline evaluation results, using hold-one-user-out cross-validation, is shown in Table 2. In these offline results, the agent is sometimes able to achieve higher points/s than our human matchers did in human-human gameplay. This is true for some image sets in all three incrementality types. In general, we also observe that simulated points/s decreases as the level of incrementality in the system decreases. Note that the total number of simulated points achieved by these policies is generally less than what human players scored; the agents optimized for points/s are less likely to score a point for each image, but make up for this in speed. These offline results led us to hypothesize that, in live interaction with users, the FullInc agent would score higher than the less incremental versions in a time-constrained game.

[4]Note that when our agent performs Request-Skip, it is still able to select its best guess image, and so it may still score a point for that image (as human players can).

	Fully Incremental		Partially Incremental		Non-Incremental		Human	
	Points/s	points	Points/s	points	Points/s	points	Points/s	points
Pets	0.185	182	0.151	188	0.151	154	0.069	227
Zoo	0.220	203	0.184	196	0.177	193	0.154	243
Cocktails	0.118	153	0.103	137	0.102	172	0.124	237
Bikes	0.077	126	0.073	147	0.071	100	0.072	223

Table 2: Offline policy evaluation results for all three incrementality types and four image sets. 14 additional image sets are omitted for space reasons.

6 Online Human-Agent Study

Our online data was captured with 125 remote participants, recruited on Amazon Mechanical Turk, who interacted entirely through their web browsers. They either conversed with each other or with one of our agents.

We captured the data using the Pair Me Up web framework (Manuvinakurike and DeVault, 2015), which enables spoken dialogues through a web browser using HTML5 libraries to stream audio between remote users and our server. In (Manuvinakurike and DeVault, 2015), we demonstrated the feasibility of collecting real-time, high quality human-human game data with this web framework. For this study, we adapted Pair Me Up to support human-agent interaction. See (Manuvinakurike et al., 2015) for a detailed discussion of our web architecture, study costs, and how we managed the Amazon HITs for this study, including steps to verify each participant's audio setup and network latency.

Of the 125 participants, 50 were paired with each other (forming 25 human-human pairs) and 25 were paired with each of the FullInc, PartInc, and NonInc agents. None participated in our study more than once. From self-disclosure of the directors, 50% were female, all were over 18 (mean age 31.01, std. 10.13), and all were native English speakers.

Excerpts of Eve's gameplay during the study are included in Figure 5 in the Appendix.

After each game, participants answered a questionnaire that included basic demographic questions and also asked for their judgments on various aspects of the interaction with their partner.

7 Human-Agent Evaluation Results

In this section, we summarize our user study results, many of which are visualized in Figure 4. We evaluate our FullInc, PartInc, and NonInc agents by game score and by user's perceptions as captured in post-game questionnaires. Users responded to a range of statements with answers on a five point Likert-scale ranging from *Totally disagree* (0) to *Totally agree* (4). We compare the responses of the director in human-human (HH) pairs to the responses of human directors playing with our agent as matcher. All significance tests in this section are Wilcoxon rank sum tests.

Score (Fig. 4a). We report scores in U.S. dollars paid to participants for correct TIs ($0.02/correct TI). The FullInc system achieved a mean score of $0.33 that is significantly better than the mean $0.25 for PartInc ($p = 0.013$) and the mean $0.23 for NonInc ($p = 0.002$). No significant difference in score was observed between the PartInc and NonInc versions. These results suggest that, beyond incorporating online decoding in the ASR to reduce ASR latency, also incorporating an incremental NLU+policy is important to score maximization.

Our FullInc agent's performance in terms of score is quite strong, and comparable to HH scores. Although the mean HH score of $0.36 was a little higher than that of our FullInc agent ($0.33), the difference is not significant. The best FullInc score of $0.50 achieved as part of the study is higher than 76% of HH teams, and its worst score of $0.14 is higher than 20% of HH teams. HH teams scored significantly higher than the PartInc ($p = 0.038$) and NonInc ($p = 0.008$) versions of the system, which underscores the importance of pervasive incremental processing to achieving human-like performance in some dialogue systems.

Satisfaction with score (Fig. 4d). Human participants were significantly more satisfied with their score when working with a human matcher than with any version of our agent (for the FullInc version, $p = 0.037$). Participants who played with the FullInc agent were significantly more satisfied with their score than those in the PartInc ($p = 0.002$) and NonInc ($p = 0.017$) conditions. These results generally mirror our findings

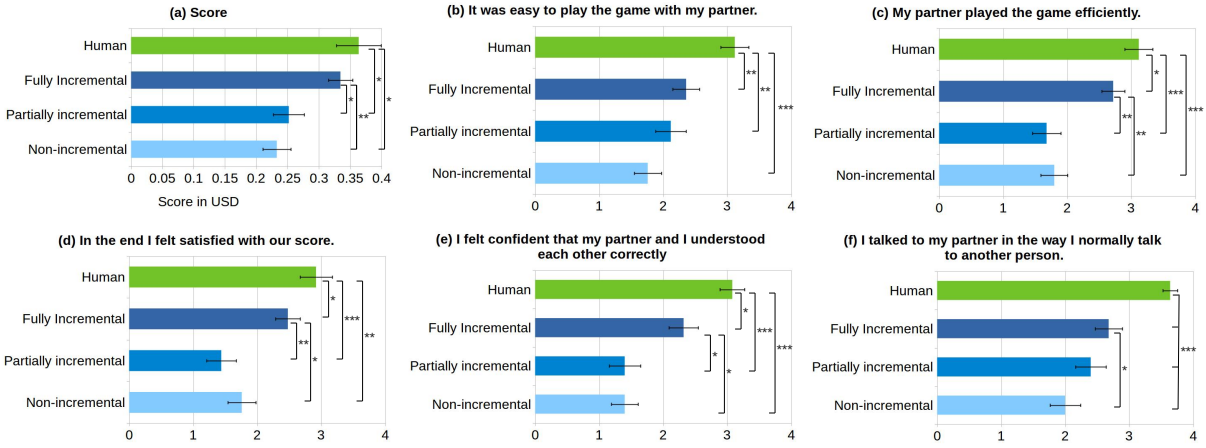


Figure 4: Scores and survey responses by condition (means and standard errors). Significant differences in Wilcoxon rank sum tests are indicated by * ($p < 0.05$), ** ($p < 0.005$), and *** ($p < 0.0005$).

for game score, and score and score satisfaction are clearly connected.

Perceived ease of gameplay (Fig. 4b). Human partners were perceived as significantly easier to play with than all agent versions. We observed a trend (not quite significant) for people to consider it easier to play with the FullInc version than with NonInc version ($p = 0.052$).

Perceived efficiency (Fig. 4c). Human partners were rated as significantly more efficient than the FullInc ($p = 0.038$), PartInc ($p < 0.0005$) and NonInc ($p < 0.0005$) agents. Among the agent versions, the FullInc agent was rated significantly more efficient than PartInc ($p = 0.001$) and NonInc ($p = 0.002$). This result echoes previous findings of increases in perceived efficiency for incremental systems, though here with a differing system architecture and task (Skantze and Hjalmarsson, 2010).

Perceived understanding of speech (Fig. 4e). Human partners elicited the most confidence that the two players were understanding each other. This perceived understanding of each other's speech was significantly higher in FullInc than in PartInc ($p = 0.010$) and NonInc ($p = 0.006$). It is interesting to consider that the NLU in these three versions is identical, and thus the level of actual understanding of user speech should be similar across conditions. We speculate that the greater responsiveness of the FullInc system increased confidence that users were being understood.

Perceived naturalness of user speech (Fig. 4f). One of our survey items investigated whether people felt they could speak naturally to their partner, "in the way I normally talk to another person". Human partners scored significantly higher than all agent versions here. The FullInc agent scored significantly higher than the NonInc agent ($p = 0.037$).

8 Conclusions

In this paper, we have presented the design, training, and evaluation of a high-performance agent that plays the RDG-Image game in the matcher role. Our policy training approach allows the system to be optimized based on its specific incremental processing architecture. In a live user evaluation, three agent versions utilizing different degrees of incremental processing were evaluated in terms of game performance and user perceptions. Our results showed that the most fully incremental agent achieves game scores that are comparable to those achieved in human-human gameplay, are higher than those achieved by partially and non-incremental versions, and are accompanied by improved user perceptions of efficiency, understanding of speech, and naturalness of interaction.

Acknowledgments

Thanks to Kallirroi Georgila and Cheng Qu. This work was supported by the National Science Foundation under Grant No. IIS-1219253 and by the U.S. Army. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views, position, or policy of the National Science Foundation or the United States Government, and no official endorsement should be inferred.

References

3rd Generation Partnership Project. 2008. Technical specification group services and system aspects; ansi-c code for the adaptive multi rate (amr) speech codec (release 12).

Gregory Aist, James Allen, Ellen Campana, Carlos Gomez Gallo, Scott Stoness, Mary Swift, and Michael K. Tanenhaus. 2007. Incremental dialogue system faster than and preferred to its nonincremental counterpart. In *Proc. of the 29th Annual Conference of the Cognitive Science Society*.

Timo Baumann and David Schlangen. 2013. Open-ended, extensible system utterances are preferred, even if they require filled pauses. In *SigDial*, pages 280–283, August.

Nina Dethlefs, Helen Hastie, Verena Riser, and Oliver Lemon. 2012. Optimising incremental generation for spoken dialogue systems: Reducing the need for fillers. In *Seventh International Natural Language Generation Conference (INLG)*.

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

Jonathan Gratch, Anna Okhmatovskaia, Francois Lamothe, Stacy Marsella, Mathieu Morales, R. van der Werf, and Louis-Philippe Morency. 2006. Virtual Rapport. In Jonathan Gratch, Michael Young, Ruth Aylett, Daniel Ballin, and Patrick Olivier, editors, *Intelligent Virtual Agents*, volume 4133, chapter 2, pages 14–27. Springer Berlin Heidelberg, Berlin, Heidelberg.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November.

Helen Hastie, Marie-Aude Aufaure, Panos Alexopoulos, Heriberto Cuayáhuitl, Nina Dethlefs, Milica Gasic, James Henderson, Oliver Lemon, Xingkun Liu, Peter Mika, et al. 2013. Demonstration of the parlance system: a data-driven, incremental, spoken dialogue system for interactive search. *Proc SIGDIAL*, pages 154–156.

Dongho Kim, Catherine Breslin, Pirros Tsiakoulis, Milica Gasic, Matthew Henderson, and Steve Young. 2014. Inverse reinforcement learning for micro-turn management. In *Interspeech*.

Ramesh Manuvinakurike and David DeVault. 2015. Pair Me Up: A Web Framework for Crowd-Sourced Spoken Dialogue Collection. In *The International Workshop on Spoken Dialog Systems (IWSDS)*.

Ramesh Manuvinakurike, Maike Paetzel, and David DeVault. 2015. Reducing the Cost of Dialogue System Training and Evaluation with Online, Crowd-Sourced Dialogue Data Collection. *Workshop on the Semantics and Pragmatics of Dialogue (SemDial)*.

Maike Paetzel, David Nicolas Racca, and David DeVault. 2014. A multimodal corpus of rapid dialogue games. In *Language Resources and Evaluation Conference (LREC)*, May.

Ondřej Plátek and Filip Jurčíček. 2014. Free on-line speech recogniser based on Kaldi ASR toolkit producing word posterior lattices. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 108–112, Philadelphia, PA, U.S.A., June. Association for Computational Linguistics.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. 2011. The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, December. IEEE Catalog No.: CFP11SRW-USB.

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

David Schlangen, Timo Baumann, and Michaela Atterer. 2009. Incremental reference resolution: The task, metrics for evaluation, and a Bayesian filtering model that is sensitive to disfluencies. In *The 10th Annual SIGDIAL Meeting on Discourse and Dialogue (SIGDIAL 2009)*.

Ethan O. Selfridge, Iker Arizmendi, Peter A. Heeman, and Jason D. Williams. 2012. Integrating incremental speech recognition and POMDP-based dialogue systems. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 275–279, Seoul, South Korea, July. Association for Computational Linguistics.

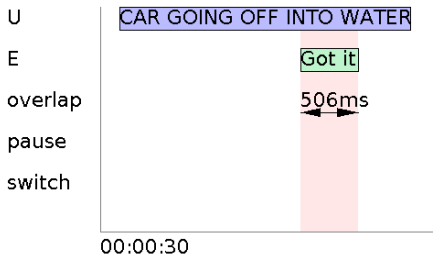
Ethan Selfridge, Iker Arizmendi, Peter Heeman, and Jason Williams. 2013. Continuously predicting and processing barge-in during a live spoken dialogue task. In *Proceedings of the SIGDIAL 2013 Conference*, pages 384–393, Metz, France, August. Association for Computational Linguistics.

Gabriel Skantze and Anna Hjalmarsson. 2010. Towards incremental speech generation in dialogue systems. In *Proceedings of the SIGDIAL 2010 Conference*, pages 1–8, Tokyo, Japan, September. Association for Computational Linguistics.

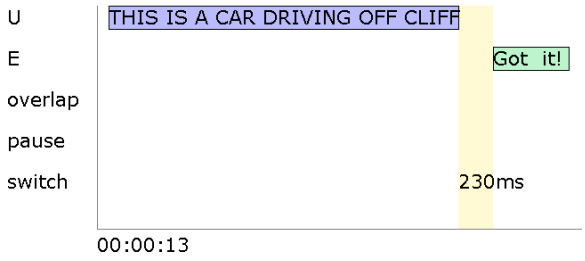
Gabriel Skantze and David Schlangen. 2009. Incremental dialogue processing in a micro-domain. In *Proceedings of the 12th Conference of the European Association for Computational Linguistics (EACL)*.

Nigel Ward and David DeVault. 2015. Ten challenges in highly-interactive dialog systems. In *AAAI Spring Symposium on Turn-taking and Coordination in Human-Machine Interaction*.

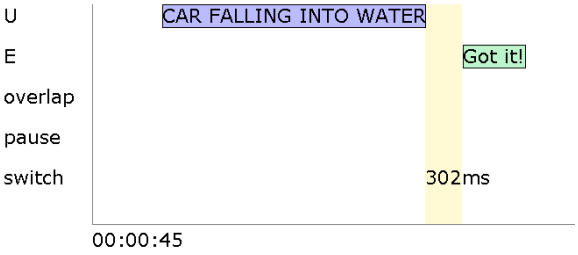
Appendix



(a) Example of Eve in fully incremental mode



(b) Example of Eve in partially incremental mode



(c) Example of Eve in non-incremental mode

(d) The target image for these examples

Figure 5: Examples of Eve's behavior in this study as different users describe the target image in (d). Seven distractor signs are also present in the display (not shown). The timing of the user's ASR results (U) and Eve's utterances (E) are indicated.

Image sources

The images of pets used in Figure 1 and of the street signs used in Figure 2 and 5 are excerpted from pictures protected by copyright and released under different licenses by their original authors. In the following attributions, we will identify the 8 images shown in the director's screen capture in Figure 1 from left-right and top-down direction, with a number from 1 to 8. Thanks to Joaquim Alves Gaspar for image 1[5] and Magnus Colossus for image 3[6], both published under CC BY-SA 3.0. Thanks to Randy Pertiet for image 2[7], Brent Moore for image 7[8] and Domenique Godbout for image 8[9], all licensed under CC-BY 2.0 and to Opacha for image 4[10] and TomiTapio for image 6[11], both licenced under CC-BY 3.0. Additionally, we kindly acknowledge Ilmari Karonen for image 5[12] and the Irish Department of Transport for the street signs shown in Figure 2[13] and 5[14], all published under Public Domain.

[5] http://commons.wikimedia.org/wiki/File:Cat_March_2010-1a.jpg
[6] http://commons.wikimedia.org/wiki/File:Canario_canary_p%C3%A1jaro_bird.jpg
[7] http://www.flickr.com/photos/34652102N04/5428922582/
[8] http://commons.wikimedia.org/wiki/File:2006_TN_State_Fair-_Guinea_Pig.jpg
[9] https://www.flickr.com/photos/dominiquegodbout/5140544743/
[10] http://commons.wikimedia.org/wiki/File:Baby-_Yellow_Naped_Amazon_Parrot_Closeup.jpg
[11] http://tomitapio.deviantart.com/art/The-bunny-says-nothing-129138755
[12] https://commons.wikimedia.org/wiki/File:Mouse_white_background.jpg
[13] http://commons.wikimedia.org/wiki/File:Ireland_road_sign_W_164.svg
[14] http://commons.wikimedia.org/wiki/File:Ireland_road_sign_W_160.svg