7-2004

Learning from Real-Time Over-the-Shoulder Instructions in a Dynamic Task

Wai-Tat Fu
Carnegie Mellon University

Daniel Bothell
Carnegie Mellon University

Scott Douglass
Carnegie Mellon University

Craig Haimson
Aptima, Inc.

Myeong-Ho Sohn
Carnegie Mellon University

See next page for additional authors

Authors

Wai-Tat Fu, Daniel Bothell, Scott Douglass, Craig Haimson, Myeong-Ho Sohn, and John Anderson

Learning From Real-Time Over-The-Shoulder Instructions in a Dynamic Task

Wai-Tat Fu[1], Daniel Bothell[1], Scott Douglass[1], Craig Haimson[2], Myeong-Ho Sohn[1], John Anderson[1]
[1]Carnegie Mellon University, Pittsburgh, PA 15213, USA
[2]Aptima, Inc, Washington, DC 20005, USA

Abstract

We extended the work of Anderson et al. (in press) by modeling how people learn from "over-the-shoulder" instructions – instructions given immediately after actions were executed – while participants were working on a Anti-Air Warfare Coordinator (AAWC) task. Specifically, we modeled the incremental top-down influence of instructions on the visual search process. We constructed a model that first responded to and converted the over-the-shoulder instructions into declarative memory chunks. These declarative chunks of instructions were strengthened with repeated exposures to these instructions. The model then incrementally learned to improve the selection of next track as the strengths of these declarative chunks of instructions increased, and performance declined as the instructions decayed over time. The model was able to fit the data well, suggesting that the model was able to capture the effects of the instructions on learning and performance in this dynamic task.

Introduction

Anderson et al. (in press) describe a system in the ACT-R cognitive architecture (Anderson & Lebiere, 1998) that takes the initial set of instructions for a dynamic task and converts the set of instructions into a declarative representation, which is interpreted by a generic set of production rules. Through the production compilation mechanism, a set of specific productions that directly perform the task are generated as the task is practiced. We extended this effort by constructing a model that learns from both initial instructions and instructions given by the system while participants are doing the task. These "over-the-shoulder" instructions are given to provide real-time feedback on participants' actions, so that eventually the initial set of instructions can be strengthened, complemented, or overridden. These instructions are given based on a cognitive tutor, a technology that has been used successfully in a variety of educational applications to facilitate learning by doing (e.g. Anderson, Corbett, Koedinger, & Pelletier, 1995).

Learning from Cognitive Tutors

Cognitive tutors were computer-based instructional technology to help students learn from instructions. The idea is that instructions should be given based on a cognitive model of the competence that the student is being asked to learn. In other words, the cognitive model should incorporate the underlying skills that allow the model to perform the task the student was expected to perform. In ACT-R, the skills will be represented as a set of production rules (procedural knowledge) for executing the correct actions under the right conditions, and as a set of declarative memory elements (declarative knowledge) for the appropriate domain knowledge. Based on the model, the tutor is able to monitor actions of the student and infer the intentions of the student. Immediate feedback can then be given to the student to facilitate learning. A major challenge in building the model is to understand how people respond and learn from immediate feedback on their actions from the tutor, and how it influences the existing declarative and procedural knowledge. In this paper, we describe a model of how people learn from immediate feedback from the tutor. The model allows better understanding of the impact of over-the-shoulder instructions on learning and performance.

The Task: The Anti-Air Warfare Coordinator

The task is the same as the one described in Anderson et. al (in press), which was constructed based on the Georgia Tech Aegis Simulation Program (GT-ASP, Hodge, Rothrock, Kirlik, Walker, Fisk, Phipps, & Gay, 1995). GT-ASP is a tactical decision-making computer game that simulates tasks facing an anti-air warfare coordinator (AAWC) on board a US Navy cruisers and destroyers. A participant assumes the role of an AAWC, which includes monitoring a radar screen for unknown aircraft, requesting and collecting information regarding the unknown aircraft, and updating the identity of the aircraft. The task we used is a simplified version of the GT-ASP, and a model-tracing[1] tutor was constructed to monitor participants' actions. We refer to it as the CMU-ASP task henceforth.

The radar screen of the CMU-ASP task (Figure 1) consists of three major areas. First, the radarscope shows various air tracks. Vectors emanating from the aircraft indicate speed and course. The AAWC is on a ship at the center of the radarscope (called ANZIO). The AAWC moves the mouse within the scope and "hooks" a target airplane by clicking the mouse button. This hooking is necessary whenever the AAWC tries to update identity of unknown aircraft. Second, there is a group of information boxes on the left of the screen where the participant can get information on tracks. Third, the menu panel shows the

---

[1] Owing to the complexities and the dynamic nature of the task, the tutor used a grammar-based representation of procedural knowledge instead of the traditional production-based representation (Douglass, 2004). The grammar representation was able to capture more variations of different courses of actions more succinctly than the traditional production representation. However, the basic tracing methodology was the same.

currently bindings of the function keys (F1 to F12 on the computer keyboard) that are used to issue commands. The AAWC spent the majority of the time in identifying the intent (friendly or hostile) of tracks on the screen and their air type (e.g., strike, or commercial). It is this identification task that the experiment focuses on. We will focus on the

first unit task in the identification task – how participants select the next track -- with and without the over-the-shoulder instructions. This unit task involves a selection and a visual search component, both of which are subject to significant improvement with the over-the-shoulder instructions. We will elaborate on this aspect below.
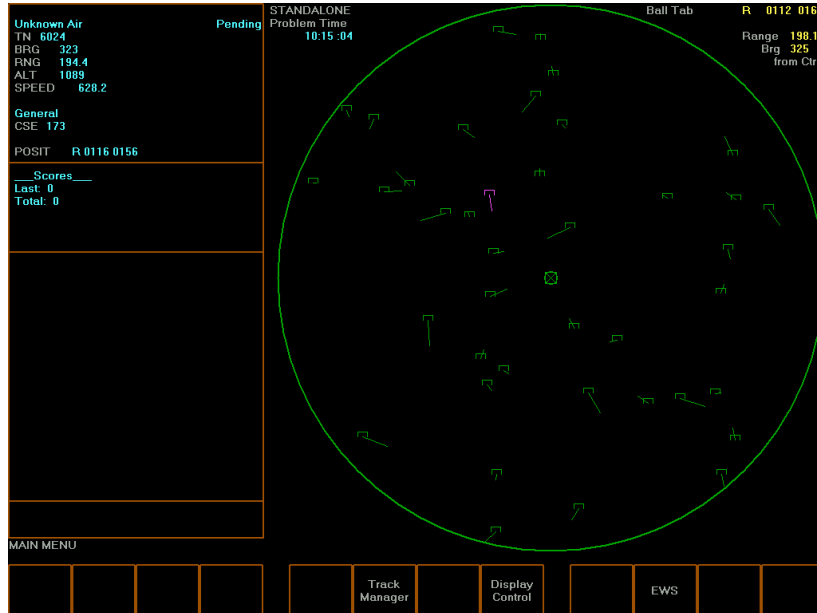

Figure 1. The display in the CMU-ASP task.

Instructions

Before the experiment participants were asked to memorize a set of initial instructions that were sufficient to finish the task (details in Anderson, et. al., in press). Half of the participants were assigned to a group where over-the-shoulder instructions were given (instruction group) and the other half did not receive any over-the-shoulder instructions (no-instruction group). The over-the-shoulder instructions were given right after the participants finished a unit task (e.g. after hooking a track or after identifying a track as a commercial profile). In this paper, we focus on the over-the-shoulder instructions that were given right after participants had hooked a track. Participants were told that their scores depended on the importance of the tracks classified, but the initial set of instructions did not explain how the importance of a track could be evaluated. After a track had been identified, a score would be displayed in the middle information box on the left of the screen as shown in Figure 1. The score was calculated by the following equation (not known to the participants): *Identification score* = (*speed* + 3 * (512 – *range*)) * *at-me*. Hence, the score was high if the *speed* was high, if it was close to the center of the screen (*range* was small), or if it was flying towards the ship. The *at-me* value ranges from 1.0 to 3.0, and it depends on the difference between the course of the track and the bearing of the track measured from the ANZIO. A small difference implies that the track is flying towards the ship, and the *at-*

me value will be high. Since the score decreases linearly over time (from 100% to 25% during the 6-min trial) and there is often insufficient time to classify all tracks in 6 minutes, the final scores (calculated as the sum of all time-weighted identification scores) depend critically on whether participants can classify tracks in the order of importance.

At any point in time, the identification scores of all unidentified tracks were calculated and tracks were ranked according to their scores. In the instruction group, after the participant hooked a track, if the hooked track was less important than 20% of all unidentified tracks the tutor would be triggered to give an audio instruction. The most important track on the screen would be highlighted, and instruction would be given to inform the participant why the highlighted track was more important (e.g. it was fast and directly approaching the ship). Participants could then learn from the instructions and improve future performance by selecting the most important unclassified tracks.

ACT-R 5.0

The basic architecture of ACT-R 5.0 consists of a set of modules, each devoted to processing a different kind of information. Coordination in the behavior of these modules is achieved through a central production system. This central production system is not sensitive to most of the activity of these modules but rather can only respond to a limited amount of information that is deposited in the buffers of these modules. The core production system can recognize patterns in these buffers and make changes to

these buffers – as for instance, when it makes a request to perform an action in the manual buffer, or a request to move attention in the visual field (e.g. to a track) or to an audio signal (e.g. an over-the-should instruction).

The Experiment

Thirty-two participants were recruited for a two-day experiment. Half the participants were assigned to the instruction group and the other half to the no-instruction group. On the first day they were given the set of initial instructions that taught them how to hook and identify a track, and how to use the F-keys to input the classifications of the tracks. On the first day participants memorized the instructions and were tested on 10 6-minute scenarios. On the second day they were tested on 10 more. Each scenario involved 40 tracks randomly placed on the screen. In the instruction group, over-the-should instructions were given except the first two and the last two scenarios given on each of the two days. In other words, in the instruction group, participants were given 2 no-instruction trials, followed by 6 instruction trials, then 2 more no-instruction trials on each of the two days of experiment.

The Model

The model uses the same set of parameters as described in Anderson et al (in press), and uses the same declarative-to-procedural system to process the initial set of instructions. The current extension processes over-the-shoulder instructions in a similar fashion, i.e. instructions are initially represented as declarative knowledge and are retrieved when needed. The over-the-shoulder instructions are also subject to the same declarative-to-procedure conversion as the initial set of instructions. However, the strength of the over-the-shoulder instructions is assumed to be much weaker than that of the initial set of instructions. To preview our results, we did find that the learning of the over-the-shoulder instructions were slower. Besides, when the tutor was turned off, performance declined quickly, suggesting that the instructions were weakly encoded and not yet proceduralized as the initial set of instructions.

To model the track selection strategies, we asked participants to freely write down the criteria they used to select the next track. The top three criteria were shown in Table 1. It shows that participants were able to learn to attend to the right features in the instruction group. It is interesting that even with no instruction, participants tended to choose tracks that were closer to the ANZIO (range), and apparently some of them learned that at-me and speed were important features.

	Instruction	No-instruction
At-me	15	4
Range	12	10
Speed	15	4

Table 1. The number of participants reported that they used the criteria to select the next track in each group.

We represented each of the above criteria as a "search-factor" and they were created as declarative chunks when the first over-the-shoulder instruction was given. The search-factors provided constraints to the visual search process. For example, a "fast" search-factor commands a search for a track that has a long vector emanating from the track, and a "range" search-factor commands a search for a track that was close to the ANZIO, and so on.

When the task began, since no search-factor was available, the model started to select the track closest to the ANZIO. This assumption was based on the apparent tendency for participants to select tracks closer to the ANZIO in both groups (see Table 1). Another assumption was that participants would tend to minimize the search time required to select a track, so that they could classify as many tracks as possible in each trial. Under this assumption, when no additional information is available, the best strategy is to select the track closest to the track that has just been classified, i.e., a nearest-neighbor strategy. This will minimize both the visual search time and the mouse movement time.
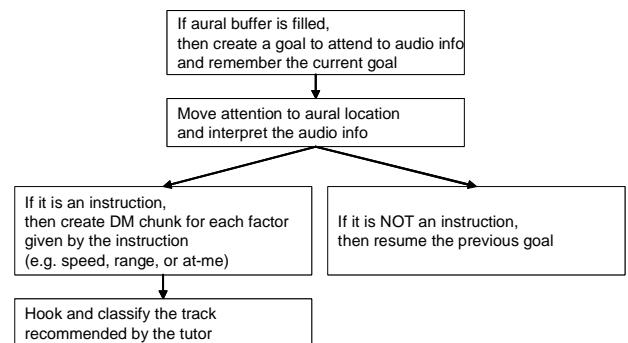

Figure 2. How the model responds to the over-the-shoulder instruction through the aural buffer. (DM = Declarative Memory)

When an over-the-shoulder instruction is given, the instruction will be stored automatically in the buffer of the aural module. A production will fire whenever there is an audio signal in the aural buffer (see Figure 2). This production will also create a new goal of moving attention to this audio signal, and stores the current goal as part of the new goal so that in case the model chooses not to follow the instruction, it can resume the current goal. After attention is moved to the audio signal, the content of the audio signal can be interpreted. If the audio signal is an instruction given by the tutor, the model will interpret the instruction (we choose not to model the interpretation, instead the interpretation is done by a lisp function). The interpretation process generates search-factors that explain why the track highlighted by the tutor is important. There are three search-factors given by the over-the-shoulder instruction: speed, range, and at-me. For example, if the instruction is: "This fast track is coming at the ANZIO and needs attention", then the interpretation process will generate the "fast" and "at-me" search-factors. For each of the search-factors

generated, the model will create separate chunks representing them in declarative memory. If there is already an identical chunk in declarative memory when the new chunk is created, the chunks will be merged and the activation of the chunk will be increased and its likelihood of being retrieved in the future increases.

After the model repeatedly processes the over-the-shoulder instructions, the activation of the search-factor chunks will be strengthened and the chunks will eventually be retrieved. These search-factors affect the selection of the next track through repeated cycles of retrieval and visual search (see Figure 3). When the model starts to select a track, it will first try to retrieve a search-factor. If a search-factor, for example, speed, is retrieved, the model will search for a fast track on the radar screen. At the same time, retrieval for the second search-factor is initiated. The visual search and the retrieval compete against each other. If the visual search process finds a track before retrieval finishes, the model will satisfice on the track and stop searching. However, if retrieval finishes first, a new visual search will be initiated based on the two search-factors retrieved. For example, if the first search-factor is speed and the second factor is range, then the model will search for a track that is both fast and close to the ANZIO. At the same time, retrieval for the third search-factor will be initiated. If, again, a track is found based on the two search-factors before the retrieval for the third search-factor finishes, the model will proceed to hook and identify the track. If retrieval finishes first, a new search will be initiated based on the three search-factors. At this point, since all factors have been retrieved, the model will proceed to hook the track when a track is found. The flow diagram in Figure 3 therefore shows how top-down influence of instructions can be incrementally combined with the visual search process.
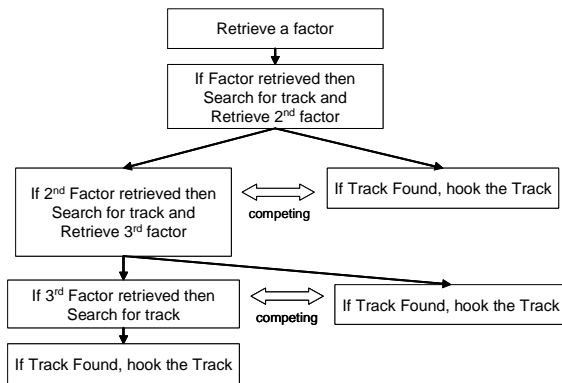

Figure 3. The model selects the next track retrieving factors that guides the visual search.

The competition of the retrieval and visual search process allows the model to incrementally improve the selection of tracks. Perhaps more importantly, it allows performance to decline gradually as over-the-shoulder instructions were dropped (e.g., see Figure 7 below). This performance decline will be difficult to be modeled by, for example, a strategy-competition approach.
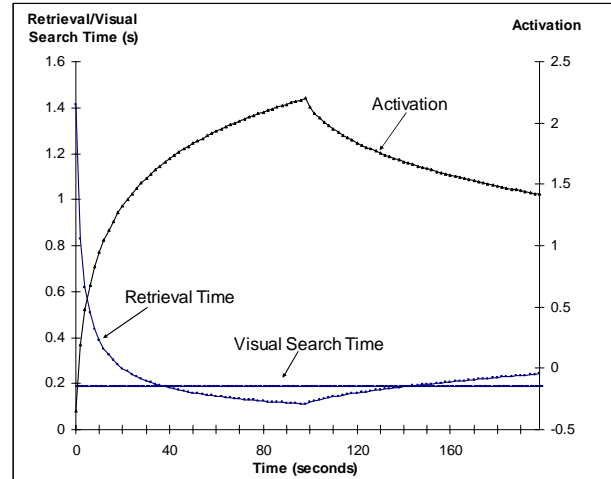

Figure 4. The activation of a chunk encountered every two seconds and its noiseless retrieval time. The visual search time is fixed at 0.185s. The encounter stops at 100s.

In ACT-R, the retrieval time (T) is calculated as T=F*exp(-A) where F is a scaling factor that defaults to 1, and A is the activation of the chunk. The calculation of the activation is more complicated, but for the current purpose, if a chunk is created t_1 before (i.e. a lag of t_1), and a new chunk was merged to the chunk at lags t_2, t_3, … t_n, then the activation of the chunk will be

$$A = \ln(\sum_{j=1}^{n} t_j^{-d}) + \varepsilon$$

where d is the decay parameter which defaults to 0.5, and ε represents normally distributed noise. Figure 4 shows how the retrieval time (without noise) decreases with time, during which the same chunk is encountered (i.e. in this case, merging of chunks) every 2 seconds. Figure 4 also shows the relationship between activation and retrieval time. After 50 encounters (i.e. after 100 seconds), strengthening of the chunk stops (i.e. the chunk is not encountered anymore). We can see that after this point, the activation of the chunk decays with time (the rate of decay is controlled by the value of d), and the retrieval time increases. The visual search time is fixed as 0.185 seconds[2]. We can see that in this simple case, as the number of previous encounters is more than 20 (i.e. after 40 seconds of regular encounters), the retrieval time is faster than the visual search time. However, when the encounter stops, the retrieval time eventually becomes larger than the visual search time.

Results

Figure 5 shows the final scores obtained after each scenario by the participants and the model in both the instruction and no-instruction groups. The final scores are the total scores

---

[2] ACT-R assumes that it takes 85ms to encode the features of a visual object. However, it often requires the firing of two productions, one for finding the location of the visual object, the second for initiating the visual encoding. Each production takes 50ms. Therefore the total time for a visual search is 185ms.

obtained for all tracks identified during each 6-min scenario. The model was able to fit the data well ($R^2 = 0.92$). However, the model did start off with slightly better scores in the first two scenarios than the participants. Apparently some participants might have picked less important tracks than the model in the beginning trials. However, given the good fit between the model and data at later trials, we consider this a good tradeoff by constructing only a single simple strategy (i.e. the nearest-neighbor strategy) to capture performance without making assumptions on the various strategies that might have been used by the participants at different stages of learning.

Participants improved steadily across trials. The final scores dropped slightly at scenario 11, when the participants came back on the second day. In general, the improvements of final scores of the model were similar to those of the participants.

Figure 5 shows that there was no significant difference in total scores between the instruction and no-instruction groups. However, Figure 6 shows that the number of identification made in each scenario was actually higher for the no-instruction group than the instruction group. Apparently the over-the-shoulder instructions had slowed down the participants. This is most obvious in scenario 3, where the over-the-instructions were first given. Combining the results from Figure 5 and Figure 6, participants in the no-instruction group had fewer scores per each track they identified. In other words, participants in the instruction group had identified more important tracks than the no-instruction group. It suggests that although participants (and the model) were slower performing the task, they were able to choose more important tracks and obtained final scores as high as participants in the no-instruction group.
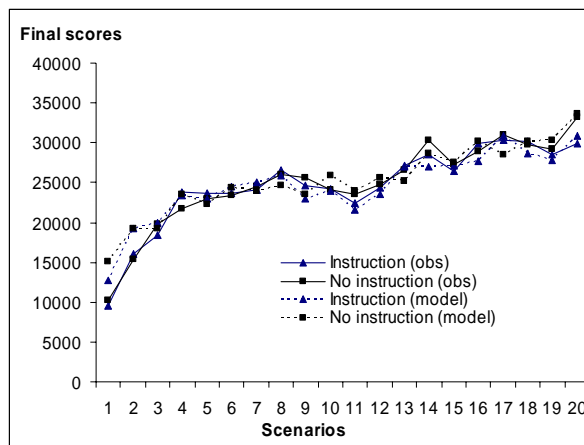

Figure 5. Final scores obtained by the participants and the model after each scenario. Note that the second day starts on scenario 11. (obs = observed data from participants)

Participants sped up steadily across scenarios, except again in scenario 11 when participants came back on the second day. The difference between the instruction and no-instruction group also became smaller with practice. It seemed that participants in the no-instruction group reached asymptotic performance earlier than the instruction group. From the analyses in Anderson et al (in press), during the early stages of learning, time to identify a track depended mostly on cognitive components. At later stage of learning, the time to identify a track depended mostly on perceptual-motor components. The difference between the two groups indicates that the over-the-shoulder instructions may require more cognitive operations to process the instructions. The smaller difference at later stage of learning indicates that the processing of instructions had also sped up. The model was able to capture the patterns of data well in both the instruction and no-instruction groups ($R^2=0.96$).

To further investigate the effect of the over-the-shoulder instructions, we derived a measure of how well participants were able to identify the tracks in the order of their importance (i.e., identify the most important first and least important track last). In each scenario, we calculated the correlation between the scores of the tracks and the order of their identification. For example, if tracks A, B, and C are identified one after the other and the scores obtained are 900, 800, and 600 respectively, then the correlation between the two arrays (900, 800, 600) and (1, 2, 3) is calculated to be -0.98. However, if the tracks were identified in the order of A, C, and B, then the correlation between the two arrays (900, 600, 800) and (1, 2, 3) will be -0.33[3]. In other words, the more negative the correlation, the better the participant is able to identify the tracks in the order of their importance.
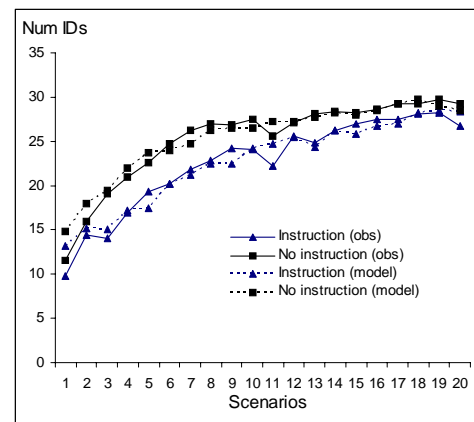

Figure 6. The number of identifications made in each scenario. (IDs = identifications)

Figure 7 shows the correlations across the scenarios for both groups. The correlations for the instruction group were lower than the no-instruction group throughout the 20 scenarios. One may wonder whether the fewer number of identifications may have contributed to the lower correlations (more negative) in the instruction group. However, after checking the data we found that the differences were too big to be caused by the fewer number of identifications (except perhaps in the first two scenarios).

---

[3] In the actual task, the scores will not be identical when tracks were identified in different orders because scores decrease linearly with time.

In fact, the difference in the number of tracks identified between the two groups was very small (see Figure 6) in the second day. We conclude that participants in the instruction group were better at identifying the most important tracks first, and the least important tracks last.
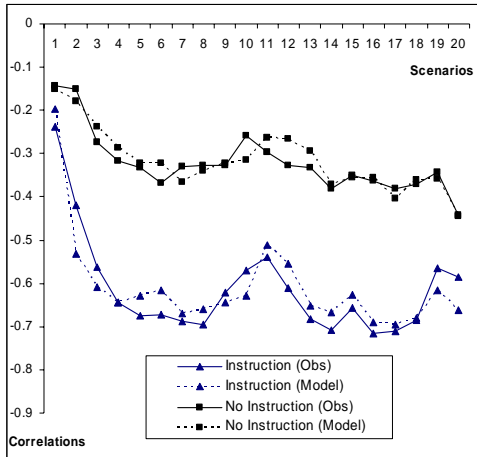

Figure 7. The correlations between scores of the track identified and their order of identification. The more negative the correlation, the better the identification was in a decreasing order of scores.

The effect of instructions can also be assessed from Figure 7 at scenarios 9 to 12, and scenarios 19 and 20 for the instruction group, during which the tutor was turned off. The instruction group shows some reduction in the magnitude of their correlations but still show stronger (more negative) correlations than the uninstructed group.

The model captures the data well ($R^2 = 0.95$). The model exhibited the same increases of correlations when the tutor was turned off as the data did. This shows that the model was able to respond and learn from the over-the-shoulder instructions as the participants did.

The nearest-neighbor strategy also did a good job capturing the small learning curve in the no-instruction group. Since the strategy always started with the track closest to the ANZIO, it captured the general tendency to classify tracks closer to the ANZIO first as reported by the participants. Since tracks close to the ANZIO tended to have a high importance, the general tendency therefore led to the sharp decrease of correlation during early trials as the number of identification increased. However, after tracks close to the ANZIO were identified, the order of identification became more random than systematic, therefore the correlation asymptote at a much less negative value than that of the instruction group.

Conclusions and Discussions

We found that participants were able to learn from the over-the-shoulder instructions and paid attention to more important tracks. To understand how people learned from these instructions, we extended previous work by Anderson et al. (in press) by constructing a mechanism that allows the model to learn from over-the-shoulder instructions. Instructions were represented as declarative memory traces. Repeated instructions strengthened these memory traces to make them more available when needed. A mechanism is constructed that allows visual search to be incrementally improved and declined based on the strengths of these memory traces of these top-down instructions. The ability to search for important tracks improved as the strengths of these memory traces increased. The strengths of these memory traces of instructions decayed when the tutor was turned off, leading to the decline in performance.

The treatment of the over-the-shoulder instructions is basically the same as the initial instructions, except that (a) they take time away from the processing, and (b) we realistically model the slow learning of the over-the-shoulder instructions whereas we assume that the initial instructions are well encoded. Intuitively, one might speculate that over-the-shoulder instructions might be easier to learn as they are presented right in the exact same situation. As least from our data, although the over-the-shoulder instructions apparently had a large impact on performance, they did take time away from the task and their effect seemed to decay quite rapidly. This may suggest that over-the-shoulder instructions may not have the privileged status (as compared to practice) as believed.

Cognitive tutors have been successfully applied in relatively static task domains such as algebra or geometry. The current line of work attempts to transfer the technology of cognitive tutors out from the protected classroom or laboratory to a real-world, complex, and dynamic task. The task requires all forms of learning and all buffers in ACT-R to interact and thus is a good test bed for the architecture. Our goal is to have the model eventually handles more complicated over-the-shoulder instructions, thus allowing automatic evaluation of a real-time instructional system for dynamic tasks.

Acknowledgments

References

Anderson, J. R., Bothell, D., Byrne M. D., Douglas S., Lebiere, C. & Qin, Y. (in press). An Integrated Theory of the Mind. *Psychological Review*.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive Tutors: Lessons learned. *The Journal of the Learning Sciences,* 4, 167-207.

Anderson, J. R., Lebiere, C. (1998) *Atomic components of though*t. Mahwah, NJ: Erlbaum

Douglass, S. A. (2004). Using the PLASTIC framework to augment scenario-based training systems with instructional agenst. In *Proceedings of the 48th Annual Meeting of the Human Factors and Ergonomics Society*.

Hodge, K., Rothrock, L., Kirlik, A., Walker, N., Fisk, A., Phipps, D., & Gay, P. (1995). *Trainings for tactical decision making under stress: Towards automatization of component skills*. Atlanta, GA: Georgia Institute of Technology.