Explaining smart systems to encourage (or discourage?) user interaction

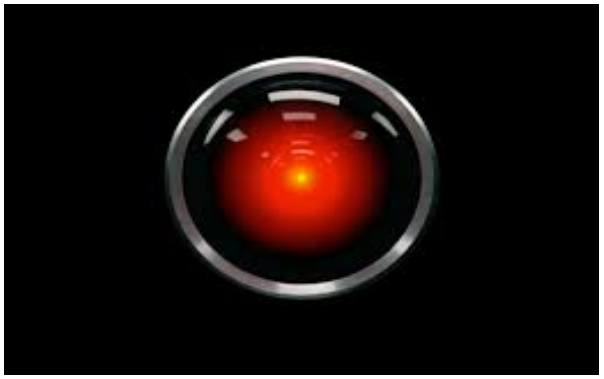
Dr Simone Stumpf
Centre for HCI Design

Simone.Stumpf.1@city.ac.uk
@DrSimoneStumpf

Bio

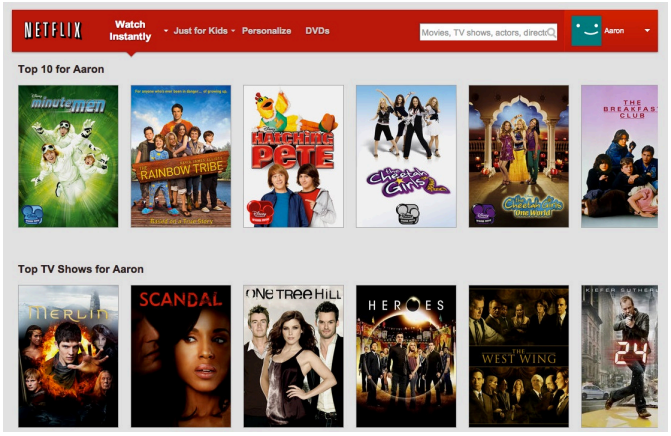
- End-user interactions with AI – explanations, user experience and trust
- Projects
  - Researcher on DARPA-funded ”CALO” project to use machine learning system to track and suggest appropriate task-based resources
  - Co-I on NSF-funded “End-user debugging of machine-learned programs“ project
  - PI for the “FREEDOM” project on smart heating system UIs
  - Co-I on EPSRC ”SCAMPI” project to develop a smart home technology for self-managing quality of life plans for people with dementia and Parkinson’s

AI?



Gmail
PRIORITY INBOX

Welcome
John


NETFLIX
Watch Instantly
Just for Kids
Personalize
DVDs
Top 10 for Aaron
Top TV Shows for Aaron
MERLIN
SCANDAL
ONE TREE HILL
HEROES
THE WEST WING
24

nest
HEATING
20

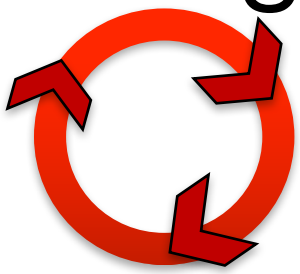
The old AI is not the new AI

- "Old" AI worked mainly on rule-based inferences that were "extracted" by a knowledge engineer e.g. expert systems

- "New" AI is typically based on machine learning using complex statistical inferences e.g. SVMs, deep learning
  - Usually system learns a function or weights from (large) data sets so it can provide some appropriate output

$$
\begin{aligned}
P(y|f_1,\ldots,f_m) &= \frac{P(f_1,\ldots,f_m|y)P(y)}{P(f_1,\ldots,f_m)} \\
&= \frac{P(y)\prod_{i=1}^{m} P(f_i|y)}{P(f_1,\ldots,f_m)} \\
\arg\max_y P(y|f_1,\ldots,f_m) &= \arg\max_y \frac{P(y)\prod_{i=1}^{m} P(f_i|y)}{P(f_1,\ldots,f_m)} \\
&= \arg\max_y P(y)\prod_{i=1}^{m} P(f_i|y)
\end{aligned}
$$

HCI issues for AI

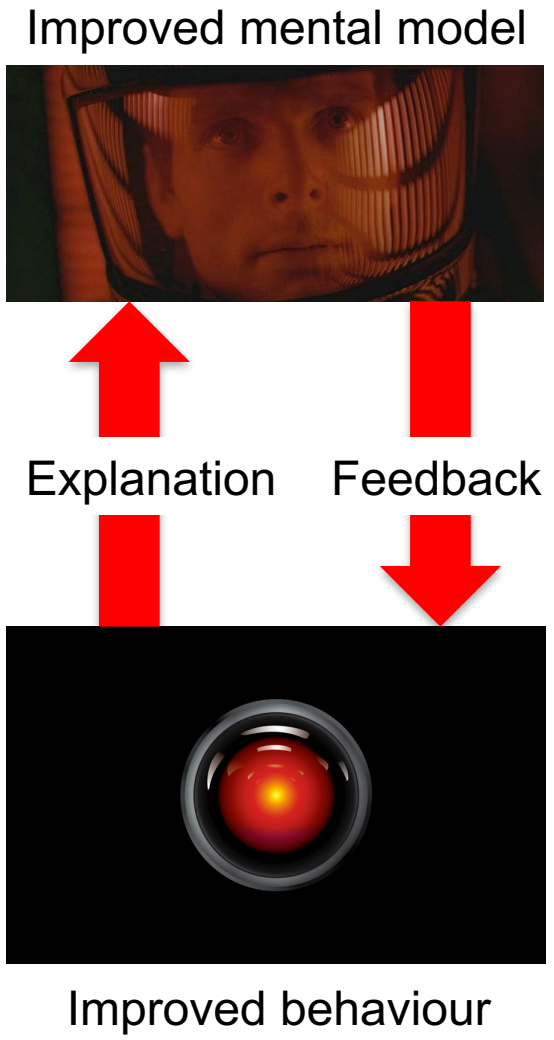


User Experience
Intelligibility
Controllability

- “Black boxes” don’t communicate how they work
- “Magic” to users so have very poor mental models of what is going on
- What happens if AI goes wrong?

Explanatory debugging [Kulesza et al. 2015]

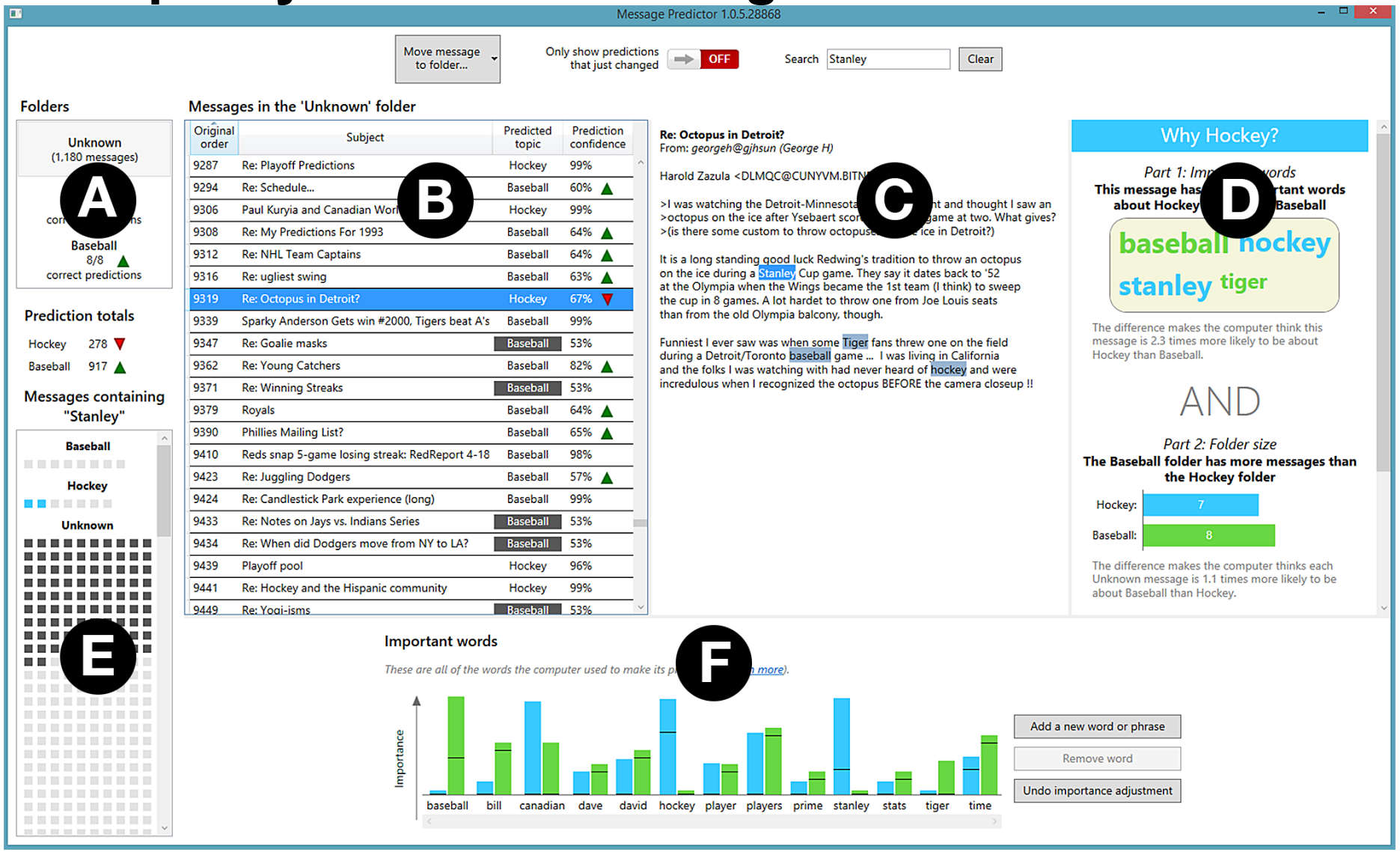
- Debugging is trying to identify and correct mistakes in a system's program = **controllability**
- Explanation-centric approach to help end users effectively and efficiently personalize machine learning systems = **intelligibility**
- *Note: explanations can be words or pictures*



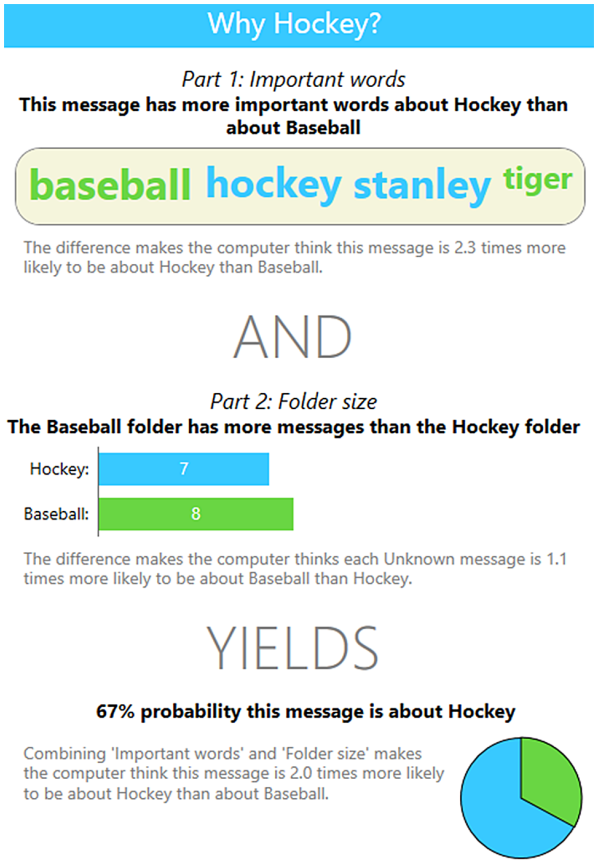
Principles of explanatory debugging

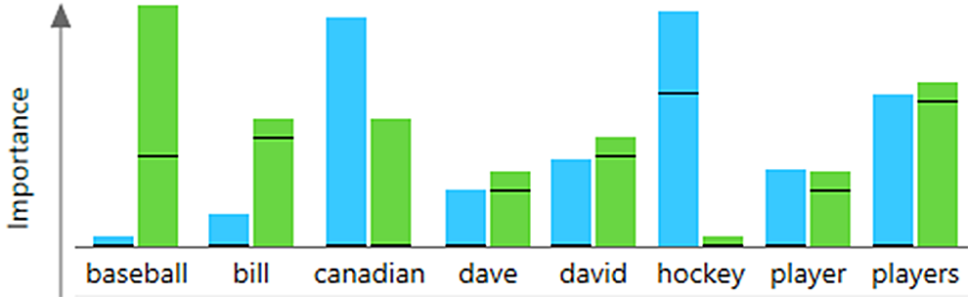
- Explainability
  - **Be iterative**: small, consumable bites as users interact
  - **Be sound**: truthful to how the system works to build trust
  - **Be complete**: include as much as possible what it uses
  - **But don't overwhelm**: tradeoff between soundness, completeness and attention
- Correctability
  - **Be actionable**: users can modify the system based on the explanations
  - **Be reversible**: users may make it worse and need a way to back out
  - **Always honour feedback**: don't disregard what the user tells the system
  - **Incremental changes matter**: need to build up system changes iteratively

Example system: EluciDebug



Explanability and correctability

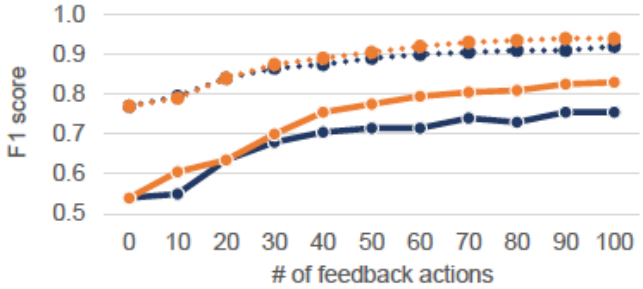




Evaluation study

- 77 participants split into two groups: 40 using EluciDebug, 37 using a version without explanations and advanced feedback
- 20 Newsgroup data set (Hockey and Baseball): initial system training on 5 messages for each subject, 1850 unlabeled messages to sort
- 30 minutes to “make the system as accurate as possible”
- Measures: accuracy, feedback given, mental model scores, perceived workload

Results

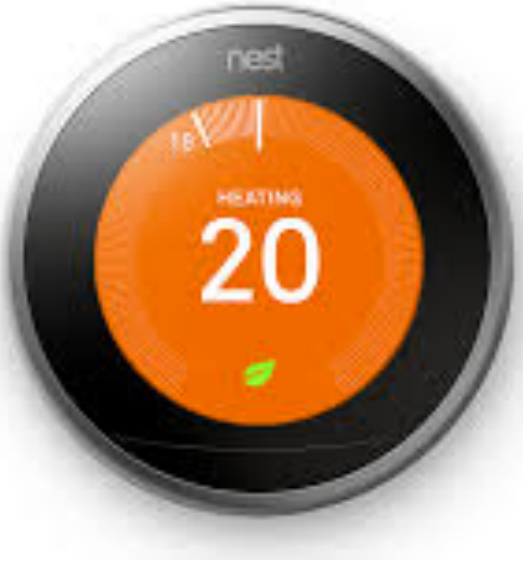


- **Better accuracy**
  - 85% for EluciDebug versus 77% at end of study
- **With less feedback**
  - On average, they added 34.5 new features, removed 8.2 features, made 18.3 feature adjustment and interacted with 47 messages (while control users had to label an average of 182 messages)
  - No difference in workload
- **With better understanding**
  - 15.8 mental model score versus 10.4
  - Correlation between mental model score and system accuracy at study end (i.e. the more you understand the better you can make it)

Not all systems are the same

- Not all systems should demand high user interactions
  - “Calm technology” [Weiser and Brown 1997]
  - “Constrained engagement” between the system and user [Yang and Newman 2013]

- Not all explanations increase users correcting the system [Bussone et al. 2015]
  - Provide explanations to discourage user interactions?



Traditional heating systems

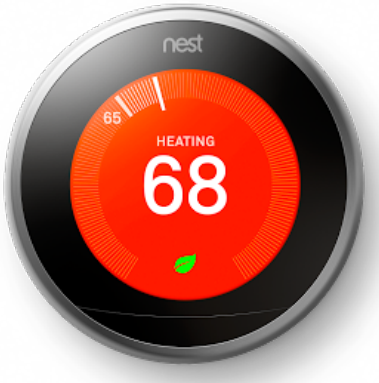
user controls heat

makes heat

emits heat

smart heating systems

system controls heat

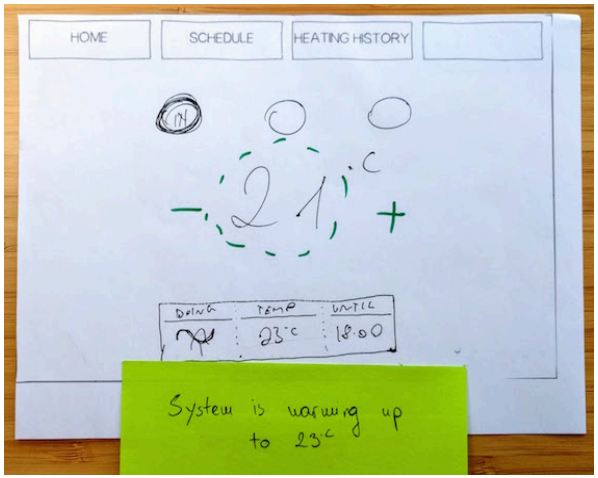


emits heat

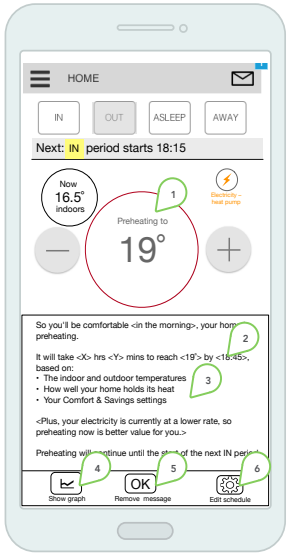
makes heat

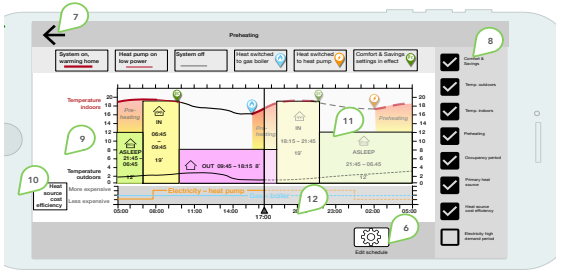
How to discourage user interactions

- Study on what users want to know [Skrebe and Stumpf 2017]
  - ”Unexpected behaviour” e.g. preheating, demand response, overshooting temp, etc
  - Detailed reasons for decisions and benefits
  - Notify before action is taken (ideally) for added control



- Design of UI
  - Text first, graph on demand
  - Explanations fit with Explanatory Debugging (mostly)





Explanation evaluation study

- 60 participants with simulated home heating scenarios
- 4 conditions
  - Control: no explanation, normal heating UI only
  - Text-only: written explanation given, no graphical element
  - Graphical-only: graphical explanation given, no written element
  - Both: full explanation given, consisting of both graphical and written elements

- Explanations increased understanding
- Explanations did not increase trust in the system
  - *Without* explanations were more concerned about predictability of system actions
  - With *any* of explanations more focused on whether system was doing the right thing

Future work

- Evaluation of smart heating UIs in field trial
- Publication on FREEDOM designs and evaluations
- Workshop on explainable smart systems (ExSS) at IUI 2018
- Research and dissemination of SCAMPI project results

Big questions:

How to design with AI as a new material?

How can we craft better explanations for use in different AI systems?

What are the impacts of explanations and user interactions with AI?