Some Challenges and Grand Challenges for Computational Intelligence

EDWARD A. FEIGENBAUM

Stanford University, Stanford, California

1. *Introduction*

When the terms “intelligence” or “intelligent” are used by scientists, they are referring to a large collection of human cognitive behaviors—*people thinking*. When life scientists speak of the intelligence of animals, they are asking us to call to mind a set of human behaviors that they are asserting the animals are (or are not) capable of. When computer scientists speak of artificial intelligence, machine intelligence, intelligent agents, or (as I chose to do in the title of this essay) computational intelligence, we are also referring to that set of human behaviors. Although intelligence means *people thinking*, we might be able to replicate the same set of behaviors using computation. Indeed, one branch of modern cognitive psychology is based on the model that the human mind and brain are complex computational “engines,” that is, we ourselves are examples of computational intelligence.

2. *Turing’s Vision and the Turing Test for Humanoid Behavior*

The idea, of course, is not new. It was discussed by Turing in the 1940s. In the play about Turing’s life, Breaking the Code [Whitemore 1987], Turing is shown visiting his old grammar school and delivering a talk to the boys, in which he offers a vision of the thinking computer. The memories of those of Turing’s colleagues of the 1940s who are still alive confirm that he spoke often of this vision. In 1950, he wrote of it, in a famous article [Turing 1950], in which he proposed a test (now called the Turing Test (TT)) for computational intelligence. In the test, a human judgment must be made concerning whether a set of observed behaviors is sufficiently similar to human behaviors that the same word—intelligent—can justifiably be used. The judgment is about behavior not mechanism. Computers are not like human brains, but if they perform the same acts and one performer (the human) is labeled intelligent, then the other must be labeled intelligent also.

I have always liked the Turing Test because it gave a clear and tangible vision, was reasonably objective, and made concrete the tie to human behavior by using the unarticulated criteria of a human judge. Turing Award winner Jim Gray, who works in fields of Computer Science other than AI, appears to agree. His list of challenges for the future includes: “The Turing test: Win the imitation game 30% of the time.” Significantly, he adds: “Read and understand as well as a human. Think and write as well as a human,” [Gray 2003]. I will have more to say about necessary conditions for these human activities later.

But there are problems with the Turing Test (TT). Human intelligence is very *multidimensional*. However, the judge must fuse all of these dimensions into a

Journal of the ACM, Vol. 50, No. 1, January 2003, pp. 32–40.

single judgment about the “humanness” of the behavior. The computer scientists who work in the areas called, or allied to, Artificial Intelligence usually must study these dimensions *separately*. Successes along any one of the dimensions can only be considered “partially intelligent” by Turing’s criterion. Imagine doing a TT of a computational intelligence that was as good as an Einstein (who, after all, was *Time* magazine’s “Man of the Century”) in inducing and creating physical theories, but was severely lacking in its ability to handle ordinary language. Still, an appropriate strategy for a scientific field to conduct its inquiry is divide-and-conquer—study the dimensions of intelligence more or less separately. We must be satisfied for a long time to come with “partial intelligence” in our artifacts as a natural consequence of this inevitable strategy.

In dividing-and-conquering, these are some examples of the many human behaviors that we either have divided out for study, or ought to (taken from a recent paper by Gentner [2003]):

—The ability to draw abstractions from particulars.
—The ability to maintain hierarchies of abstraction.
—The ability to concatenate assertions and arrive at a new conclusion.
—The ability to reason outside the current context.
—The ability to compare and contrast two representations for consistency/inconsistency.
—The ability to reason analogically.
—The ability to learn and use external symbols to represent numerical, spatial, or conceptual information.
—The ability to learn and use symbols whose meanings are defined in terms of other learned symbols.
—The ability to invent and learn terms for abstractions as well as for concrete entities.
—The ability to invent and learn terms for relations as well as things.

3. *Partially Intelligent Artifacts and What We Have Learned from Them*

Although Turing’s vision is far from being achieved, very substantial progress has been made in creating computational models of most of the dimensions of cognition and in constructing software to realize these models. I will give two examples briefly, and then follow with a third that is discussed at length.

In natural language processing, though our computational artifacts can not yet “read and understand as well as a human,” computational linguists have developed superb models for the processing of human language grammars. Where they have lagged is in the “understand” part: the semantics that attach real-world meanings to the word- symbols, then use those meanings for knowledge organization and inference.

In computer vision, scene understanding is excellent in specific situations for which knowledge of the context and the objects of the scene can be explicated. Indeed, because computer “eyes” can be multispectral (and human eyes are not), some computer vision programs can exceed human visual abilities. What is missing is keyed by the term, *specific situations*. Vision programs lack a breadth of knowledge of contexts and objects.

4. *Computational Intelligence Working on Very Hard Problems*

As I chose my own path using the divide-and-conquer strategy, I chose to model intelligent behavior of the most complex sort—reasoning tasks that were difficult for skilled humans and indeed impossible for most humans. These were the software artifacts that I later called *Expert Systems* (*ES*). ES are called that because their behavior, their performance on solving difficult tasks, rivals the best human experts in certain specific areas (called domains).

In 1965, when this work began, this basic idea of high-performance problem-solving was not new. In the mid 1950s, a statistician who was a colleague of Turing in World War II, I. J. Good, speculated on progress toward "an ultraintelligent computer" [Good 1965]. Only a year or two later, Gelernter's program for proving theorems in plane geometry exhibited a better performance on a standardized test in plane geometry than any of the thousands of high-school students taking the same test [Gelernter et al. 1963]. Four years later, Samuel's program for checker playing beat one of the best human players in the USA [Samuel 1963]. And, decades later, a chess-playing program beat the world's chess champion.

The ES of 1965 and beyond were in one important dimension much more complex. The ES examples of computational intelligence were constructed to perform their expert-level behavior in difficult real world areas whose context includes large bodies of knowledge. Examples include: medicine, various areas of physical science and engineering, and the analysis and control of many business and manufacturing processes. The ES for planning complex operations were especially powerful when compared with the best human behavior in the application areas. As an example of high performance, one of these programs found important use in the NATO Air Operations Center during the Bosnian campaign. Two others formed the basis for successful software companies in manufacturing planning.

All of these ES examples of computational intelligence are examples of *partially intelligent* artifacts. They were developed to satisfy the scientific need to understand complex problem solving behavior and the engineering need to apply such understanding. None would pass Turing's Test. They are designed to be specific to domains and to certain tasks within those domains. They lack breadth in their available behaviors and have limited flexibility in their ability to interact with people. Yet their task performance has been, in some cases, world-class. There have been tens of thousands of them built. Each one constitutes an experiment on the path to human-level complex problem solving—and beyond, to Good's concept of an "ultra-intelligent" computer.

What is the most important thing we have learned from all of these experiments? The answer turns out to be simple, seemingly obvious in retrospect (but many important scientific truths seem obvious in retrospect).

For an artifact, a computational intelligence, to be able to behave with high levels of performance on complex intellectual tasks, perhaps surpassing human level, it must have extensive knowledge of the domain. Knowledge means things likes terms for entities, descriptions of those entities, relationships that organize the terms and entities for reasoning, symbolic concepts, abstractions, symbolic models of basic processes, fundamental data, a large body of remembered instances, analogies, heuristics for "good guessing," among many other things.

Computer scientists, with their typical backgrounds from logic and mathematics, enjoy creating elegant and powerful reasoning methods. But the importance of these methods pales in comparison with the importance of the body of domain knowledge—the artifact's knowledge base.

This is so important a result that it is almost a principle of AI. To push the point, let me illustrate the obvious. As I write, I can look out my window and see the Stanford Mathematics Department on my left and the Stanford Medical School on my right. In the Math building are some of the most powerful reasoners in the world, at least one of whom has won the Fields Medal. If I were, at this moment, to have a threatening medical event, I would ask my CS colleagues to rush me to the Medical School, not the Math Department. The powerful reasoning of the mathematicians would be essentially totally ineffective compared with the knowledge of medicine of the doctors in the Emergency Room of the Medical School's hospital. For the doctors to apply their knowledge effectively to my case will require some simple reasoning, but not logically powerful and elegant methods.

In fact, I believe that we now have an overabundance of such methods that have yet to receive adequate test of utility and integration into our AI software systems. To use these methods effectively, the field faces a major challenge of experimentation and system integration.

But the field faces a much greater challenge in the sphere of knowledge: knowledge representation, and (especially) knowledge acquisition.

As early as 1977, two well-known MIT AI researchers wrote about AI's "shift to the knowledge-based paradigm" [Goldstein and Papert 1977]. The understanding of what needs to be done is widespread, but the practice is not. Ask a computational linguist why her natural language-processing system is not performing well on some task, or what her system would require to understand the text in documents on the WWW, and she will usually answer correctly "knowledge of the domain of discourse." But it is only in the last few years that the community of computational linguists has begun to codify knowledge, developing the volunteer-built semantic net known as Wordnet.

Computer scientists do not like the seemingly endless and painstaking task of representing knowledge for use by computational intelligence. It is in small part the logic of knowledge representation (they like that part) and in large part epistemology ("that's somebody else's problem"). The enormous effort of the CYC research group (www.cyc.com), led by Lenat, to codify a useable body of "common sense" general knowledge has largely been done by graduates in philosophy, religion, and many other disciplines, but not computer science. Yet CYC-like knowledge is precisely the knowledge that will enable a computational intelligence to pass TT—if one ever does, and I believe eventually one will.

5. *Challenges and Grand Challenges*

5.1. TURING'S TEST REVISITED. As I mentioned, Gray proposed the challenge of a computational intelligence passing TT in its original form 30% of the time. The facet of an intelligence being examined by TT is largely one of breadth, or span, of human activity and concerns that are represented, implying for the computational intelligence a huge knowledge base. Acquiring such a large computer-useable knowledge base is a Very Grand Challenge.

I propose this alternative to TT. Suppose we choose to test the facet of quality (the complexity, the depth) of reasoning, that which distinguishes the Einstein from you and me. This is a more manageable task, since computer scientists are much better masters of reasoning processes than of knowledge bases. Here is a reformulation of the TT that attempts to capture the facet of quality. To avoid possibly polluting Turing's legacy with this revision, let me call it the Feigenbaum Test (FT).

Two players play the FT game. One player is chosen from among the elite practitioners in each of three preselected fields of natural science, engineering, or medicine. (The number could be larger, but for this challenge not greater than ten). Let's say we choose the fields from among those covered in the US National Academy Complex, and the player must be a member of the National Academy. For example, we could choose astrophysics, computer science, and molecular biology.

In each round of the game, the behavior of the two players (elite scientist and computer) is judged by another Academy member in that particular domain of discourse, for example, an astrophysicist judging astrophysics behavior. Of course, the identity of the players is hidden from the judge, as it is in TT. The judge poses problems, asks questions, asks for explanations, theories, and so on—as one might do with a colleague. Can the human judge choose, at better than chance level, which is his National Academy colleague and which is the computer?

The game can be played several times to enhance the statistics of the test, using different pairs of Academy members from the selected domain, one being a player and one being a judge. To factor out the facet of intelligence related to full natural language understanding, the framers of the game might decide at the outset that the interactions will be conducted in the heavily jargonized and stylized language with which practitioners in the selected domains usually communicate their problems and issues.

Referring back to Gray's criterion for success of the computational intelligence, the challenge will be considered met if the computational intelligence (CI) "wins" one out of three disciplinary judging contests, that is, one of the three judges is not able to choose reliably between human and computer performer. Relative to what has been accomplished in the Expert Systems area until now, in the building of knowledge bases for science, engineering, and medicine, this is a formidable Grand Challenge. But it is still far from the extraordinary grand challenge of the ultraintelligent computer (UIC). Paradoxically, the UIC would be easily discernible from the elite human performer. It would be offering inductions, problem solutions and theories that were not yet reached by any human, yet were plausible, rigorous upon explanation, and either correct or interesting enough to subject to experimental test.

My guess is that, if we seriously attack the AI and epistemological issues underlying success in the FT, the 100$^{\text{th}}$ anniversary issue of the *JACM* will be able to publish some UIC results.

5.1.1. *Grand Challenges* 2 *and* 3. It is clear from what I have already said that I believe the key, and indeed central, component of an intelligent system (a "CI") is its large knowledge base. Building it is the bottleneck engineering problem for constructing CI. That fact has been known for more than two decades. The missing science in this area of AI led to an explosion of research in machine learning.

Initially, it was believed that the machine learning processes would learn symbolic concepts built up out of symbolic entities, relations, and ontologies. For example, in the late 1960s, I collaborated on the Meta-DENDRAL machine learning project, the result of which was a publication in the mainline literature of chemistry of a hitherto undiscovered symbolic model of the mass-spectral fragmentation of an interesting family of organic molecules [Buchanan et al. 1976]. It was published because it was a contribution to mass-spectral fragmentation theory.

In the years following, the machine learning field moved away from symbolic concepts toward the border of statistics, where the entities and relations were statistical in nature. This work had a large impact and helped to fuel the new field of data mining, but it had essentially no impact on the construction of the large knowledge bases at the core of CI. A team like Lenat's CYC team continued to encode symbolic concepts manually, one by one. So the "missing science" is still mostly missing. Challenges 2 and 3 below are my attempts to state lines of research to fill this important, indeed crucial, near-void.

5.2. Build a Large Knowledge Base by Reading Text, Reducing Knowledge Engineering Effort by One Order of Magnitude. This Grand Challenge is not novel. It is indeed implied by Gray's subchallenge to read and understand. It even has been tried in limited contexts, by Buchanan and others. Reddy has stated the challenge this way: "Read a chapter of a text and answer the questions at the back of the chapter."

The intent here is to "educate" a knowledge base in the same way that we receive most of our education. We "inherit" from our cultures a large part of what we know via the written word (although some of our knowledge comes from experience and apprenticeship). For most of what we know, symbolic learning is built up from prior learned symbolic structures represented in our knowledge bases.

The challenge I want to pose is partly a *machine learning challenge* and partly a challenge of engineering an *educational strategy*.

Let me try to make the challenge a concrete one by reference to my Grand Challenge 1. If the performance of the CI competing in the FT is excellent, this will be largely attributable to its knowledge of its particular domain. For some specific domain, later to be used in an FT, start this grand challenge by doing two things.

First, manually encode a novice-level understanding (symbolic representation) of the domain, that is, humans will do the knowledge engineering. The novice-level "view" of the domain can be taken directly from a well-regarded elementary text of the domain, for example, an introduction to molecular biology.

Second, write the software for a system that will read the "next" text in the field, augmenting as it reads the kernel novice-level knowledge base.

To make this a more practical prospect, some human intervention will be allowed. The program's natural language capabilities may not be fully able to cope with the language used, so queries for clarification may need to be answered. Occasional direct intervention into the symbolic structures may be needed to introduce a missed concept or to correct a "misunderstanding" that found its way into the knowledge base. (Note in passing that it is not my view that a CI must be perfect. To paraphrase a famous quotation: "To err is the fate of all intelligence." It is a consequence of bounded rationality.)

The amount of knowledge base development and change introduced by the human intervention should not exceed ten percent of all the symbolic structures. This translates into speeding up the development of large knowledge bases by approximately one order of magnitude, which is grand enough for this grand challenge.

The introduction of domain concepts to the CI must occur in some order. There will surely be preferred orderings (but probably not a unique one) in which the more advanced readings should be presented. This is the part of the challenge that involves the educational strategy. I view this as a challenge of engineering design of an educational strategy because we have the opportunity of studying at all times what is “inside the black box.” This is a luxury that educators do not normally have.

Progress on this grand challenge will be measured by the performance of the CI being educated by its reading. The grand challenge can be achieved incrementally. Ultimate “victory,” of course, is to bring the CI to the state of knowledge with which it can pass the FT. But other midpoints would indicate significant progress: human-level performance at the end of a college major; at the end of a master’s degree; at the granting of a Ph.D.; and so on up the normal levels of human performance.

If this were to succeed, it would lead to another Grand Challenge, but one that is manageable given what had to have been done before. The educated CI would continue to educate itself by reading the emerging literature of the domain. That is, it would “keep up with the literature.” Human assistance will still be allowed, but less than was allowed earlier. Indeed, one could think of this phase as “collaboration,” since both human and CI will be learning the new material at the same time.

The test in this Grand Challenge would be simply an extension of the FT (EFT) Biannually, the educated CI would take the FT. Do it twice. The criterion for passing the FT would remain the same. To pass the EFT, the CI must pass the FT one time in the four years (two repetitions) of the extension.

5.3. Distilling from the WWW a Huge Knowledge Base, Reducing the Cost of Knowledge Engineering by Many Orders of Magnitude. The WWW can be thought of as the world’s largest database, especially if one includes all the information that is accessible via links. It contains much of the world’s current events, cultural history, and accumulated knowledge in textual and other informational types. It is truly a *mirror* of our human knowledge, or perhaps more accurately a transformation of our knowledge. For knowledge engineers trying to build CI that will pass TT or FT, it is the tempting apple on the tree of knowledge.

However, the WWW is not a knowledge base. Except for supporting data, most of what the WWW contains can not participate directly in inference making, problem solving, and decision making that a CI must do. The WWW, simply put, does not represent knowledge using any of the standard tools of knowledge engineering, logic, and probabilistic inference. Therefore, the Grand Challenge for plucking this treasured apple (perhaps the ultimate apple) from the tree would be to “make it so,” to use the language of a popular science-fiction epic.

I would like to call this a Grand Vision, because the word “Challenge” suggests that I have in mind a test for when the Challenge is met. Perhaps the methods being studied in this Grand Vision can eventually be used to try to create the CI artifacts that will pass my Grand Challenges 1 and 2, and that would convert the Grand Vision into a Grand Challenge 3.

Work on this Grand Vision began in the late 1990s on an international scale. The work is vigorously supported by the US government (DARPA) and the European

Union (Framework), and has the considerable support of the WWW Consortium (W3C) under the label of the Semantic Web. The conception is that the "sources of knowledge" are widely distributed, and number in the hundreds of thousands, perhaps millions of web page owners and their pages.

Tools are being created to (among other things):

—Facilitate the building of general and specific ontologies (logical structures for organizing knowledge).

—Facilitate the integration of the ontologies built by many people (or programs) in many places into logically consistent representations that will be highly efficient, and will probably have to pass a human "editing" examination.

—Give the multitude of web page creators a markup language in which each can do an extensive semantic markup of his/her textual submission (and perhaps other information types). XML, of course, is a foundation stone. For AI scientists, RDF (Resource Description Framework) follows naturally. The present work of the DAML+OIL international project has the promise of eventual distribution of user-friendly semantic processing and markup tools to all web page builders. For more, consult www.semanticweb.org.

The semantic markups are just the raw material for a huge global knowledge base. To implement the Grand Vision, knowledge engineers must build a system of "semantics scrapers" that will access the semantic markups, integrate them appropriately into the growing knowledge base, and set up the material for the scrutiny of an editorial process.

Will this work? Or is the apple too far up the tree to be plucked in our era of computer science and engineering research? We will know within the next decade.

6. *Concluding Remarks: Manifest Destiny*

In this essay, I have given challenges and Grand Challenges (only a few) for researchers doing Computational Intelligence, that is, in the AI area of computer science. It would have been an uncomfortable high-wire act for me to have tried to set challenges in other areas of CS. But far beyond personal questions of intellectual sure-footedness, I hold a strong belief that Computation Intelligence is the destiny of CS. I hold no professional belief more strongly than this. I call computational intelligence the manifest destiny of computer science.

I learned the term "manifest destiny" when I studied American History as a young student. In the early 19th Century, when the small population of the new United States extended only to the Appalachian Mountains of the east, great visionaries like Thomas Jefferson imagined a USA that encompassed all territories to the far ocean at the continent's western edge. That vision, motivating generations of settlers and policy makers, was called the Manifest Destiny.

Computational Intelligence *is* the manifest destiny of computer science, the goal, the destination, the final frontier. More than any other field of science, our computer science concepts and methods are central to the quest to unravel and understand one of the grandest mysteries of our existence, the nature of intelligence. Generations of computer scientists to come must be inspired by the challenges and grand challenges of this great quest.

REFERENCES

BUCHANAN, B. G., SMITH, D. H., WHITE, W. C., GRITTER, R., FEIGENBAUM E. A., LEDERBERG, J., AND DJERASSI, C. 1976. Applications of artificial intelligence for chemical inference. XXII. Automatic rule formation in mass spectrometry by means of the meta-DENDRAL program. *J. ACS 98*, 61–68.

GELERNTER, H, ET AL. 1963. Empirical explorations of the geometry theorem proving machine. In *Computers and Thought*, E. Feigenbaum and J. Feldman, Eds. McGraw-Hill, New York, pp. 153–167 (reprinted, AAAI Press 1995).

GENTNER, D. 2003. Why we're so smart. In *Language in Mind: Advances in the Study of Language and Thought*, D. Gentner and S. Goldin-Meadow, Eds. MIT Press, Cambridge, Mass.

GOLDSTEIN, I., AND PAPERT, S. 1977. Artificial intelligence, language, and the study of knowledge. *Cog. Sci. 1*, 84–123.

GOOD, I. J. 1965. Speculations concerning the first ultraintelligent machine. *Adv. Comput. 6*, 31–88.

GRAY, J. 2003. What Next? A Dozen Information-Technology Research Goals. *JACM 50*, 1 (Jan.), 41–57.

SAMUEL, A. 1963. Some studies in machine learning using the game of checkers. In *Computers and Thought*, E. Feigenbaum, and J. Feldman, Eds. McGraw-Hill, New York (reprinted, AAAI Press 1995).

TURING, A. M. 1950. Computing machinery and intelligence. *Mind 59*, 433–460.

WHITEMORE, H. 1987. *Breaking the Code*. Amber Lane Press.