The AAAI-17 Workshop on
Crowdsourcing, Deep Learning, and Artificial Intelligence Agents
WS-17-07

# Complementing the Execution of AI Systems with Human Computation

**Ece Kamar**[1], **Lydia Manikonda**[*2]
[1] Microsoft Research, Redmond, WA
[2] Arizona State University, Tempe, AZ
eckamar@microsoft.com, lmanikonda@asu.edu

## Abstract

For a multitude of tasks that come naturally to humans, performance of AI systems is inferior to human level performance. We show how human intellect made available via crowdsourcing can be used to complement an existing system during execution. We introduce a hybrid workflow that queries people to verify and correct the output of the system and present a simulation-based workflow optimization method to balance the cost of human input with the expected improvement in performance. Through empirical evaluations on an image captioning system, we show that the hybrid system, which combines the AI system with human input, significantly outperforms the automated system by properly trading off the cost of human input with expected benefit. Finally, we show that human input collected at execution time can be used to teach the system about its errors and limitations.

## Introduction

Artificial intelligence has pursued the construction of systems that can accomplish tasks that come naturally to humans. In recent years, human computation has emerged as a resource for developing AI systems. To date, interactions between human computation and AI systems have been mostly limited to providing training data for predictive modeling in an offline fashion (e.g., (von Ahn and Dabbish 2004; Russell et al. 2008; Alonso, Rose, and Stewart 2008)).

Despite advances in algorithms and representations, the performance of many AI systems are inferior to human-level performance for tasks that come naturally to humans (e.g., (Cui et al. 2015)). When AI systems take on roles typically served by people without human supervision, their shortcomings may lead to errors, which can be drastic in critical domains, and may negatively affect user trust. We investigate principles, models and algorithms for developing *hybrid intelligent systems*, in which human input is incorporated into the execution of an AI system to address its shortcomings (See Figure 1). In a hybrid intelligent system, the AI system and the crowd form a single processing unit to generate a single output together in response to a problem instance input. In each execution, first the AI system generates

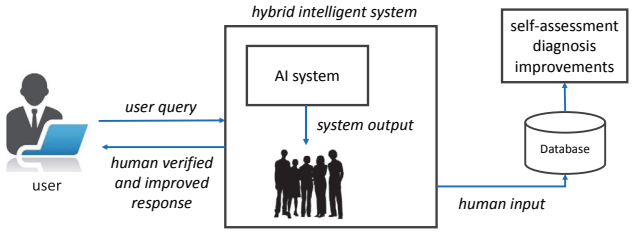


Figure 1: Flow for hybrid intelligent systems.

an output for the input problem instance. Instead of directly delivering this output to the user, the crowd is queried on demand for verifying and then improving the system output so that mistakes from AI systems can be prevented, and a reliable experience can be delivered to the end user. The crowd input is incorporated to the execution through an application specific workflow of crowdsourcing tasks. Human input collected from the crowd during execution for verifying system output and improving it can be logged. This input can be used for assessing the performance of the AI system, for diagnosing its errors through observing human improvements and for improving system performance.

Our studies on hybrid intelligent systems focus on an existing system designed for captioning images; the system is tasked with creating a single sentence summary of the prominent information in a given image (Fang et al. 2015). It serves as an ideal setting for our studies as the task is easy and intuitive for people to perform but challenging for machines. We describe a hybrid workflow for verifying and improving the outputs of the image captioning system that combines three types of crowdsourcing tasks – to verify captions, to fix system generated auto-captions, and to generate new captions.

A hybrid intelligence system is faced with a challenge of optimizing the workflow parameters to trade off the expected improvement from having more human input with the time and monetary costs associated with it. We address this challenge by combining data collection with a simulation-based optimization algorithm. The algorithm estimates the net utility of a workflow by simulating its execution on human inputs collected for a training set and selects the static workflow that offers the highest net expected utility.

[*] Lydia Manikonda contributed to this research during an internship at Microsoft Research.

We empirically evaluate the effectiveness of the hybrid workflow in complementing the execution of the image captioning system. Our experiments show that the hybrid workflow, when optimized with the simulation-based algorithm, can increase the net utility of executing the system alone by 50%. They also show that a small training set is sufficient for the simulation-based algorithm to optimize workflow parameters. The results also demonstrate that humans fixing machine output generates higher efficiency than humans generating captions alone, highlighting opportunities from humans and machines working together. In addition, the input collected during execution through humans verifying and fixing machine output can be used for assessing and diagnosing the automated system without additional need for data collection. The results demonstrate that hybrid execution can successfully overcome the shortcomings of AI systems and call attention to the virtuous loop it creates to guide the continuous improvements of AI systems.

## Background and Related Work

Our work relates to several efforts in the human computation literature on designs, techniques, models and algorithms for using human intelligence to accomplish tasks that machines cannot do alone. We group the related work under three sections as described below:

### Workflow Design and Optimization

Workflows are commonly used in human computation for accomplishing tasks reliably through microtasking with the crowd. The simplest workflows are proposed for labeling tasks, where labels collected from multiple workers are aggregated to infer the ground truth answer of a task (Sheng, Provost, and Ipeirotis 2008). The Soylent System introduced *Find-Fix-Verify* workflows for workers to build on each others' work for word processing tasks (Bernstein et al. 2010). Other researchers developed workflows for carrying out complex tasks by decomposing them into multiple, smaller tasks (Lasecki et al. 2013; Kittur et al. 2011; Zhang et al. 2012).

Decision-theoretic optimization techniques have been proposed for optimizing the allocation of human effort for labeling tasks (Kamar, Hacker, and Horvitz 2012; Kamar, Kapoor, and Horvitz 2013). Other thread of research (Lin, Mausam, and Weld 2013; Dai et al. 2013) addressed the optimization of complex workflows by employing partially observable Markov decision processes (MDP) and reinforcement learning where the tasks are completely accomplished by the human input with no baseline AI system in existence. Our work builds on existing approaches by introducing a simulation-based algorithm for optimizing the parameters of static workflows that complement the execution of an existing AI system.

### Human Computation for AI Systems

Previous work on complementary computing highlights the promise of using the different strengths of humans and machines for problem solving tasks (Horvitz and Paek 2007). Researchers investigated approaches for dividing responsibilities among humans and machines in machine learning (Cheng and Bernstein 2015; Chang, Kittur, and Hahn 2016) and in machine translation (Shahaf and Horvitz 2010). Other line of work explored different combinations of human and machine involvement to identify the shortcomings of a machine learned pipeline (Parikh and Zitnick 2011).

Multiple lines of work have investigated methods to incorporate human input to verify the decisions of machine learning systems for simple perception tasks such as image labeling (Kamar, Hacker, and Horvitz 2012) and image search (Yan, Kumar, and Ganesan 2010). Researchers studied MDP-based algorithms for acquiring human input to correct image annotations created by an automated system (Russakovsky, Li, and Fei-Fei 2015). In the Zensors system, crowd was asked to carry out perception tasks so that human input can be used later to train an automated system for gradual transition to automation (Laput et al. 2015). Our work extends this line of work by investigating hybrid workflows in which crowd not only verifies but also improves the output of an existing sophisticated machine learning pipeline during execution.

### Real-time Crowdsourcing

Although this paper does not address issues related to real-time acquisition of human input during execution, the recent advances in real-time crowdsourcing can be incorporated with hybrid workflows for supporting the execution of AI systems in real-time. Bernstein et al., showed that the latency of accessing crowd input can be reduced to under a second using the retainer model (Bernstein et al. 2011). Researchers have also shared insights and general architectures for developing successful real-time crowd-powered systems (Lasecki, Homan, and Bigham 2014). Models developed in previous work on predicting delay and cost of acquiring real-time human input (Yan, Kumar, and Ganesan 2010) can be incorporated with the optimization techniques presented in this paper through formalizing richer cost functions.

## Methodology for Complementing AI Systems

We now explore image captioning as a case study to describe our methodology for the integration of crowd input in hybrid intelligence systems.

### Image Captioning System

Image captioning is proposed in recent years as a challenge problem for AI researchers to promote advances in image understanding and language generation (Cui et al. 2015). The goal of this challenge is to summarize the salient content in a given image in a single sentence. As a test bed in our investigations, we use the image captioning system (Fang et al. 2015), which was one of the 2015 CVPR challenge winners. The system is composed of three main machine learned components: (1) the object recognizer component for detecting words, (2) the Language Model (LM) component for generating sentences from detected words,

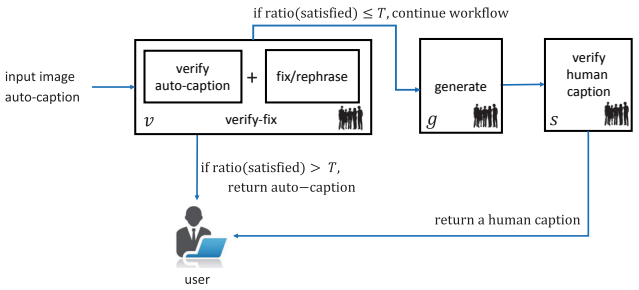


Figure 2: Hybrid workflow for complementing image captioning with human crowd input.

and (3) the re-ranking component for ranking sentences produced by the LM component. The system was trained and evaluated on the MS COCO captioning data set (Chen et al. 2015). The evaluations comparing machine generated captions with human generated captions show that 66% of the time, the auto-generated captions are unsatisfactory which makes this system an ideal candidate for our studies.

## Workflow Design

Figure 2 presents the workflow that we designed to complement the image captioning system with human intellect. The workflow takes as input an image and the corresponding caption generated by the image captioning system (the *auto-caption*). The workflow has access to a population of workers through crowdsourcing that accomplishes micro-tasks on demand. It uses worker overlap at each step to increase the quality of human involvement. The output is a caption that is verified and when needed improved by the crowd.

The workflow sequences three types of human computation tasks to produce an output caption. As a first step, the workflow assigns $v$ workers to the *verify-fix* task. In this task, workers are shown the input image and the auto-caption and are asked to evaluate if the auto-caption is satisfactory for the given image (*verify auto-caption* step). Depending on their answer, they are either asked to correct the unsatisfactory auto-caption or they are asked to rephrase the satisfactory auto-caption to another satisfactory alternative (*fix/rephrase* step). After $v$ workers verify the quality of the auto-caption, the workflow makes a decision about terminating the workflow or using further human input in generating a human caption and/or verifying human captions. If the ratio of workers assessing the auto-caption as satisfactory is above threshold $T$, the workflow delivers the auto-caption to the user. If not, the workflow asks $g$ workers to generate a satisfactory caption for the input image without showing the auto-caption (*generate* step). Finally, the workflow asks workers to verify the quality of human captions, which is the collection of $v$ fixes/rephrases and $g$ newly generated captions (*verify human caption* step). For each human caption, the workflow assigns $s$ workers to verify if the human caption is satisfactory. This step collects $s \times (v + g)$ assessments in total. The workflow outputs the human caption that receives the highest ratio of satisfactory evaluations from $s$ workers. If $s$ is 0, a random human caption is delivered.

Let workflow $w$ be specified by the tuple $< v, T, g, s >$. Values assigned to these parameters affect the behavior of the hybrid workflow. When $v$, $g$ and $s$ are all 0, the workflow delivers the auto-caption to the user without human involvement. When $v$ is 0 and $g > 0$, the workflow does not utilize the auto-caption in generating the final caption and delivers a caption generated by a human. It is important to note that the design space of possible hybrid workflows is not limited to the one in Figure 2. For example, the verify auto-caption and fix/rephrase steps can be divided to be given to separate workers in alternative workflows. We prefer the workflow in Figure 2 as verifying the auto-caption is the first step of fixing and we aim to eliminate redundant work from workers by combining *verify auto-caption* and *fix/rephrase* tasks under the *verify-fix* task. Alternative workflows may support iterative improvement steps instead of applying overlap to fix and generate steps. Or, they may achieve additional efficiencies by conditioning the hiring of workers for verify steps based on previous workers' responses.

Setting workflow parameters high is likely to increase the output quality but it also increases the monetary and time costs of executing the workflow. Achieving the highest gains from the workflow hinges on successfully trading off the expected improvements from human input at each step of the workflow with the associated cost. We present a workflow optimization procedure that aims at finding the best static workflow for a given input set. Our decision to focus on optimizing static workflows rather than dynamic workflows is due to the large amount of data collection needed to learn dynamic workflows and practical challenges in implementing them in current crowdsourcing marketplaces. Our optimization procedure seeks the set of parameters that achieve the highest average net value across the set of input images.

Our optimization procedure follows a simulation-based approach: First, we collect a data set of the hybrid workflow execution on a small subset of the input data, which is later used in workflow optimization. The data set is collected by running the workflow with parameters large enough that workflows of smaller parameter combinations can be simulated using this data set. Second, our optimization algorithm performs grid search over possible parameter values to select the workflow that offers the highest expected net utility over the training data set.

## Task Design and Data Collection

The hybrid workflow in Figure 2 is composed of three micro-tasks; *verify*, *fix/rephrase* and *generate*. In all tasks, we inform workers about the captioning challenge by telling them that a satisfactory caption summarizes the prominent information in a given image with a single sentence so that a blind user can understand the contents of the image. *Verify* is a binary labeling task in which we present a worker an image and a caption, and ask workers to tell us whether the given caption is satisfactory for the image. The *fix/rephrase* task immediately follows a worker verifying an auto-caption. If the worker finds the caption unsatisfactory, we ask the worker to rewrite a caption by turning the original caption into a satisfactory caption. If the worker finds the caption satisfactory, then we ask workers to provide another satis-

factory caption that is significantly different from the original caption. We designed the task to have similar workloads in both fixing and rephrasing so that this step does not bias the worker's assessment in the prior verification step. The *generate* task shows an image and asks workers to provide a caption in a text box.

For our data collection, we randomly sampled 1000 images from the validation portion of the MS COCO data set and obtained the corresponding auto-captions generated by the captioning system. In each experiment, a small, randomly sampled subset of this data is used in workflow optimization and the rest is used in evaluation. To enable the simulation of different workflows in the optimization algorithm, in the data collection, we set the workflow parameters $< v, g, s >$ to values $< \bar{v}, \bar{g}, \bar{s} >$, which are at least as large as the maximum of parameters we consider in searching for the best workflow. Data collected from fix/rephrase and generate tasks are used in the optimization algorithm to simulate the corresponding steps. Whereas, data collected from verify tasks are used for two purposes; a subset of this data for each image is used for simulating verify steps, and the remainder is used for evaluating the quality of a caption if that caption is the output of the hybrid workflow. Therefore, in our data collection the overlaps of the verify steps are larger than the maximum of $v$ and $s$ parameters we consider in searching for the best workflow. While collecting our data set, we set $\bar{v}$ and $\bar{s}$ to 10 and set $\bar{g}$ to 5. We paid $1¢$ for verify tasks, an additional $4¢$ for the fix/verify tasks and $8¢$ for generate tasks. The data collection was performed on Amazon Mechanical Turk.

Our efforts for high-quality data collection in crowdsourcing focused on the following strategies: Each task contained example tasks and clear and detailed instructions based on a feedback loop with workers. We performed spam detection based on worker agreement. For *verify* tasks, we reviewed the work of workers who disagreed with other workers consistently and blocked their work if our analysis reached the same conclusion. For tasks that asked workers to fix or generate captions, the assessments of other workers through *verify* human caption tasks were used to identify workers who consistently provide low-quality captions.

The resulting data set is structured as follows: For each image $i \in I$, the set of captions are $C =< c^a, \{c^h_1, ..., c^h_{\bar{v}+\bar{g}}\} >$, where $c^a$ is the auto-caption and $\{c^h_1, ..., c^h_{\bar{v}+\bar{g}}\}$ is the set of human captions. The first $\bar{v}$ elements of human captions are fixes/rephrases and the remaining $\bar{g}$ are generates. For each image $i$, caption assessments are represented in $V =< V^a, V^h >$, where $V^a = \{v^a_1, ..., v^a_{\bar{s}}\}$ are binary assessments (satisfactory, unsatisfactory) of $c^a$ and $V^h = \{v^h_{1,1}, ..., v^h_{1,\bar{s}}; ...; v^h_{\bar{v}+\bar{g},1}, ..., v^h_{\bar{v}+\bar{g},\bar{s}}\}$ is a matrix, where $v^h_{j,k}$ is the $k^{th}$ binary assessment of $c^h_j$.

## Workflow Optimization

The goal of workflow optimization is to select the set of parameters $< v, T, g, s >$ that maximizes the net utility of executing the hybrid workflow for a given set of images $I$. The net utility is a combination of the quality of captions created by the workflow and the cost associated with executing the workflow with humans in the loop.

Let $I_{tr} \subset I$ be the set of images to be used in workflow optimization. We assume that $I_{tr}$ comes from the same distribution that test images are from. $W$ is the set of all workflows to be considered in the search for the optimal workflow. For the purpose of image captioning scenario, $W$ includes all tuples $< v, T, g, s >$ such that $v, g, s$ are integers between 0 and 5 and threshold parameter $T$ varies between 0.0 and 1.0 with 0.2 increments. $c_v, c_f, c_g$ are the costs in \$ for verify, fix/rephrase and generate tasks respectively. $u_{sat}$ is the utility of a satisfactory outcome. We assume $u_{unsat}$, the utility of an unsatisfactory caption, to be 0 for simplification. The task of workflow optimization is finding $w^* \in W$ that optimizes the expected net utility as follows:

$$\begin{aligned} w^* &= \arg\max_{w \in W} \mathbb{E}\big[NU_w\big] \\ &\approx \arg\max_{w \in W} \sum_{i \in I_{tr}} \sum_{e \in E} NU_w(i, e) \end{aligned}$$

where

$$NU_w(i, e) = p_{sat}(w, i, e) \times u_{sat} - cost(w, i, e)$$

The challenge of optimizing $w^*$ is estimating $NU_w(i)$, the expected net utility of workflow $w$ for a given image $i$. This value may vary across different executions of the workflow based on the quality of work produced by workers. The main idea of our optimization algorithm is using sampling to generate $E$, a set of possible executions of workflow $w$ on a given input image $i$, and approximating $NU_w(i)$ by aggregating over $NU_w(i, e)$, the net utility of $w$ on image $i$ over execution $e$. $NU_w(i, e)$ can be computed by simulating the execution and estimating $p_{sat}(w, i, e)$, the probability of producing a satisfactory caption, and $cost(w, e, i)$, the cost of executing workflow $w$ on image $i$.

The simulation-based workflow optimization algorithm is presented in Algorithm 1. For each workflow $w$, the algorithm computes $p_{sat}(w)$ and $cost(w)$ by simulating $w$ on each image in $I_{tr}$ $\varepsilon$ many times. The simulation process has two main steps: First, it simulates the execution of the workflow by sampling human captions and assessments about captions from the data set (lines 5-7). $\sigma_v, \sigma_g, \sigma_s$ are set of indices of the sampled assessments for verify-fix, generate and select tasks respectively. Once the workflow terminates – either by outputting the auto-caption or one of the human captions – the algorithm evaluates how satisfactory the output is based on $\sigma^c_v$ and $\sigma^c_s$, the assessments that are *not* used in simulating the workflow (lines 8-13). After each execution, the order of elements of each row of $V^h$ are randomized so that the set of assessments for each human caption are sampled independently (lines 14-15). Once $p_{sat}$ and $cost$ values are estimated for each workflow, the output of the algorithm, is $w^*$, the workflow that maximizes the average net utility over images in $I_{tr}$.

The main challenge that Algorithm 1 addresses is estimating the net utilities of any workflow in $W$ through steps 2-15. Once workflow utilities are estimated, the algorithm performs simple exhaustive search over $W$ to select $w^*$. For problems in which the space of possible workflows makes exhaustive search infeasible or some workflow parameters

**Algorithm 1:** Workflow optimization algorithm

1 **foreach** $w =< v, T, g, s >$ *in* $W$ **do**
2 $\quad p_{sat}(w) \leftarrow 0, cost(w) \leftarrow 0$
3 $\quad$ **foreach** *image* $i$ *in* $I_{tr}$ **do**
4 $\quad\quad$ **for** $e \leftarrow 1$ *to* $\varepsilon$ **do**
5 $\quad\quad\quad < \sigma_v, \sigma_g, \sigma_s > \leftarrow SampleExecution(w)$
6 $\quad\quad\quad \sigma_v^c \leftarrow \{1 : \bar{v}\} \setminus \sigma_v, \sigma_v^s \leftarrow \{1 : \bar{s}\} \setminus \sigma_s$
7 $\quad\quad\quad \sigma_h \leftarrow \sigma_v \bigcup \sigma_g$
8 $\quad\quad\quad$ **if** $\sum_{k \in \sigma_v} v_k^a / v > T$ **then**
9 $\quad\quad\quad\quad p_{sat}(w) \leftarrow p_{sat} + (\sum_{k \in \sigma_v^c} v_k^a / |\sigma_v^c|)$
10 $\quad\quad\quad\quad cost(w) \leftarrow cost(w) + (c_v + c_f) \times v$
$\quad\quad\quad$ **else**
11 $\quad\quad\quad\quad h^* \leftarrow \arg\max_{j \in \sigma_h} \sum_{k \in \sigma_s} v_{j,k}^h / |\sigma_s|$
12 $\quad\quad\quad\quad p_{sat}(w) \leftarrow p_{sat} + (\sum_{k \in \sigma_s^c} v_{h^*,k}^h / |\sigma_s^c|)$
13 $\quad\quad\quad\quad cost(w) \leftarrow cost(w) + (c_v + c_f) \times v$
$\quad\quad\quad\quad\quad + c_g \times g + (v + g) \times s \times c_v$
14 $\quad\quad\quad$ **for** $j \leftarrow 1$ *to* $(\bar{v} + \bar{g})$ **do**
15 $\quad\quad\quad\quad$ randomize order of elements in $V_j^h$
16 $w^* \leftarrow \arg\max_{w \in W}(p_{sat}(w) \times u_{sat} - cost(w))$

are continuous, the exhaustive search step be replaced with more sophisticated approaches from the optimization literature such as randomized search or gradient-based optimization (Bergstra and Bengio 2012; Chapelle et al. 2002).

The optimization procedure can be applied to other workflows as long as the individual micro-tasks are independent of the parameters chosen for the workflow. For example, in our workflow, the content of the verification task for human captions is independent of the number of human captions collected with fix/rephrase and generate steps as each human caption is verified separately. When this condition holds, a dataset collected by running the workflow with large enough parameters at each step can be used to simulate the execution of the workflow with other parameter combinations of smaller values.

## Empirical Evaluation

This section presents an empirical evaluation of the hybrid workflow for image captioning and the workflow optimization. The evaluations are performed on the data set collected for the 1000 images sampled from the validation portion of MS COCO data set. The significance of the results are tested with the Wilcoxon signed-rank test (Wilcoxon 1945).

The main hypothesis of this work is that human input can improve the auto-captions generated by the image captioning system. Figure 3 analyzes this hypothesis by comparing the qualities of auto-captions with the average qualities of captions generated and fixed/rephrased by humans. We define the quality of a caption as the ratio of the workers assessing the caption to be satisfactory. The figure bins the images in our data set according to the quality of the auto-

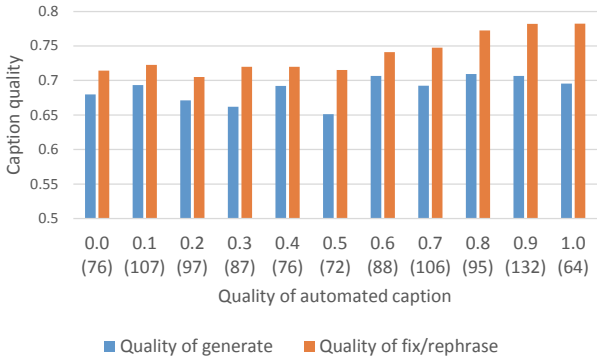


Figure 3: Comparison of the qualities of captions generated by the system and with human input.

caption. The number of images in each bin is given in parenthesis. The analysis shows that both types of human input result in significantly higher quality captions than the system alone ($p < 0.01$). In addition, the fix/rephrase condition lead to significantly higher quality captions than humans independently generating captions. This observation can be explained in a number of ways: Fixing an existing caption may have a lower cognitive load for humans than generating one. Even when the input auto-caption is unsatisfactory, it may set an example of what form of a caption is acceptable from workers in terms of its structure and detailedness.

Figure 4 shows performance improvements gained from executing hybrid workflows rather than executing the automated system alone for different values of $u_{sat}$. It reports results for the hybrid workflow when its parameters are optimized using Algorithm 1 and when they are chosen randomly in $W$. It also reports results for a *human generate workflow*, which is a simplified version of the hybrid workflow in Figure 2 that is composed of *generate* and *verify human caption* steps and does not make use of the auto-caption. Human generate workflows are optimized using a simplified version of Algorithm 1 that searches the best tuple $< g, s >$ for given $u_{sat}$ and $c_g$ and $c_v$.

In evaluating the optimized workflows, we randomly sample a small subset of the data set to be given to Algorithm 1 as input for optimization ($I_{tr}$) and use the remaining data set for evaluation. We vary the size of $I_{tr}$ between 5 images and 100 images and repeat the experiment 10 times for each condition. The confidence bars on Figure 4 reports the performance difference of the optimized workflows when the size of the input data ($I_{tr}$) is varied. Sampling size ($\epsilon$) in Algorithm 1 is set to 100. Testing of workflows follow the simulation steps of Algorithm 1 between lines 5-13.

For all $u_{sat}$ values, the net utility of the optimized hybrid workflow is significantly higher than the optimized human generate and automated workflows ($p < 0.01$). The relative performance of the hybrid workflow improves as $u_{sat}$ increases as the optimization algorithm can afford to acquire more human input to improve the auto-caption. When $u_{sat}$ is \$6, the net utility of the hybrid workflow is as high as 1.5 times of the net utility of the automated system. We see a

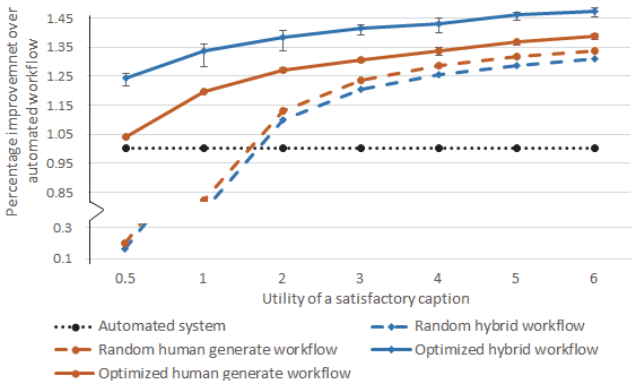


Figure 4: Performance improvements from hybrid and human generate workflows over the automated system.

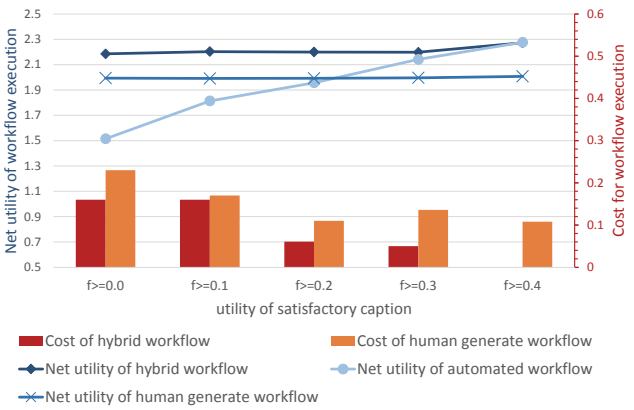


Figure 5: Behavior of hybrid workflow for improving levels of automated system performance ($u_{sat} = \$3$).

significant difference between the performance of optimized *hybrid* and *human generate workflows* as a result of hybrid workflows employing fix-verify tasks over generate tasks.

Our optimization procedure requires collecting a data set of workflow execution with large enough overlap such that the data set can be used in Algorithm 1 in optimizing workflow parameters. A practical consideration for implementing the procedure is the amount of data needed to get effective results. The confidence bars on optimized workflow conditions in Figure 4 show that the performance of Algorithm 1 is not sensitive to the size of the input data set ($I_{tr}$). The algorithm is able to identify effective workflows for data sets as small as 5 images, which makes the cost of data collection small compared to the benefit from executing the optimized hybrid workflow over the automated system for the remaining data set. When training data set is collected for 5 images, the cost of data collection is recovered after executing the hybrid tasks for 50, 22 and 13 tasks when $u_{sat}$ is set to \$1,\$ 2 and \$3 respectively. Being able to optimize workflows with a small data collection is due to our decision to focus on static workflows rather than dynamic workflows, as optimizing dynamic workflows would incur significantly larger demands on data collection. Additional experiments also revealed that the performance of the algorithm is insensitive to the sampling size; comparable results are obtained for sampling size of 20.

Workflow optimization can be repeated during the lifecycle of an automated system to get acquainted with the system changes. We expect an automated system executing within a hybrid workflow to become less dependent on human input as the performance of the automated system improves as it learns from human input. Figure 5 shows how the demands of executing the hybrid workflow decreases as the performance of the automated system improves while maintaining a stable quality. In these experiments, we simulate system improvements by filtering out images from the data set that have auto-captions of quality below threshold $f$ and repeat workflow optimization for each level of system performance. The figure shows that the cost of the hybrid workflow decreases as the performance of automated system increases while maintaining a stable quality. These results suggest that hybrid workflows can allow for gradual transition towards complete automation as the underlying AI system improves.

## Human Input as Feedback to AI Systems

In the previous section, we showed that human input in hybrid workflows can significantly help to improve the quality of the system output. Humans verifying and fixing system output during execution has a secondary benefit in terms of providing feedback to the system and its designers about the system performance and how it can be improved.

In this section, we present two scenarios of how human input collected through hybrid workflows is used to develop metareasoning capabilities for the system. First, we show how labels collected from *verify auto-caption* tasks can be used to train a self-assessment model for the image captioning system. Then we study how the fixes collected for unsatisfactory auto-caption can help with self-diagnosis.

### Training a Self-assessment Model

Every time a worker completes a *verify auto-caption* task, the worker provides a binary label about the system performance for a given input image. This data can be used to train a self-assessment model for the system predicting how satisfactory is the system output for any given input image. Even in the case when a self-assessment model exists, the data can be used to update or re-calibrate the model as the training data may not match the distribution of data the system faces in execution time. An accurate model can be incorporated into the hybrid workflow to replace *verify auto-caption* tasks to make the workflow more efficient.

The image captioning system lacks a predictive model of its performance. Training such a model is the focus of our next set of experiments. For our experiments, we expanded our existing data set with worker assessments for *verify auto-caption* tasks for 9000 other images sampled from the validation portion of the MS COCO data set and corresponding auto-captions. The resulting data set has 5 bi-

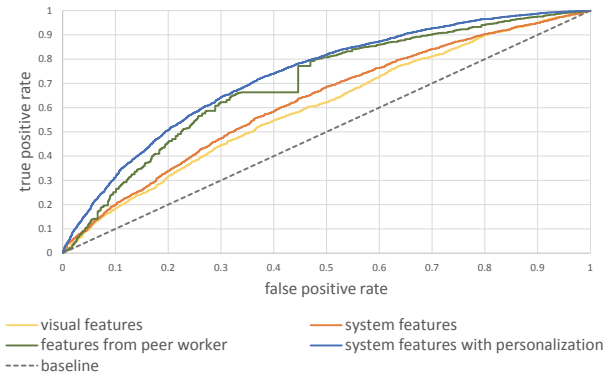


Figure 6: Comparison of self-assessment models trained from worker input.

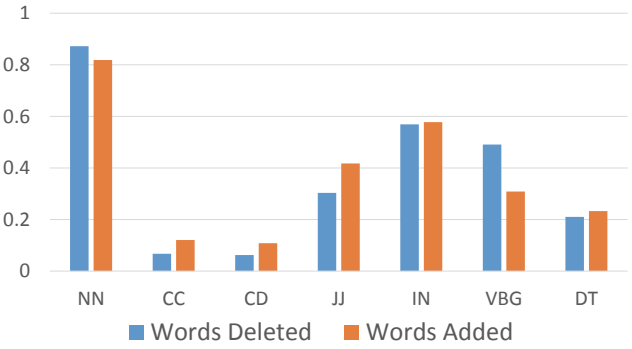


Figure 7: Frequency of different fixes for turning an unsatisfactory caption to satisfactory. Two categories of caption modifications – added tags; deleted tags. NN – *Noun*; CC – *Conjunctions*; CD – *Numbers*; JJ – *Adjective*; IN – *Preposition*; VBG – *Verb*; DT – *Determiner*

nary annotations for 10000 image-auto-caption pairs. We divide the dataset into training, validation and testing portions such that all assessments for the same image-auto-caption pair belong to the same portion. The ratios of training, validation and testing portions are 40%, 30% and 30%.

The task of training a self-assessment model is a binary classification problem. Each instance of the data set has features, which may describe the input image or the execution of the system for the image, and a binary label, which is the assessment of a user (or a target worker in these experiments). We use boosted decision trees in our experiments for prediction. For each image, our *visual features* come from the layers of the deep neural network used for object recognition in the image captioning pipeline. The extraction of these *fc7* features are described in (Fang et al. 2015). The *system features* are extracted from the execution of each component of the system, including the scores of captions from the re-ranker and the language model components, the recognition scores from the object recognizer, statistics about the distribution of recognition results of objects and activities, and statistics about how recognition results translate to the captions produced by the system.

The analysis of worker assessments shows that people differ in their assessments of captions captions, which opens up the possibility of personalizing the predictions of the self-assessment model if the identity of the user whose assessment we are predicting is known. To explore this possibility, we used the training data to develop profiles of workers providing assessments. The features for worker profiles include statistics about their assessments such as their average assessment, their agreements with other workers and their confusion matrices as computed by the Bayesian Classifier Combination aggregation model (Kim and Ghahramani 2012). In training and testing of the model with *personalization*, the feature set includes features about the worker that we are predicting the assessment of (i.e., target worker) derived from the corresponding profile.

We wondered how visual and system features compare with another assessment from a peer worker in predicting a worker's assessment. As a baseline, we trained a classifier that has access to *features from peer worker*. To compute this feature set, we sample a peer worker, who is different than the target worker, who provided an assessment for the same image-auto-caption pair. The feature set includes the assessment of the peer worker for the same image-auto-caption pair in addition to the features describing the peer worker's profile. This model does not have personalization as it does not have access to the profile of the target worker.

Figure 6 compares the performance of different classifiers by showing the ROC curve associated with each classifier. The figure shows that models that have access to visual (yellow line) and system features (orange line) perform better than the baseline (dashed line) – a random classifier. The model with system features perform slightly better than the model with visual features only, showing that the signals collected from components may offer value for predicting system performance. Comparison of the classifier with access to peer worker assessment with the classifier that has access to system features shows that a peer worker is still a better predictor than the system signals of another worker's assessment. However, when the classifier with system features has personalization (blue line), we see it being as accurate as the peer worker in predicting the target worker's assessment.

## System Diagnosis from Fixes

The information about the fixes can be logged and used to analyze the types of mistakes the system is making. In this diagnostic analysis, we pick all pairs of the original and fixed captions in our data set such that the original caption is assessed to be unsatisfactory but the fixed caption is satisfactory based on the majority voting of 10 workers. This data set includes original and fixed caption pairs for 44.3% of the 1000 images in our original data set. Among all such pairs, we take the difference of the two sentences and perform part-of-speech tagging on the difference to analyze the different steps workers took to fix an unsatisfactory caption (Toutanova et al. 2003). The analysis identifies the most common type of mistake as misrecognizing objects followed by the proposition errors and misrecognition of activities.

The results of such an analysis can help system designers in prioritizing future steps in system improvement. For example, Figure 7 provides evidence that improving object detection is a more promising next step than improving the count or attributes of objects.

## Discussion and Conclusions

We presented an intuitive human-in-the-loop methodology to address how human computation can be used to complement an existing image captioning system during the execution to overcome its limitations. To this end, we proposed a hybrid workflow that combines system output with human input and presented a simulation-based algorithm for optimizing the workflow parameters. Our experiments highlight the benefits of the hybrid workflow and emphasize how human input can be employed to refine the behavior of the system. The hybrid execution methodology is useful at execution time for preventing mistakes of existing AI systems and also provides valuable feedback to system developers for continuous system improvement.

The particular hybrid workflow studied in this paper demonstrated the benefits of hybrid intelligence systems. Improvements in the design of hybrid workflows, such as conditional branching based on worker agreement, can further increase the effectiveness of hybrid execution. Improvements in the design of human workflows can lead to better hybrid workflows which subsume them. A POMDP approach for managing dynamic workflows that can adjust decisions based on task difficulty, worker quality or worker responses is a promising direction with challenges about efficiently learning model parameters and making decisions. Hybrid execution may provide additional benefits to future systems in continuous retraining with human input.

**Acknowledgements** We would like to thank Eric Horvitz and Besmira Nushi for valuable discussions and feedback.

## References

Alonso, O.; Rose, D. E.; and Stewart, B. 2008. Crowdsourcing for relevance evaluation. *SIGIR Forum* 42(2).

Bergstra, J., and Bengio, Y. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research* 13(Feb):281–305.

Bernstein, M. S.; Little, G.; Miller, R. C.; Hartmann, B.; Ackerman, M. S.; Karger, D. R.; Crowell, D.; and Panovich, K. 2010. Soylent: A word processor with a crowd inside. In *Proc. UIST*.

Bernstein, M. S.; Brandt, J.; Miller, R. C.; and Karger, D. R. 2011. Crowds in two seconds: Enabling realtime crowd-powered interfaces. In *Proc. UIST*.

Chang, J. C.; Kittur, A.; and Hahn, N. 2016. Alloy: Clustering with crowds and computation. In *Proc. CHI*.

Chapelle, O.; Vapnik, V.; Bousquet, O.; and Mukherjee, S. 2002. Choosing multiple parameters for support vector machines. *Machine learning* 46(1-3):131–159.

Chen, X.; Fang, H.; Lin, T.-Y.; Vedantam, R.; Gupta, S.; Dollár, P.; and Zitnick, C. L. 2015. Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

Cheng, J., and Bernstein, M. S. 2015. Flock: Hybrid crowd-machine learning classifiers. In *Proc. CHI*.

Cui, Y.; R., R. M.; Lin, T.; Dollar, P.; and L., Z. 2015. COCO Captioning Challenge. http://mscoco.org/dataset/#captions-challenge2015.

Dai, P.; Lin, C. H.; Mausam; and Weld, D. S. 2013. POMDP-based control of workflows for crowdsourcing. *Artificial Intelligence* 202:52 – 85.

Fang, H.; Gupta, S.; Iandola, F.; Srivastava, R. K.; Deng, L.; Dollár, P.; Gao, J.; He, X.; Mitchell, M.; Platt, J. C.; et al. 2015. From captions to visual concepts and back. In *Proc. CVPR*.

Horvitz, E., and Paek, T. 2007. Complementary computing: policies for transferring callers from dialog systems to human receptionists. *User Modeling and User-Adapted Interaction* 17(1-2):159–182.

Kamar, E.; Hacker, S.; and Horvitz, E. 2012. Combining human and machine intelligence in large-scale crowdsourcing. In *Proc. AAMAS*.

Kamar, E.; Kapoor, A.; and Horvitz, E. 2013. Lifelong learning for acquiring the wisdom of the crowd. In *Proc. IJCAI*.

Kim, H.-C., and Ghahramani, Z. 2012. Bayesian classifier combination. In *Proc. AIStats*.

Kittur, A.; Smus, B.; Khamkar, S.; and Kraut, R. E. 2011. Crowdforge: Crowdsourcing complex work. In *Proc. UIST*.

Laput, G.; Lasecki, W. S.; Wiese, J.; Xiao, R.; Bigham, J. P.; and Harrison, C. 2015. Zensors: Adaptive, rapidly deployable, human-intelligent sensor feeds. In *Proc. CHI*.

Lasecki, W. S.; Thiha, P.; Zhong, Y.; Brady, E.; and Bigham, J. P. 2013. Answering visual questions with conversational crowd assistants. In *Proc. SIGACCESS*.

Lasecki, W. S.; Homan, C.; and Bigham, J. P. 2014. Architecting real-time crowd-powered systems. *Human Computation Journal*.

Lin, C. H.; Mausam; and Weld, D. S. 2013. Towards a language for non-expert specification of POMDPs for crowdsourcing. In *Proc. HCOMP*.

Parikh, D., and Zitnick, C. L. 2011. Finding the weakest link in person detectors. In *Proc. CVPR*.

Russakovsky, O.; Li, L.-J.; and Fei-Fei, L. 2015. Best of both worlds: Human-machine collaboration for object annotation. In *Proc. CVPR*.

Russell, B. C.; Torralba, A.; Murphy, K. P.; and Freeman, W. T. 2008. Labelme: A database and web-based tool for image annotation. *Int. J. Comput. Vision* 77(1-3).

Shahaf, D., and Horvitz, E. 2010. Generalized task markets for human and machine computation. In *Proc. AAAI*.

Sheng, V. S.; Provost, F.; and Ipeirotis, P. G. 2008. Get another label? Improving data quality and data mining using multiple, noisy labelers. In *Proc. KDD*.

Toutanova, K.; Klein, D.; Manning, C. D.; and Singer, Y. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. NAACL*.

von Ahn, L., and Dabbish, L. 2004. Labeling images with a computer game. In *Proc. CHI*.

Wilcoxon, F. 1945. Individual comparisons by ranking methods. *Biometrics bulletin* 1(6):80–83.

Yan, T.; Kumar, V.; and Ganesan, D. 2010. Crowdsearch: Exploiting crowds for accurate real-time image search on mobile phones. In *Proc. MobiSys*.

Zhang, H.; Law, E.; Miller, R.; Gajos, K.; Parkes, D.; and Horvitz, E. 2012. Human computation tasks with global constraints. In *Proc. CHI*.