Learning and Reusing Dialog for Repeated Interactions with a Situated Social Agent

James Kennedy[1] ✉, Iolanda Leite[1,2], André Pereira[1], Ming Sun[1], Boyang Li[1], Rishub Jain[1,3], Ricson Cheng[1,3], Eli Pincus[1,4], Elizabeth J. Carter[1], and Jill Fain Lehman[1]

[1] Disney Research, Pittsburgh, PA 15213, USA,
[2] KTH Royal Institute of Technology, SE-100 44, Stockholm, Sweden
[3] Carnegie Mellon University, Pittsburgh, PA 15213, USA,
[4] USC Institute for Creative Technologies, Los Angeles, CA 90094, USA
`james.kennedy@disneyresearch.com, jill.lehman@disneyresearch.com`

Abstract. Content authoring for conversations is a limiting factor in creating verbal interactions with intelligent virtual agents. Building on techniques utilizing semi-situated learning in an incremental crowdworking pipeline, this paper introduces an embodied agent that self-authors its own dialog for social chat. In particular, the autonomous use of crowdworkers is supplemented with a generalization method that borrows and assesses the validity of dialog across conversational states. We argue that the approach offers a community-focused tailoring of dialog responses that is not available in approaches that rely solely on statistical methods across big data. We demonstrate the advantages that this can bring to interactions through data collected from 486 conversations between a situated social agent and 22 users during a 3 week long evaluation period.

Keywords: Verbal chat; social robot; repeated interactions; borrowing dialog

1 Introduction

Traditional dialog systems rely on domain experts to manually define structure, rules, and goals to navigate through conversations, e.g., [2], imposing considerable costs for content authoring. Hand-crafted dialog knowledge also risks the introduction of personal bias – while system builders may have absolute certainty about what the agent can do, they may not fully anticipate what people will say to effect action or what the agent should say to keep people on task. As a result, statistical techniques with big data have been an increasing focus for learning dialog without the content authoring expense [11,12]. Not surprisingly, statistical techniques are most successful when the distribution of language phenomena in the underlying data match the distribution of language phenomena in the desired interaction. Such approaches are promising for a number of important applications, however they do not address the problem of efficiently authoring content when prior corpora do not exist. This work makes a contribution toward a scenario that remains a challenge for purely statistical approaches: conversation situated in natural environments with relationships that persist over time.

The current work seeks to explore this scenario by creating a Persistent Interactive Personality (PIP) that can engage in verbal social chat interactions as part of a community. Although the particular agent we focus on here, Kevin, engages only in social chat, the mechanisms used for self-authoring dialog build on existing techniques for task-driven discourse introduced in an earlier PIP; specifically, the generation of narrative descriptions of future task situations to elicit dialog lines from crowdworkers [6]. Kevin learns new dialog through face-to-face interaction and the crowdworking pipeline, then generalizes the conditions of use by borrowing across dialog states. In the following, we briefly review previously described capabilities as they occur in Kevin, then focus on when and how borrowing occurs as a function of experience. The paper contributes a description of the implemented system, with an evaluation used as a proof-of-concept. We demonstrate that the technique has a number of advantages for users during interactions, particularly in the context of repeated interactions within a community. In addition, we posit that the combination of mechanisms offers analysis opportunities for understanding natural language that are not possible with purely statistical approaches.

2 Related Work

Manual definition of dialog structure, rules or action space [2,13] incurs high cost and tends to work well when domains are small, i.e., task-oriented dialog such as when an agent has to guide users in a shopping mall [4] or interact in limited virtual worlds [1]. The less cumbersome and increasingly popular approach to learning dialog structure is to use machine learning techniques. Machine learning techniques are commonly used to translate user input directly into a system response [11, 12]. Models are typically trained on huge amounts of data that is difficult to adapt to specific situations. For example, a model trained using hundreds of movie scripts is unlikely to be applicable when talking to a close friend in an office setting. Such systems may also have problems in generating a variety of responses, and in utilizing history over repeated interactions with the same users. As a solution, Mori and Araki [9] propose a method that combines a statistical model with rule-based and transition-oriented approaches. Each of the three methods seeks to cover for shortcomings in another. For example, the rule-based element generates appropriate responses, but over a narrow set of inputs, whereas the machine learning element is broad but sometimes inappropriate in response. All three methods are employed, with utterance selection based on an approximation of naturalness and the likelihood of conversation continuation. The approach we take is similar in combining both statistical and non-statistical methods, but supplants rule authoring with a kind of learning from examples via autonomous deployment of human crowdworkers.

Commercial approaches that mix rule-based systems with machine learning approaches from leading artificial intelligence companies are also starting to power several question/answering chatbots for specific domains. However, in socially-oriented conversations where there are various ways to address user input depending on the current situation, these systems are still not very capable. For example, in response to “Good morning!”, one may say “Good morning. My name is ___” (a new hire greets a colleague), “How was your weekend?” (a friend greets another on Mondays), or simply “Hi!” (one greets a stranger on the street). Accounting for the context of the interaction becomes an important part of the content to be generated; in the work presented here, we use an explicit dialog state to explore and generalize language use across contexts, as defined by our state variables.

In social chat domains, where people generally possess sufficient knowledge to continue a conversation, we suggest that crowdworkers could be a useful resource to tailor the responses. The approach taken in [5] also uses crowdworkers but in a ‘live’ interaction, providing dialog responses to users in the moment. The content is ephemeral, requiring continual access to crowdworkers and the associated cost. It is also likely that to sustain interactions with users, some use of user history will be required to build rapport, as suggested by [7]. Guo et al. [3] describe a system where a concierge robot systematically improves its dialog capabilities in a set of categories. The robot automatically updates thresholds to decide when to respond to a user or when to ask for clarification. After asking for clarification if the user’s utterance can still not be understood, it is marked for further processing. In these cases, the development team or crowdworkers can help the system to add new utterances to existing classes or to create new ones. This work is similar to ours as we propose an agent that can learn from crowdworkers, that engages in face to face interactions and that draws on prior experience to select dialog responses. However, our work differs in several key aspects. Our agent is not a question/answering system, but is solely interested in social chit chat. We therefore do not restrict the categories that users can talk about; the dialog is completely open. Also, it does not ask the users or the development team for additional clarification, but improves its dialog capabilities in an autonomous manner using generalization or crowdworkers as necessary.

3 System Design

The goal for Kevin was to create an embodied agent capable of engaging in social chit-chat over repeated interactions with users in an office. When a valid response was available, he would continue a conversation; otherwise, the conversation would fail. The failure point was then used to drive expansion and acquire a response, so failure never occurred twice at the same point. Kevin was introduced to our user community as a casual acquaintance at work. Users were requested to be natural but ‘benevolent’ in conversations, i.e., not to try to intentionally cause dialog failure. This section will describe the system implementation, including the two methods used for generating dialog, the instantiation of the agent, and the technologies used to enable smooth human-agent interactions.

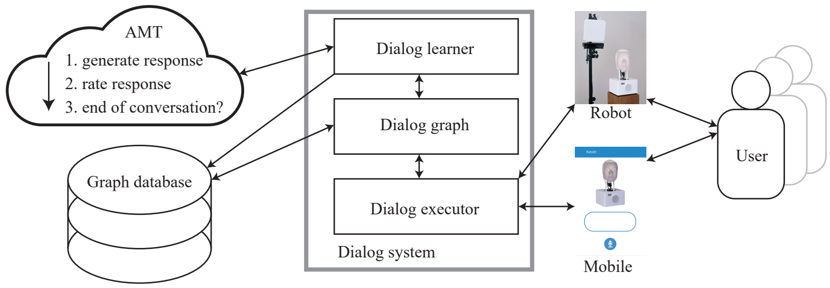

Fig. 1. System diagram showing key components and the flow of information between them. The dialog components are independent of the agent instantiation.

3.1 AMT Expansion

Kevin's dialog graph was learned both by the agent's autonomous deployment of a pipeline of Human Intelligence Tasks (HITs) on Amazon Mechanical Turk (AMT) and through face-to-face interactions with users. The full AMT pipeline consisted of three stages: generating a line of dialog, rating the line, and evaluating the line as a conversational ending (as in [6]). Crowdworkers' decisions were contextualized by a story narrative that described some or all of the agent's state and up to 5 lines of previous dialog (details below). Dialog graph nodes that arose from face-to-face interaction did so either through *direct addition* of an Automatic Speech Recognition (ASR) transcription or by *borrowing*, both described more fully below and in Fig. 2. Because each source could introduce errors – due to poor ASR or inappropriate generalization – nodes that arose from fully-situated face-to-face interaction were evaluated for quality control by Kevin's autonomous deployment of the latter two portions of the AMT pipeline.

The initial dialog graph was generated by situating AMT workers with a story narrative that conveys the values of a small number of state variables: time of day, day of week, and familiarity. For example, "Mai is a friendly person who enjoys her job in an office downtown. It's Wednesday afternoon. Kevin and Mai have never met. They run into each other at the office". The values for the variables are as follows: time of day - morning/afternoon; day of week Monday/Wednesday/Friday[5]; familiarity: never met/known each other for a few weeks. The state and number of situational features was intentionally small in order to be explicit in representation, enabling straightforward attribution of changes in the graph. All names in the exposition were changed to protect participant anonymity.

To create the initial graph, root nodes were generated for all combinations of the state variables. Then, each root node was expanded breadth-first by repeated

[5] Pilot studies informed us that this constituted a minimal set of states to begin social chat with some linguistic variety. These studies also indicated that Tuesday and Thursday contained similar language to Wednesday, so the state space was reduced and these days would resolve to using the state of Wednesday.

use of the AMT pipeline with one node randomly selected per level as the graph grew, until one path of depth 9 existed for each root. Each HIT was sent to 5 independent crowdworkers at each stage of the pipeline. Authored lines that were above the quality threshold given the editors' ratings were grouped into a single node if they had near-identical semantic similarity scores (see below). This grouping both helps to provide linguistic variation when the agent is speaking and greater coverage of the input when listening.

Nodes contain dialog and the connections between nodes represent confirmed continuation pathways allowing dialog behavior to emerge via graph traversal. When approached, Kevin randomly chooses whether to initiate a conversation or wait, but uses the same graph independent of speaking and listening roles. Thus, the simplest form of generalization occurs when Kevin acquires and speaks lines it earlier heard from a user, in situ. A conversation in which Kevin said, "How you doing?" and heard "I'm good, how about you?" as a reply, may later produce the reply "I'm good, how about you?" to the human's "How's it going today?". Given multiple options for speech at a turn, a random choice of responses is made in order to encourage growth of the graph (a strategy such as always picking the highest rated response would force growth in some graph areas to the detriment of others).

Kevin had 632 nodes in the initialized graph prior to beginning face-to-face interactions. Once deployment began, all conversation failure points (i.e., no response available yet) were marked for expansion during an overnight AMT run so that 5 responses would be available at those points the next morning. Thus, Kevin would progress further in the chat over time.

3.2 Dialog Borrowing and Selection

Dialog utterances can be 'borrowed' across state space within the graph as a form of generalization. This is done by replicating a node and updating the vertices to connect the copy to the previous point in the conversation. Borrows were rated using 3 workers and were made permanent if they made sense or, if not, stored as unusable to avoid repeating the work in the future. Although borrowing does not introduce new language to Kevin's repertoire, the mechanism does provide a way to both continue the conversation in the moment and expand dialog behavior at lower cost than a full AMT expansion. The same process is applied to utterances added from transcribed user speech.

Borrowing is based on a calculation of semantic similarity between utterances. Similarity was computed by $1 - cos(\theta)$, where θ is the angle between two feature vectors utt_1 and utt_2 that represent utterances. To compute an utterance vector representation, we started by training a Word2vec model [8] on a corpus containing over 6,500 scripts of soap operas and TV series (approximately 43 million words). Using the Gensim package [10], we trained a Skip-gram model – more appropriate for predicting the surrounding words in a sentence – with window size 5 and excluding words with frequency less than 10. The extracted feature vector for an utterance was computed by the average vectors of each word present in that utterance, excluding the vectors of stop words such as

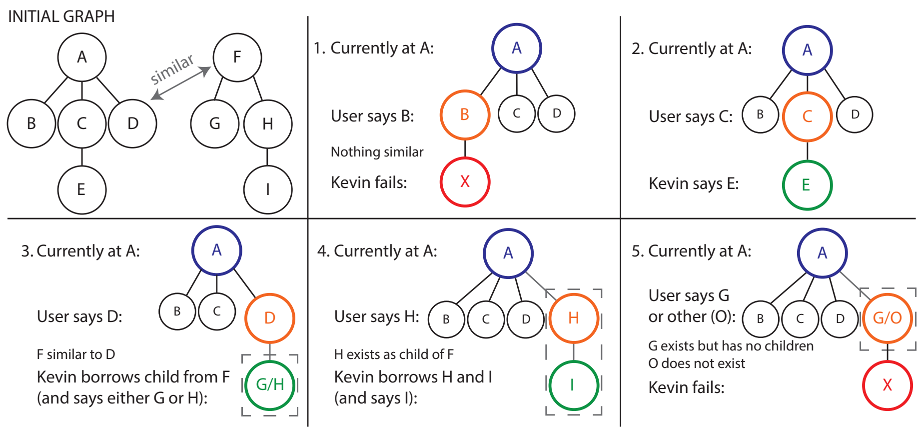

Fig. 2. To select the next utterance for Kevin to say, the dialog tree is traversed from the current node (node A in the example). Depending on the user utterance, five scenarios are possible. Two of these result in failure (red X node), one results in following a confirmed pathway, and the final two borrow from elsewhere in the graph to continue the conversation. The temporary modifications (indicated through gray dotted boxes) are marked for validation in the last two stages of the AMT pipeline.

On Monday
Kevin : How was your weekend?
User : It was great.
Kevin : What did you do?
User : Went to the movies with friends.

On Wednesday
User : Do anything after work yesterday?
Kevin : Went to the movies with friends.

Fig. 3. Example of how Kevin may borrow a user utterance.

"the" or "and"[6]. The threshold for two utterances to be considered similar was 0.6 (based on experimenter judgment of data collected from pilot studies). This cosine similarity metric is also used to determine whether what the user says is similar to any of the child nodes of the node used by the agent in the non-failure cases (see Fig. 2).

Combining the borrowing technique with the existing learned graph structure provides three possible successful routes to a response from a user utterance, along with 2 failure routes. Whenever possible, the utterance Kevin selects as a response is from the current context, based on traversing the graph from the root note that instantiates that context. If the user utterance is already in that context and it has one or more children, then Kevin will respond from that set (Fig. 2: 2). If the utterance exists in the current context, but nothing is similar according the the similarity metric just described, then Kevin will fail

[6] We started our experiments with a pre-trained Word2vec model based on the Google-News dataset, but the performance of that model for computing semantic similarity between social dialog lines was substantially worse.

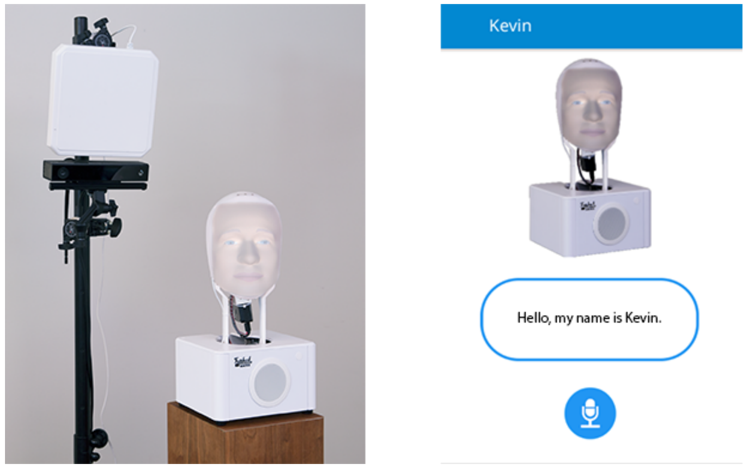

Fig. 4. Our agent, Kevin, in both robot (*left*) and mobile app (*right*) embodiments.

(Fig. 2: 1). Failure will also occur if the user says something that exists in another context, but that does not have children, or if something new is said that has no similarity to something already in the graph (Fig. 2: 5). In cases where the user says something from the current context that has no response, but is similar to something elsewhere in the graph with children, then a child will be borrowed as a response (and later validated; Fig. 2: 3). In the pair of conversations in Fig. 3, for example, a line from the user that is originally tied to the day after the weekend is borrowed to a different portion of the week via the semantic similarity of "What did you do?" and "Do anything after work?".

In the final case, (Fig. 2: 4) if the user utterance does not exist in the current context, but is similar to an utterance from another context, then the parent and child pair can be borrowed from that context. In all conversations, Kevin continues to select a dialog response until two simultaneous utterances (i.e., one from Kevin and one from the user) are both suitable 'end-of-conversation' points, based on crowdworker evaluations when the dialog was added to the graph.

3.3 Agent Instantiation

The agent was embodied as both a robot head and a virtual character in a mobile phone app that displayed the head (Fig. 4). Kevin was placed in a public space in an office for 12 days; members of the office could also choose to have the Kevin app on their phones to verbally chat with him. Kevin randomly either waited for the user to start talking or initiated the conversation himself. When the interaction began, the state was resolved based on the current time, day, and user history to provide the starting root node for dialog graph traversal. When the user said an utterance, Kevin sought to match this to something in the graph, prioritizing by the previous position as described above, borrowing if necessary. If no response was found, Kevin would fail and say "Oops, gotta go!".

The social robot version of Kevin is a Furhat retro-projected robot head, augmented with a Microsoft Kinect v2, a webcam, and a long-range RFID reader. The Kinect is used for skeleton tracking, allowing the robot to detect when a user is approaching. Users were provided with RFID badges to wear when interacting with the robot, enabling seamless user identification as well as storage and retrieval of information pertaining to that user. The webcam was used to record video logs of all interactions, and to provide an audio input for automatic speech recognition (ASR) provided by Microsoft Cognitive Services.

The mobile version of Kevin was implemented using Unity3D to provide a cross-platform mobile front-end to the dialog system. The phone used speech-to-text from IBM Watson, available as a Unity3D plug-in. Both the robot and mobile versions of the system connected to the same server running the dialog logic code and the graph database where the structure was stored. All conversations were logged in the graph database for recall in resolving starting conversation states and for subsequent analysis.

4 Evaluation and Results

The overall aim of the evaluation was to provide a proof-of-concept that an agent utilizing the expansion and borrowing capabilities described above could indeed have longer conversations with less failures over time. In addition, the evaluation allowed the collection of data to study how language was used within the user community, demonstrating the value of the approach and providing insights into improvements in the system design moving forwards.

4.1 Participants

Kevin was deployed in an office for 12 days with 22 users (age *M(SD)*=32.5(9.0) years). The study was conducted with IRB approval; participants provided informed consent and were paid. Participants were asked to interact multiple times daily; the robot was located in the kitchen to make this a convenient occurrence. User name, gender and time of participation was stored along with each conversation; names were replaced in the AMT hits and all narratives were generated for both genders. Participants averaged 22.1 interactions (*SD*=13.6), for a total of 486 conversations. The initialized graph grew from 632 to 4292 nodes.

4.2 Graph Expansion and Utterance Use

Over time, significantly more conversations were completed successfully; day 12 failure (82%) was significantly lower than day 1 failure (100%); $z = 2.913, p = .004$. Interactions also showed a trend of increasing length (Fig. 5), indicating that the approach achieved the main goal of extending conversations based on learned material. The system instantiation was intentionally designed to allow explicit exploration of how the graph expanded and the impact that this had on conversations (and vice versa). Through examining the graph node origins (user

Origin	**Created**	**Created Used**	**Borrowed**	**Borrowed Used**
ASR	270	293	207	230
AMT	2751	362	599	814
Total	3021	655	806	1044

Table 1. Origin of nodes in the final graph, with the number of times each type was used subsequent to its initial creation. The full graph consists of 4292 nodes: 3021 'created' via AMT, 806 'borrowed' across contexts, 13 roots, and 452 nodes that were rejected or borrowed to a position below a temporary node before it was rejected.

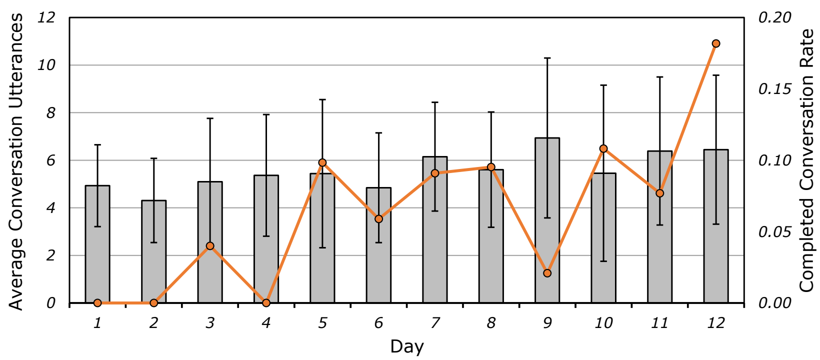

Fig. 5. Over time, our approach leads to longer conversations *(primary vertical axis; bars)*. The completed conversation rate is calculated from the number of conversations ended by the agent based on both the user and the agent saying an utterance that could end the conversation in sequence. The rate of successful conversations increases over time *(secondary vertical axis; line)*. *Error bars* show *SD*.

speech transcriptions – ASR nodes – and AMT expansions, and borrowing of both), evidence suggests that both of the learning mechanisms were beneficial to extending the conversations. Table 1 shows the breakdown of node origins and the times they were used at the completion of the study. Average use of ASR nodes (M=1.09, SD=0.38) was greater than the use of AMT nodes (M=0.13, SD=0.88), however 5 AMT nodes at a time were generated at a failure point, compared to 1 ASR node.

Our method of generalization (dialog borrowing) is supported by examining how individuals interact with Kevin. *Idiosyncrasy* is calculated as the proportion of times an individual re-visits a node that was created by him/her as opposed to other users or the agent (via AMT). Users had a tendency to revisit nodes they created, with the mean idiosyncrasy equal to 0.71 (SD=0.21). Thus, 71% of ASR node visits were by the user that created them. This suggests that users like to follow up on topics that were previously discussed and encourages personalization for users that cannot be gained through big data use.

Generated by:	this user	any user	AMT	combined
First heard	11	119	809	939
Heard before	1	2	45	48
Combined	12	121	854	987

Table 2. Utterances heard by users (i.e., only considering nodes spoken by Kevin, not heard by Kevin) split by the origin of the utterance and excluding "Oops, gotta go!".

Depth difference	-4	-3	-2	-1	0	1	2	3	4
Borrow count	27	33	74	128	196	161	117	44	40
Success rate (%)	89	85	92	94	94	94	91	77	90
SD **(%)**	6	6	3	2	2	2	3	6	5

Table 3. Borrow counts and success by depth difference in the graph. Most borrows go to a graph depth within 1 level of the borrow origin. This suggests a semantic structure to conversations based on depth.

The content that Kevin learns from users is made permanent if it makes sense (as judged using the rating portion of the AMT pipeline). As a result, users can hear back things that they have said to Kevin, or that other users have said to Kevin. This aspect of the system is particularly advantageous as it results in Kevin gradually acquiring utterances that are well-suited for use in the context in which interactions are taking place (as the utterances originate directly from those interacting in this context). Table 2 summarizes the utterances that users hear during interactions, separated by their origin. It can be seen that novel utterances are typically used, with a favorable proportion stemming from user input (13.5%) considering the overall portion of nodes originating from ASR input (9.8%). This further supports the idea that users follow up on previously discussed topics, as ASR nodes are more likely to be part of previously raised topics.

4.3 Emergent Graph Structure

Analyzing the depth relationships between borrows (which are performed based on semantic similarity), it is clear that most borrows occurred at equivalent depths in interactions (Table 3). This suggests that the conversations take some kind of structure; semantically similar utterances were produced at roughly the same depth in the conversations. This is likely to be in part a product of the shorter length of conversations occurring in this context; most of the conversations covered a greeting and health inquiry near the beginning. However, when conversations became longer, successful borrowing across depths could be used to help further generalize responses, thereby improving the efficiency of expanding dialog options and again offering analysis approaches not available using purely statistical methods.

Day 1, Interaction 1

Kevin : Hello, my name is {{ speakerA }}.
User : Hi, how are you?
Kevin : I'm good, how about you?
User : I'm very good, what do you think about the weather today?
Kevin : Oops, gotta go!

Day 4, Interaction 9

Kevin : Good afternoon <user name>.
User : How are you today Kevin?
Kevin : I'm good, How are you?
User : I'm good.
Kevin : You sounded hesitant. You sure?
User : Yes
Kevin : Oops, gotta go!

Day 8, Interaction 25

User : Hi Kevin.
Kevin : Hello <user name>.
User : How was your weekend?
Kevin : It was great
User : What did you do?
Kevin : Went Christmas shopping.
User : Did you buy anything fun?
Kevin : I was mostly shopping for others.
User : Or what. <asr error>
Kevin : Oops, gotta go!

Day 12, Interaction 33

User : Hi Kevin.
Kevin : Hello <user name>.
User : Would you like a cup of coffee?
Kevin : That would be great, thanks.
User : What's your plan for Christmas?
Kevin : I have some family flying in, so just spending it with them, you?
User : Do you have any plans this weekend?
Kevin : Not really.
User : OK enjoy your holidays.
Kevin : Oops, gotta go!

Fig. 6. Examples of conversations with the same user over the course of the evaluation. The conversations increase in length, nearing a natural conclusion by the final example, but also vary in content. Minor punctuation corrections have been made in ASR transcriptions to ease readability.

5 Discussion

Conversations were extended over time and conversations increasingly ended through reaching crowdworker marked 'end-of-conversation' points, rather than through failure (Fig. 5; Sec. 4.2). Given the completely open nature of the dialog, this kind of improvement was not necessarily guaranteed despite it being the goal of the system design. Fig. 6 shows an example of how turns evolve as the agent learns. This improvement provides the proof-of-concept for the system design that the evaluation sought to demonstrate. The results show that the application of Kevin in a persistent user community allows reuse of material generated by users through the borrowing mechanism. This can be an advantage in producing dialog responses that are well-suited to the environment. Simultaneously, the crowdworking pipeline broadens the response range and can be used to validate borrowed responses.

Kevin's use of explicit state provides the semantic anchors for organizing the dialog knowledge acquired. The state is used to generate the narratives in the AMT pipeline, thereby eliciting responses that are indexed for future use in exactly the ways that Kevin senses and represents his world. The same state descriptions are extant when Kevin acquires language through face-to-face interaction. While the result is a contextualization of language that is stronger than that which can be achieved by methods whose only representation of context is the co-occurrence statistics of words, it is by no means perfect. In particular, it will invariably be the case that small, engineered states will fail to include features that always make relevant distinctions among subsets of utterances. The current implementation of Kevin does not, for example, include any notion of time of year. With a December deployment, it is not surprising that some of the language learned is quite specific to the holiday season. As these utterances are contextualized only with respect to day of week, time of day, and length of relationship between conversants, they may well be used again in June. The decision to restrict the state to a small number of relevant features would lead to a less-coherent experience for users in this instance, one that might or might not have occurred in a purely statistically-driven approach.

Interestingly, we have some evidence that Kevin's simple mechanisms ameliorate the small-state problem to a degree, at least in a limited community setting. An early implementation decision was to not include gender of the conversant as part of the state. Nevertheless, narrative generation requires instantiating the dialog partners and assigning gender implicitly through names or pronoun choice. Because Kevin uses his dialog model in both speaker and listener roles, the lack of gender feature resulted in acquiring a response to "How are you today?" of "Great! I just found out I am pregnant". Kevin's pregnancy became enough of a topic within the office community that subsequent interactions with different users pursued the topic until, finally, Kevin's failure-driven learning resulted in an AMT-authored response that resolved the situation: "it was just a joke <user name>, guys can't get pregnant". Whether Kevin can learn to recognize such paths through his knowledge graph as opportunities for discovering important distinctions is a topic for future work, but one that is made possible by being able to inspect the dialog traces and their accompanying states over time.

In the future, we intend to reintroduce explicit goal structure to Kevin, including both conversational and non-conversational task goals. Furthermore, we will explore the trade-off between exploitation of re-using utterances that are known to be valid, against exploring a greater number of the available but untested utterances. We also expect to replace the current "copy and paste" method for growing the graph with a more efficient dialog representation in which utterance nodes store vertices connected to the states in which they have been validated. Such a change would still permit observing relationships between state variables and their effects on dialog use.

6 Conclusion

Our agent, Kevin, uses face-to-face dialog, semi-situated elicitation of dialog, and generalization of dialog context via semantic similarity to bring about advantages for users during interactions. The generalization method enables the agent to continue conversations in the moment, whilst the AMT pipeline provides breadth in responses to help further extend conversations over time. The generalization method additionally enables reuse of user spoken utterances, allowing idiosyncrasy and appropriateness in conversations with individual users, and with the larger user community. This paper described the system design in detail and provided a proof-of-concept for the approach. An evaluation consisting of 486 conversations with 22 users over a 3 week period demonstrated that while the generalization is not perfect, it provides a practicable and beneficial option when large corpora are not available to use with purely statistical approaches in an interaction scenario.

References

1. Aylett, R.S., Louchart, S., Dias, J., Paiva, A., Vala, M.: FearNot!–an experiment in emergent narrative. In: Proc. of the 5th International Conference on Intelligent Virtual Agents. pp. 305–316. Springer Berlin Heidelberg (2005)
2. Bohus, D., Rudnicky, A.: The RavenClaw dialog management framework: Architecture and systems. Computer Speech & Language 23(3), 332–361 (2009)
3. Guo, S., Lenchner, J., Connell, J., Dholakia, M., Muta, H.: Conversational Bootstrapping and Other Tricks of a Concierge Robot. In: Proc. of the 2017 ACM/IEEE International Conference on Human-Robot Interaction. pp. 73–81. ACM (2017)
4. Kanda, T., Shiomi, M., Miyashita, Z., Ishiguro, H., Hagita, N.: An affective guide robot in a shopping mall. In: Human-Robot Interaction (HRI), 2009 4th ACM/IEEE International Conference on. pp. 173–180. IEEE (2009)
5. Lasecki, W., Wesley, R., Nichols, J., Kulkarni, A., Allen, J., Bigham, J.: Chorus: a crowd-powered conversational assistant. In: UIST 2013. pp. 151–162. ACM (2013)
6. Leite, I., Pereira, A., Funkhouser, A., Li, B., Lehman, J.F.: Semi-situated Learning of Verbal and Nonverbal Content for Repeated Human-robot Interaction. In: ICMI 2016. pp. 13–20. ACM, New York, USA (2016)
7. Matsuyama, Y., Bhardwaj, A., Zhao, R., Romero, O.J., Akoju, S.A., Cassell, J.: Socially-aware animated intelligent personal assistant agent. In: 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue. p. 224 (2016)
8. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
9. Mori, H., Araki, M.: Selection method of an appropriate response in chat-oriented dialogue systems. In: 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue. p. 228 (2016)
10. Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proc. of the LREC 2010 Workshop on New Challenges for NLP Frameworks. pp. 45–50 (2010)
11. Shang, L., Lu, Z., Li, H.: Neural Responding Machine for Short-Text Conversation. CoRR abs/1503.02364 (2015)
12. Vinyals, O., Le, Q.: A neural conversational model. arXiv preprint arXiv:1506.05869 (2015)
13. Young, S., Gašić, M., Thomson, B., Williams, J.D.: POMDP-based statistical spoken dialog systems: A review. Proc. of the IEEE 101(5), 1160–1179 (2013)