Making intelligent systems understandable and controllable by end users

Simone Stumpf[1], Weng-Keen Wong[2], Margaret Burnett[2], Todd Kulesza[2]

[1] City University London
Centre for HCI Design, School of Informatics
London EC1V 0HB, United Kingdom
Simone.Stumpf.1@city.ac.uk

[2] Oregon State University
School of EECS
Corvallis, Oregon 97333, USA
{wong, burnett, kuleszto}@eecs.oregonstate.edu

ABSTRACT
Pervasive systems for end users are becoming mainstream yet ways to make them transparent and controllable by users are still in their infancy. In this position paper we describe our work with other kinds of intelligent systems to make them intelligible and adaptable by end users. Our results could hold useful lessons for pervasive systems to better support their use.

Categories and Subject Descriptors
H.5.m [Information interfaces and presentation]: Miscellaneous

General Terms
Human Factors

Keywords
Explanatory debugging; intelligent user interfaces; machine learning; personalization; intelligent assistants.

1. INTRODUCTION
Many intelligent systems, such as email inbox filters, object recognition systems, and music recommenders, learn from data to personalize themselves to specific end users. These kinds of adaptations are also found in pervasive systems, such as smart home systems and context-aware mobile applications. Interacting with these systems is, however, currently limited and often uninformative for the end user because of the internal complexity and "black box" nature of most of these pervasive systems. With a few exceptions (e.g., [7]), research into making them transparent and controllable by end users is still in its infancy.

We view the process of end-user interaction with intelligent systems from an *explanatory debugging* perspective [5] (see Figure 1). First, the intelligent system must provide an explanation to the end user, in order for the end user to form a correct mental model of the "source code" and behavior of the system. Second, the end user then provides feedback to the intelligent agent in order to fix the "bugs" in the system. Our work with intelligent systems, specifically text classifiers and recommender systems, could hold lessons for adopting the same approach for pervasive systems.

2. EXPLANATIONS
Our approach rests on the assumption that end users will be better at debugging if they have deeper and better knowledge of how the intelligent system works. Our recent work has shown that, indeed, soundness of mental models impacts end users' ability to efficiently and effectively steer a system's behavior and their perceptions of benefit, satisfaction and user experience [3]. The mental models of end users in this study were shaped through brief scaffolded instruction sessions that explained how the system, a music recommender, worked "underneath the hood". However, it may be feasible to build this instructional ability into intelligent systems, in order for them to explain themselves better.

Explaining intelligent systems is challenging because at their heart are complex statistical machine learning algorithms, which even experts in machine learning find hard to understand [1]. To help address this problem, we have been exploring how different kinds of explanations impact end users' reasoning.

First, we looked at the understandability of three explanation approaches for text classifiers [10]. We compared Keyword-based, Similarity-based and Rule-based explanations. Our results indicate that, while there was not one perfect way to explain the behavior, Rule-based and Keyword-based approaches were easier to understand compared with similarity-based explanation mechanisms, and they also led to end users being able to understand the behavior more correctly. Factors that played a part in understanding were the perceived soundness of the reasoning and how this reasoning was communicated.

We have also explored what specific elements need to be explained. We adapted the Whyline approach [2] for a naïve Bayes text classifier to provide explanations about different elements of an intelligent system's reasoning and a visual explanation of keyword weights [6, 4]. We found that participants had particular problems with understanding *where* to make changes and how these changes would *affect* other parts of the system.

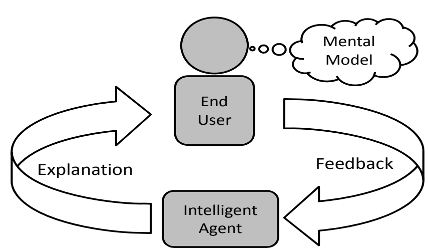

Figure 1. An "explanatory debugging" perspective of end-user interaction with an intelligent agent. Interaction consists of two parts: 1) the agent provides an explanation to the end user who then 2) provides corrective feedback to the agent.

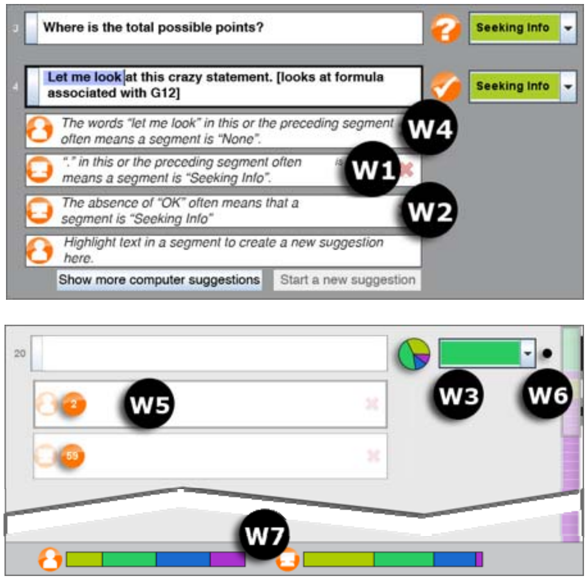

Figure 2: Explanations: Machine-generated explanation (W1); Absence explanation (W2); Prediction confidence (W3); User-generated suggestion (W4); Impact count (W5); Change history markers (W6); Popularity bar (W7).

We then investigated the benefits of explaining the machine's reasoning versus its runtime behaviors to make debugging choices more transparent to end users (Figure 2). We found that providing support for explaining "run-time" behavior (widgets 3, 4 and 7) had a significantly positive impact on both end users' effectiveness of debugging and their attitude toward the system [5].

3. CORRECTIVE FEEDBACK

Along with explaining, we have also made some headway helping end users with debugging itself. A simple way to "debug" (fix) a statistical machine learning system is to provide more training examples from which it can learn. However, this approach is time consuming and we have investigated alternatives to give end users more control so that these types of systems quickly heed their feedback.

First, we investigated what types of corrective feedback end users would like to give to machine learning systems [10]. The results were a wide variety of feedback, including regarding reweighting features, creating new features (such as by combining features or creating features based on relational information) and even wholesale changes to algorithms.

We then focused on incorporating feature reweighting feedback into learning algorithms. One approach for feature reweighting is to present a user with a visual explanation of an algorithm's prediction and then allow the user to modify the weights of the features through this visual explanation [6]. In our experiments with naïve Bayes, we found that this approach was difficult for end users because of its insensitivity to change; reweighting a small subset of features among thousands of features produced little or no change to the original prediction. A more successful approach, which we called user co-training, uses a co-training framework in which the user's feedback is treated as if it were a second classifier [8]. However, user co-training can suffer from an unstable period early in the training that can frustrate users [9].

Last, we developed an approach based on locally-weighted logistic regression which allows end users to label *features* rather than *instances* [11]. This algorithm has shown promise in both simulated studies and studies involving actual end users.

4. CONCLUSION

We have explored how end users who are not trained in software engineering or machine learning could better interact with and fix their intelligent systems. Our work has focused on both providing explanations of how these systems work as well as guiding end users to make informed choices when debugging a system. We have also worked on making the system heed the end user when they attempt to debug it. We believe that our work could hold important lessons for the intelligibility and control of pervasive systems, which are built on similar machine learning foundations.

5. ACKNOWLEDGMENTS

This work has been in part supported by NSF grant 0803487.

6. REFERENCES

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

[2] Ko, A. and Myers, B. Designing the Whyline: A debugging interface for asking questions about program behavior. *Proc. CHI*, ACM (2004), 151-158.

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

[4] Kulesza, T., Stumpf, S., Wong, W.-K., Burnett, M., Perona, S., Ko, A., Oberst, I. Why-oriented end-user debugging of Naïve Bayes text classification. *Transactions on Interactive Intelligent Systems* 1(1), ACM (2011).

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

[6] Kulesza, T., Wong, W.-K., Stumpf, S., Perona, S., White, R., Burnett, M., Oberst, I., Ko, A. Fixing the program my computer learned: Barriers for end users, challenges for the machine. *Proc. IUI*, ACM (2009).

[7] Lim, B.Y., Dey, A.K., Avrahami, D. Why and why not explanations improve the intelligibility of context-aware intelligent systems. *Proc. CHI*, ACM (2009), 2119–28.

[8] Stumpf, S., Rajaram, V., Li., L., Burnett, M., Wong, W.-K., Dietterich, T., Sullivan, E., Drummond, R., Herlocker, J. Interacting meaningfully with machine learning systems: Three experiments. *IJHCS* 67(8), (2009), 639-662.

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

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

[11] Wong, W.-K., Oberst, I., Das, S., Moore, T., Stumpf, S., McIntosh, K., Burnett, M. End-user feature labeling: A locally-weighted regression approach. *Proc. IUI*, ACM (2011).