Evaluation of an Integrated Multi-Task Machine Learning System with Humans in the Loop

Aaron Steinfeld, S. Rachael Bennett, Kyle Cunningham, Matt Lahut,
Pablo-Alejandro Quinones, Django Wexler, Dan Siewiorek*
Jordan Hayes†, Paul Cohen‡, Julie Fitzgerald**, Othar Hansson†, Mike Pool††,
Mark Drummond‡‡
* School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA
† Bitway, Inc; ‡U. of Southern California; **JSF Consulting; ††IET; ‡‡SRI International
Correspondence: steinfeld@cmu.edu & radar@bitway.com

***Abstract*—** Performance of a cognitive personal assistant, RADAR, consisting of multiple machine learning components, natural language processing, and optimization was examined with a test explicitly developed to measure the impact of integrated machine learning when used by a human user in a real world setting. Three conditions (conventional tools, Radar without learning, and Radar with learning) were evaluated in a large-scale, between-subjects study. The study revealed that integrated machine learning does produce a positive impact on overall performance. This paper also discusses how specific machine learning components contributed to human-system performance.

***Keywords*:** *machine learning, intelligent systems, mixed-initiative assistants, evaluation*

I. Introduction

The RADAR (Reflective Agents with Distributed Adaptive Reasoning) project within the DARPA PAL (Personalized Assistant that Learns) program is centered on research and development towards a personal cognitive assistant. The underlying scientific advances within the project are predominantly within the realm of integrated machine learning (ML). These ML approaches are varied and the resulting technologies are diverse. As such, the integration result of this research effort, a system called Radar, is a multi-task machine learning system.

Annual evaluation on the integrated system is a major theme for the RADAR project, and the PAL program as a whole. Furthermore, there is an explicit directive to keep the test consistent throughout the program. As such, considerable effort was devoted towards designing, implementing, and executing the evaluation. This paper summarizes efforts to validate the hypothesized beneficial impact of the integrated machine learning present in Radar.

It is also important to note that the RADAR project differs from the bulk of its predecessors in that humans are in the loop for both the learning and evaluation steps. Radar was trained by junior members of the team who were largely unfamiliar with the underlying ML methods. Generic human subjects were then recruited to use Radar while handling a simulated crisis in a conference planning domain. This allowed concrete measurement of human-ML system performance. It is important to consider personal assistance systems in the context of human use due to their inherent purpose.

There have been past attempts at creating digital assistants to aid users in the performance of complex activities. Possibly the most memorable and infamous example of these is the animated paperclip accompanying Microsoft Word. Agents such as these are usually most valuable to a novice, as opposed to an experienced user.

On the opposite end of the spectrum of assistants, we can find those that are human. While human assistants are malleable, intuitive, accommodating, and are able to expand their knowledge, they lack certain characteristics present in an ideal digital assistant. Humans assistants lack perfect recall, incur latencies on time critical tasks, cannot rapidly compute optimizations and execute other taxing algorithms, are more susceptible to periodic performance losses due to turnover and constrained availability, and cannot operate continuously. Furthermore, human assistants do not scale well – providing an assistant to every human in an organization is cost prohibitive on several metrics.

Radar is an attempt to achieve the best of both worlds by focusing on a cognitive digital assistant. The presence of learning is the main distinction when using the prefix "cognitive." The knowledge it obtains can be used to automate and prep tasks, thus providing the assistance of a human without the limitations of a human and making digital assistance more adaptable and suitable for the user.

A. The Radar System

Radar is specifically designed to assist with a suite of white-collar tasks. In most cases, the specific technologies are designed to be domain agnostic (e.g., email categorizing, resource scrounging, etc). However, for the purposes of the evaluation, the base data present in Radar and used for learning is centric to the domain of conference planning. As such, certain components appear to be domain-specific but their underlying technologies are more extensible (e.g.,

Table 1. Radar components

LITW	Component	Capability
X	CMRadar-Rooms (Room Finder)	Resource scrounging by learning room reservation owner behaviors
X	Email Classifier	Task-oriented label assignment to email messages based on prior activity
X	Space-Time Planner (STP)	Elicitation of facts about the world in order to do better optimizations
X	Virtual Information Officer (VIO)	Classification and extraction to assist information updates on websites
X	Workflow by Example (WbE)	Batch website updates from training on input files
	Annotations Database (AnnoDB)	Email parsing and related natural language processing
	Scone	Knowledge representation support for the AnnoDB
X	Briefing Assistant (BA)	Summarization of activity based on prior activity *(Note: not deployed)*

conference-related email categories, room finding, etc).

While evaluation testing was performed on several Radar 1.x versions, they generally contained the same machine learning components (Table 1). The major variations were due to engineering and user interaction improvements in a number of components and the removal of the Briefing Assistant for engineering reasons. Again, the individual ML technologies will not be described in detail here – the focus here is to show that such integrated systems can provide real benefit and evaluation can be accomplished in a manner robust to unforeseen synergies and use.

An important distinction is whether a ML component "learns in the wild" or requires special interaction to gain knowledge. Learning in the wild (LITW) is a primary mission of the RADAR project and is specific to learning that occurs through the course of daily use. Brute force spoon-feeding and code-driven knowledge representation is not LITW. To count as LITW, learning must occur through regular user interaction and user interfaces present in Radar.

An example of brute force encoding would be asking someone to copy the campus building specifications into Radar all at once. However, learning is LITW if Radar decides knowing the capacity of a certain room is really important, Radar asks the user for the capacity, and the user looks it up and enters the specific value.

Table 1 details which components in Radar 1.1 were LITW and what their specific assistance entails. Note that this list is continuously growing and more components are expected in the next major release of Radar. Likewise, the next release is expected to include tighter integration between ML components. Additional detail on Radar components and capabilities is deferred to other papers.

B. Test Conditions and Hypotheses

In order to show the specific influence of learning on overall performance, there were two Radar conditions – one with learning (+L) and one without (-L). In the context of the evaluation test, learning was only LITW. Learning acquired through knowledge engineering by a programmer or through brute force encoding would be available in both the +L and -L Radar conditions.

To the user, Radar was essentially a system layered into Outlook. The components in Table 1 are either behind the scenes (e.g., Scone, AnnoDB) or visible as modified Outlook views (e.g., Email Classifier, VIO) or separate windows (e.g., STP). In many ways, the user interaction development aspect of Radar lagged behind the learning components. This was largely due to limitations in Outlook and user interaction will be improved in the next version of Radar.

A third condition where subjects utilize conventional off the shelf tools (COTS) allowed estimates to be made on the overall benefit of integration, optimization, engineered knowledge, and improvements in user interaction as compared to the current state of the art. For this application, this toolset consisted of an unaltered version of Outlook, the schedule in an Excel spreadsheet instead of the STP, a web portal to the room reservation system, and the conference website which could be manually updated.

The primary mission of the evaluation test was to examine two top-level hypotheses. These were:

1. Radar with learning (+L) will do better than Radar without learning (-L)
2. Radar will do better than conventional tools (COTS)

The comparison in Hypothesis 1 is commonly called the Learning Delta. Additional hypotheses, detail on methods, and findings can be found in [1].

C. Related Work

As previously mentioned, this was a multi-task ML system and therefore required a complex scenario for rigorous evaluation. Unfortunately, research utilizing human subjects to evaluate multi-task cognitive digital assistants with demanding tasks of this nature is limited, and so few comparison cases are available.

Furthermore, evaluations of ML systems are largely based on simulation (e.g., [2, 3]), comparison to traditional methods (e.g., [4]), subject judgments on system performance (e.g., [5]), or have sparse details on human subject evaluation (e.g., [6]). It is quite possible that this is generally the result of the kind of system that is built – something that is not meant to be an assistant but, rather, is designed to perform a task that has specific rules. An assistance system, when designed and evaluated, should be tested with humans in the loop (e.g., [7]).

As far as the rest of literature is concerned, there is relatively little literature on evaluation results of cognitive digital assistants and their focus tends to be specific to a

narrow range of learning (e.g., [8, 9]). This may be because most of assistants of this nature are design exercises, lack resources for comprehensive evaluation, not evaluated with humans in the loop, and/or proprietary and unpublished.

II. Method and Materials

A key requirement for the annual evaluation test was repeatability and a consistent level of difficulty so that performance improvements can be measured across years. At a fundamental level, this is nearly impossible to achieve in a complex test of this nature. As such, the goal was to start with a test scenario that was challenging enough to accommodate synergistic learning effects, component advances, and new research directions for the out-years. A common condition, working the problem with conventional off the shelf tools (COTS) is run for each test, thus permitting benchmarking of small changes to the protocol and each test's stimulus package (e.g., specific crisis, additional tasks, etc). Furthermore, the stimulus package for the test is bound by parameters that are broad enough to prevent training to the test, but narrow enough to ensure that the stimulus package will measure the ML technologies present in the version of Radar being tested.

As mentioned, this is a system consisting of Radar and a human. At a high level this means that human subjects may need, or be required, to perform specific tasks manually. The utilization of a COTS condition where there are no Radar tools makes the ability for full manual execution a requirement. This nuance also allows for tasks and stimuli that are currently difficult for strictly software tools to complete autonomously – mixed effort towards task completion is perfectly acceptable and expected. Removal of manual control can occur if Radar technology replaces the manual inputs. For example, a user interface that allows subjects to manually scrounge for resources can be removed if a Radar component can be used to perform this task.

A. Storyline and Simulated World

The general scenario for the evaluation was that the subject was filling in for a conference planner who was indisposed, to resolve a crisis in the current conference plan. This crisis was major enough to require a major shuffling of the conference schedule and room assignments that, in turn, triggered secondary tasks. These included supporting plans (e.g., shifting catering, AV equipment delivery, adjusting room configuration, etc), reporting (e.g., make changes to the website, issue a daily briefing, etc), and customer handling (e.g., "here is the campus map"). Noise stimuli were also present in the form of unrelated email, unusable rooms, unrelated web pages, and other clutter content.

The materials included an email corpus and simulated world content. The need for repeatability over time led to the requirement for a simulated world. This consisted of facts about the world (e.g., characteristics of a particular room) and conference (e.g., characteristics of each event).

The simulated world and the initial conference were designed to provide clear boundaries on the types of tasks subjects would need to complete, yet also permit large-scale information gathering, precise measurement of learned facts, and the opportunity to induce a substantial crisis workload. The conference itself was a 4-day, multi-track technical conference complete with social events, an exhibit hall, poster sessions, tutorials, workshops, plenary talks, and a keynote address. The conference was populated with over 130 talks/posters, each with a designated speaker and title. All characters were provided with email addresses and phone numbers. Many were also given fax numbers, website addresses, and organizations.

The physical space was a modification and extension of the local university campus. In addition to modifying the student union, two academic buildings and a hotel were created and populated. These latter three buildings were instantiated to protect against campus entry knowledge in the subject pool. This information was presented to the subject in the form of revised university web pages easily accessible from the subject's home page.

Other static web content included a conference planning manual (complete with documentation of standard task constraints), a read-only file with the original schedule, and manuals for the tools used by the subjects.

Subjects were also given access to a working, realistic "university approved" vendor portal where goods and services could be ordered for the conference. These included audio-visual equipment, catering, security, floral arrangements, and general equipment rentals. Email receipts, complete with computed prices and hyperlinks to modification/cancellation pages, were delivered to the subject's mail client in real time. All vendor interactions were via web forms since automatic or Wizard of Oz handling of subject e-mails can lead to problems with stimulus consistency and realism. This had face validity since many real-life counterparts are web-based, including the subject signup website used during recruitment.

The corpus initialization for each experiment included:

- The predecessor's conference plan in the file format of the condition toolset
- Other world state information – e.g., room reservation schedule, web pages detailing room characteristics, etc. (Figure 1, top and middle)
- The vendor portal, loaded with the initial orders (Figure 1, bottom)
- Stored e-mail from the original conference planner, including noise messages and initial vendor orders
- Injected e-mail, including details of the crisis, new tasks, and noise (e.g., Table 2)

Cost is a major barrier for experimental research and a large portion is attributable to stimuli and artifact development. We have made the commitment to provide much of the stimuli and supporting content described here to external parties for re-use. This occurs through the Airspace website [10].

B. Email Corpus

The email corpus was constructed but occasionally utilized anonymized real content where appropriate (e.g., noise messages). There were initial attempts to acquire an existing email corpus centric to a conference planning activity but this posed significant challenges in the realm of Institutional Review Board (IRB) approval due to the need to anonymize all content – including subtle cues that would reveal identities. Prior attempts within the project to perform such a step produced haphazard results where entity anonymization was not sufficient.

Even a real conference planning email corpus free of IRB constraints would not be entirely adequate. A real corpus would still require considerable alignment with a simulated world (e.g., websites, rooms, etc.) and would not necessarily match the ML technologies present in the system. For example, the corpus for the real conference may completely lack website update tasks and focus heavily on what local tours to include in the registration packet.

This early investigation led to the determination that the corpus should be fabricated with an eye towards realism and the ML being tested. A team of undergraduate English majors was employed to create a detailed backstory corpus, independent messages detailing one or more tasks, and noise messages. The students were given a series of story arcs, guidelines, and a handful of characters with some specific assigned personalities (e.g., formal, annoying, etc). This effort included a directive to the email authors to let natural errors occur in their writing (e.g., signal message in Table 2). Some characters were assigned personality types that would also lead to different writing styles and email body structure (e.g., terse, bad spelling, etc). Other directives included the utilization of event, paper, and room descriptor variations (e.g., "Dowd in Stever"). Resulting content was screened for fit to the specifications, alignment with world facts, and template syntax adherence.

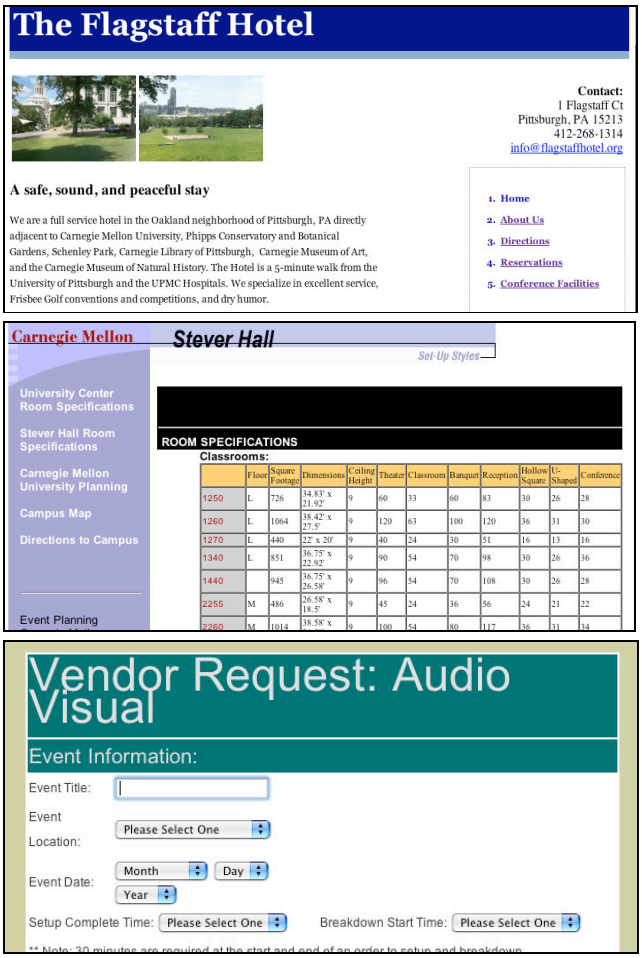

Figure 1. Static web and vendor portal examples

All email corpus content was in a structure which supported date shifting and variable substitution (e.g., Table 2, sender of the noise message). Date anchors and variables were stored in a separate file. These allowed for easy modification of key

Table 2. Sample messages

Signal Message	Noise Message
From: jpsontag@ardra.org To: bor@cs.cmu.edu Subject: Lucia di Lamermoor I hate to be a pest, but I finally got tickets to the opera, Lucia di Lamermoor for my wife on our aniversary. It is wednesday night. I want the whole day to ourselves, so I can avoid crashing out plans, that would be great! Let me know. The other days are fine. Thank! J.P.	From: var="kimMail" To: bor@cs.cmu.edu Subject: Hey Uncle Blake! I have a favor to ask you--Mom and Dad's anniversary is coming up, and I wanted to do something special for them, especially since they've been so supportive of the whole wedding concept. I was thinking about getting them tickets to go see "The Phantom of the Opera" when the Broadway Series came to Pittsburgh. I know that sometimes you can get cheaper tickets through work, so I was wondering if that was possible for this show. Please let me know asap so that I can make arrangements! Thanks, you're the best! Kim

values by the external program evaluators and time shifting of the corpus for experiment execution.

C. Objective Performance Measurement

As experiment-friendly conference planning performance measures are not readily available, a new method was utilized. It was extremely important that this measurement be tied to objective conference planning performance rather than a technology-specific algorithm (e.g., F1 for classification). This technology agnostic approach also permits accurate measurement of component synergies and human use strategies.

Creation of this measurement was largely achieved through an evaluation score designed and developed by the external program evaluators (authors JF, MP, and PC). This complex score function summarized overall performance into a single objective score ("Final_Score" range from 0.000 to 1.000). Performance was in terms of points collected by satisfying certain conditions coupled with penalties for specific costs. These included quality of conference schedule (e.g., constraints met, special requests handled, etc), adequate briefing to conference chair, accurate adjustment of the website (e.g., contact information changes, updating the schedule on the website, etc), and costs incurred while developing schedule. Such costs included both the budget and how often subjects asked fictional characters to give up their room reservations. Additional detail on scoring is deferred to other documents. At the top level, the score coefficients were 2/3rd for the schedule (including penalties for costs incurred), 1/6th for website updating, and 1/6th for briefing quality.

In addition to this measure, subjects also completed a post-test survey designed to measure perception of system benefit, assistance, and other related metrics. Details on the survey design and results are reported elsewhere [11].

D. Procedure

Each subject was run through approximately 3 hours of testing (1 for subject training and 2 for time on task). Each cohort of subjects for a particular session was run on a single condition (COTS, Radar -L, or Radar +L). When possible, cohorts were balanced over the week and time of day to prevent session start time bias. Follow-up analyses on this issue revealed no apparent bias. The nominal cohort size was 15 but was often lower due to dropouts, no-shows, and other subject losses (e.g., catastrophic software crash). Cohorts were run as needed to achieve approximately 30 subjects per condition.

Motivation was handled through supplemental payments for milestone completion (e.g., the conference plan at the end of the session satisfies the constraints provided). Subjects were given general milestone descriptions but not explicit targets. These milestones roughly corresponded to the top-level coefficients in the score function.

III. Results

A. Data Source for this Example

There were several test windows during the run-up to the data shown here. This corresponds to COTS and Radar 1.1 tested with a stimulus package of 107 messages, 42 of which were noise.

The crisis for this package was a loss of the bulk of the conference rooms for 1.5 days (out of 4 total). A variety of other small perturbations rounded out the task set. These

Table 3. RADAR 1.1 means and t-test comparisons

Condition	Mean	Comparison	p-value
COTS	0.452	Overall Delta (With Learning > COTS)	<0.0001
No Learning (-L)	0.492	Learning Delta (With Learning > No Learning)	<0.0001
With Learning (+L)	0.605	Nonlearning Delta (No Learning > COTS)	<0.041

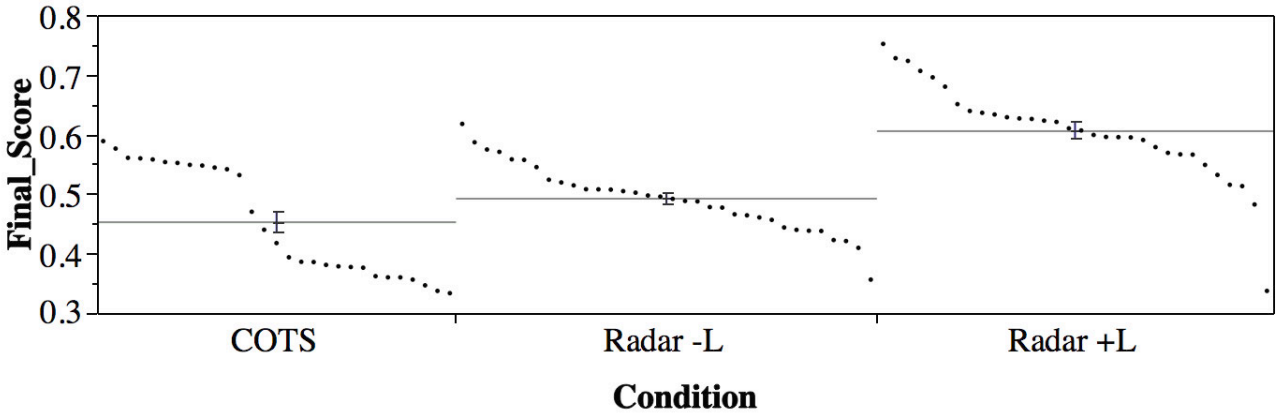

Figure 2. Radar 1.1 results on Crisis 1 (Score 2.0)

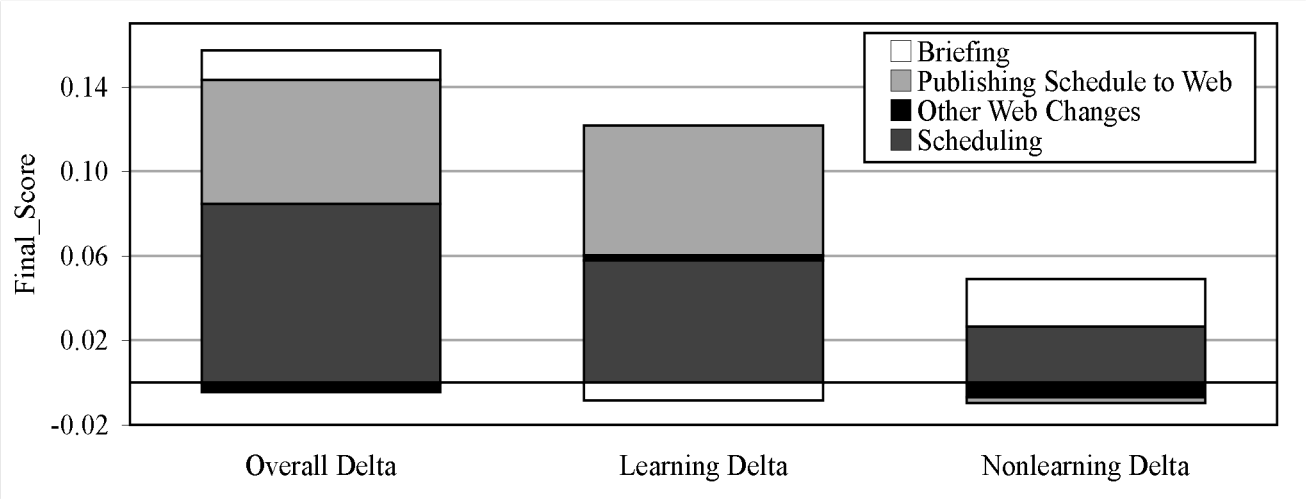

Figure 3 Score component impacts on the overall score (Score 2.0)

included changes to room details, speaker availability, session preferences, and website details. This stimulus package (aka Crisis 1) was designed by the external evaluators. As of this paper, the external evaluators have designed three different crisis packages.

The subject pool used for analysis, after exclusions and dropouts, was 29, 34, and 32 (COTS, Radar -L, and Radar +L). As such, this test accumulated 64 cumulative hours worth of time on task by subjects with a multi-ML system.

Scheduling and scoring for the conditions shown here was not in parallel. COTS data was collected in the fall of 2005 and the Radar data was collected in the spring of 2006. The data described here were scored with version 2.0 of the external evaluator's scoring algorithms (aka Score 2.0).

B. Final_Score Results

Figure 2 shows between subject performance across the three conditions. The Learning Delta (the difference due to the inclusion of machine learning) is 0.113, which is approximately 74% of the Overall Delta (improvement over COTS). This suggests that machine learning was the prime contributor to the performance gains. In this graph, all condition differences are significant and in the expected direction for the initial hypotheses (Table 3).

The need for an integrated evaluation with humans in the loop becomes especially apparent when examining the makeup of the Deltas (Figure 3). Subjects noticeably altered their strategies and use of assistance technology based on the presence/absence of specific features. For example, COTS subjects clearly focused on updating individual website corrections (e.g., "my name is spelled wrong") over other activities – probably due to familiarity with website form manipulations. Likewise, subjects in the Radar conditions took full advantage of autonomous components to relieve time pressure (i.e., schedule optimizer in both -L and +L, batch website updating in +L, etc).

Table 4. Learning contributors to score component

Score Component	Learning Contributors
Scheduling	STP, CMRadar-Rooms, Email Classifier
Publishing Schedule to Web	WbE
Other Web Changes	VIO, WbE, Email Classifier
Briefing	Email Classifier

Gains due to publishing the schedule to the website can be tied explicitly back to WbE, but is not the only place where WbE can contribute/detract from overall performance (Table 4). Note that while the Email Classifier contributes to many factors of the score function, its role is to surface the task and not to assist with the completion of the task itself. As such, the negative Learning Delta for the briefing component (Figure 3) is not solely due to a deficiency of the Email Classifier. In fact, this difference is due to human decision making related to task allocation – almost twice as many subjects in the nonlearning condition as in the learning condition compiled a briefing (56% vs. 28%). Task identification is not the same as task prioritization, hence the importance of an overall task performance measurement.

IV. Discussion

The results clearly show that Hypothesis 1 (ML helps) holds true. Likewise, Hypothesis 2 (Radar is better than COTS) is also true. Furthermore, it is clear that component value was highly dependent on how subjects allocated effort – some technologies were underutilized based on strategic decisions.

The initial concern at the start of this endeavor was that the methods and materials would not be adequately sensitive to measure mixtures of ML technologies that were still being formulated. This concern is still valid in that there are new ML components being developed for the next version of

Radar. The decision to measure at the top human-Radar system level was an attempt to be robust to unknown ML technologies. While this limits the ability to directly account for specific component benefit, this approach clearly captures high-level benefits and use patterns for human in the loop multi-task ML.

While not shown here, there have been other human subjects tests with other versions of the system and the protocol. These have shown changes in performance due to variations in ML, HCI, engineering, crisis difficulty, and human training. As such, the test method and materials have also been shown to be suitable for measuring shifts in performance due to a variety of system and scenario effects.

ACKNOWLEDGMENTS

Othar Hansson and Mike Pool joined Google and Convera (respectively) after contributing to this work.

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. NBCHD030010. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the DARPA or the Department of Interior-National Business Center (DOI-NBC).

REFERENCES

[1] Steinfeld, A., Bennett, R., Cunningham, K., Lahut, M., Quinones, P.-A., Wexler, D., Siewiorek, D., Cohen, P., Fitzgerald, J., Hansson, O., Hayes, J., Pool, M., and Drummond, M., The RADAR Test Methodology: Evaluating a Multi-Task Machine Learning System with Humans in the Loop. 2006, Carnegie Mellon University, School of Computer Science: Pittsburgh, PA. http://reports-archive.adm.cs.cmu.edu/anon/2006/abstracts/06-125.html

[2] Clymer, J. R. Simulation of a vehicle traffic control network using a fuzzy classifier system. In Proc. of the IEEE Simulation Symposium. 2002.

[3] Clymer, J. R. and Harrsion, V. Simulation of air traffic control at a VFR airport using OpEMCSS. In Proc. IEEE Digital Avionics Systems Conference. 2002.

[4] Zhang, L., Samaras, D., Tomasi, D., Volkow, N., and Goldstein, R. Machine learning for clinical diagnosis from functional magnetic resonance imaging. In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2005.

[5] Hu, Y., Li, H., Cao, Y., Meyerzon, D., and Zheng, Q. Automatic extraction of titles from general documents using machine learning. In Proc. of ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL). 2005.

[6] Allen, J., Chambers, N., Ferguson, G., Galescu, L., Jung, H., Swift, M., and Taysom, W. PLOW: A Collaborative Task Learning Agent. In Proc. Conference on Artificial Intelligence (AAAI). 2007. Vancouver, Canada.

[7] Schrag, R., Pool, M., Chaudhri, V., Kahlert, R., Powers, J., Cohen, P., Fitzgerald, J., and Mishra, S. Experimental evaluation of subject matter expert-oriented knowledge base authoring tools. In Proc. NIST Performance Metrics for Intelligent Systems Workshop. 2002. http://www.iet.com/Projects/RKF/PerMIS02.doc

[8] Shen, J., Li, L., Dietterich, T. G., and Herlocker, J. L. A hybrid learning system for recognizing user tasks from desktop activities and email messages. In Proc. International Conference on Intelligent User Interfaces (IUI). 2006.

[9] Yoo, J., Gervasio, M., and Langley, P. An adaptive stock tracker for personalized trading advice. In Proc. International Conference on Intelligent User Interfaces (IUI). 2003.

[10] Airspace: Tools for evaluating complex systems, machine language, and complex tasks. http://www.cs.cmu.edu/~airspace

[11] Steinfeld, A., Quinones, P.-A., Zimmerman, J., Bennett, S. R., and Siewiorek, D. Survey measures for evaluation of cognitive assistants. In Proc. NIST Performance Metrics for Intelligent Systems Workshop (PerMIS). 2007.