Human-Centered Distributed Conversational Modeling: Efficient Modeling of Robust Virtual Human Conversations

Brent Rossen[1], Scott Lind[2], and Benjamin Lok[1]

[1] CISE
University of Florida
Gainesville, FL 32611, USA
{brossen,lok}@cise.ufl.edu
[2] Dept of Surgery, Oncology
Medical College of Georgia
Augusta, GA 30912, USA
dlind@mail.mcg.edu

Abstract. Currently, applications that focus on providing conversations with virtual humans require extensive work to create robust conversational models. We present a new approach called Human-centered Distributed Conversational Modeling. Using this approach, users create conversational models in a distributed manner. To do this, end-users interact with virtual humans to provide new stimuli (questions and statements), and domain-specific experts (e.g. medical/psychology educators) provide new virtual human responses. Using this process, users become the primary developers of conversational models. We tested our approach by creating an example application, Virtual People Factory. Using Virtual People Factory, a pharmacy instructor and 186 pharmacy students were able to create a robust conversational model in 15 hours. This is approximately 10% of the time typical in current approaches and results in more comprehensive coverage of the conversational space. In addition, surveys demonstrate the acceptability of this approach by both educators and students.

Keywords: Virtual Humans, Agents and Intelligent Systems, Human-centered Computing, Distributed Knowledge Acquisition, End-user Programming.

1 Introduction

Preparing a Virtual Human (VH) to conduct a free-form conversation can take months. As an example, our research group recently created Vic, a VH who plays the role of a patient having stomach pain. Vic was created to be capable of a 10-minute free-form conversation about his symptoms with a pharmacy student. Vic's development took approximately 6 months and 200 hours of work to develop a conversational model with a 75% accuracy rate. This time requirement restricts the scalability of VH applications. This paper presents a distributed end-user approach, the use of which

Zs. Ruttkay et al. (Eds.): IVA 2009, LNAI 5773, pp. 474–481, 2009.

results in a more complete and accurate conversational model in a significantly shorter time.

Vic is one of the many VHs created in the last few years for domain-specific conversations. These VHs [1-3] conduct natural language conversations using unannotated corpus retrieval approaches [2, 4]. In order to function well, these models require VH developers to acquire a large conversation specific corpus [1-5]. The corpus consists of what the users will say to a VH (stimulus) and what the VH will say back (response). The current methods for acquiring these corpuses are logistically difficult and time-consuming [1-5].

We propose that VH users (as opposed to developers) generate the model using Human-centered Distributed Conversational Modeling (HDCM). HDCM applies ideas from Crowdsourcing [6] and Human-Computation [7] to the problem of enumerating the stimuli-response space of a conversation. HDCM acquires the knowledge from the people who have it, the domain novices and experts. Our evaluation results show that HDCM shortens the time to model the conversation (efficient) and the resulting VH conversational model is more comprehensive (robust).

2 Motivation and Related Work

VHs for natural language conversations are increasingly popular for communication skills training. Projects in military [1], psychology [3], and medicine [2, 8] have been created with significant collaborative effort from both domain experts and computer science experts. These publications report that it is logistically difficult and time consuming to create the necessary conversational corpora [1-3, 8].

Standard resources for creating conversational corpora include -- recordings of people in "natural" or staged interactions, asking experts, Wizard of Oz (WoZ) interactions, and automated spoken interactions [9]. From a survey of projects in this area [1-4], we see that the standard approach of VH developers creating these corpora is to:

1. *Gather Starting Stimuli:* VH Developers create a starting set of stimuli and responses by asking experts, and watching recordings of natural and/or staged interactions.
2. *Refine with Users:* To find additional stimuli, they bring end-users to the lab to use WoZ interactions, where users interact with a VH controlled by a human operator, or automated spoken interactions, where users interact with an automated VH.
3. *Validate with Experts:* VH Developers collaborate with experts to validate new stimuli and create responses to those stimuli.
4. *Repeat:* Iteratively repeat steps 2 and 3 until the domain expert and VH developers conclude that the accuracy of the conversational model is acceptable.

We will hereafter refer to this method as Centralized Conversational Modeling (CCM) because of the VH developer's role as the hub for transferring information from experts and novices to the conversation corpus. CCM is limited by the following three challenges:

Challenge 1: Corpus retrieval requires a corpus detailed enough for generalization. Recordings of "natural" or staged interactions and asking experts directly provide a good "starting point," but they are not detailed enough for generalization [5].

Unanticipated stimuli account for the majority of errors (51%) in a conversation modeled using CCM [2].

Challenge 2: There are logistical issues in the use of the corpora sources regarding legal use of existing material, monetary cost, required time, and end-user availability.

Challenge 3: VH developers may not know the domain, so time is needed with domain experts to validate new stimuli and create new responses via phone/email.

In practice, these three challenges result in few iterations of user testing, and each iteration having a limited number of users. Thus, the resulting conversation corpus has significant gaps in its stimuli coverage (many unanticipated stimuli). This causes increased response errors and a decreased ability for the VH interaction to achieve educational and training objectives. The HDCM method addresses these challenges by directly engaging end-users in the process of knowledge acquisition for conversational modeling.

The idea of directly engaging end-users for knowledge acquisition was explored in *Open Mind Common Sense* [6], the Open Directory Project, and Wikipedia. These projects fall under headings of crowdsourcing and community based-design, and they embody the idea of distributed collaborative work. Collaborative work implies that the contributors for these projects are motivated to *work* on the project itself. While these projects have found great success, their approach would not succeed for VH conversation training applications; novices (e.g. students) are not generally motivated to engage directly in the process of conversational modeling [8]. We find the solution to motivating users in Lois von Ahn's *ESP Game* [7]. Von Ahn pointed out that human-based computation can solve problems that are still untenable for computers to solve, e.g. searching images. Just as Von Ahn hid computer vision in the *ESP Game,* in HDCM we hide conversational modeling within interactions.

3 Human-Centered Distributed Conversational Modeling

HDCM applies the ideas of human-based computation and crowdsourcing to the challenge of conversational modeling. We saw in section 2 that the VH developer's role in creating the conversational model is collecting knowledge from the end-users and using that knowledge to "teach" the conversational model. Using HDCM, domain experts and novices collaborate to teach the VH how to converse. They collaborate asynchronously through a GUI that is useable without any knowledge of the technical details of conversational modeling, such as XML. End-users engage in the following iterative process to create a VH conversational model.

1. *Gather Starting Stimuli:* A domain expert primes the Conversational model with best guesses as to what will be said to the VH and what the VH should say back.
2. *Refine with Novice-Users:* Multiple novices have a typed conversation with the VH. The system collects new stimuli when the VH does not have a response, and when it responds incorrectly (details in section 3.1).

3. *Validate by Expert-User:* A domain expert asynchronously enters responses to which the VH could not respond, or to which the VH responded incorrectly.
4. *Repeat:* Phase 2 and 3 are repeated until an acceptable accuracy is reached as determined by the domain expert and VH developers.

Through interactions with a VH, the domain novices enumerate the space of what will be said to the VH; while domain experts enumerate the space of what the VH will say back. Compared to CCM, iterations of HDCM are completed faster, and can involve a greater number of end-users. This process generates a corpus that enumerates the space of a conversation. That corpus forms the basis of a VH conversational model for corpus retrieval conversations.

3.1 Virtual People Factory: An Implementation of HDCM

To evaluate HDCM we created Virtual People Factory (VPF). VPF is a web-application that implements the HDCM process described in section 3 and a web service that provides support for presentation in multiple display mediums (Fig. 1).

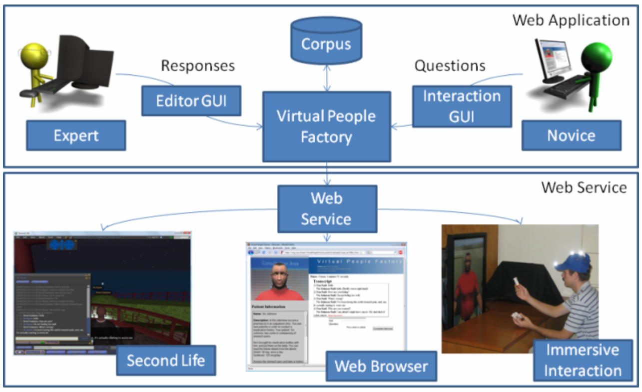

Fig. 1. Virtual People Factory System Overview[1]

We built VPF using open source software components: Apache Web Server, PHP Scripting Language, Javascript/JQuery, and MySQL Database. The system runs on a single server containing a Core2 Duo Quad-Core processor and 4GB of RAM. The application provides interfaces for both expert and novice users. Novice users perform interactions using the browser-based interview GUI. The interaction is similar to an instant messaging conversation[1].

To respond to user utterances, VPF uses an un-annotated corpus retrieval approach [2]. This approach uses keyword matching to find a list of corpus stimuli that are most similar to the current input stimulus[2]. During these interactions, VPF gathers three types of errors – *true negative, false negative*, and *false positive*. A *true negative* error

[1] Video Addendum: http://vpf.cise.ufl.edu/vpf_iva09/
[2] Details of our corpus retrieval approach found at: http://vpf.cise.ufl.edu/forum/

is when a user types a stimulus, and the system cannot find any response because there is no appropriate response. With a *false negative*, there is an appropriate response, but the corpus did not have a stimulus to locate that response. These errors are automatically logged in a list of new stimuli. However, VPF cannot reliably identify false positives. *False positives* result from a mismatched stimulus, where the VH did respond, but incorrectly. For example, the user asks, "Do you take *Tums* regularly?" and the character responds, "I take *Aspirin* all the time." Accordingly, when the VH responds incorrectly, the instructions ask users to press the "Mark Incorrect" button as seen in the video[1]. Pressing that button logs the false positive error as a new stimulus. After VPF gathers errors as new stimuli, the expert uses the resulting list of stimuli to add new stimuli and responses to the conversation corpus.

To validate new stimuli and create responses, experts use the Editor GUI[1]. In the Editor GUI, VPF shows the expert new stimuli one at a time. For each new stimulus, VPF provides its best guesses as to an existing correct response. VPF provides this list using the ordered list of responses from the corpus retrieval approach [2, 4]. The expert can choose one of those responses, or type in a new response. If they type in a new response, VPF provides a list of similar existing responses. The expert then selects an existing response, or uses their new response.

The resulting VPF conversational models are accessed in other applications using the VPF SOAP XML web-service. This web-service has been used to deploy VPF models in instant message interactions through a web-browser, spoken interactions through the *Interpersonal Simulator* [2], and typed interactions using *Second Life*. The web-service provides speech understanding, audio file serving, body and facial animations, and dialogue acts.

4 Evaluation Study: A Virtual Patient with a Stomach Ache

We used VPF to examine if the HDCM approach:

1. Enables an expert to create a conversational model,
2. Reduces conversational modeling time requirements,
3. Results in a conversational model with increased accuracy for spoken interactions.

To evaluate HDCM, a virtual patient named Vic was developed for an Introduction to Pharmacy Communications course taught by Dr. Carole Kimberlin in Spring of 2008. At minimum, Vic needed to discuss the following topics: his age, weight, gender, medical history (Hypertension, Hypothyroidism), and medication history (Zestril, Synthroid, Aspirin, Tums), and current stomach pain. To converse about these topics, Vic needed extensive domain specific knowledge. The pharmacy expert and pharmacy students provided this domain knowledge using the HDCM process.

The expert participant is the Pharmacy Instructor. She has computer experience on par with most medical professionals. The novice participants consisted of 12 teaching assistants (TAs) and 164 2nd year students from a Pharmacy Communication Skills course. Participant ages range from 20 to 60 with an average of 25.44.

Results

Conversational Modeling Time: There were three iterations of conversational modeling improvement. The first included the 12 TA interactions (group **TA**). The second included the first 44 student interactions (group **S1**). The last included the remaining 120 student interactions (group **S2**). Participants interacted for an average of 20 minutes, making the total student time 62 hours. These three rounds of user-testing required 15 hours of expert time over 2 weeks and created a conversational corpus consisting of 2655 stimuli and 595 responses.

Conversation Accuracy Improvements: We evaluated the interaction transcripts for accuracy by reviewing the response to each participant question. We marked the response as accurate if there was a semantic link between the stimuli and response[4]; meaning there was a response and it was correct according to Vic's symptoms and medical history. We analyzed the percentage of responses that were accurate for all of group TA, and a random 10 transcripts from groups S1 and S2. Results -- TA: *60.6%* (s.d. = 13.3%), S1: *71.2%* (s.d. = 6.7%), S2: *79.6%* (s.d. =5.3%). Thus, response accuracy improved with each cycle of testing/error correction.

Accuracy with Spoken Inputs: We examined the performance of the HDCM model with spoken transcripts and compared that to the performance of a conversational model created using CCM. To run the comparison, we analyzed Interpersonal Simulator [2] transcripts from 33 spoken interactions between pharmacy students and a VH patient. We removed inaccurate utterances due to speech recognition errors from the transcripts (16.7%), and analyzed the responses to the remaining utterances using both HDCM and CCM. Accuracy analysis revealed *74.5%* accuracy (s.d. = 11.1%) for the conversational model created with CCM while the one created with HDCM has *78.6%* accuracy (s.d. = 9.7%). Using a T-test on the raw accuracy numbers, we see a significant difference at $p < .05$ with $t = 2.4$.

Table 1. Results Overview: CCM vs HDCM

Method	Creators	Interactions	Expert Time	Novice Time	Stimuli	Responses
Centralized	VH Experts, Pharmacy Educators, 51 Students	Spoken Interactions	~200 Hrs	11 Hrs (13 Min Avg)	1418	303
Distributed	Pharmacy Educator, 186 Students	VPF Typed Interactions	15 Hrs	62 Hrs (20 Min Avg)	2655	595

Discussion

The results of this case study show that HDCM reduces the time needed to create the speech-understanding portion of a conversational model. Using HDCM, the Pharmacy Instructor was able to develop Vic in fifteen hours over 2 weeks, compared to the VH developers and experts creating Vic in ~200 hours over 6 months.

We see in Table 1 that there is a decrease in the required expert time by ~92.5% and increase in the total novice time by 545.5%. Given such a large amount of novice data and an effective method for processing this data, the pharmacy instructor was able to create a corpus of nearly double the size of the CCM method. This larger corpus yielded a significant 4.1% improvement in accuracy.

Feedback from both the pharmacy educator and pharmacy students stated that the experience was educationally beneficial. Surveys show that 32% of the students felt the experience was “very valuable” (ratings 8-10), with an average rating of 6.2/10. The Pharmacy Instructor expressed that “building this scenario was relatively easy with minimal training, and that the effort is worthwhile because the scenario can be used over and over.”

5 Conclusions

The results of this study show that HDCM is an efficient method for generating VHs with the ability to recognize and respond to user speech. Since these studies, healthcare educators have begun using VPF to integrate VHs into the curricula of four medical schools in the United States. To accommodate curricular integration, they have created six additional medical scenarios for teaching interview skills.

In August 2008, we opened VPF to the public: http://vpf.cise.ufl.edu. VPF currently has 41 active users outside of our research group, including VH researchers, healthcare practitioners, psychologists, and high-school students. These users have explored creating characters outside of the healthcare domain including as educators and tour guides. From their work, as of March of 2009, VPF has facilitated 1600 interactions consisting of more than 35,000 utterances.

Acknowledgements. Special thanks go to Dr. Carole Kimberlin and Dr. Diane Beck for their participation in the pharmacy study. We also thank Dr. Andrew Raij, Aaron Kotranza, Joon Chauh and Dr. Kyle Johnsen for their advice and assistance during VPF’s development. This work was made possible by a University of Florida Alumni Fellowship and National Science Foundation Grants.

References

1. Kenny, P., et al.: Building interactive virtual humans for training environments. In: ITSEC 2007, NTSA (2007)
2. Dickerson, R., et al.: Evaluating a Script-Based Approach for Simulating Patient-Doctor Interaction. In: SCS 2005 International Conference on Human-Computer Interface Advances for Modeling and Simulation, pp. 79–84 (2005)
3. Kenny, P., Parsons, T.D., Gratch, J., Rizzo, A.A.: Evaluation of justina: A virtual patient with PTSD. In: Prendinger, H., Lester, J.C., Ishizuka, M. (eds.) IVA 2008. LNCS (LNAI), vol. 5208, pp. 394–408. Springer, Heidelberg (2008)

4. Leuski, A., et al.: Building effective question answering characters. In: Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue (2006)
5. Reiter, E., Sripada, S., Robertson, R.: Acquiring correct knowledge for natural language generation. Journal of Artificial Intelligence Research 18, 491–516 (2003)
6. Singh, P., et al.: Open Mind Common Sense: Knowledge Acquisition from the General Public. In: On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE, pp. 1223–1237 (2002)
7. von Ahn, L., Dabbish, L.: Labeling images with a computer game. In: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 319–326 (2004)
8. Villaume, W.A., Berger, B.A., Barker, B.N.: Learning Motivational Interviewing: Scripting a Virtual Patient. American Journal of Pharmaceutical Education 70(2) (2006)
9. Ruttkay, Z., et al.: Evaluating Embodied Conversational Agents. Evaluating Embodied Conversational Agents 4121 (2004)