Using Human Computation to Acquire Novel Methods for Addressing Visual Analogy Problems on Intelligence Tests

David A. Joyner[1,2], Darren Bedwell[1], Chris Graham[1], Warren Lemmon[1], Oscar Martinez[1], Ashok K. Goel[1]

Design & Intelligence Laboratory[1]
Georgia Institute of Technology
Atlanta, GA 30332 USA

Udacity[2]
2465 Latham Street
Mountain View, CA 94040 USA

{djoyner3, dbedwell3, cgraham36, wlemmon3, omartinez8}@gatech.edu; goel@cc.gatech.edu

Abstract

The Raven's Progressive Matrices (RPM) test is a commonly used test of intelligence. The literature suggests a variety of problem-solving methods for addressing RPM problems. For a graduate-level artificial intelligence class in Fall 2014, we asked students to develop intelligent agents that could address 123 RPM-inspired problems, essentially crowdsourcing RPM problem solving. The students in the class submitted 224 agents that used a wide variety of problem-solving methods. In this paper, we first report on the aggregate results of those 224 agents on the 123 problems, then focus specifically on four of the most creative, novel, and effective agents in the class. We find that the four agents, using four very different problem-solving methods, were all able to achieve significant success. This suggests the RPM test may be amenable to a wider range of problem-solving methods than previously reported. It also suggests that human computation might be an effective strategy for collecting a wide variety of methods for creative tasks.

Introduction

The Raven's Progressive Matrices (RPM) tests are a group of intelligence tests based on visual analogy problems (Raven, Raven, & Court 1998). In these problems, a matrix of visual frames is presented with a blank space; six or eight options are presented for filling in this space. Performance on RPM has been shown to correlate well with other intelligence tests (Snow, Kyllonen, & Marshalek 1984). Thus, although wholly visual, the RPM tests measure *general* human intelligence, and are often used as the psychometric measure of choice in educational and clinical settings.

Hunt (1974) suggested that humans use multiple problem-solving methods to address RPM problems, including "analytical" and "Gestalt" methods. Bringsjord & Schimanski (2003) have proposed intelligence tests such as RPM as a method of measuring the effectiveness of AI techniques. AI research has developed a variety of methods for addressing RPM and similar visual analogy problems, including both "analytical" methods that typically use propositional representations (Evans 1968; Lovett, Forbus, & Usher 2009; O'Donoghue, Bohan & Keane 2006; Prade & Richard 2011; Ragni & Neubert 2014), and "Gestalt" methods that often use imagistic representations (Dastani, Induskhya & Scha 2003; Kunda, McGreggor, & Goel 2013; McGreggor & Goel 2014; Schewring et al. 2009). Another way of classifying the various methods is by control of processing. For example, some methods for addressing RPM problems, such as the affine method (Kunda, McGreggor & Goel 2013), first generate an answer based on the (partial) matrix, and test this answer by comparing it with each available choice; other methods, such as the fractal method (McGreggor, Kunda & Goel 2014), test each available answer by computing the degree of fit in the matrix. While it may appear that generation of answers is a necessary part of creativity, we posit that generating explanations for available answers is also creative.

The Raven's Test and Creativity

One major component in the value of the RPM test is its connection not only to intelligence, but also to creativity. Hunt (1974) laid the foundation for the creative nature of problem-solving methods on this test in identifying the two broad categories of methods mentioned previously, "Gestalt" and "analytical". Kirby & Lawson (1983) argued further that it is the diversity of problem-solving methods that makes the RPM test a valuable tool for assessing intelligence in humans. If creativity is in part the ability to develop novel, useful, and effective methods to a problem, then the RPM test's admission of multiple methods adds to its value as a tool for studying creative problem solving.

Second, Keating & Bobbitt (1998) argue that addressing many RPM problems requires metacognitive abilities to select among the available problem-solving methods, to monitor the progress of the selected method, to suspend or abandon the current method and move to a different method, and to combine insights from the use of multiple meth-

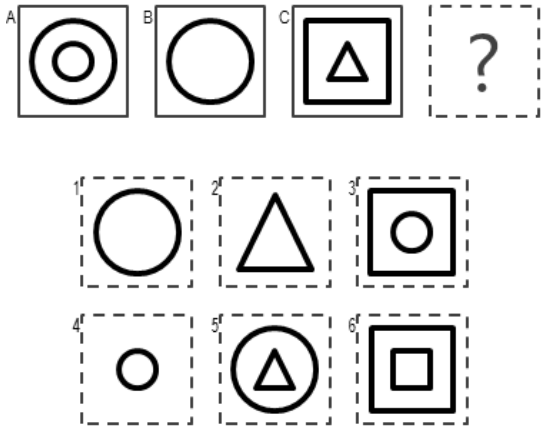

Figure 1: A 2x1 visual analogy problem. Although RPM tests do not have 2x1 problems, 20 2x1 problems are used as a soft introduction to solving visual analogies.

ods into one final answer choice. Third, the normatively correct choices for some RPM problems are often non-obvious, sometimes even unexpected, such as in the problem shown in Figure 1. Thus, from the perspective of both process (metacognitive processing) and product (unexpectedness of the answer), the RPM test measures not only intelligence, but also creativity.

One potential critique of the RPM test for studying creativity is that a set of answer choices are presented to the test-taker. However, this implies that the creative task necessarily entails generating a novel answer. The structure of the RPM problems turns this notion of creativity around: rather than generating an answer, the test-taker instead creatively generates an explanation for a particular answer choice. In Figure 1, for example, the most obvious answer would be a large square; however, none of the answer choices match this obvious answer. The presence of answer choices constrains the activity and forces the test-taker to creatively generate not an answer, but an explanation for why one of the presented choices is most compelling. This explanation is as much the output of the creativity process as the answer itself.

From the perspective of computational creativity, the above analysis makes the RPM test an excellent choice for designing, evaluating, and comparing new AI methods not only for intelligence, but also for creativity: the task admits a wide variety of AI methods characterized by different knowledge representations and different controls of processing. The question then becomes: how can we identify the novel techniques that may effectively address RPM problems?

We postulate that one strategy for acquiring new methods for addressing visual analogy problems on the RPM test is through crowdsourcing (Howe 2008), or, more accurately, human computation (Law & von Ahn 2011). Although crowdsourcing has typically been used for acquiring domain knowledge, human computation also admits acquisition of problem-solving methods. Yet, it is also important to acquire new methods for addressing visual analogy problems not from any crowd, but from intelligent, educated, high-achieving humans who themselves are likely to do well on the RPM test.

The Experiment

In Fall 2014, we offered a new online Georgia Tech graduate-level CS 7637 course titled "CS 7637 Knowledge-Based AI: Cognitive Systems" as part of the new Georgia Tech Online MS in CS Program (Goel & Joyner 2014; Goel & Joyner 2015). We also offered an in-person class in parallel, with the two classes sharing the same syllabus and structure. The course describes its learning goals as, "to develop an understanding of (1) the basic architectures, representations and techniques for building knowledge-based AI agents, and (2) issues and methods of knowledge-based AI." Toward this end, students cover several knowledge representations (semantic networks, frames, scripts, formal logic), reasoning strategies (case-based reasoning, rule-based reasoning, model-base d reasoning), and target domains (computational creativity, design, metacognition). More comprehensive information on the structure and content of the class is available at the link above.

In previous offerings of the in-person class, we had used variants of problems on the RPM test to motivate the class projects (Goel, Kunda, Joyner, & Vattam 2013). Thus, we knew class projects based on the RPM test stimulated student engagement while providing an authentic opportunity to explore cutting-edge research. Therefore, in Fall of 2014, we again designed the class projects based on variants of problems on the RPM test. Students in both the online and in-person sections were asked to complete four projects that addressed 123 RPM-inspired problems in all, culminating in Project 4, wherein students designed agents that could answer all 123 problems using visual input. 224 students completed Project 4, addressing all the problems using the raw imagistic input. We collected all the data on these 224 Project 4 submissions, including the designs of the agents and their performance on the 123 problems.

In this paper, we will describe the results of this experiment. First, we will present at a high level the results of the 224 agents that were developed to address these RPM-inspired visual analogy problems. Second, we will examine in greater detail the design of four of the most creative and effective agents developed for the project. These agents operate according to four significantly different methods for reasoning about these problems. In describing these agents, we will clarify their relationship to elements of human creativity operationalized and instantiated in AI agents.

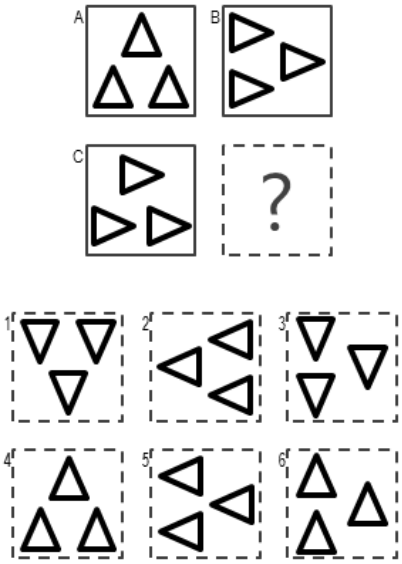

Figure 2: A 2x2 visual analogy problem, inspired by Raven's Progressive Matrices. In this paper, individual squares in a problem are called 'frames', while individual shapes within each frame are called 'objects'.

RPM-Inspired Visual Analogy Problems

The standard set of Raven's Progressive Matrices test is made of 60 visual analogy problems: 24 of the problems are 2x2 matrices, and 36 of the problems are 3x3 matrices. For copyright reasons, we have not yet been able to use actual RPM in these class projects. Instead, we have developed a set of 123 RPM-inspired problems. These problems are broken into three categories: 27 2x1 matrices (as shown in Figure 1), 48 2x2 matrices (as shown in Figure 2), and 48 3x3 matrices (as shown in Figure 3). Although there are no 2x1 matrices in the actual RPM test, these are included in our set to provide a simpler initial set of problems for students to address before moving on to more difficult problems.

To develop these RPM-inspired problems, we examined individual problems on the actual RPM tests (both the standard and the advanced test) and wrote problems to have a close correspondence with the problems on the actual tests. Although the individual shapes and their properties differ, these RPM-inspired problems mimic the same transformations and problem types as the actual standard and advanced RPM tests. These correspondences, however, only exist at the level of individual problems; not every RPM has a corresponding RPM-inspired problem in our problem sets, and some types of problems are present more often in our problem sets than in the actual RPM tests. Therefore, no claim is made that our RPM-inspired problem sets are equivalent to the RPM tests as a whole; we only claim that the individual problems capture the same reasoning as problems on the original RPM tests. We are presently running two previously-designed agents (Kunda. McGreggor & Goel 2011; McGreggor, Kunda & Goel 2014) for solving the actual RPM tests against these new RPM-inspired problems in order to establish a conversion factor between the two sets.

The Projects

In the Fall 2014 version of the KBAI class, students completed a series of four projects. In the first three projects, students designed agents that could address 2x1, 2x2, and 3x3 matrix problems. During these projects, the input into these agents was propositional representations of the 123 RPM-inspired visual analogy problems. The propositional representations were written by the instructors of the course to prevent students from building inferential advantages into the representations. During the design of their agents, students could see 83 of these problems: the remaining 40 were designated 'Test' problems and were hidden from students in order to test their agents for generality. Thus, students were encouraged to construct agents with general problem-solving ability rather than agents that would tightly fit a small set of previously-seen problems.

By the end of project 3, students had completed an agent that could solve 2x1, 2x2, and 3x3 visual analogy problems based on propositional input. In project 4, students designed an agent that could solve these same problems using visual input. Here, students' agents read in the images directly from .PNG files, with one file representing each frame from the problem. Students' agents were run against the same 123 problems. Students' grades were dependent on performance on 100 of these problems (the remaining 23 were provided as challenge problems with no credit granted for correct answers), and 40 of these 100 problems were withheld as 'Test' problems. This paper focuses only on the agents designed in project 4, which took visual input.

Table 1: Performance on the eight sets of RPM-inspired problems (123 problems in all). "*n*" gives the number of problems in that set. "Avg." gives the average number of correct answers in that set for the 224 agents. "1", "2", "3", and "4" give the performance of the four agents described in further detail under 'Four Agents', below.

	n	Avg	1	2	3	4
2x1 Basic	*20*	8.8	18	14	17	12
2x1 Extra	*7*	1.5	4	1	7	2
2x2 Basic	*20*	8.8	18	16	20	14
2x2 Extra	*8*	2.5	7	4	7	7
2x2 Test	*20*	7.2	17	16	14	12
3x3 Basic	*20*	11.0	19	17	20	15
3x3 Extra	*8*	1.5	2	0	6	4
3x3 Test	*20*	7.9	16	15	11	13

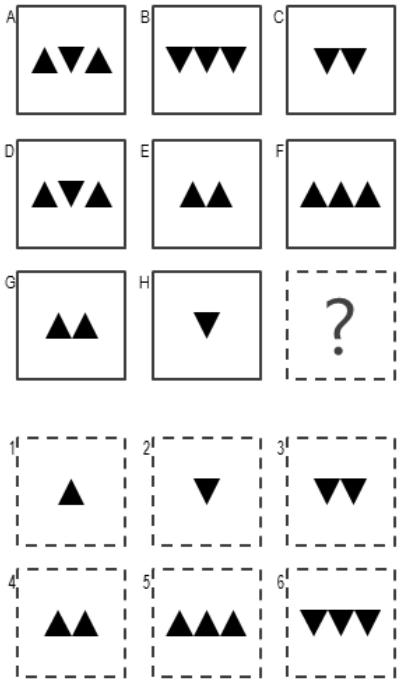

Figure 3: A 3x3 visual analogy problem, inspired by Raven's Progressive Matrices. Individual objects within frames in an RPM can be said to have 'properties'; for example, some of the triangles in this problem have a 180° rotation as a property.

Aggregate Results

Students in the KBAI class submitted 224 agents, each of which ran against the 123 problems. The percentage of agents answering an individual problem correctly ranged from 87% (fo r the easiest 3x3 problem, which involved no transformations between frames) to 8% (for the hardest 3x3 problem, which demanded reasoning about the sum of the number of sides of multiple shapes). Among the problems completed for credit, one Test problem was correctly answered by only 10% of agents; this 2x2 problem involved two transformations – change-fill and remove-shape – that conflicted with one another.

Table 1 previously shows the performance of the agents as a whole, as well as the performance of the four agents highlighted below. The table is broken up by the eight distinct problem sets students addressed: 'Basic' sets *were* provided to students during the design of their agents and *were* evaluated for the project grade; 'Test' sets *were not* provided to students for the design of their agents and *were* evaluated for the project grade; 'Extra' sets *were* provided to students during the design of their agents but *were not* evaluated for the project grade. Agents' scores on the Basic and Test sets comprised 70% of students' project grades.

Perhaps surprisingly, students' agents performed better on 3x3 problems than on 2x2 problems. While 3x3 problems allow more complex problem structures, such as transformations in which two frames together determine the contents of a third, students noted that 3x3 problems gave their agents more information with which to work. With more information, their agents performed better, even on more complex problems.

Four Agents

After evaluating the aggregate results, we examined the problem-solving methods of several of the best-performing agents and identified a number of particularly novel and successful methods for addressing these RPM-inspired problems. The majority of the 224 submitted agents operated by first writing a propositional representation based on shape recognition, and then solving the problem propositionally; we describe the most successful agent using this method below, which combines contour recognition with problem classification. However, we also identified several other methods to solving these problems. Here, we describe three additional creative methods to solving RPM-inspired problems based on imagistic representations.

Agent 1: Contour Recognition & Reasoning

Agent 1 uses an intermediate propositional knowledge representation for working memory. In the agent's representation, each frame in an RPM consists of objects, and each object consists of the following attributes: shape, size, fill, rotation, and relative-position to other shapes. A library of shapes was available to the agent, storing 20 basic shapes and features such as symmetry and corner count. Agent 1's method has three phases: symbol extraction, top-down recognition, and bottom-up recognition.

Phase 1 uses image processing to extract a propositional representation for each problem. First, objects are found by isolating connected components, after which they are classified into shapes based on attributes of the object like corner count, edge lengths, and convexity. Other object attributes, including fill, rotation, size, and relative position are also computed in this phase.

Phase 2 uses top-down pattern finding. 19 pattern recognizers look for simple patterns that will be combined to form a pattern fingerprint. Recognizers include "constant rotation across objects in frame" (as seen between frames A and C in Figure 2) and "object count arithmetic sequence." For each problem matrix, patterns are found and combined for all in-row, -column, and -diagonal relationships. The agent then chooses the answer with the largest set of matchers. In the event of a tie, Phase 3 begins.

Phase 3 performs bottom-up reasoning by splitting each problem into 2x1 sub-problems: 2 for 2x2 matrices and 29 for 3x3 matrices (including diagonal sub-problems). The agent solves each sub-problem, producing multiple answer choices, then uses majority-rule to make a final answer selection.

To solve a 2x1 sub-problem, (1) all object pairs from frame A to frame B are created; (2) all object pairs from frame C to the answer choices are created; (3) all mappings between object pairings from step one and step two are created; and (4) each mapping is given a score. The scoring function includes the intuitiveness of the transformation in step two and the strength of analogy in step three. For example, a mapping would be scored highly for intuition for mapping a triangle from frame A to a triangle in frame B. However, if a triangle in frame A instead mapped to a square in frame B, the best analogy would map triangles from frame C to squares in frame D. The highest scoring mapping is the most intuitive analogy. In the worst case, phase 3's runtime is $O((n!)^3)$, where n is object count per frame. To offset this, time limits were imposed.

To take the problem shown in Figure 3 as an example: during Phase 1, 31 shapes and 14 frames would be represented in a fashion similar to the following: frames: [{id: 1, objects:[{id: 1; shape: triangle; fill: yes; angle: 0; left-of: [2, 3]; size: medium},{id: 2; shape: triangle; fill: yes; angle: 180; left-of: [3]; size: medium}…]}, …].

During Phase 2, each potential answer is inserted into the last cell of the matrix, and each pattern matcher runs. Here, the matcher labelled "remaining shapes after pairing" will match: each upright triangle in the first cell of a row or column is paired with a flipped version in the second cell, and the remaining triangles are checked to see if they match those of the third cell. Other matchers may also match the inserted choice, creating a more complex pattern. In the end, each potential answer will have a list of matchers associated with it, and the one with the longest list of matchers is selected. For this problem, the agent would choose the first answer choice. Because the problem would be solved in Phase 2, Phase 3 would not execute.

Agent 1 performed exceptionally well, correctly answering 101 of the 123 problems (88 of the 100 problems for credit). Agent 1's general method of generating a representation based on prior shape knowledge also reflects the most common approach used in the class (as well as an approach used in prior literature, e.g. O'Donoghue, Bohan, & Keane 2006); however, Agent 1's classification of multiple problem types goes beyond what the majority of agents attempt and plays a large role in its success.

Connecting with computational creativity, Agent 1 possesses the ability to creatively generate its own answers. Presently, Agent 1 operates by substituting each answer choice in the empty frame and evaluating its degree of fit to the problem's transformations; however, implicit here is the idea of an 'optimal' fit for the remaining frame. Were the agent deprived of the answer choices, it could instead generate the optimal solution for the empty frame. Agent 1 is limited in this regard, however, in that it could only produce solutions that are comprised of the shapes in its shape library; Agent 1 cannot deal with novel shapes.

Agent 2: Shape-Agnostic Transformation Recognition

The second agent, Agent 2, operates in two stages. First, the agent detects and analyzes individual objects to produce a propositional representation, similar to Agent 1. The agent uses the individual properties to find relationships between objects in pairs of frames, and chooses the answer that best fits the relationships that are found. Agent 2's high-level process thus resembles Agent 1's in its initial phase of translating imagistic representations into propositional ones; however, it differs in that it does not rely on prior shape knowledge. Agent 2 derives the structure and content of the problem from within the problem, rather than based on prior knowledge of shapes and features.

The agent begins by recording visual measurements for each object in the problem and using a simple clustering method to partition similar objects into shape groups. The agent records the width/height ratio of an object and the amount of whitespace "outside" of the object's boundaries in its cropped region. Without predefined knowledge of triangles and squares, the agent instead categorizes shapes based on these properties and gives them arbitrary names. For example, the agent may label all triangles as "shape1" and all squares as "shape2", even if the individual objects vary in size and other properties across the problem, based on these measurements. To account for variations in the measurements, objects are rotated to optimize an arbitrary scoring function. This also helps determine relative rotation angles between objects which are necessary in certain problems.

To take an example, in Figure 1, there are no overlapping objects in the frames. Individual objects are easily isolated, and the shapes of these objects are distinguished by the relative outside whitespace. Other properties, such as relative size and position, are also computed. In frames A and B, the agent records as the target relationship that the single object in frame B has the same shape (shape2) as both of the objects in frame A and the same size as the larger object in frame A. The agent then compares frame C with each answer frame to find the closest match to this relationship. An exact match is not possible because frame C contains two different shapes (shape1 and shape3) rather than a single shape. The correct answer, frame 2 with the large triangle (shape3), is chosen because it matches all aspects of the target relationship other than the object matching the shape of the smaller object. Thus, the concept of shape is used to mark objects as being different from or similar to other objects, and as long as the agent correctly observes those differences in the visual analysis portion it will have enough information to solve the problem.

The process for the problems in Figures 2 and 3 is similar, although the addition of rotating objects demands the

agent's rotation logic. For example, in the first frame of Figure 3, the two outer triangles are already at the "ideal" rotation angle and are given an angle value of 0 degrees, whereas the middle triangle would reach the same "ideal" value after being rotated 180 degrees. As noted before, the primary difference between Agent 1 and Agent 2 is that while Agent 1 relies on prior knowledge of shapes and their potential properties, Agent 2 takes a grounded method to identifying shapes in a frame. Thus, while Agent 1 will fail to recognize previously unseen shapes, Agent 2 is equipped to address previously unidentified shapes.

Agent 2 performed exceptionally well, correctly answering 83 of the 123 problems (78 of the 100 problems for credit). It is notable, though, that Agent 2's performance lagged behind on the 'Extra' problem sets; many of these sets included transformations, such as counting the sides of a shape, for which Agent 2's more visually-oriented method does not account. We also hypothesize Agent 2 would show greater success on problems featuring previously unseen shapes that humans could similarly address, but no such problems were included here.

Like Agent 1, Agent 2 can also generate novel answers rather than select them from a set of possible answers. The paragraph above acknowledged that on the problem presented in Figure 1, the most-obvious answer to Agent 2 is not present among the answer candidates. To have a 'most obvious' answer prior to examining the choices, Agent 2 must generate its own solutions. This also reveals how the presence of candidate answers can encourage creativity by introducing new constraints. It is creative to generate novel solutions from scratch, but it is also creative to generate arguments for available non-obvious solutions.

Agent 3: Visual Heuristics

In contrast to Agents 1 and 2, Agent 3 does not derive any representation of the visual analogy problems. Agent 3 begins from the supposition that it is fundamental to reduce the input space to something both *manageable* and *meaningful* for the agent to be able to *compute* and *correctly guess* an answer from the given choices. Agents 1 and 2 do so by reducing the input space to a propositional representation; Agent 3 reduces the input space to sets of contiguous non-white pixels.

Agent 3 takes each possible answer choice and computes the likelihood it is correct. To do so, the agent takes a series of measurements capturing the relationship between each training pair, which is described by any two adjacent cells in the matrix. It then compares those measurements against each of the test-answer pairs, the combinations of any cell adjacent to the empty slot and each answer choice. Each comparison, if significant enough, casts a vote for the current answer as the likely answer with a weight directly proportional to the believed similarity of the cells. The most-voted answer is selected as the agent's answer.

Many relationship measurements were evaluated, such as grid-based similarity, histogram-based similarity, and affine transformations. After multiple iterations, few measures were needed to yield the best performance. In the final design, the agent only uses the following two measurements:

- **Dark pixel ratio**: the difference in percentage of the number of dark-colored pixels with respect to the total number of pixels in the contiguous pixel sets of two matrix cells.
- **Intersection pixel ratio**: the difference in percentage of the number of dark-colored pixels present at the same coordinates with respect to the total number of dark-colored pixels in both matrix cells for a given set of contiguous pixels.

For example, in Figure 1, the intersection pixel ratio would lead the agent to vote for the answers containing an outer square; this is analogous to the most logical answer to the problem, an outer square with the inner object removed. Counterintuitively, the correct answer is just the expanded triangle, but the agent would also vote for that answer based on the dark pixel ratio's similarity to the most logical answer. Hence, thanks to the simple metrics used, the agent is "immune" to problems that may appear deceiving at first glance or may involve convoluted transformations. Although for this particular example, the agent picked answer 6, the correct answer was evaluated to be only 6.76% less likely to be correct.

Agent 3 performed exceptionally well, correctly answering 102 of the 123 problems (82 of the 100 problems for credit). Agent 3 gave the most correct answers of any agent, although a greater proportion of its correct answers were previously-seen problems than Agent 1's similarly high performance. This may suggest that the iterations examining the effectiveness of multiple measures of similarity may have overfit the agent's reasoning to those problems, and that further development with more problems may expand the set of desirable measurements.

Unlike Agents 1 and 2, Agent 3 does not have the capability of generating an answer choice rather than selecting from a set of presented answer choices. This is because while Agents 1 and 2 operate under an implicit ranking of possible choices culminating in an ideal choice, Agent 3 might find numerous options equally ideal, and thus could generate thousands of candidate selections.

Agent 4: Hybrid Reasoning

Agents 1 and 2 use propositional representations of the target problem while Agent 3 uses purely imagistic representations; Agent 4, by contrast, leverages both and takes a hybrid method. This method asks the question: can an agent quickly find patterns and relationships in a problem through a high-level visual comparison? If the agent can find high-level visual relationships quickly, it can efficient-

ly formulate a solution without any further propositional understanding of the problem. If no such visual relationships are found, the agent may look for lower level propositional relationships present in the problem.

Thus, Agent 4 starts by examining frames for visual relationships and transformations that can be quickly detected by visual inspection. The agent uses image similarity to detect rotation, vertical and horizontal reflection, the identity transformation, image addition, XOR, and NOR. If this process detects the presence of one of these relationships within a matrix problem, the agent generates a prospective solution and looks for a matching answer. For example, in Figure 1, the transformation between frame A and frame B would be identified through the XOR transformation, which searches for pixels present in only one of two frames. Similarly, in Figure 2, the transformation between frame A and frame B would be identified through the rotation transformation; the agent would (successfully) identify frame 3 as a frame that would complete the same rotation transformation when paired with frame C.

This imagistic method was successful in finding solutions to over 20% of the problems, and it was much more computationally efficient compared to extracting propositional representations from the images; this is notable in that it acknowledges the different levels of effort applied by humans in solving these problems. Results could be further improved by searching for more types of high-level relationships and transformations, by applying transformations at a lower granularity than at the image level, and by improving the image comparison. For example, at present, Agent 4 is unable detect the visual transformations between parts of frames in Figure 2.

This visual method has difficulty finding relationships that cannot be represented through affine transformations, such as problems involving prior knowledge of shapes and properties represented in the frames. When the agent is confronted with problems like these, it will try to find low-level relationships using contour recognition to identify shapes and object properties, ultimately leading to a method similar to Agent 1.

Agent 4 performed exceptionally well, correctly answering 79 of the 123 problems (66 of the 100 problems for credit). Although these scores are the lowest among these four agents, they are in the top 10% of agents submitted. Moreover, Agent 4 may represent the best approximation of human reasoning; humans can discuss problems in both visual and propositional terms (Kunda, McGreggor & Goel 2011), and Agent 4 similarly can do both.

As noted in the description above, during the first phase of its reasoning, Agent 4 generates prospective solutions and compares those prospective solutions to the answer choices. Thus, it already engages in creative answer generation and compares the generated answers to the candidate solutions.

Discussion

Agents 1 and 2 above exemplify Hunt's (1974) analytical, propositional reasoning strategies for addressing RPM problems. Agent 1 extracts propositonal representations that describe the shapes, spatial relations, and transformations from the input images, and then operates on those representations. Agent 2 also extracts propositional representations, but these representations are grounded in the transformations between objects: it has no prior knowledge of shapes, but rather the ability to generate representations of the transformations themselves. Agents 3, on the other hand, exemplifies Hunt's "Gestalt" visual reasoning strategy for RPM. It uses visual abstractions over problems to approximate the answer even without precise knowledge of the transformations between frames. Agent 4 combines the two methods: it first leverages the immediately-identifiable "intuitive" answer that can be established from accessible visual transformations before resorting to more complex propositional reasoning strategies. Thus, Agent 4 demonstrates the possibility of creatively combining methods. As far as we know, the precise strategies used by these agents have not appeared in the literature on the RPM test.

These four agents, along with the 220 other agents developed over the course of this project, reflect the ability of AI agents to succeed on a test of human intelligence that relies on creative and flexible problem-solving. This experiment suggests that there may be no one single "right" problem-solving strategy for the RPM test, that creativity on the RPM test may entail a large number of problem-solving strategies, and that we have so far discovered only a subset of creative problem-solving strategies. Future research along these same lines will test future agents against the authentic RPM test; examine patterns of errors in agents' performance for comparison to human performance (Kunda et al. 2013) including atypical cognition (Kunda & Goel 2011); and better articulate the strengths and weaknesses of different methods (Lynn, Allik, & Irwing 2004; Kunda et al. 2013). We will also examine merging multiple agents into a single agent equipped with metacognitive ability to select among the different strategies, thus more closely approximating factors that determine human success on such tests (Keating & Bobbitt 1978).

Conclusions

The RPM test admits many problem-solving methods, which in part is what makes it a good test of intelligence and creativity. The various problem-solving methods differ in both the knowledge representations and control of processing they use. In this paper we described a human computation strategy for acquiring novel problem-solving methods for addressing RPM-inspired visual analogy problems. This strategy resulted in the design of 224 AI agents for addressing 123 visual analogy problems. Some of the

agent designs were both novel and effective: we described four of these agent designs.

An important issue in computational creativity is how to acquire knowledge of creative methods. Our research suggests that human computation may be a useful strategy for this acquisition, especially when the computation comes from intelligent, educated, high-achieving humans who themselves are likely to do well on a creative task.

Acknowledgements

We thank all 224 students in both the in-person and online sections of CS 7637 KBAI course at Georgia Tech in Fall 2014. Goel was the primary instructor of both sections; Joyner was the course developer and head TA of the online section; Lemmon, Graham, Martinez, and Bedwell were four students in the online course and developed agents 1, 2, 3, and 4, respectively.

We are grateful to Maithilee Kunda and Keith McGreggor for their prior work on which this project builds. We also thank the course's teaching team: Lianghao Chen, Amish Goyal, Xuan Jiang, Sridevi Koushik, Rishikesh Kulkarni, Rochelle Lobo, Shailesh Lohia, Nilesh More, and Sriya Sarathy. We also thank the anonymous reviewers of this paper: their comments truly helped improve the discussion.

References

Bringsjord, S., & Schimanski, B. (2003). What is Artificial Intelligence? Psychometric AI as an answer. In *Procs*. *18th IJCAI*, 887-893.

Dastani, M., Indurkhya, B., & Scha, R. (2003). Analogical Perception in Pattern Completion. *JETAI 15*(4), 489-511.

Evans, T. (1967). A Program for the Solution of a Class of Geometric Analogy Intelligence-Test Questions. In M. Minsky (ed.) *Semantic Information Processing*. MIT Press.

Goel, A. & Joyner, D. (2014). CS7637: Knowledge-Based AI: Cognitive Systems [Online Course]. Retrieved from http://www.omscs.gatech.edu/cs-7637-knowledge-based-artificial-intelligence-cognitive-systems/

Goel, A. & Joyner, D. (2015). An Experiment in Teaching Cognitive Systems Online. Technical Report, Georgia Institute of Technology.

Goel, A., Kunda, M., Joyner, D., & Vattam, S. (2013). Learning about Representational Modality: Design and Programming Projects for Knowledge-Based AI. In *Fourth AAAI Symposium on Educational Advances in Artificial Intelligence*.

Howe, J. (2008). *Crowdsourcing: Why the Power of the Crowd is Driving the Future of Business*. Crown.

Hunt, E. (1974). Quote the raven? Nevermore! In L. W. Gregg (Ed.), *Knowledge and Cognition*. 129-158. Hillsdale, NJ: Erlbaum.

Keating, D. , & Bobbitt, B. (1978). Individual and developmental differences in cognitive-processing components of mental ability. *Child Development*, 155-167.

Kirby, J., & Lawson, M. (1983). Effects of strategy training on progressive matrices performance. *Contemporary Educational Psychology*, *8*(2), 127-140.

Kunda, M., & Goel, A. (2011). Thinking in Pictures as a Cognitive Account of Autism. *Journal of Autism and Developmental Disorders, 41*(9), 1157-1177.

Kunda, M., McGreggor, K., & Goel, A. (2013). A Computational Model for Solving Problems from the Raven's Progressive Matrices Intelligence test using Iconic Visual Representations. *Cognitive Systems Research, 22*, 47-66.

Kunda, M., Soulieres, I., Rozga, A., & Goel, A. (2013). Methods for Classifying Errors on the Raven's Standard Progressive Matrices Test. In *Proceedings of the 35th Annual Meeting of the Cognitive Science Society*, 2796-2801. Berlin, Germany.

Law, E., & von Ahn, L. (2011). *Human Computation*. Morgan & Claypool.

Lovett, A., Tomai, E., Forbus, K. & Usher, J. (2009). Solving geometric analogy problems through two-stage analogical mapping. *Cognitive Science 33*(7), 1192-1231.

Lynn, R., Allik, J., & Irwing, P. (2004). Sex differences on three factors identified in Raven's SPM. *Intelligence, 32*, 411-424.

McGreggor, K., Kunda, M., & Goel, A. (2014). Fractal and Ravens. *Artificial Intelligence 215*, 1-23.

O'Donoghue, D., Bohan, A., & Keane, M. (2006). Seeing Things: Inventive Reasoning with Geometric Analogies and Topographic Maps. *New Generation Computing 24* (3), 267-288.

Prade, H. & Richard, G. (2011). Analogy-Making for Solving IQ Tests: A Logical View. In *Procs*. *19th International Conference on Case-Based Reasoning*, 561-566. London, UK: Springer.

Ragni, M. & Neubert, S. (2014). Analyzing Raven's Intelligence Test: Cognitive Model, Demand, and Complexity. In H. Prade & G. Richard (Eds.) *Computational Approaches to Analogical Reasoning: Current Trends*, 351-370. Springer.

Raven, J., Raven, J. C., & Court, J. (1998). *Manual for Raven's Progressive Matrices and Vocabulary Scales*. San Antonio, TX: Harcourt Assessment.

Schwering, A., Krumnack, U., Kuhnberger, K-U, & Gust, H. (2009). Spatial cognition of geometric figures in the context of proportional analogies. In *Procs*. *Spatial Information Theory, Lecture Notes in Computer Science Volume 5756*, 18-35.

Snow, R., Kyllonen, P., & Marshalek, B. (1984). The topography of ability and learning correlations. *Advances in the Psychology of Human Intelligence*, *2*, 47-103.