The Thirty-Third International
FLAIRS Conference (FLAIRS-33)

Using Reinforcement Learning to Manage Communications Between Humans and Artificial Agents in an Evacuation Scenario

Skanda Vaidyanath*
BITS Pilani
Hyderabad Campus
India

Kallirroi Georgila
Institute for Creative Technologies
University of Southern California
USA

David Traum
Institute for Creative Technologies
University of Southern California
USA

Abstract

In search and rescue missions, robots can potentially help save survivors faster than human emergency responders alone would. In our experimental virtual reality simulation environment we have a system which comprises a swarm of unmanned aerial vehicles (UAVs) and a virtual "spokesperson". The system and a human operator work together on locating and guiding survivors to safety away from an active wildfire encroaching on a small town. The UAVs and the spokesperson are equipped with natural language capabilities through which they can communicate with the survivors to convince them to evacuate. If they fail to do so they can ask the human operator to intervene. We use reinforcement learning to automatically learn a policy to be followed when a UAV has located survivors. The system learns the best course of action to help the survivors evacuate. We vary the distance of the fire, the level of cooperativeness of the survivors, and how busy the human operator is, and we report results in terms of percentage of survivors saved in each condition.

Introduction

Using unmanned aerial vehicles (UAVs) or swarms of UAVs for search and rescue during emergencies is a well established idea (Kolling et al. 2016). In a swarm of UAVs, the UAVs need to coordinate based on shared information and distributed algorithms. The UAVs should also be able to communicate with survivors and relay information to first responders. Currently it is common practice to have multiple human operators control multiple UAVs, which can be inefficient. However, the alternative of having one human operator control a swarm of UAVs is an open research problem, and a major issue is the cognitive complexity involved in having one human control multiple UAVs simultaneously (Kolling et al. 2016). Therefore, there is a strong need for UAVs with autonomous capabilities and virtual agents to ease the burden on human operators.

We have built a virtual reality environment depicting an active wildfire encroaching on a small town (Chaffey et al. 2019). A human player assumes the role of a swarm operator and is tasked with deploying the UAVs at certain areas of the town, and directly communicating with residents in danger from the fire. There is also a virtual human ("spokesperson"), who assists with tasks when multiple incidents occur simultaneously, to reduce the cognitive load on the human operator. The system (UAVs and spokesperson) and the human operator work together on locating and guiding survivors to safety away from the wildfire. The UAVs and the spokesperson are equipped with natural language capabilities through which they can communicate with the survivors to convince them to evacuate. If they fail to do so they can ask the human operator to intervene.

Currently the spokesperson is implemented in a Wizard of Oz (WOz) setup where a human wizard plays the role of the spokesperson, while the human operator thinks that she interacts with a real system. The human wizard follows a predefined set of capabilities and language response protocol. The wizard's responses are converted to audio via a speech synthesizer. The civilians' responses are in the form of prerecorded audio and their activation is also controlled by the human wizard. The WOz setup enables us to collect realistic interaction data between the human operator and the system without the constraints posed by speech recognition and natural language understanding limitations (Marge et al. 2017). Our ultimate goal is to have a fully automated system where both the spokesperson and the UAVs have autonomous capabilities. Thus, in future work, the data collected in the WOz setup will be used for building models for automated natural language processing and swarm management.

This paper focuses on one specific aspect of our system automation, namely, when the system should rely on the UAVs and the spokesperson to guide the survivors to safety, and when it should prompt the human operator to intervene and communicate directly with the survivors. The latter would happen if the survivors refused to move despite multiple warnings. We use reinforcement learning to learn a policy to be followed when a UAV has located residents in danger from the fire in order to help the survivors evacuate.

Our experiments are performed in simulation and, given that we allow for potential randomness in setting up the problem, our results are promising (see results section).

*This work was done during the first author's internship at the University of Southern California.

Related Work

While there is a plethora of work on designing robot swarms, less attention has been placed toward human-robot interaction in the search and rescue domain.

Casper and Murphy (2003) did a post-hoc analysis on the data collected during the World Trade Center crisis response, and made recommendations which called for more research on perceptual and assistive interfaces to help emergency personnel handle robots more effectively, and in turn reduce the cognitive load on first responders.

Sycara and Lewis (2012) argue that, as the number of UAVs increases, the need for coordination among UAVs as well as for assistance to the human operator exceeds current state-of-the-art capabilities. Thus, in order to achieve a wider deployment of robots for practical tasks in various areas we need to expand human span of control over teams of robots. They have developed a taxonomy of human-robot tasks and appropriate human-robot interaction modes based on complexity of control and cognitive demands placed on human operators. According to their theory, there are 3 types of human-robot interactions: (1) autonomous coordination of UAVs that can be controlled as a swarm by a human operator which works only in limited scenarios, (2) control of each of the UAVs independently by the human operator, and (3) not only control of UAVs but also direct involvement of the human operator in their coordination. The third type of human-robot interaction can be very difficult and performance deteriorates when a human operator is asked to control and coordinate the decisions of more than 8-12 robots (Wang et al. 2009). This result motivates our use of a virtual spokesperson as a means to reduce the cognitive load on the human operator, and in turn allow for the simultaneous control and coordination of a large number of robots.

Reinforcement Learning

Reinforcement Learning (RL) is a method for learning the policy of an agent that takes a sequence of actions to maximize some notion of a "reward". Here we model the RL problem as a Markov Decision Process (MDP). An MDP is defined as a tuple $< S, A, P, R, \gamma >$ where S is the set of all states that the agent may be in, A is the set of all actions that are available for the agent to take, $P : S \times A \rightarrow P(S, A)$ is the set of transition probabilities between states after taking an action, $R : S \times A \rightarrow \Re$ is the reward function, and $\gamma \in [0, 1]$ is a discount factor weighting long-term rewards. At any given time step t, the agent is in a state $s_t \in S$ and chooses an action $a_t \in A$, for which it receives a reward r_{t+1} according to the reward function R, and transitions to state s_{t+1} according to the transition probability $P(s_{t+1}|s_t, a_t)$. The agent selects an action based on the average reward it has previously observed after having performed that action in similar contexts (during training), the so-called Q-function.

We experimented with two categories of model-free RL algorithms – Monte Carlo simulations (on-policy and off-policy Monte Carlo) and Temporal Difference Learning methods (Q-learning, SARSA, and expected SARSA). The on-policy Monte Carlo method performed the best in our setup, so below we focus on this algorithm and its results. With the Monte Carlo algorithm, we make updates to the Q-values only at the end of each episode. We implement the *first-visit* version of the Monte Carlo algorithm, which means that we sample an episode from the environment and for every unique state-action pair in the episode, we calculate the total returns from *the first occurrence* of that state-action pair until the end of the episode. We then set the Q-value of that state-action pair to the average of all such observations over many episodes. For exploration, we use an ϵ-greedy policy and we gradually decay ϵ from a value very close to 1 to a value very close to 0, encouraging exploration in the initial episodes and exploitation in the later ones. We also set γ to be equal to 0.95.

Experimental Design

In our setup there is an active wildfire encroaching on a small town and a UAV from the swarm is currently in contact with a civilian group. The UAV has to employ a series of actions (e.g., warnings) to convince the civilian group to leave their homes before the fire reaches their location. Furthermore, once the group has been convinced, the UAV also has to guide the group to safety in the manner that the group prefers. There is one human operator that is available for the UAV to interrupt when it feels that it cannot convince the group in time. However, the UAV must only interrupt the human operator when she is not busy and when the situation is hopeless (i.e., the fire is too close). There is also a virtual spokesperson that can negotiate with the group through the UAV. The UAV itself is only capable of issuing pre-recorded warning messages. Any sort of "negotiation" must happen through the operator or the spokesperson. Here we are not concerned with how an actual negotiation between the spokesperson or the human operator and the civilians may unfold. Dealing with negotiation and persuasion strategies is part of our future work. Currently our goal is to learn a policy regarding when to warn the civilians through the UAV, when to warn them through the spokesperson, and when to have the human operator intervene.

There are 3 possible levels of communication with the survivors. In the first level the UAVs warn the survivors that they need to evacuate. If the first level fails then in the second level the spokesperson attempts to persuade the survivors to evacuate. If the second level fails too then in the third level it is the human operator's turn to assume responsibility for convincing the residents to move. The cost of having the spokesperson engage in conversation with the survivors is higher than the cost of just warning them through the UAV. This is because the spokesperson is responsible for multiple tasks whereas the UAV is already occupied with monitoring the survivors. Similarly, the cost of having the human operator engage in dialogue with the survivors is higher than the cost of having the spokesperson or the UAV try to persuade the survivors to evacuate, especially when the human operator is busy handling more urgent situations. Thus, the human operator should be asked to intervene only when there is no other choice. At the same time, the proximity of the fire is an unpredictable factor that should also be taken into account. If the fire is rapidly approaching and the residents refuse to

evacuate then there may not be enough time for warnings through the UAV or conversation between the residents and the spokesperson. In that case, the human operator should be asked to intervene no matter how busy she is. So we need to optimize the use of our resources and at the same time ensure that the civilians are guided to safety.

We simulate 3 different types of civilian groups – the stubborn couple, the old couple, and the babysitter and the child. The 3 groups behave differently when engaging with the UAV (or the spokesperson or the human operator), and also have different ways in which they would like to be led out to safety. Note that *the system does not know which of the 3 groups it is currently dealing with* so the policy that we learn must generalize to all 3 groups. We only consider the interaction of *a single UAV with a single civilian group*. Multiple instances of the same policy can be initialized for different UAVs in the swarm. Since the policy works irrespective of the type of civilian group, this is an acceptable setup. Below we provide details about how we set up the RL problem.

Actions

The policy actions that we consider are as follows:

1. *Warn*: The UAV issues a pre-recorded warning message.
2. *Allow-spokesperson-to-negotiate*: The UAV opens a communication channel for the virtual spokesperson to negotiate with the civilian group.
3. *Interrupt-operator*: The UAV interrupts the human operator and opens a communication channel for the operator to negotiate with the civilian group.
4. *Query-for-guidance-info*: Ask the civilian group for information about how they would like to be guided to safety. The possible guidance options are given below as additional actions.
5. *UAV-guide*: The UAV guides the group to safety.
6. *Vehicle-guide*: The UAV calls for a vehicle to guide the group to safety.
7. *Wait*: The UAV waits and does nothing.

State Variables

1. *Operator-busyness-level*: This variable gives an indication of how busy the human operator is and when it is appropriate to interrupt her. Values are integers ranging from 0 (not busy) to 3 (very busy). This variable is initialized to a random value.
2. *Group-status*: This variable indicates the current status of the group or status of the negotiation and takes one of the following values, 0: Being-monitored, 1: Being-warned, 2: Spokesperson-negotiating, 3: Operator-negotiating, 4: Group-convinced, 5: Group-saved. An episode always starts with the value of the *Group-status* variable being equal to 0.
3. *Fire-approach-time*: The time taken for the fire to reach the group. This variable takes integer values from 4 (fire is far away) to 0 (fire is at the location of the group and the episode ends).
4. *Preferred-guidance-type*: The preferred way in which the group would like to be guided to safety and takes one of the following values, 0: Unknown, 1: Self-guided, 2: Guided-via-UAV, 3: Guided-via-vehicle. This variable is always initialized to 0, and set to value 1, 2, or 3 once the agent selects the *Query-for-guidance-info* action after the group has been convinced (*Group-status* = 4).
5. *Negotiation-status*: This variable gives us an indication of the negotiation strategies we have tried already and can take one of the following values, 0: no form of negotiation/warning attempted, 1: warnings issued but no negotiations from the spokesperson, 2: the spokesperson has attempted a negotiation. The variable is initialized to 0.

Thus, overall we have a total of 1440 states, 7 actions, and 10080 state-action pairs.

Reward Function

We give the agent a reward of $+5000$ for every group it manages to save in time and a -5000 for every group that it does not. To discourage the agent from constantly interrupting the operator, we give it a reward of $-(300 + op\_busy * 500)$ where op_busy is the *Operator-busyness-level* state variable. We also include some reward shaping by providing a reward of -3000 for selecting one of the guide actions (*UAV-guide* or *Vehicle-guide*) or the *Query-for-guidance-info* action when the group has not been convinced yet.

To simulate these actions accurately, we need to recognize that the actions *Warn*, *Allow-spokesperson-to-negotiate*, and *Interrupt-operator* are not instantaneous. A warning or a negotiation takes time and to account for that in our simulations we use probabilities. We assume that once a warning has been issued, the agent goes into a state where the *Group-status* variable is set to *Being-warned*. Now for every action that the agent plays from this state, there is a 90% chance that the warning ends, and the *Group-status* variable goes back to *Being-monitored* or *Group-convinced*. If the warning does not end then the *Group-status* variable remains set to *Being-warned*. For the negotiation-related actions (*Allow-spokesperson-to-negotiate* and *Interrupt-operator*), we take it one step further in trying to model the time taken for the actions as accurately as possible. We recognize that the human operator and the virtual spokesperson are intelligent and know how much time is left before the fire reaches the location of the group. Hence, they would engage in a longer negotiation if they had more time left and vice versa. We also note that it is the easiest to convince the babysitter and hardest to convince the stubborn couple. This means that the time taken for a negotiation with the babysitter would be less than the time taken for the old couple which would be less than the time taken with the stubborn couple. We define the probabilities required to model the time taken for a negotiation (with the spokesperson or the operator) as follows:

$$P = \frac{factor}{(1+\textit{Fire-approach-time})}$$

$$\text{where } factor = \begin{cases} 1.75, & \text{if babysitter and child} \\ 1.50, & \text{if old couple} \\ 1.25, & \text{if stubborn couple} \end{cases}$$

and P is the probability

Fire approach time	Babysitter and child	Old couple	Stubborn couple
Percentage of civilians saved (%)			
2	75.89	72.52	72.42
3	93.22	91.30	84.18
4	99.26	96.11	95.14
Average number of system actions per episode			
2	4.41	4.64	4.87
3	5.05	5.38	6.73
4	4.58	7.17	7.52

Table 1: Percentage of civilians saved and average number of system actions for each one of the civilian groups and *Fire-approach-time* values, during testing (10000 episodes).

So once the agent plays the action *Allow-spokesperson-to-negotiate* or *Interrupt-operator*, then the *Group-status* variable changes to *Spokesperson-negotiating* or *Operator-negotiating* respectively, and there is a probability P of the *Group-status* variable getting set back to *Being-monitored* or *Group-convinced*. Most likely the babysitter and the child, the old couple, and the stubborn couple are going to be convinced by a warning, the spokesperson, or the operator, respectively, unless there is no time and the fire reaches them first. If the negotiation does not end then the *Group-status* variable remains unchanged. For every action the agent plays, there is a 20% chance that the *Fire-approach-time* is reduced by one and a 70% chance that the *Operator-busyness-level* is reduced by one if it is not already 0. Thus these variables are updated through the episode.

We learn 3 different policies by initializing the *Fire-approach-time* variable to values $2, 3$ and 4. Each policy works for all 3 civilian groups. We train each policy for 1 million episodes and test it for 10000 episodes. Also, in each episode the level of busyness of the human operator is randomly initialized.

Results

Table 1 shows our results for each one of the 3 policies (*Fire-approach-time* = 2, 3, 4) in terms of percentage of civilians saved and average number of system actions, during testing (10000 episodes). Note that the average number of system actions required to save the babysitter and the child is marginally higher for *Fire-approach-time* = 3 than for *Fire-approach-time* = 4. Although this seems counter-intuitive, we believe that it is a side-effect of the fact that the agent must learn a policy that adapts to all 3 groups and possibly due to some variation (because of probabilistic updates) when generating episodes during testing. Our results are promising. When the interaction starts with the fire not being in the immediate vicinity, the system saves the civilians more than 95% of the time. When the interaction starts with the fire being not too far but not too close either, the success rate drops to about 90% on average. Finally, when the interaction starts with the fire being very close, the success rate drops to about 74% on average. These success rates depend on the level of cooperativeness of the civilians and on the location of the fire, which are randomly initialized and updated probabilistically. So there are many cases where the system cannot save the civilians even if it does everything perfectly because there is simply not enough time or the civilians refuse to cooperate.

Conclusion

We used RL to learn a policy to be followed when a UAV has located survivors. The system learned the best course of action to help the survivors evacuate. We did not make major assumptions about the behavior of the survivors, the distance of the fire, or the human operator's level of busyness. To be tested in more realistic conditions, more advanced models could be used but the basic methodology would still apply.

Although there has been previous research on optimizing human operation of UAVs, this work is unique in also including a model of other humans interacting with the UAVs (i.e., the survivors) and the use of a virtual spokesperson as a means of reducing the cognitive load on the human operator.

Acknowledgments

This work was supported by the U.S. Army. Statements and opinions expressed do not necessarily reflect the policy of the United States Government, and no official endorsement should be inferred. The first author was supported by the IUSSTF-Viterbi program.

References

Casper, J., and Murphy, R. R. 2003. Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 33(3):367–385.

Chaffey, P.; Artstein, R.; Georgila, K.; Pollard, K. A.; Gilani, S. N.; Krum, D. M.; Nelson, D.; Huynh, K.; Gainer, A.; Alavi, S. H.; Yahata, R.; and Traum, D. 2019. Developing a virtual reality wildfire simulation to analyze human communication and interaction with a robotic swarm during emergencies. In *Proceedings of the 9th Language and Technology Conference*.

Kolling, A.; Walker, P.; Chakraborty, N.; Sycara, K.; and Lewis, M. 2016. Human interaction with robot swarms: A survey. *IEEE Transactions on Human-Machine Systems* 46(1):9–26.

Marge, M.; Bonial, C.; Foots, A.; Hayes, C.; Henry, C.; Pollard, K. A.; Artstein, R.; Voss, C. R.; and Traum, D. 2017. Exploring variation of natural human commands to a robot in a collaborative navigation task. In *ACL Workshop on Language Grounding for Robotics*.

Sycara, K., and Lewis, M. 2012. Human control strategies for multi-robot teams. In *WSEAS International Conference on Computers*.

Wang, H.; Lewis, M.; Velagapudi, P.; Scerri, P.; and Sycara, K. 2009. How search and its subtasks scale in N robots. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*.