Faster Responses are Better Responses: Introducing Incrementality into Sociable Virtual Personal Assistants

Vivian Tsai, Timo Baumann, Florian Pecune and Justine Cassell

Abstract Speech-based interactive systems, such as virtual personal assistants, inevitably use complex architectures, with a multitude of modules working in series (or less often in parallel) to perform a task (e.g., giving personalized movie recommendations via dialog). Add modules for evoking and sustaining sociability with the user and the accumulation of processing latencies through the modules results in considerable turn-taking delays. We introduce incremental speech processing into the generation pipeline of the system to overcome this challenge with only minimal changes to the system architecture, through partial underspecification that is resolved as necessary. A user study with a sociable movie recommendation agent objectively diminishes turn-taking delays; furthermore, users not only rate the incremental system as more responsive, but also rate its recommendation performance as higher.

1 Introduction

We present a way to improve turn-taking responsiveness in a social, multimodal dialog system [9] that builds a relationship with users while recommending movies, and we show that increased responsiveness is also perceived as improved performance regarding those recommendations. The base movie agent first prompts users to specify preferred genres, directors, and actors; a knowledge graph-based recommendation system [7] consequently produces suitable movie titles and explanations of their relevance. The agent delivers not only this task-based information, but also conversational utterances intended to build rapport with the user (Table 3).

Vivian Tsai
Johns Hopkins University, e-mail: viv@jhu.edu

Timo Baumann, Florian Pecune and Justine Cassell
Carnegie Mellon University, e-mail: tbaumann@andrew.cmu.edu,
e-mail: fpecune@andrew.cmu.edu, e-mail: justine@cs.cmu.edu

When comparing human-dialog system interactions with human-human interactions, Ward et al. [12] found that a crucial issue with the former is "responsiveness," exceeded in importance only by "recognition and understanding" and "time-outs." Indeed, the response delay in our legacy application frequently disrupts interactions: users become frustrated when the agent fails to respond promptly and consequently end the interaction prematurely or even abandon the interaction mid-exchange. We therefore seek to improve the responsiveness of the movie agent.

An analysis of our legacy system showed that the most significant source of delay is the recommendation system, which takes 1.7 s on average to return the requested movie content. A user study demonstrates that our incorporation of incrementality not only objectively improves response time by eliminating this delay, but also positively impacts users' evaluations of their interactions with the movie agent in interesting ways.

2 Related Work

Incremental speech analysis and speech generation have previously been used as effective means of ensuring fast responses for spoken dialog systems [5, 6]. Specifically, the required processing times in a dialog system can take place *while* speech is delivered by the user or the system. Skantze and Hjalmarsson [11] implemented this strategy via a system that played a filler ('uhm') to give the appearance of reduced turn-taking delays (by 0.6 s on average); the resulting system was rated by users as significantly more efficient than a non-incremental version. However, the integration of incremental processing into larger existing dialog system architectures has been limited (e.g., requiring switching between different modes for certain dialog states [3]).

The incremental speech synthesis system in [2] uses only partially specified descriptions of utterances when initiating the delivery of those utterances: the beginning of each utterance must always be known for the system to start speaking it, but later parts may be underspecified as long as they are fully specified by the time the system requires them for synthesis. This concept of underspecification has not yet been applied in the context of full incremental dialog systems, to the best of our knowledge. Bąk et al. [1] present the concept of partial specification for object-oriented modeling, which we use similarly in our implementation.

3 Implementation

Our system is split into a frontend client, for speech recognition and synthesis, and a server backend (Figure 1), which involves a multiuser framework for managing multiple users and a pipeline architecture for dialog processing. Pipeline modules include a natural language understanding (NLU) module, a dialogue manager (DM), a social reasoner (SR), and a natural language generator (NLG).

For each user-agent exchange, the NLU/DM interprets user input, then returns a response intent for the agent (e.g., *ask for favorite genre*, or *give movie rec.*) and, if applicable, a movie recommendation object (comprised of suitable movie titles and reasonings behind these selections) from the recommendation system. The SR [10] selects a conversational strategy based on the DM response intent, and the NLG uses both the response intent and the SR strategy to formulate an appropriate response. Finally, the NLG replaces any variables in this response (i.e., [*movie title*]) with the corresponding content, thus producing a complete response to be output to the frontend.

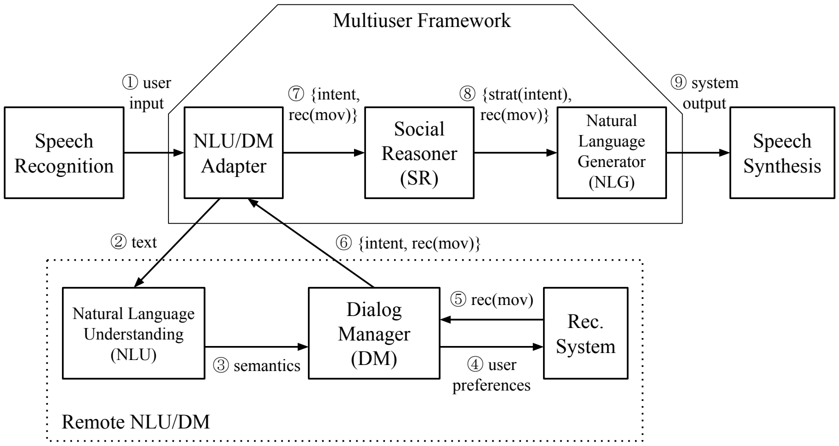

Fig. 1 Task flow for legacy, non-incremental system: modules are arranged in a pipeline; NLU, DM, and recommendation are outsourced to a separate service. The order of the information flow is shown by the circled numbers.

Our incremental solution (Figure 2) integrates the concept of partial specification while retaining the pipeline of the legacy system. Since the recommendation system creates the most significant delay, we adapt the architecture so that during interactions involving recommendations, processes for which that recommendation content is unnecessary can continue while the delay occurs. Rather than waiting for the recommendation system's output, the DM immediately outputs the response intent and an *underspecified* recommendation object. The former allows the SR to produce a strategy, which in turns allows the NLG to generate its response.

While the legacy system considers each NLG response as a whole, the incremental system splits it at unit markets, then considers each unit in sequence and incrementally sends results to the frontend. (On the frontend side, these units are placed in a queue and synthesized only when all preceding units have already been verbalized.) If a unit contains a variable and the value for that variable is currently

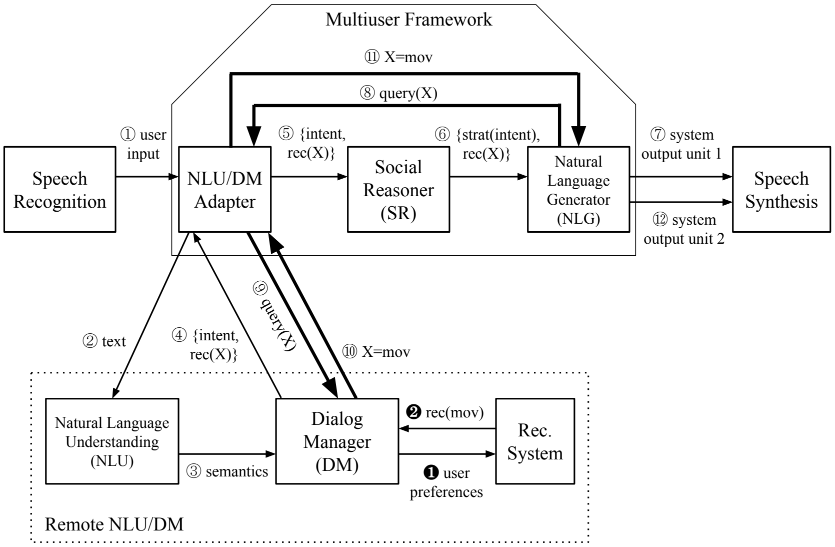

Fig. 2 Task flow for incremental system, where the response shown is split into two separate units. Movie recommendation querying is performed concurrently (inversely circled numbers). The first unit of system output is spoken while details of the second are being determined (see strong arrows).

underspecified, the NLG waits until that value can be provided; in the case of an underspecified recommendation object, the NLG queries the DM for the fully specified recommendation object (and is blocked until it can be provided).

By incrementally generating responses, our solution allows the agent to begin speaking as soon as the first unit is complete and thus before the movie recommendation is available. In other words, if a response contains enough units prior to a unit with the movie title variable, the recommendation system latency can be folded into the time required to verbalize the preceding units.

4 User Study

To evaluate the efficacy of our incremental solution, we compared users' interactions with the agent under both non-incremental (N) and incremental (I) conditions. Three dependent variables were assessed: *responsiveness of agent*, *attentiveness of agent*, and *enjoyability of conversation*. We also assessed the *recommendation quality* (i.e., quality of the agent's movie recommendations) as a presumably unaffected control variable. This gave us the following hypotheses:

- **H-Inc.** The incremental system (I) will positively impact the dependent variables: under the incremental system (I), users will rate the agent as more responsive,

more attentive, and more enjoyable than they will under the non-incremental system (N).
- **H-Rec.** The movies recommended are not affected by the I/N condition and their perceived quality should thus remain the same.

Our system's incrementality is presently limited to whole sentences as units. We thus structured NLG response patterns to ensure that all responses are comprised of two units: a "social" sentence (e.g., "Wow, here's one I'd love to go to!") and a "recommendation" sentence, the latter of which contained the movie title variable. Each social sentence was constructed such that its verbalization by the agent took more than 1.7 s, thus covering the average latency of the recommendation system.

4.1 Procedure

The experiment was conducted with a total of 24 subjects (12 female, 12 male; mean age 20-25) recruited from the university community and randomly split into four equally sized groups. Each subject held two conversations, one with the non-incremental (N) system and one with the incremental (I), with two different scenarios (each giving a specific genre, director, and actor to feed to the agent). The system-scenario, as well as N/I ordering combinations, was counter-balanced for the four groups to remove ordering and/or scenario preference effects. Subjects were informed only that the conversations would differ in scenario; no information about the difference in the systems was given.

For each conversation, subjects began by using scenario content to answer the movie agent's initial questions; continued the conversation by commenting on the agent's responses ("I've already seen that movie") and/or specifying new preferences ("Actually, I'd prefer comedies"); and ended the conversation once they were satisfied with the quantity/quality of the recommendations and the overall interaction. They then evaluated their experience by completing a questionnaire (eight randomly ordered statements, two (one original, one reverse-coded) for each of the four variables) on the conversation. To conclude, subjects completed an additional final questionnaire on whether or not they noticed a difference between the two conversations; afterwards, they were debriefed on the differing N/I system conditions.

Data from all conversations were collected in the form of log files, which included system internals, conversation transcripts, and timestamps for both frontend and backend processes, as well as audio files.

4.2 Results

The average duration of subjects' conversations with the agent was 200 s, with a mean of 6 movie recommendations given per conversation. We used timestamps

Table 1 Mean durations (in seconds) for user-agent exchanges.

	Non-Incremental System	Incremental System
Non-Recommendation Exchanges	0.7	0.7
Recommendation Exchanges	2.2	0.5

from the log file data to calculate the latency between each pair of the user's request and the agent's subsequent response.

While the differing systems had no impact on exchanges that did not involve movie recommendations (Table 1), the incremental system saved an overall average of 1.7 s per user-agent exchange in which a movie was recommended (cutting down the response time to nearly 20%) and was thus objectively more responsive than the non-incremental system. To measure participants' subjective evaluations of the non-incremental and incremental system, we performed sign tests on each of the four dependent variables (see Table 2).

The significant advantage of the incremental system supports our hypothesis **H-Inc**, particularly when focusing on users proficient in English. Most subjects maintained during their debriefings that they did not notice any time differences between the two systems, but many noted in their final questionnaire that their conversation with the incremental system went "more smoothly" and seemed to "flow better."

The incremental system was also rated higher for recommendation quality (the quality of movie recommendations), even though we used the same recommendation system for both conditions. We therefore reject hypothesis **H-Rec**. This result indicates that turn-taking delays have a significant influence on perceived quality, pointing to the need to attend to turn-taking latencies in system implementation.

In sum, the study results support our hypothesis that an incremental system is preferred and more positively rated by users. We also find that incrementality improves the perceived recommendation quality.

5 Conclusion and Future Work

Our findings indicate that the impact of incremental processing extends beyond latencies and can significantly influence users' perception of a system's capabilities.

Table 2 P-values of sign tests for each variable from non-incremental vs. incremental conditions.

Variable	All subjects (N=24)	$\geq$7 years of English (N=18)
Responsiveness	.038*	.033*
Attentiveness	.058	.038*
Enjoyability	.084	.038*
Recommendation Quality	.017*	.015*

* Statistical significance ($p < .05$)

However, our current solution is only effective when responses are structured such that social utterances are long enough to cover the recommendation system delay; furthermore, when no variables exist, these social utterances unnecessarily prolong user-agent interactions.

In the future, we envision an incremental generation system that delivers information to the user as quickly as is reasonable and otherwise buys time [8] using social talk until information becomes available. Using anytime processing [14], we could additionally consider expected recommendation performance to determine whether or not to buy time (i.e., the system could delay its response if it believes that a better recommendation may still be produced through a more extensive search). We also envision using incremental speech synthesis [4, 13] in order to break down the granularity of speech output to smaller units than sentences for higher flexibility.

References

1. K. Bąk, Z. Diskin, M. Antkiewicz, K. Czarnecki, and A. Wąsowski. *Partial Instances via Subclassing*, pages 344–364. 2013.
2. T. Baumann. Partial representations improve the prosody of incremental speech synthesis. In *Proceedings of Interspeech*, 2014.
3. T. Baumann, M. Paetzel, P. Schlesinger, and W. Menzel. Using affordances to shape the interaction in a hybrid spoken dialogue system. In *Proceedings of ESSV*, pages 12–19, 2013.
4. T. Baumann and D. Schlangen. INPRO_iSS: A component for just-in-time incremental speech synthesis. In *Proceedings of ACL System Demonstrations*, 2012.
5. H. Buschmeier, T. Baumann, B. Dorsch, S. Kopp, and D. Schlangen. Combining incremental language generation and incremental speech synthesis for adaptive information presentation. In *Proceedings of SigDial*, pages 295–303, 2012.
6. O. Buß, T. Baumann, and D. Schlangen. Collaborating on utterances with a spoken dialogue system using an isu-based approach to incremental dialogue management. In *Proceedings of SigDial*, pages 233–236, 2010.
7. R. C. Kanjirathinkal and W. Cohen. Personalized recommendations using knowledge graphs: A probabilistic logic programming approach. In *Proceedings of ACM RecSys*, pages 325–332, 2016.
8. S. López Gambino, S. Zarrieß, and D. Schlangen. Beyond on-hold messages: Conversational time-buying in task-oriented dialogue. In *Proceedings of SigDial*, pages 241–246, 2017.
9. Y. Matsuyama, A. Bhardwaj, R. Zhao, O. Romeo, S. Akoju, and J. Cassell. Socially-aware animated intelligent personal assistant agent. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 224–227, 2016.
10. A. Papangelis, R. Zhao, and J. Cassell. Towards a computational architecture of dyadic rapport management for virtual agents. In *Proceedings of the International Conference on Intelligent Virtual Agents*, pages 320–324, 2014.
11. G. Skantze and A. Hjalmarsson. Towards incremental speech generation in conversational systems. *Computer Speech and Language*, 27(1):243–262, 2013.
12. N. G. Ward, A. G. Rivera, K. Ward, and D. G. Novick. Root causes of lost time and user stress in a simple dialog system. In *Proceedings of Interspeech*, pages 1565–1568. ISCA, 2005.
13. M. Wester, D. A. Braude, B. Potard, M. P. Aylett, and F. Shaw. Real-time reactive speech synthesis: Incorporating interruptions. *Proceedings of Interspeech 2017*, pages 3996–4000, 2017.
14. S. Zilberstein. Using anytime algorithms in intelligent systems. *AI magazine*, 17(3):73, 1996.

Appendix

The following table shows an audio transcription of a conversation between a user and our movie agent. During the first part of this conversation, the movie agent asks for the user's preferences; during the second part, the movie agent suggests relevant movie titles (adding any new specifications to the list of preferences) until the system determines that the user is satisfied.

Note that each of the movie agent's recommendation sentences is preceded by a "social" sentence, which is generated in accordance with a conversational strategy in order to build rapport with the user.

Table 3 Conversation example.

	Speaker	Utterance
Preferences	*Movie Agent*	*Hi, I'm Sara. I can help you find a good movie to watch. Would you mind telling me your favorite genres?*
	User	my favorite genres are comedy and action
	Movie Agent	*I like the way you think! Who are your favorite directors?*
	User	Christopher Nolan and Christopher McQuarrie
	Movie Agent	*Okay! Who are your favorite actors?*
	User	Christian Bale and Amy Adams
Movie Rec.	*Movie Agent*	*I think this movie fits your tastes. How about Interstellar (2014)?*
	User	oh that's a good movie but I've already watched it do you want to recommend another one for me
	Movie Agent	*Wow, here is one I'd love to go to. It's called The Dark Knight Rises (2012).*
	User	yeah that's a good movie too I've seen that too so do you have another recommendation
	Movie Agent	*I think this movie fits your tastes. How about American Hustle (2013)?*
	User	oh that's a good recommendation I haven't watched that before
	Movie Agent	*If I can be of further assistance, I'll be here.*