The Role of Explanations in Assessing and Correcting Personalized Intelligent Agents

Todd Kulesza
Oregon State University
School of EECS
Corvallis, OR 97331
kuleszto@eecs.oregonstate.edu

Weng-Keen Wong
Oregon State University
School of EECS
Corvallis, OR 97331
wong@eecs.oregonstate.edu

Margaret Burnett
Oregon State University
School of EECS
Corvallis, OR 97331
burnett@eecs.oregonstate.edu

Simone Stumpf
City University London
Centre for HCI Design
London EC1V 0HB, U.K.
Simone.Stumpf.1@city.ac.uk

CHI'12, May 5–10, 2012, Austin, Texas, USA.
ACM 978-1-4503-1016-1/12/05.

Abstract
Intelligent agents are becoming ubiquitous in the lives of everyday users, from simple recommenders like Google Suggest to the complex face recognition techniques used in modern photo albums. The research community, however, has only recently begun to study how people (1) assess the reliability, and (2) correct the mistakes, of these agents. This paper outlines the potenial role for explanations to help end users accomplish each of these tasks.

Author Keywords
Intelligent agents; end user programming;

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous;

General Terms
Design, Experimentation, Human Factors

Introduction and Research Interests
Intelligent agents have moved beyond mundane tasks like filtering junk e-mail. Search engines now exploit pattern recognition to detect image content (e.g., clipart, photography, and faces); Facebook and image

editors take this a step further, making educated guesses as to *who* is in a particular photo. Netflix and Amazon use collaborative filtering to recommend items of interest to their customers, while Pandora and Last.fm use similar techniques to create radio stations crafted to an individual's idiosyncratic tastes. Simple rule-based systems have evolved into agents employing complex algorithms. These *intelligent agents* are computer programs whose behavior only becomes fully specified *after* learning from an end user's training data. Because these programs continue to adapt after being deployed, they present two unique challenges for their end users.

First, end users of intelligent agents need to assess when they can rely on—or trust—their agent's work. Such trust is highly contextual—some of the agent's predictions may not matter at all to an end user, while others may matter a great deal. Further, the agent's reasoning is constantly changing as it learns from the user's behavior, so a system that was reliable yesterday may not be trustworthy today.

The second challenge is about corrections. When an intelligent agent's reasoning causes it to perform unexpectedly in the field, only the end user is in a position to correct—or more accurately, *to debug*—the agent's flawed reasoning. Here, debugging refers to *mindfully and purposely* adjusting the agent's reasoning (after its initial training) so it more closely matches the user's expectations. Recent research has made inroads into supporting this type of functionality [1, 2, 5, 7], but debugging can be difficult for even trained software developers—helping end users, who have knowledge of neither software engineering nor machine learning, is no trivial task.

We believe these two challenges—establishing an appropriate level of trust in an agent, and aligning its reasoning with a specific end user's—are inherently linked. A sound understanding of an agent's reasoning is a logical prerequisite for both assessing the agent's reliability and providing useful corrections to its reasoning. We hypothesize that explaining an agent's reasoning and capabilities to end users will enable them to form better judgments of the agent's reliability and help users to provide feedback that can substantially improve the agent's future predictions, as illustrated in Figure 1.

Background

Our prior research has begun to explore end-user interactions with intelligent agents, particularly focusing on end user attempts to assess an agent's reliability and correct its mistakes.

Using a paper prototype, we investigated three

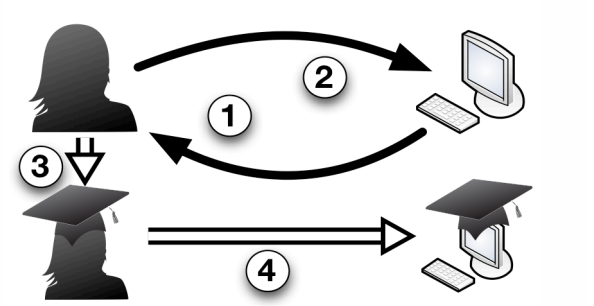

Figure 1. We envision a cyclic, explanation-based approach for users to learn about the agent's reasoning (1) and interactively correct it (2). In the process, the user learns more about what the agent can be relied upon to do, and how to effectively align its reasoning with the user's own (3), with the eventual outcome of "more intelligent" intelligent agents (4).

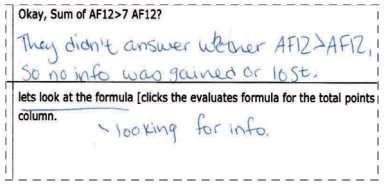

Figure 2: We used a paper prototype to elicit participant feedback in a "natural" manner.

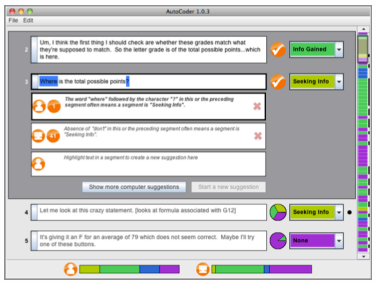

Figure 3: A prototype designed to help end users debug an intelligent agent using their "natural" vocabulary [5].

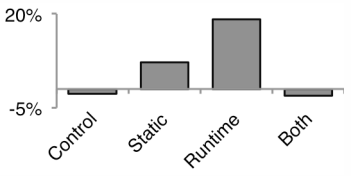

Figure 4: Participants exposed to runtime explanations debugged their intelligent agent significantly better than other participants.

different types of explanations (keyword-based, rule-based, and similarity-based) that a machine learning systems could provide to end users regarding why it was behaving in a particular manner, as well as user reactions to these explanations [9]. This paper prototype was also used to elicit corrections to the agent's reasoning from participants (e.g., adjusting feature weights), allowing us to design an interactive prototype supporting the explanations best understood by participants and the types of corrections they most requested [11]. This interactive prototype permitted us to run offline experiments studying the effects of the corrections provided by end users on prediction accuracy versus traditional label-based corrections. The results suggest that even when very simple corrections are incorporated into an agent's decision-making process, it has the potential to increase the accuracy of the resulting predictions [10, 11].

Some participants in the above study, however, markedly worsened the quality of their assistant's predictions; they encountered barriers that prevented them from successfully debugging their agent. Thus, we conducted a follow-up study to categorize the barriers and information needs that end users encounter when debugging an intelligent agent's reasoning [6]. To support users in overcoming these barriers, we conducted a formative study using the Natural Programming methodology [8] (Figure 2), identifying the types of corrections end users want to provide text-classifying intelligent agents and a natural vocabulary for the agent to use when explaining its reasoning [5]. We instantiated our approach in an online prototype (Figure 3) and evaluated it with a user study, finding that presenting explanations of the agent's current reasoning (similar to a runtime debugger) helped participants significantly improve its accuracy compared with participants who either received no explanations, or who received explanations of its static capabilities and features [5] (Figure 4).

We have also investigated methods for supporting end-user assessment of intelligent agents. This work included an exploration of the different methods for an agent to identify and explain which of its predictions are most in need of assessment by an end user (prioritization), as well as a technique for the agent to extend each user assessment to very similar predictions (coverage) [3]. We conducted a user study with three prototypes (Figure 5 illustrates one variant), each evaluating one of these prioritization methods; our findings revealed that each of our prioritization methods helped participants find significantly more of the agent's mistakes than the traditional (sans-explanation) ad-hoc assessment approach (Figure 6). Further, our coverage technique helped participants assess more than twice as many predictions as the control group [3].

Most recently, we have begun to explore the impact mental models play when end users assess and correct an intelligent agent. Our prior work has identified the potential benefits of supporting end-user assessment and end-user corrections of intelligent agents; now, we are attempting to determine how explanations of the agent's reasoning and capabilities influence users' mental models, and how these mental model support end users in such tasks. Our initial work in this area has illustrated the practicality of faithfully explaining an agent's reasoning to end users, and identified that as participants learned more about the working of an agent, they not only became more aware of the

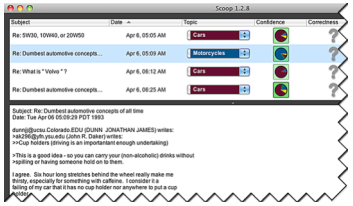

Figure 5: A prototype instantiating our approach for helping end users assess intelligent agents [4].

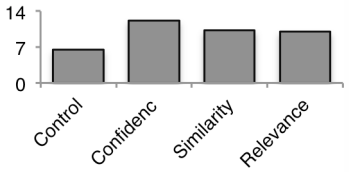

Figure 6: Participants working with our methods for prioritizing the relative importance of assessing each prediction found significantly more of the agent's mistakes than the control group.

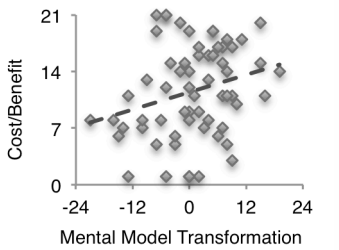

Figure 7: As participants' mental models grew sounder, they increasingly found the benefits of debugging to outweigh the costs.

benefits of debugging its reasoning—they also became more willing to do so (Figure 7) [4].

References

[1] Amershi, S., Fogarty, J., Kapoor, A. and Tan, D. 2010. Examining multiple potential models in end-user interactive concept learning. *Proc. CHI* (2010), 1357–1360.

[2] Kapoor, A., Lee, B., Tan, D. and Horvitz, E. 2010. Interactive optimization for steering machine classification. *Proc. CHI*. (2010), 1343–1352.

[3] Kulesza, T., Burnett, M., Stumpf, S., Wong, W., Das, S., Groce, A., Shinsel, A., Bice, F. and McIntosh, K. 2011. Where are my intelligent assistant's mistakes? A systematic testing approach. *Proc. IS-EUD* (2011), 171–186.

[4] Kulesza, T., Stumpf, S., Burnett, M. and Kwan, I. 2012. Tell me more? The effects of mental model soundness on personalizing an intelligent agent. *Proc. CHI* (2012) (*to appear*).

[5] Kulesza, T., Stumpf, S., Burnett, M., Wong, W.-K., Riche, Y., Moore, T., Oberst, I., Shinsel, A. and McIntosh, K. 2010. Explanatory Debugging: Supporting end-user aebugging of machine-learned programs. *Proc. VL/HCC* (2010), 41–48.

[6] Kulesza, T., Stumpf, S., Wong, W.-K., Burnett, M., Perona, S., Ko, A. and Obsert, I. 2011. Why-Oriented End-User Debugging of Naive Bayes Text Classification. *ACM Transactions on Interactive Intelligent Systems*. 1, 1 (Oct. 2011).

[7] Lim, B. and Dey, A. 2010. Toolkit to support intelligibility in context-aware applications. *Proc. Ubicomp* (2010), 13–22.

[8] Pane, J., Myers, B. and Miller, L. 2002. Using HCI techniques to design a more usable programming system. *Human Centric Computing Languages and Environments*. (2002).

[9] Stumpf, S., Rajaram, V., Li, L., Burnett, M., Dietterich, T., Sullivan, E., Drummond, R. and Herlocker, J. 2007. Toward harnessing user feedback for machine learning. *Proc. IUI* (2007), 82-91.

[10] Stumpf, S., Rajaram, V., Li, L., Wong, W., Burnett, M., Dietterich, T., Sullivan, E. and Herlocker, J. 2009. Interacting meaningfully with machine learning systems: Three experiments. *International Journal of Human-Computer Studies*. 67, 8 (Aug. 2009), 639–662.

[11] Stumpf, S., Sullivan, E., Fitzhenry, E., Oberst, I., Wong, W.-K. and Burnett, M. 2008. Integrating rich user feedback into intelligent user interfaces. *Proc. IUI* (2008), 50–59.