Proceedings of the Eighth AAAI Conference on Human Computation and Crowdsourcing (HCOMP-20)

Effective Operator Summaries Extraction

Ido Nimni,[1] **David Sarne**[1]
[1]Bar-Ilan University, Israel
ido.nimni12@gmail.com, sarned@cs.biu.ac.il

Abstract

This paper proposes a heuristic algorithm for effectively summarizing the work of novice robot operators, e.g., ones recruited through crowdsourcing platforms, in search and rescue-like tasks. Such summaries can be used for many purposes, perhaps most notably for monitoring and evaluating an operator's performance in settings where information gaps preclude automatic evaluation. The underlying idea of our method is dividing the task timeline into intervals, and extracting a subset of high-scoring and low-scoring segments within, using a heuristic scoring function. This results in a short effective summary of the operator's work, based on which several other crowdworkers can evaluate her performance. The effectiveness of the proposed method was extensively evaluated and compared to a large set of alternative methods through a series of experiments in Amazon Mechanical Turk. The analysis of the results reveals that the proposed method outperforms all tested alternatives. Finally, we evaluate the performance one may achieve with the use of machine learning for predicting the operator's performance in our domain. While this approach manages to reach a performance level similar to the one achieved with summaries, it requires an order-of-magnitude greater effort for training (measured in terms of crowdworkers time).

Introduction

With recent advances in robotics, the interest in human-operator interfaces for robotic-based missions (Polin et al. 2016), primarily for exploration and search missions (Wang et al. 2009c; 2011; Lewis, Sycara, and Nourbakhsh 2019) has grown significantly. In this kind of missions—such as Urban Search and Rescue (USAR) (Nourbakhsh et al. 2005) or planetary exploration (Jakuba et al. 2018)—robots are used to remotely carry out various tasks which heavily rely on the ability to effectively navigate in the plain or space. In some complex environments, however, their sophisticated logic, computational skills, and advanced sensors are still insufficient for fully understanding event happenings. Here, the assistance of human operators who can interpret visual and other sensory data and improve, or at times fully take control over the robot's navigation, becomes invaluable (Wang et al. 2009c; Machlev and Sarne 2020).

Much research has focused on improving the operator's control, suggesting various synchronous and asynchronous interfaces for operating the robots and managing the task (Wang et al. 2011; Kosti, Sarne, and Kaminka 2014). Still, not all human operators possess the same level of expertise, and the effectiveness of their navigation can highly vary (McGinn, Sena, and Kelly 2017). Therefore the ability to evaluate operators' performance in retrospect is crucial.

Evaluating the performance of an operator becomes even more acute when the operator is a novice or a crowdworker. Various real-life tasks involving robot-operation can nowadays be handled by untrained operators, on an ad-hoc basis. This is often the case whenever the robot navigation is quite intuitive and the main challenge lies in identifying events that call for its operation, a task which is complex for the robot yet quite easy for a person. For example, consider the case of operating an autonomous drone to identify the black box of an airplane that crashed. The drone's image processing capabilities are often insufficient for effectively identifying items in the clutter and a human operator (even a novice one) would be more capable guiding the drone's search, provided a proper interface. The use of crowdworkers in that sense is highly advantageous due to their high availability and relatively cheap cost (compared to a trained operator).

Alongside the many advantages of using crowdworkers for robot navigation, one needs to take into account several disadvantages that hinder their performance: lack of experience, as well as failure to understand the task, setting or goals may result in ineffective (and sometimes damaging) operation. Also, in the absence of proper monitoring, workers may be tempted to work on other tasks in parallel, to increase revenues, becoming less attentive overall (Yin, Chen, and Sun 2014). Being able to evaluate the operator's performance can thus be useful for various purposes, e.g., rewarding those that did very well, or preventing those that perform poorly from taking part in the task in the future.

The operator evaluation problem becomes highly challenging, both with expert operators and with novices or crowdworkers, whenever performance cannot be directly measured by the system, e.g., when it is fully subjective, or when information gaps preclude an objective evaluation. For example in USAR applications, the system may realize whether there was a survivor at a certain location to which an operator navigated a robot. However, it cannot tell, even in retrospect, if

there were any other survivors missed by the operator (as otherwise, if having such abilities, the system could have effectively navigated the robot by itself and spare the human operator). In such cases, operators can be evaluated only based on the opinion of an external expert that monitors the process (either in real-time or offline). Such a process is highly resource consuming.

In this paper, we propose the use of short summaries of robot-operated sessions as a means for facilitating offline operator's evaluation. Given a trace (e.g., a video or a schematic re-run) of the robot-operation session, our algorithm generates a sequence of sub-segments of that trace, aiming to effectively capture the highlights of the session. The main challenge in producing such a summary is overcoming the lack of information related to the continuous changes taking place in the environment (and hence any measure of success in carrying out the task). The algorithm can rely only on the operation of the robot. The summary can then be shown to experts for evaluation, or to crowdworkers, taking advantage of 'the wisdom of the crowd', requiring a reduced amount of work either way (compared to watching the full trace).

We evaluate the effectiveness of using the summaries generated with our proposed method through a set of comprehensive experiments using Amazon Mechanical Turk (AMT), based on an infrastructure that emulates the application of robotic boats for deterring birds in fish ponds. The achieved performance is compared to several alternative evaluation methods, including evaluation based on the complete traces, evaluation based on watching the trace of events in increased speed (fast-forwarding), and evaluation based on summaries combined of random segments. The results analysis reveals that, overall, our summary outperforms all other methods tested in terms of the accuracy of the evaluation obtained and the amount of effort required.

Finally, we report the results of our efforts aiming to develop machine-learning based models for evaluating the performance of a human operator in the above described settings. This attempt turned successful in the sense that we managed to converge to a similar level of evaluation accuracy as with summarization. However, the training of the methods required an order of magnitude greater effort in terms of the amount of data that had to be collected for training (measured in operators' minutes). Interestingly, we find that the two methods can co-exist and be used together as a means for providing an even better assessment of operator performance—in our experiments this latter approached resulted in an 11% improvement in the accuracy of the valuations generated.

Related Work

The research described in this paper relates primarily to three research areas: human-robot interaction, primarily in aspects of control by non-experts and effective collaboration, crowdsourcing, in particular emphasizing crowdworkers' attentiveness and collective evaluation, and strategy summarization.

Robot Control. Various mechanisms have been suggested for enhancing robot control, in particular robot navigation (Wang et al. 2009c; Velagapudi et al. 2008; Wang et al. 2011). While most work considered and experimented with expert operators, some work explicitly addressed the problem of operating a robot by novices. For example, Bruemmer et al (Bruemmer et al. 2004) develop a robot control system for a mixed-initiative setting, providing substantial experimental evidence for the success of a very broad spectrum of novice users to operate the robot in an urban search and rescue scenario. McGinn et al (McGinn, Sena, and Kelly 2017) investigate the performance and control behavior of novice robot operators in a home environment. They find a substantial variance in performance, indicating a wide spectrum of abilities among novice operators.

Crowdsourcing. Our use of crowdworkers is twofold—both for robot operation and for operator evaluation. Crowdsourcing for robot navigation, or more broadly, robotics and web-based robotics, is not a new idea (Toris, Kent, and Chernova 2014). For example, Crick et al (2011) present an online interface that can be used by crowdworkers to navigate a mobile robot through a maze, focusing on the effectiveness of different types of image data streams for teleoperating and training the robot, when provided to operators. Schulz et al (2000) study the use of web interfaces to remotely operate mobile robots in public places, enabling remote users to interact with humans within the robots' environment. They present much evidence for the effectiveness of such interfaces for web-based monitoring and control.

The use of crowdworkers for operators evaluation relates to an extensive literature on using crowdsourcing for generating some 'collective wisdom' (Gao and Zhou 2013; Wang et al. 2012). One factor that may highly influence the quality of evaluations to be received in our case is the relatively long and tiring nature of the task. There is much for workers' attention degradation with time in monotonous tasks (Rahman 2012), often leading to losing interest and switching to other tasks back and forth (Elmalech et al. 2016). Prior work has suggested various remedies for this problem, such as designated monetary compensation mechanisms (Mason and Watts 2010; DiPalantino and Vojnovic 2009; Finnerty et al. 2013), breaking a long task to a series of smaller sub-tasks to diversify the work or to avoid fatigue or boredom (Yin, Chen, and Sun 2014) and introducing dummy events to improve crowdworkers' attention (Elmalech et al. 2016).

Strategy Summarization. While to the best of our knowledge there is no work on human operator's task performance summarization, there is a broader literature relating to explaining robot behavior, interpretable machine learning, as well as studies concerned with users' mental models of systems that are relevant (Amir, Doshi-Velez, and Sarne 2019). For example, in the context of human-robot interaction, several methods have been suggested for supporting users in debugging a robot or improving the ability of the human and the robot to collaborate effectively (Nikolaidis and Shah 2013; Lomas et al. 2012; Brooks et al. 2010). In particular, Hayes & Shah (2017) proposed several methods for explaining

robot policies to people using past execution traces, enabling users to query the agent's behavior in different states and request explanations. These approaches are complementary to ours, in the sense that they provide different ways of examining the behaviors of agents, yet they do not attempt to generate summaries of the agent's behavior. Several prior works suggested methods for explaining recommendations given by MDP-based intelligent assistants (Dodson, Mattei, and Goldsmith 2011; Elizalde 2008). Yet, these aim to explain a specific action taken (or a suggested action) rather than reason about the task as a whole. Similarly, the many approaches that have been recently proposed for developing interpretable machine learning models (Doshi-Velez and Kim 2017; Vellido, Martín-Guerrero, and Lisboa 2012; Ribeiro, Singh, and Guestrin 2016) typically seek to explain a decision made by a machine learning model rather than provide a description of a strategy or behavior of an agent in different situations.

Model

We use a common robot-operation model, of wide use in prior work on search and rescue (Wang et al. 2009b; 2009a). It considers settings where a robot needs to be continuously navigated by its operator for handling events. Each event is characterized by an arrival time (at which it becomes active and visible), location in the plane and the amount of time it remains active. An event is considered 'handled' if the robot reaches its location while it is still active. The robot's operator is a priori unaware of the total number of events to appear, their locations and the times they will appear in, as well as the time window they will remain active. The operator will be able to identify an event and its location only at the time it appears.

The robot's navigation is carried out simply by having the operator draw routes that the robot will follow. The model assumes the robot's speed is constant, and at times where there is no active drawn route, the robot stops in place (idle). At any time the operator can change or even fully override the currently active route, e.g., in response to changes in the environment. The goal of the operator is to navigate the robot, based on the information being unfolded along with the session, such that the overall number of events eventually handled is maximized.

Once the session is finished, the effectiveness of the robot-navigation needs to be evaluated. Since the only reliable information available to the system is the routes generated along with the session, the evaluation needs to be carried out by a human evaluator/s. On top of the routes information, the latter can also extract the information about events, including their appearance time and location (e.g., by watching the video stream of events, similar to the way the operator made use of this knowledge for navigating the robot). Each evaluator will watch the session as a whole (or selected parts, i.e., a summary) and assign a score. The success of the evaluation will then be measured based on the correlation between the score assigned by the evaluator (or the average score in case of several evaluators) and the number of events handled in the session.

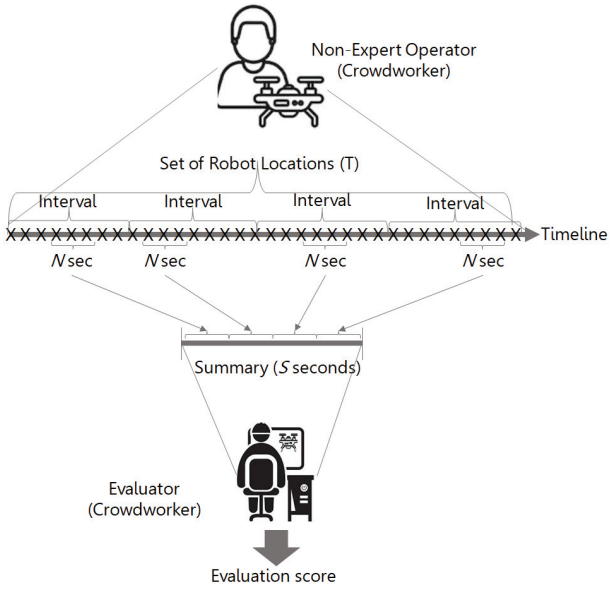

Figure 1: The summary extraction process.

Proposed Method

The summarization process relies on dividing the session into fixed intervals and intelligently selecting a short continuous trace-portion (a 'segment') from each interval (see Figure 1 for illustration). The underlying idea is that the performance of the operator may substantially vary along time, hence for completeness, it is important to consider segments that span the entire session. The segment picked from each interval is either the one we believe to reflect the best or the worst performance of the operator in that interval.

In order to decide which segment to choose from each fixed interval along the timeline, we use Algorithm 1. The algorithm takes as an input the parameters T and S. The first is the robot's trace along time, decoded as a set of tuples storing for each time point the robot's location in the plane and its current active route. The second is the length of the summary to be produced. In addition, the interval takes as an input the parameter N, denoting the length of any atomic continuous segment to be presented, such that it will provide enough context to evaluate performance at a given time.

For exposition purposes we divide Algorithm 1 into three conceptual stages: extracting attributes of the different time points, calculating a performance-related score to each point based on the extracted attributes and finally selecting a point (and its corresponding segment) from each interval based on the calculated scores.

Extracting Point Attributes. For each point $p \in T$, we calculate the following measures (Step 3): (a) *idle time* - time elapsed without having an active route; (b) *time from last decision* - time elapsed since the most recent update or generation of a route; (c) *time till next decision* - from current time till next update or generation of a route; (d) *existence of active route* (binary) - whether there is currently an active route; (e)

Algorithm 1: Summarization algorithm

input: T - robot's trace along time; N - segment's length; S - summary length;
output: $summarized$ - the summarized timeline;

1 $scoredTimeline = []$; $summarized = []$;
2 **for** $p \in T$ **do**
3 $\quad p = ExtractAttributes(p)$;
4 $\quad scoredTimeline.push(Evaluate(p))$;
5 **end**
6 **for** each $I \subset scoredTimeline$ *of size* $|T| \cdot \frac{N}{S}$ **do**
7 $\quad$ **if** $Rand() < p_{select}$ **then**
8 $\quad\quad p =$ Lowest scored point in I;
9 $\quad$ **end**
10 $\quad$ **else**
11 $\quad\quad p =$ Highest scored point in I;
12 $\quad$ **end**
13 $\quad summarized.push(\frac{N}{2}$ points before $p)$;
14 $\quad summarized.push(\frac{N}{2}$ points from p and on);
15 **end**
16 *return* $summarized$;

Algorithm 2: Evaluating user performance in a given point along timeline.

input: $p \in T$ - robot location
output: $score$ - the score assigned to point p

1 **function** $Evaluate\ (p)$ **:**
2 $\quad score = (p.idleTime)^{\alpha_{idle}} +$
$\quad\quad (p.timeFromLastDecision)^{\alpha_{elapsed}} +$
$\quad\quad (p.timeTillNextDecision)^{\alpha_{remaining}}$;
3 $\quad$ **if** *no active route* **then**
4 $\quad\quad score = score * \alpha_{path}$;
5 $\quad$ **end**
6 $\quad$ **if** *current route is eventually interrupted* **then**
7 $\quad\quad score = score * \alpha_{pre\_change}$;
8 $\quad$ **end**
9 $\quad$ **else if** *current route overrides a previous one* **then**
10 $\quad\quad score = score * \alpha_{post\_change}$;
11 $\quad$ **end**
12 $\quad$ *return* $score$;
13 *end*

route to be interrupted (binary) - whether the current route will be updated before completed; (f) *route is an update* (binary) - whether the current route resulted from an update to an existing route. These will be used for extracting performance score and are stored in the array $scoredTimeline$. The above set of attributes attempts to provide a mixture of measures for the attentiveness of the operator (e.g., idle time and the existence of an active route) and measures for the operator's reaction to the continuous changes in the environment the robot is operating in (e.g., time till next decision and route to be interrupted).

Evaluation Phase. Next, we assign a score to each point based on its attributes (Step 4). The score aims to capture the suitability and effectiveness of the user's actions (or lack of) at that specific time. The greater the score, the more likely it is that the operator is inattentive to happenings or made poor choices around that time, and vice versa. The score calculation is done through the heuristic function *Evaluate(point)* as given in Algorithm 2. It is essentially weighing the different attributes specified above according to their nature - for measurable attributes power coefficients are used (α_{idle},$\alpha_{elapsed}$,$\alpha_{remaining}$), whereas for binary attributes additive coefficients are used (α_{path},α_{pre_change},α_{post_change}).

Selection Phase. Finally, we select points of interest. As explained above, the selection methodology relies on dividing the timeline into fixed-length intervals and selecting a segment from each. Since the segment size is N and the total summary is of length S, the number of segments to be presented is $\frac{S}{N}$. The interval length is therefore $|T| \cdot \frac{N}{S}$. The algorithm iterates over the intervals of such size spanning T (Step 6). For each such interval it chooses whether to pick the point of maximum or minimum score (using the pre-specified probability p_{select}, see Step 7)[1] and then takes all points within a segment of size N spanning that point (Steps 13-14). The output is a summary of length S, containing different continuous segments of points.

Experimental Infrastructure

For evaluating our operator's automatic summarization method we used an experimental framework simulating boats' navigation in fish ponds. The framework was developed as part of a larger multi-institute collaboration aiming to develop innovative capabilities for autonomous systems designed to deter massive fish-eating birds from the depredation of fish ponds. The system is based on a small robotic boat, guided through location (GPS-based), computer-vision and human (crowd-sourcing-based) sensors. When birds land on the pond, they are scared away by a boat heading to their direction as quickly as possible, thus limiting the damage.

The performance of vision processing algorithms is highly affected by the water movement in the background in this domain, hence the automatic discovery of birds and automatic navigation of the boat is commonly poor. A human operator, on the other hand, even a novice one, can handle boat operation quite easily, as all that is required is identifying the birds upon arrival and pointing the boat in their direction. Therefore boat navigation is planned to be carried out using either expert operators or crowdworkers, based on schematic pond maps and the transfer of cameras output (installed on poles around the fish ponds).

We used a web-based simulated version of the above system, which was primarily developed for experimentation.[2] The interface provided to the operator enables clicking on a location in the pond area to initiates a straight-line route from the current boat's position or clicking on the boat to convey a

[1] Several other selection criteria can be applied. For example, choosing according to the average score of adjacent points.

[2] The testbed was developed using IIS for the server-side and Angular framework (HTML, CSS, Typescript) for the client-side.

route by drawing it (freestyle). With both methods the current route can be disrupted at any time and change immediately takes effect. New birds appear in the pond according to a pre-defined scenario which specifies their arrival time, specific location at the fish pond, and the time they will leave by themselves if not deterred by the boat by then. Birds are deterred whenever the boat gets near them. The deterrence distance is pre-set by the system administrator. To emphasize this capability the boat is enclosed with a circle representing its deterrence radius and once a bird is within the circle, it will be deterred immediately (indicated by the change of its color to red) and will not be presented anymore. The goal of the boat operator is to navigate the boat in a way that deters as many birds as possible within the time allotted for the session.

The birds' deterrence testbed is a good representation of our problem domain: it contains a dynamic environment where new events of a spatial nature (represented by the birds that arrive and leave) need to be handled by a human-operated robot (the robotic boat), where operators are commonly novices and employed on an ad-hoc basis. Furthermore, evaluating an operator requires knowledge of the environment which is very difficult to extract (birds locations and timings) and the system can only rely on the robot's location along time for this task and traces of the generated routes.

For evaluating the performance of individual operators based on their recorded sessions of operating the boat, we developed a complementary GUI. With this GUI we could load a session and present it to a user (i.e., a crowdworker evaluator) either as is or only selected parts of it. The GUI enabled asking the evaluator at any preferred time to provide a numerical evaluation for the current (or overall so far) effectiveness of the robot navigation. For supporting the experimental design we also enabled controlling the speed of playing the session (or parts).

Experimental Design

Our experiments were based on twenty-minute of boat-operating sessions. The fish farming pond's shape was arbitrarily set to be a circle of 427 pixels radius. Birds' appearance events were randomly set for each session (both arrival time and location). Each arriving bird left after 10 seconds, unless deterred within that time.

The robotic boat's speed was set to $\frac{1}{5}$ pixel per millisecond and the deterrence radius was set to 25 pixels. The total amount of birds was set to 930. This amount, which is greater than the overall deterrence capacity, guaranteed that not even the best operator would be able to deter all birds, based on the boat's speed and pond's size, hence the operator's quality becomes an issue.

We generated 68 different scenarios of the above type. The reason for generating that many scenarios is that we wanted to enable a comparison to machine-learning based methods, as we explain later on, hence we needed enough data for training. Operators were recruited and interacted through AMT. Each participant received thorough instructions on how to operate the boat, the game rules and her goal in the game, followed by a short qualification quiz. Compensation included a small show-up fee and a bonus which linearly depended on the number of deterred birds. Each operator was assigned one of the 68 scenarios described above (with no repetitions), so we ended up with 68 recorded game sessions for which we had every operation taken by the operator recorded. Out of the 68 sessions we picked eight that were used as the set of sessions requiring operator evaluation. The selection of the eight sessions was carried out such that the set will provide as much variance as possible in terms of the real score achieved, the average idle time and average navigation route length - which are considered measures with some correlation to operator's performance. The number of birds deterred in these sessions varied between 403-671.

For each of the eight twenty-minute sessions we produced a four-minute summary, using Algorithm 1 with parameters: $N = 6$ seconds (based on a small scale experiment testing with $N \in [2, 4, 6, 8, 10]$, choosing the value resulting in the best correlation between evaluated score and actual score), $p_{select} = 0.5$ (to obtain a balanced mixture of operations), $\alpha_{idle} = \alpha_{elapsed} = \alpha_{remaining} = 2$, $\alpha_{path} = 1.5$, $\alpha_{pre\_change} = 1.1$ and $\alpha_{post\_change} = 0.9$. We note that while the above parameter choices do not result in a fully tuned summary, the idea is to provide a proof of concept for the effectiveness of the proposed method, rather than to find the optimal configuration. Consequently, we did not check additional configurations, as even with these parameters the method was found to perform better than all other methods we compare to. Still, a close examination of the scores assigned to each of the 1200 seconds of each of the eight test cases, aligning them with the events that took place along the session, reveals that these are indeed good indicators for performance.

For each of the eight sessions that needed to be evaluated, we recruited evaluators (using, once again, AMT) that provided a score for the effectiveness of the operation. Evaluators received a thorough explanation regarding the task and had to pass a short qualification quiz. The evaluators were explicitly asked to evaluate the quality of the operator's work, knowing that it is measured by the number of deterred birds out of the total. Our compensation scheme suggested a small payment for participation and a $2 bonus for providing detailed answers. This choice aligns well with results reported in prior research favoring performance-based payments (PBPs) over fixed hourly rate (Gneezy and Rustichini 2000) as long as the bonus was sufficiently high (Gneezy and Rustichini 2000), and in particular in evaluation tasks (Harris 2011).

Overall, we had 600 evaluators in our experiments. To avoid a carry-over effect, each evaluator was assigned one session only out of the eight, and experienced one of the following tested treatments:

- *Full Session (single score)* - watching the full session and providing a single score at the end (10 participants for each of the eight sessions, 80 participants overall).
- *Full Session (interval scores)* - watching the full session and pausing every two minutes throughout to provide a score for that two minutes period. Also, at the end of the session a single score for the complete session was requested (10 participants for each of the eight sessions, 80 participants overall).
- *Fast Forward (X2 and X5)* - watching the full session

in a fast forward mode, i.e., presenting the changes in the environment at an increased pace, providing a single score at the end. Here we experimented with replaying the session in an X2 (twice-faster) speed and an X5 (five times faster) speed, resulting in sessions of ten-minute and four-minute length, respectively (10 participants for each of the eight sessions and each of the two speeds, 160 participants overall).

- *Random Summary* - watching a four-minute summary of the session composed of random six-second segments from each interval (as opposed to the selection based on Algorithm 1) and providing a single score at the end (10 participants for each of the eight sessions, 80 participants overall).
- *Heuristic Summary* - watching the four-minute summary we created for the session, as specified above, and providing a single score at the end (25 participants for each of the eight sessions, 200 participants overall).

In all the above treatments, the numeric score provided was discrete, between 1-10, where 1 is the worst possible performance, and 10 is the best.[3] Also, in all treatments, in addition to the numerical evaluation, subjects were requested to provide a textual evaluation justifying their score.

Overall, the set of methods we compare to enables both reasoning about the performance that can be achieved by providing evaluators the complete trace (as is, in segments and at a greater pace) and when providing them only a random subset of the trace. This enables a better understanding of the influence different aspects of our proposed heuristic summary method have on the achieved performance (e.g, the shortened trace presented, the scoring function affecting the selection of segments to present).

Results

We randomly draw a single evaluation out of those it received for each of the eight sessions, and calculated the correlation between the set of evaluations (on a scale of 1-10) and the actual number of birds deterred in each. This process was repeated 1000 times. Table 1 depicts the average correlation (over those calculated in each of the 1000 draws) in all five treatments.

Full Single/Interval	*Fast Fwd X2/X5*	*Rand*	*Heuristic*
0.09 / 0.28	0.15 / -0.02	0.13	0.35

Table 1: Average correlation between evaluators' scores and actual performance.

From the table, we observe that while our heuristic summary method uses less or the same amount of evaluator's time, compared to the other methods, it produces more accurate evaluations. Still, the correlation it obtains is somewhat modest. One possible solution for improving evaluation accuracy, which applies to all tested methods, is averaging several evaluations of different evaluators when evaluating a given session. Therefore in the following paragraphs, we focus our analysis on this approach, emphasizing the tradeoff between the overall evaluators' time spent in the evaluation and the performance (in terms of average correlation) obtained.

[3]This scale was set arbitrarily and any other set could be used, as we are only interested in the correlation between the evaluation score and the actual number of birds deterred.

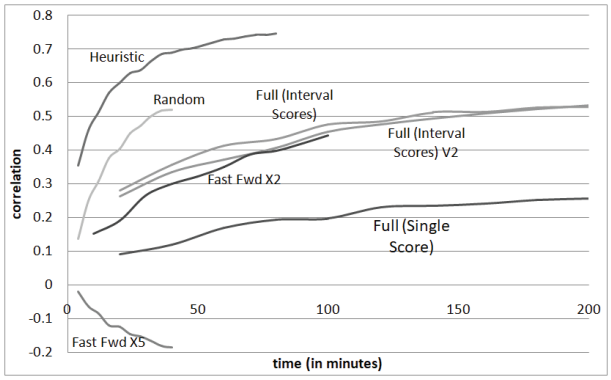

Figure 2: Average correlation between evaluations and actual performance using the eight test case sessions, as a function of the amount of time invested by the evaluators (minutes).

Figure 2 depicts the average correlation between evaluators' average scores and actual performance for the eight sessions used, in all five treatments, as a function of the overall amount of human labor invested (horizontal axis, measured in minutes). Note that data points of different methods that correspond to the same amount of invested effort may be based on a different number of valuations for each session (hence the difference in points' granularity and the extent of the different curves). For example, the investment of twenty minutes is equivalent to a single valuation in Full Session, two valuations in Fast Forward X2, and five valuations in Fast Forward X5, Heuristic Summary and Random Summary. For each amount of operators' time on the horizontal axis, we calculated the corresponding number of evaluators needed with each method and randomly draw that many valuations (with no repetitions) for each of the different sessions. Then, calculated the average valuation for each session and consequently the correlation with actual performance. This was repeated 1000 times, hence each data point represents the average correlation of the 1000 draws. Overall, we observe from Figure 2 that the heuristic summary method dominates all other tested methods by far. It provides a substantially greater correlation to the actual score, for any amount of invested human effort. In the following paragraphs, we provide some additional important insights revealed from the results summarized in Figure 2.

Full Session. The results of the Full Session based on interval scores treatment are presented in the graph using two variants. The first is the correlation with the average overall score received at the end of the session and the second with the average of scores received in the two-minute intervals. Surprisingly, the performance with all three variants tested is quite poor, and even when using 10 evaluators for evaluating each session (equivalent to 200 minutes) the best of the three

achieved a 0.53 correlation between prediction and actual score. This level of performance can be achieved using a random summary with a fifth of the effort (40 compared to 200 minutes). These results are particularly disappointing because when watching the complete session the evaluator can potentially count the number of birds deterred and provide an exact answer, i.e., reaching a correlation of 1 even with a single evaluator.

Interestingly, the two variants in which the evaluators were requested to provide an evaluation every two minutes resulted in 2 to 3 times better performance compared to the variant where evaluators had to watch the complete session before providing their valuation. One possible explanation for this is that it is somehow difficult for people to recall and equally address in terms of the weights assigned to all the events that took place throughout the session. It is also possible that evaluators found the task of focusing on the boat's route for twenty minutes to be somewhat boring, and some of them did not pay attention, at times, to the happenings on the screen. Shortening the evaluation session enables overcoming these problems as the evaluation is made with all events still fresh in one's memory. This can explain the dominance of averaging the scores given to two-minute intervals over getting one overall score after watching the twenty-minute footage. However, our results suggest that an even better performance is obtained if the two-minute valuations results received are discarded, relying only on the additional global score provided. Apparently, requesting the two-minute scores helps in getting the worker's attention to events throughout the session and in 'summarizing' some of the intervals in a structured manner (i.e., through the scores assigned).

Fast Forward. The performance of the Fast Forward treatment seems to be highly influenced by the speeding factor. Indeed, with this mechanism, the evaluator still gets to see the entire sequence of events, and apparently, with rather a moderate speeding (X2), she does not lose much compared to watching at a normal speed, as performance is better than with Full Session with a single score. Still, performance falls short compared to the two variants of the Full Session with interval scores. This can either be explained by the fact that much like with Full session based on a single score the evaluators may be bored and somehow inattentive when having to watch a long session, or by some additional cognitive load incurred from having to watch events in a pace quicker than their actual happening. To test the first explanation some additional experimentation is required, asking evaluators to provide sub-scores every minute when watching the session in X2 speed. However, for the second explanation, we can find strong support in the results of the X5 treatment. Here we observe a reverse correlation which increases as the number of evaluations received increases. Meaning that evaluators could not map good performance to high score whatsoever.

Random Summary. This method is found to be second only to our proposed heuristic summary approach. The performance it converges to is the same as the best of the Full Session methods converges to, except that it reaches this level of performance with fifth(!) of the evaluators' time investment. Meaning that the summary by itself is instrumental in achieving an effective score, possibly by keeping the evaluators focused and preventing loss of important insights. Even though missing a large portion of the happenings, the result seems to capture a somewhat effective sample of the operator's attitude towards the task and representatively reflect her performance. Still, for any amount of evaluators' time invested, the average scoring received with our heuristic summary method is better correlated with actual performance, compared to the use of Random Summary. In fact, our method converges to a 50% greater average correlation (0.75 compared to 0.51). This suggests that our selection of events of interest to be included in the summary is highly effective and accounts, in large, to the high quality of the summary produced. This can also be learned from the textual feedback received from Turkers, e.g., random summary reported that it was hard to evaluate the boat operator 'at times it did seem like some important parts were missing' (in the random summary) vs. 'I thought it was adequate to provide enough context for the operator's actions and not be overly long' (in the heuristic summary).

Heuristic Summary. As already mentioned above, we observe from Figure 2 that the heuristic summary dominates all other tested methods, for any amount of invested operators effort, by far. Furthermore, the variance in the correlations obtained when using our method is significantly smaller than with the other methods. This latter attribute is of great importance due to the consequences of a wrong valuation (e.g., continue hiring a wrong operator or terminating the work of a good one).

ML-Based Prediction

Our heuristic summary approach for solving the operator evaluation problem can be considered as behavioral-based—it relies on analyzing operator's actions according to some behavioral model (i.e., based on behavioral features) for generating a summary that can then be used for enhancing the collective wisdom of crowdworkers in producing a proper evaluation. One alternative approach for solving the problem would be the use of machine learning for constructing a prediction model that could predict scores based on those behavioral features.

Machine Learning algorithms were developed for formal domains, which are materially different from social domains (Shmueli 2017; Yahav, Shehory, and Schwartz 2018). Still, recent research has found much merit in integrating data science and social science, in the context of predicting human choice behavior (Plonsky et al. 2017). Therefore, we took a similar approach and used several standard machine learning techniques (K-Nearest Neighbors, Neural Network and Random Forest) for predicting the operator's effectiveness based on observed behavioral features. The method found to perform best is Random Forest (RF) hence we focus our analysis on its results. Random Forest is an ensemble (or set) of classification or regression trees. Each tree in the ensemble is built based on the principle of recursive partitioning, where

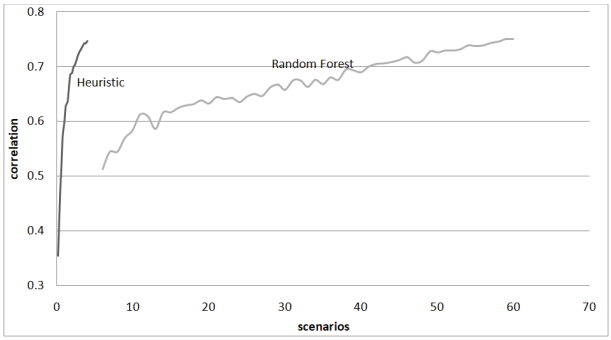

Figure 3: Average correlation when using Random Forest prediction and our Heuristic Summary, as a function of the training set size (or its equivalent crowdworker time).

the feature space is recursively split into regions containing observations with similar response values.

Our implementation used the 'scikit-learn' Python module (Pedregosa et al. 2011) with 100 trees. The best predicting features were found to be the overall boat idle time and the average route length the boat traveled within the session. These, together with the number of birds deterred out of the total amount introduced within the sessions, were used as an input for the learned model. For the training set, we used 60 full-lengths (twenty-minute) sessions. For the test set, we used the eight sessions that were used for testing the alternative methods as described above.

Figure 3 depicts the average correlation between actual and predicted performance using Random Forest, as a function of the training set size. The procedure used for calculating each point is similar to the one used in Figure 2 - for each training set size we randomly sampled that amount of sessions out of the 60 available and used those for training the model. Then, we used the model for predicting performance in each of the eight sessions of the test set, and calculated the correlation with the actual number of deterred birds. This was repeated 1000 times for each training set size, hence each point in the graph is the average of 1000 calculated correlations. The figure also includes a curve depicting the performance achieved with the heuristic summary for the equivalent amount of crowdworkers' time (i.e., each twenty-minute operator's session is equivalent to five four-minute-based evaluations). From the Figure we observe that the use of 60 sessions is sufficient for performance to converge, reaching a correlation of 0.75. This level of performance is similar to what we manage to achieve with our behavioral-based summarization approach. Still, with the proposed heuristic summary, we reached this level of performance with 80 minutes of evaluators' time. With Random Forest we had to learn from 60 sessions, which are equivalent to 1200 operators-minutes (15 times more!).

We emphasize that despite the similar results obtained, the two approaches are completely different and associated with different kinds of advantages and disadvantages. For example, with the ML-method training data needs to be collected only once, whereas in our method evaluators need to be employed separately for each session. On the other hand, our method results can improve if using better evaluators (which is irrelevant when relying on ML). Better evaluators can be identified based on repeated interaction, comparing valuations and actual performance, in retrospect or by checking how close one's valuations are to the consensus (e.g., to the average of valuations received from others). Moreover, we note that the summary generated with our method is self-contained, meaning that it can be used for various other purposes, e.g., for training the operators themselves or prospective operators through examples.

Finally, we note that since the ML-based prediction and the proposed heuristic summary method are so different in their essence, they can be used together as a means for providing an even better prediction. In our domain, taking the average of the score predictions produced by the two methods for each tested setting we end up with a correlation of 0.83 with the actual number of deterred birds, compared to 0.75 with each of the methods separately.

Conclusions

The encouraging results reported in the former sections suggest that indeed the heuristic summary method is a highly effective alternative to evaluations produced based on watching full sessions, fast-forwarding the session or observing random pieces from the session. With the summaries generated, one can achieve the same quality of evaluation (as with the other methods) with a substantially fewer evaluators' effort, or a substantially more accurate evaluation for a similar effort. The comparison to random summaries may seem somewhat naive, and indeed that choice was made merely due to lack of any proven segment selection heuristic in prior work. Still, we emphasize that the results indicate that the random selection achieves better correlation than all other non-summary-based methods tested, hence its importance.

One apparent dominating behavioral factor influencing a segment's score in the decision whether to include it in the summary is the worker's activity level. While this may seem more adequate to scenarios where events are frequent, we note that by properly setting the weight assigned to idle events or the length of the summary to be produced (and consequently the length of the intervals considered), one can capture sufficient activity even when events are infrequent.

While ML-based methods are found to produce similar evaluation accuracy, the heuristic summary method requires substantially less human effort to deliver. Indeed with the ML-based algorithms, this effort is a one-time investment. Still, there are several other advantages in favor of the heuristic summary as discussed in the former section. In particular, as we demonstrate, the two methods can co-exist and complement each other, yielding an even better prediction.

We emphasize that the results obtained with the heuristic summary as reported in this paper, are actually lower bounds for the performance one may achieve if using it fully tuned, in terms of the different parameters used. In particular, we hypothesize that by further increasing the portion of high-score events (at the expense of low-score events) better results may be obtained as people are known to be more critical of mistakes than successes.

We see many directions for future research emerging from this paper, out of which we detail three. First, as mentioned above, much work is still needed for finding good ways for tuning the proposed method to perform optimally in different application domains. Second, we believe the summary generation can benefit from dynamic segment length selection, adjusting the number of seconds to show before and after an event to its nature. This, however, would require further optimization as the number of 'events' to show will now need to depend on the nature of events picked. Finally, we see a great potential in using machine-learning-based methods for predicting the most informative segment within each interval, as an alternative for our behavioral-based scoring function.

Acknowledgements

This research was partially supported by the ISRAEL SCIENCE FOUNDATION (grant No. 1162/17) and the Israeli MINISTRY OF SCIENCE & TECHNOLOGY (grant No. 89583).

References

Amir, O.; Doshi-Velez, F.; and Sarne, D. 2019. Summarizing agent strategies. *Autonomous Agents and Multi-Agent Systems* 33(5):628–644.

Brooks, D. J.; Shultz, A.; Desai, M.; Kovac, P.; and Yanco, H. A. 2010. Towards state summarization for autonomous robots. In *AAAI Fall Symposium: Dialog with Robots*, volume 61, 62.

Bruemmer, D. J.; Boring, R. L.; Few, D. A.; Marble, J. L.; and Walton, M. C. 2004. " i call shotgun!": an evaluation of mixed-initiative control for novice users of a search and rescue robot. In *Systems, Man and Cybernetics*, volume 3, 2847–2852.

Crick, C.; Osentoski, S.; Jay, G.; and Jenkins, O. C. 2011. Human and robot perception in large-scale learning from demonstration. In *Proceedings of the 6th international conference on Human-robot interaction*, 339–346.

DiPalantino, D., and Vojnovic, M. 2009. Crowdsourcing and all-pay auctions. In *Proc. of ACM-EC*, 119–128.

Dodson, T.; Mattei, N.; and Goldsmith, J. 2011. A natural language argumentation interface for explanation generation in markov decision processes. *Algorithmic Decision Theory* 42–55.

Doshi-Velez, F., and Kim, B. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Elizalde, F. 2008. Policy explanation in factored markov decision processes. In *Proceedings of PGM*, 97–104.

Elmalech, A.; Sarne, D.; David, E.; and Hajaj, C. 2016. Extending workers' attention span through dummy events. In *Fourth AAAI Conference on Human Computation and Crowdsourcing*, 42–51.

Finnerty, A.; Kucherbaev, P.; Tranquillini, S.; and Convertino, G. 2013. Keep it simple: Reward and task design in crowdsourcing. In *Proc. of SIGCHI*, 14:1–14:4.

Gao, C., and Zhou, D. 2013. Minimax optimal convergence rates for estimating ground truth from crowdsourced labels. *arXiv preprint arXiv:1310.5764*.

Gneezy, U., and Rustichini, A. 2000. Pay enough or don't pay at all. *The Quarterly journal of economics* 115(3):791–810.

Harris, C. 2011. You're hired! an examination of crowdsourcing incentive models in human resource tasks. In *Proceedings of Crowdsourcing for Search and Data Mining (CSDM)*, 15–18.

Hayes, B., and Shah, J. A. 2017. Improving robot controller transparency through autonomous policy explanation. In *Proceedings of HRI*, 303–312.

Jakuba, M. V.; German, C. R.; Bowen, A. D.; Whitcomb, L. L.; Hand, K.; Branch, A.; Chien, S.; and McFarland, C. 2018. Teleoperation and robotics under ice: Implications for planetary exploration. In *2018 IEEE Aerospace Conference*, 1–14.

Kosti, S.; Sarne, D.; and Kaminka, G. A. 2014. A novel user-guided interface for robot search. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3305–3310.

Lewis, M.; Sycara, K.; and Nourbakhsh, I. 2019. Developing a testbed for studying human-robot interaction in urban search and rescue. In *Proceedings of HCII'03*, 270–274.

Lomas, M.; Chevalier, R.; Cross II, E. V.; Garrett, R. C.; Hoare, J.; and Kopack, M. 2012. Explaining robot actions. In *Proceedings of HRI*, 187–188.

Machlev, N., and Sarne, D. 2020. Predicting crowdworkers' performance as human-sensors for robot navigation. In *Proceedings of AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*. forthcoming.

Mason, W., and Watts, D. J. 2010. Financial incentives and the performance of crowds. *ACM SigKDD Explorations Newsletter* 11(2):100–108.

McGinn, C.; Sena, A.; and Kelly, K. 2017. Controlling robots in the home: factors that affect the performance of novice robot operators. *Applied ergonomics* 65:23–32.

Nikolaidis, S., and Shah, J. 2013. Human-robot cross-training: computational formulation, modeling and evaluation of a human team training strategy. In *Proceedings of HRI*, 33–40.

Nourbakhsh, I.; Sycara, K.; Koes, M.; Yong, M.; Lewis, M.; and Burion, S. 2005. Human-robot teaming for search and rescue. *Pervasive Computing, IEEE* 4(1):72–79.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Plonsky, O.; Erev, I.; Hazan, T.; and Tennenholtz, M. 2017. Psychological forest: Predicting human behavior. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Polin, M. R.; Siddiqui, N. Y.; Comstock, B. A.; Hesham, H.; Brown, C.; Lendvay, T. S.; and Martino, M. A. 2016. Crowdsourcing: a valid alternative to expert evaluation of

robotic surgery skills. *American journal of obstetrics and gynecology* 215(5):644–e1.

Rahman, D. 2012. But who will monitor the monitor? *The American Economic Review* 2767–2797.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *Proc. of ACM International Conference on Knowledge Discovery and Data Mining*, 1135–1144. ACM.

Schulz, D.; Burgard, W.; Fox, D.; Thrun, S.; and Cremers, A. B. 2000. Web interfaces for mobile robots in public places. *IEEE Robotics & Automation Magazine* 7(1):48–56.

Shmueli, G. 2017. Analyzing behavioral big data: methodological, practical, ethical, and moral issues. *Quality Engineering* 29(1):57–74.

Toris, R.; Kent, D.; and Chernova, S. 2014. The robot management system: A framework for conducting human-robot interaction studies through crowdsourcing. *Journal of Human-Robot Interaction* 3(2):25–49.

Velagapudi, P.; Wang, J.; Wang, H.; Scerri, P.; Lewis, M.; and Sycara, K. 2008. Synchronous vs. asynchronous video in multi-robot search. In *First International Conference on Advances in Computer-Human Interaction*, 224–229.

Vellido, A.; Martín-Guerrero, J. D.; and Lisboa, P. J. 2012. Making machine learning models interpretable. In *ESANN*, volume 12, 163–172.

Wang, H.; Chien, S. Y.; Lewis, M.; Velagapudi, P.; Scerri, P.; and Sycara, K. 2009a. Human teams for large scale multirobot control. In *2009 IEEE International Conference on Systems, Man and Cybernetics*, 1269–1274.

Wang, H.; Lewis, M.; Velagapudi, P.; Scerri, P.; and Sycara, K. 2009b. How search and its subtasks scale in n robots. In *Proceedings of HRI*, 141–148.

Wang, H.; Velagapudi, P.; Scerri, P.; Sycara, K.; and Lewis, M. 2009c. Using humans as sensors in robotic search. *FUSION'09* 1249 – 1256.

Wang, H.; Kolling, A.; Brooks, N.; Owens, S.; Abedin, S.; Scerri, P.; Lee, P.-j.; Chien, S.-Y.; Lewis, M.; and Sycara, K. 2011. Scalable target detection for large robot teams. In *HRI'11*, 363–370.

Wang, J.; Kraska, T.; Franklin, M. J.; and Feng, J. 2012. Crowder: Crowdsourcing entity resolution. *arXiv preprint arXiv:1208.1927*.

Yahav, I.; Shehory, O.; and Schwartz, D. 2018. Comments mining with tf-idf: the inherent bias and its removal. *IEEE Transactions on Knowledge and Data Engineering* 31(3): 437–450.

Yin, M.; Chen, Y.; and Sun, Y.-A. 2014. Monetary interventions in crowdsourcing task switching. In *Proc. of HCOMP*, 234–241.