Faculty Perspective offers a space for McCormick professors to comment on issues facing their fields of research. In “Watson for Everyone,” three computer science professors lay out how we can turn software from tool into collaborator.

The problems we face are growing ever more complex, but our human cognitive capacities remain unchanged. People and organizations are deluged with a rising tide of potentially relevant information, ranging from books, documents, and magazines to blog postings and tweets. Search technologies help, but even at their best, they provide only candidates, often far too many candidates, relying on people to do their own filtering and synthesis. What we need is systems that complement us, absorbing the tsunami of potentially relevant information, winnowing out the most relevant parts of it, and synthesizing material from multiple sources to produce actionable knowledge. These systems will embody a movement away from the traditional model of software as tool to a new model, software as collaborator, leading to a revolutionary impact of computing on all spheres of human life.

IBM’s Watson provides a useful example. Watson’s performance was revolutionary: it showed how a synergistic combination of artificial intelligence (AI) techniques could be used to perform fact-based question answering at a level that no one thought possible even a few years ago. (Fact-based question answering involves retrieving facts or combining facts in straightforward ways to select possible answers.) Watson used machine-reading techniques to assimilate vast collections of documents (more than 100 million pages) into internal representations that supported integration and reasoning. Machine-learning techniques helped Watson determine from experience which strategies were likely to succeed for different types of questions. Real-time response was provided by using massive hardware resources, capable of performing at the level of the best humans at its task.

Consider a Watson-like system applied to your documents, learning how to answer questions that matter to you. If you’re a scientist, the documents would include the wide-ranging technical literature and potentially your laboratory notebooks, emails, and other records your organization maintains. If you’re an intelligence or business analyst or a journalist, the documents would include a vast array of information sources, as well as detailed documents concerning particular subjects of interest. If you’re a manager, the documents would include your organization’s records and relevant news sources. If you’re a teacher, they would include materials in the areas you teach, journals and forums describing the latest advances in techniques and practices, and school records. Even just fact-based question answering, if extremely accurate and fast, based on your documents and answering questions relevant to you, would be quite valuable. For that reason, we fully expect that a variety of Watson-like systems will be constructed by many organizations. It may seem

cases people find clever ways to do it with fewer resources. Deep Blue, after all, required what, for its time, was substantial parallel hardware, yet within a few years there were chess programs operating at almost its level of play while running on stock hardware.

Such systems will be an initial step toward the software-as-collaborator model: the software is starting to adapt to your world, instead of you adapting to it. However, it would still lack many of the capabilities that we expect out of our human collaborators. Here are what we see as the core problems that must be solved to achieve our vision of software collaborators:

Reading for deeper understanding: Watson’s reading processes can be viewed as a kind of skim reading, gathering factual material about entities and relationships in the world in order to answer questions by retrieving, and occasionally combining, facts about them. To answer deeper questions, material being read must be assimilated into coherent models. This remains an open problem; in fact, reading a textbook and answering its questions based on what was learned has been proposed as a grand challenge for artificial intelligence.

Many real-world problems involve tracking situations and problems over time. For example, keeping up with progress in areas of science and technology or unfolding political situations requires assimilating material being read into ongoing, accumulated conceptual models. This requires deeper reading than Watson used. Moreover, all of the sources given to Watson were reasonably authoritative (e.g., encyclopedias, the complete works of Shakespeare). Most of the time, though, our information sources contain more errors. And sometimes journalists, intelligence analysts, and business analysts must deal with dissembling and disinformation, as well as with the usual errors in sources. This means our software collaborators must help us distinguish fact from fiction.

Teachers, too, could benefit from software that can read and understand student work more deeply. Already natural-language techniques are being used to detect plagiarism and to score certain essay tests, but these rely on fairly crude statistical techniques. Being able to track which students exhibited particular misconceptions would provide a more fine-grained analysis of student progress, which could be used to tailor instruction (both inside and outside the classroom) more effectively.

Richer interaction: Software collaborators need better conversational skills. When interacting with people, Watson only took in questions as input, and each question was answered independently. Human conversation is far richer; we build up a shared context and shared models, ask follow-up and
clarification questions, and pose hypotheticals and alternatives. Our software collaborators need the same skills in order to maintain the shared state of conversations. This context includes both the immediate conversational context and the shared knowledge that collaborators build together concerning their joint problems, plans, and interests.

Interactions are multimodal: people often sketch when they interact, to communicate both spatial ideas (e.g., maps, layouts) and plans and ideas (e.g., concept maps). Software collaborators should be able to participate in sketching, both understanding what is drawn and conveying spatial aspects of information by drawing as required. Understanding gestures and facial expressions is also important as a means of responding to the subtle signals that people tacitly use during conversation to keep things on track.

Collaborators also adapt over time to each other’s communication styles and learn new tasks via interaction using natural modalities. This requires significantly more fluency in language than just question answering: A software collaborator needs to understand instructions and commands. It needs to be able to seek help when it is stuck, explaining what the problem is and taking advice about how to solve it.

**Efficient reasoning at scale:** Filtering and combining information to produce useful knowledge requires combinations of deductive, statistical, abductive, and analogical reasoning. Deductive reasoning involves using logic to determine what does, or does not, follow from what is known about a situation. The person in a story who is pregnant cannot be a male, for example. Statistical reasoning helps determine which logically possible alternatives are more likely: given an ambiguous reference to someone who is pregnant, it is more likely the 20-year-old woman than the 80-year-old woman. Abductive reasoning concerns finding plausible explanations for a situation. For example, a scientist who suddenly stops publishing for a while might have changed institutions, gone into administration temporarily, or spent time doing classified research. Analogical reasoning involves using prior examples to reason about new situations and to construct generalizations based on similar situations. Historical cases are commonly used in political analysis, for example, and one mark of experts is their distillation of practical knowledge from experience. Assimilating knowledge into conceptual models requires reasoning to figure out how the new material fits into what is already known. This can, of course, require scrutinizing already accepted knowledge and rejecting it, if the weight of new evidence indicates that it is incorrect. Answering questions beyond simple fact retrieval also requires more reasoning. Subtle conclusions rest on either deeply reasoning or combining large numbers of disparate facts to reveal hidden patterns, or both. A software collaborator should support both interactive-time question answering and offline reasoning to handle more complex analyses.

**Self-guided learning:** Watson required the services of a large team of technically trained experts to hand-tune its algorithms and reading matter. Software collaborators should require no more routine maintenance than does a human collaborator, i.e., reading and conversation to keep in sync. They need to automatically identify their own conceptual gaps and formulate learning plans to increase their understanding. Unlike Watson, which was constructed to answer trivia questions on general knowledge, software collaborators should be capable of adapting to new tasks and subject areas automatically as their workloads change. They should automatically prioritize investigation of new source material and solicit specific input from their human collaborators when needed.

These four areas include many hard scientific problems; progress on them will move Watson-like systems from grand experiments toward software collaborators that will be useful in all walks of life. Scientists and engineers will be able to more easily focus their attention on aspects of the research literature that are relevant to their current problem. Analysts and journalists will be better able to “connect the dots” and spot patterns and problems that currently escape detection. Decision makers will have relevant precedents ready to hand, as well as help in generating scenarios describing possible outcomes. Teachers will have help in assessing student performance and finding new materials that could improve their classes.

The rising tide of big data can either become a deluge that leaves us gasping for air or the wellspring of information that our software collaborators sift, sort, filter, and organize to provide the information we need, when we need it. We think that creating Watson-like systems—and beyond—for everyone will be an even bigger benefit to humanity than Internet search.

What we need is systems that complement us … leading to a revolutionary impact of computing on all spheres of human life.