Robert S. Engelmore Award Article

Building AI Applications: Yesterday, Today, and Tomorrow

Reid G. Smith, Joshua Eckroth

■ *AI applications have been deployed and used for industrial, government, and consumer purposes for many years. The experiences have been documented in IAAI conference proceedings since 1989. Over the years, the breadth of applications has expanded many times over and AI systems have become more commonplace. Indeed, AI has recently become a focal point in the industrial and consumer consciousness. This article focuses on changes in the world of computing over the last three decades that made building AI applications more feasible. We then examine lessons learned during this time and distill these lessons into succinct advice for future application builders.*

AI Applications of Yesterday and Today

As with the AAAI itself, the Innovative Applications of Artificial Intelligence conference (IAAI) was the brainchild of Raj Reddy. Howie Shrobe summarized the context: "… the emergence of scientific achievements had triggered opportunities to tackle new problems … The point of the conference was to exchange information about what really works and what the real problems are. The goal was to lead to better technology, to find and remedy current deficiencies, and to solve real problems" (Shrobe 1996).

In the preface to the proceedings of the first IAAI conference in 1989, Herb Schorr, program chair, made some interesting comments about the 1989 state of several of the AI technologies that are now well established (Schorr and Rappaport 1989):

  ISSN 0738-4602

Problems and System Types	Specific Applications
Rule-Based Systems: *Widely applied base technology*	TurboTax
Credit Card Fraud Alert	Netflix Recommender
Insurance	FareCast, Google Flights, Kayak price predictor
Scheduling: *Maintenance, Crew, Gate*	Narrative Science GameChanger
Video Games	IBM Watson
Search Engines	Dragon Speech Recognition
Augmented/Virtual Reality	Amazon Robotics / Kiva Systems
Photo Face Recognition	Roomba
Handwriting Recognition: *Mail Sorting, ATM-Checks*	Kinect
Translation	Driver-Assist / Self-Driving Vehicles
Deep Learning	Siri, Cortana, Amazon Echo
Robotics	

Table 1. AI in Use.

> *Expert Systems:* "Nearly all [applications] are expert systems because it is in this form that AI is most rapidly coming into widespread use."
>
> *Robotics:* "[N]o robot software system for complex tasks is commercially available ... robots seem to be stuck with their early applications and have made small commercial progress in the last few years."
>
> *Neural Networks:* "[W]e know of no neural networks in practical day-to-day use ... while this technology appears to possess vast potential ... we leave it for this book's successor to cover such applications."
>
> *Natural Language Processing (NLP):* "[NLP] has been constrained historically by limitations of computational power, but the fantastic progression of computational cost/performance has eliminated this bottleneck. ... [But] today's applications ... are very limited and very few low-level natural language functions are being deployed."

Expert systems were common and successful in the late 1980s in large part because they were able to incorporate domain- and task-specific knowledge; their reasoning engines were relatively simple and, consequently, these systems could be deployed on computer hardware available at the time.

Herb Schorr's comments about robotics, neural networks, and NLP are prescient. In fact, in each area, the story has completely flipped since 1989: today, robots are common in industrial and service applications such as factory automation and farming, and their deployment continues to grow; neural networks make up significant portions of vision, speech, and text-processing systems, and deep learning is one of the more popular research and application areas in AI today; and natural language processing can be found in many applications that billions of people use every day, such as search engines, personal assistants, and web-connected speakers.

Today, AI is everywhere. By contrast with 1989, when very few AI companies were in existence, today many companies, from early stage startups to mature enterprises, are developing AI applications (Zilis 2015).

The world of AI apps is very different as well. In the early days, AI was viewed with suspicion in industry as only the latest hype. Today, AI apps are all around us. Indeed, AI and machine learning are expected in almost every app.

Many, perhaps most, large organizations are making use of AI technologies for market forecasting, customer support, recruiting, fraud detection, scheduling and planning, and other uses. Consumer-oriented examples of AI include Google's search engine, self-driving cars, and Google Now; Apple's Siri (Cheyer 2014); Microsoft's Cortana and Bing; Amazon's Echo; Facebook's automatic photo tagging; Netflix's movie recommendations; and automated check deposits using one of many mobile banking applications. Table 1 shows even more problem and system types, plus specific applications, several of which have been presented at IAAI or AAAI over the years. Of course, not all of these examples are commonly recognized as AI applications — the AI features have disappeared into the fabric. Modern search engines are a good example of this phenomenon.

Although interest in computer science in general dropped after the dot-com crash of the early 2000s (Thibodeau 2008), the last few years have seen a steady growth in the number of news stories about AI appearing in popular media, as discovered by AAAI's automated AI in the News weekly news bot (Eckroth et al. 2012). Figure 1 shows this trend. Com-

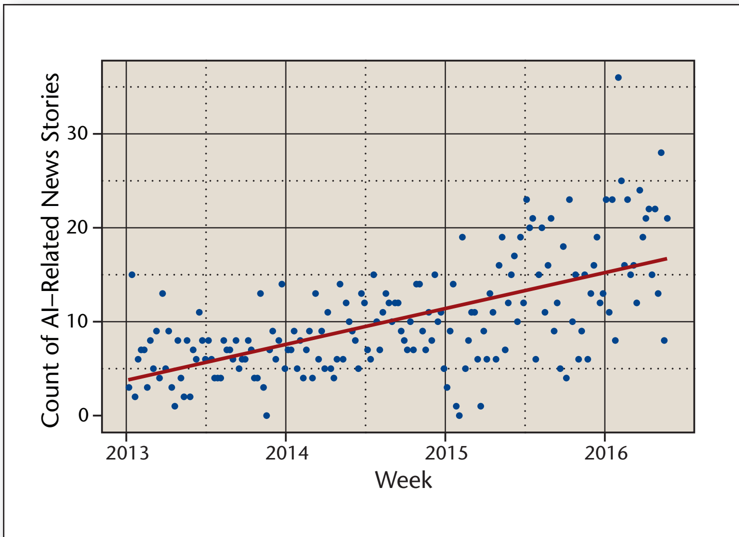

Figure 1. Count of News Stories Found by AI in the News for Each Week.

The line visualizes a linear regression.

puter science undergraduate enrollments have exhibited a similar trend, and as of 2014, more Ph.D. graduates are employed in the field of AI, across academe and industry, than any other subfield of computer science (Zweben and Bizot 2015).

We believe there are several factors contributing to the growth since the first IAAI conference in 1989.

Moore's Law

One of the most important changes is the growth in hardware performance. To illustrate, we will consider a deployed system from the first IAAI conference. Clancy, Gerald, and Arnold (1989) developed an expert system that "assisted attorneys and paralegals in the closing process for commercial real estate mortgage loans." Their system was required to work on IBM PCs with Intel 80286 processors and 640 KB memory. They wrote that the PC's limited memory posed a "critical technical consideration" and they programmed their system to swap subsets of the knowledge base in and out of memory during normal operation. Their solution, and more arcane memory management techniques, are likely familiar to AI system builders who were active in the early days.

Today, consumer hardware is 2500 times faster (from 1.5 million instructions per second, or MIPS, on an 80286 compared to 3783 MIPS on an Intel i7-3770K, quad core).[1] It is now common to find more powerful servers, with more than 10 cores, which means that we can take advantage of a speed-up of 10,000 — and growing.

In addition, consumer hardware contains 25,000 times as much memory (640 KB to 16 GB), and 50,000 times the disk capacity (40 MB to 2 TB). (See also Preshing [2012].) This explosive growth of computing power can be attributed to Moore's law, which summarizes Gordon Moore's observation that the number of transistors in integrated circuits doubles approximately every two years. The diversity of integrated circuits has also grown, resulting in general-purpose GPUs (with multiteraflop performance; that is, trillions of floating-point operations per second) that have helped usher in the era of practical machine learning.

The Internet

The global impact of the Internet on science and society in general cannot be overstated. One of the more interesting effects of the Internet, and the web in particular, on AI systems was noted in Halevy, Norvig, and Pereira's (2009) article The Unreasonable Effectiveness of Data. They note that the web enables easy acquisition of massive amounts of data from billions of web pages, provided by billions of users.

Halevy and colleagues further argue that sophisticated knowledge representation and reasoning systems may be unnecessary, even detrimental when a massive corpus such as the web is available. For example, in the case of the semantic web, they suggest that writing an ontology, adding metadata markup for web pages, and building a complex reasoning system is likely to be more expensive and error prone than simply querying the vast, unstructured corpus with shallow parsing and straightforward statistical analysis. The long tail of real-world concepts defeats any effort to develop a grand model of everyday reasoning, but the long tail is well represented in massive data sets such as the web.

Open Source Software

Frustrated with a trend toward proprietary development practices at the Massachusetts Institute of Technology (MIT), Richard Stallman started the GNU Project in 1983 to create a free and open source UNIX-like operating system. The idea spread and has been harnessed by various groups, resulting in an abundance of high-quality open source software. The internet played a large role in the distribution and development of open source software. In particular, development of the Linux operating system, which was built with GNU project tools, grew rapidly in the 1990s due to the availability of newsgroups, email, and file sharing. As of November 2015, 99 percent of the 500 most powerful supercomputers in the world run the open source Linux operating system. Most software development environments in use today are open source (Oracle's JVM, Microsoft's .Net, C/C++ compilers, Python, and others), and many open source libraries and toolkits are available for AI-specific tasks, a sampling of which are shown in table 2.

Machine Learning

Because available hardware did not allow large-scale

TensorFlow (Google)	Machine learning toolkit
OpenCV (itseez)	Computer vision library
Sphinx (CMU)	Speech recognition toolkit
Drools (Red Hat)	Rule-driven expert system shell, planning engine
GATE (University of Sheffield)	Natural language processing toolkit
Robot Operating System (Open Source Robotics Foundation)	Platform for integrating various algorithms and libraries related to robotics

Table 2. Sample Open Source AI libraries and Toolkits.

numeric computation, early AI systems relied on heuristics encoded symbolically, even in data-heavy tasks such as computer vision. Though significant progress continues to be made in symbolic reasoning, it is clear that the power available to process vast amounts of data — even at high data rates — has enabled practical deployment of machine-learning techniques and resulted in a wide diversity of successful applications from speech recognition and face recognition to self-driving cars. Additionally, it is interesting to note that machine learning seems to dominate the popular perspective of AI today, just as it was considered by early AI researchers to be an essential component of intelligence (Minsky 1961).

We find evidence of this in Google Trends data, shown in figure 2. Interest in AI appeared to wane after the dot-com crash and hit a low around 2009. The search term *computer science* follows a similar trend. Recently, interest has renewed and appears to be supported by machine learning and recent work in deep learning.

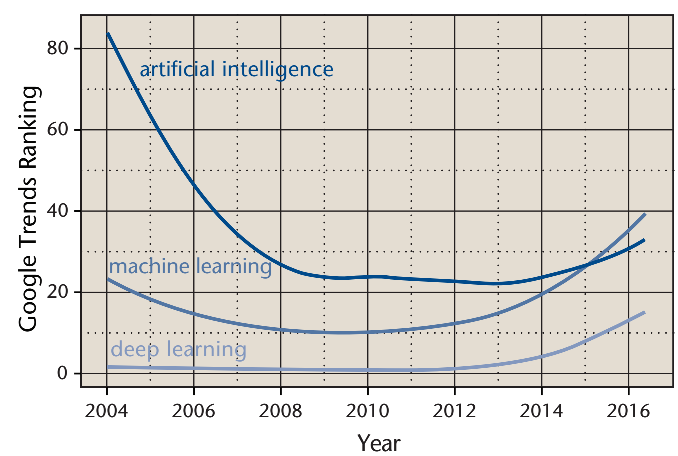

Figure 2. Google Trends Rankings for Various Search Terms.

The y-axis represents smoothed relative interest.

Reduced Business Risk

These points add up to another change. Because of the greater computing power and more readily available data sets and software, today there is less need to build massive technology platforms. Hence, it is cheaper to build AI systems. More effort can be spent on solving specific business problems, thereby reducing the risk associated with artificial intelligence.

Compared to 1989, today it is orders of magnitude easier to integrate AI systems into a company's overall IT portfolio. The reasons include: modern AI systems utilize standard hardware and software (in many cases); they integrate more easily into existing architectures; the iterative development process pioneered in AI projects has become common across IT; and, the success of high-profile AI systems such as Watson and Siri means that most people know that AI can work in the real world. (The authors thank Neil Jacobstein for this insight.)

Distributions and Trends from the IAAI Conferences

At the outset, IAAI included only applications that had been deployed; that is, for which there was experience based on actual use, and for which payoff could be estimated. In 1997, an emerging applications track was added to bridge the gap between AI research and AI application development. The goal was to support information sharing among researchers and system builders: researchers could see which techniques proved fruitful in deployed applications, and builders could learn of emerging techniques that had yet to be proven in the field but showed promise.

An analysis of the topics covered by IAAI articles is shown in figures 3 and 4. This analysis is provided by i2k Connect (i2kconnect.com), whose goal is help organizations find, filter, and analyze unstructured

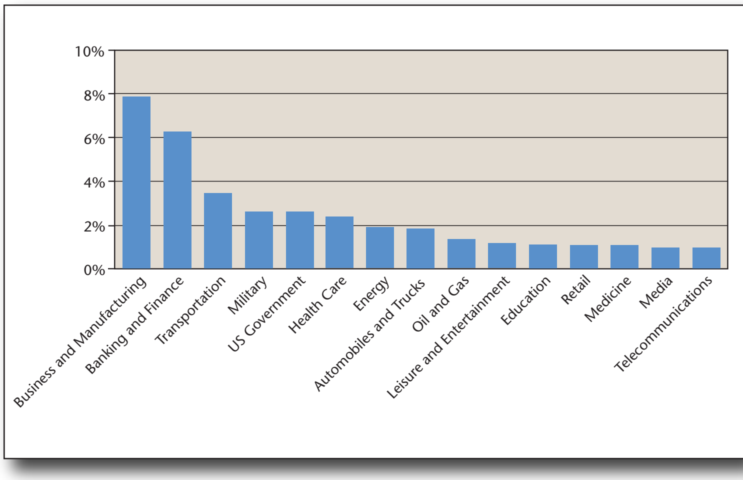

Figure 3. Top 15 Industry Topics in IAAI Articles, 1989–2016.

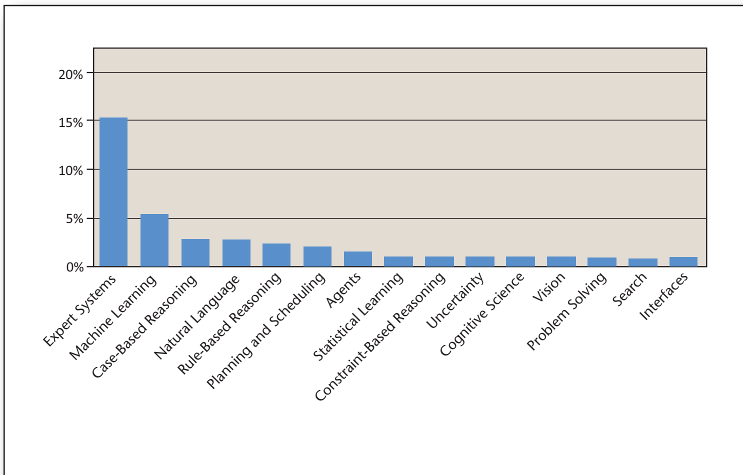

Figure 4. Top 15 Technology Topics in IAAI Articles, 1989–2016.

data by transformation into structured data. The platform automatically tags documents with accurate and consistent metadata, guided and enriched by subject matter expertise. The figures show data from deployed applications only — 316 of them. The figures include only 15 of a long tail of more than 100 industry and technology topics that have been covered in IAAI. Figure 3 also excludes information technology applications, that is, AI applied to our own business, which would otherwise be number one.

Another way of analyzing IAAI topics over the years is shown in figure 5. The figure shows that the technology mix has evolved. Expert systems clearly dominated the early days of IAAI. Machine learning is notably absent early on. Over time, however, the mixture has become more diverse, with no topic clearly dominating in recent conferences. We note that it is not the case that expert systems died. Rather, after a few years, they became more standard practice than innovation. Fewer papers were published about novel applications of expert systems. They disappeared into the fabric, now applied everywhere, from the high-end emulation of rare human experts, to the embedding and application of rule books and procedure manuals. However, we will likely see more hybrid machine-learning technologies that can automatically update their reasoning engines as the application data change over time.

Some technologies have not been represented much at IAAI, like speech understanding and robots. They do appear, just not in the top 15. In the recent 2016 deployed papers track, four technologies are applied: spatial reasoning, crowdsourcing, machine learning, and ontologies. We also note that there have been two papers on deep learning, one in 2015 and one in 2016, neither documenting a deployed application. Some of this may be due to self-selection in that our data are limited to IAAI conferences, which may not accurately reflect how often these technologies are utilized in the overall application world.

In our final analysis of IAAI articles, figure 6 shows a quick overview of the top concepts mentioned over the years. The analysis was done with a modified form of the C-value/NC-value method (Frantzi, Ananiadou, and Mima 2000), which extracts significant concept names found in text, as opposed to just the most frequently used phrases. Note that there may be some temporal bias in this analysis due to the data set reflecting the past decades of IAAI papers, versus trends in the most recent papers.

High-Impact AI Applications

Many of the past IAAI program chairs and cochairs and AAAI Fellows kindly responded to a request for their views on what have been the high-impact applications, including some that opened up a new area, presented at IAAI conferences over the years.

Because we have selected high-impact applications and it takes time to establish whether an application has had high impact, some of the examples may look a bit dated. Note, however, that in several cases, a recent update has been presented at IAAI.

A few of the applications that were singled out by several respondents as being high impact are summarized in the following.

1983: Process Diagnosis System (PDS)

The Process Diagnosis System (Fox, Lowenfeld, and Kleinosky 1983) started out as an expert system shell. It has been in active use and continuous development since 1985. 1985! Though the origin of PDS predates IAAI, it serves as an early example of deployed AI. It started with a presentation by Mark Fox at Westinghouse. Over the 30-year period, Westinghouse sold the business to Siemens, where it is now at the heart of their Power Diagnostics Center that performs centralized rule-based monitoring of over 1200 gas tur-

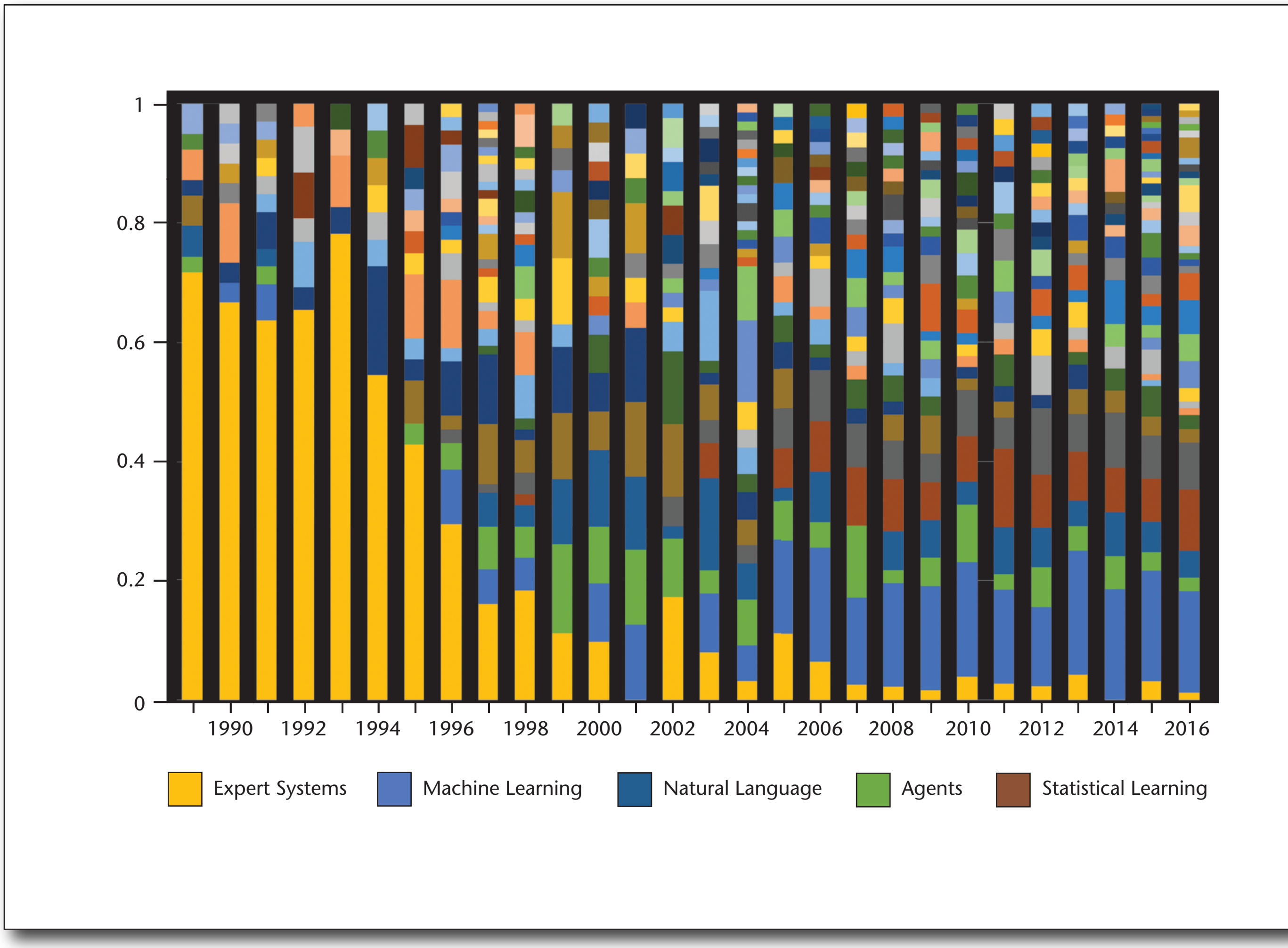

Figure 5. Mix of Technologies Deployed and Emerging in IAAI Articles, 1989–2016.

The dominant technologies include expert systems, machine learning, agents, natural language, and statistical learning, and are included in the figure legend.

bines, steam turbines, and generators. Ed Thompson and Ben Bassford celebrated the 30th anniversary of the system with the IAAI community when they presented an update at the 2015 conference in Austin (Thompson et al. 2015). Their paper summarizes the many changes that have been incorporated into the system over its lifetime, to deal with change in requirements, the customer business organization, and underlying computer technologies.

1989: Authorizer's Assistant

A knowledge-based credit-authorization system for American Express, the Authorizer's Assistant (Dzierzanowski et al. 1989) was the forerunner of now standard credit card transaction analysis. It created a capability that we all take for granted today — and complain about every time we are called to verify a charge — until at least we ourselves are the victims of fraud.

Expansion, improvement, and testing were planned from the start to ensure consistency as the knowledge base changed as well as ensure general system performance. The team found that consistency, audit tracking, and evaluation were key to acceptance and return on investment (ROI). They observed, "the [Authorizer's Assistant] proved to be better than all but the most expert credit card authorizers ... and that translated directly into huge ROI." The system's internal expert system incorporated 890 rules and ran on rack-mounted Symbolics Lisp machines connected to an IBM mainframe.

Phil Klahr generously provided these retrospective insights.

1989: Applications of Artificial Intelligence to Space Shuttle Mission Control

This NASA application originated in the Mission Control Center for STS-26 as a rule-based real-time

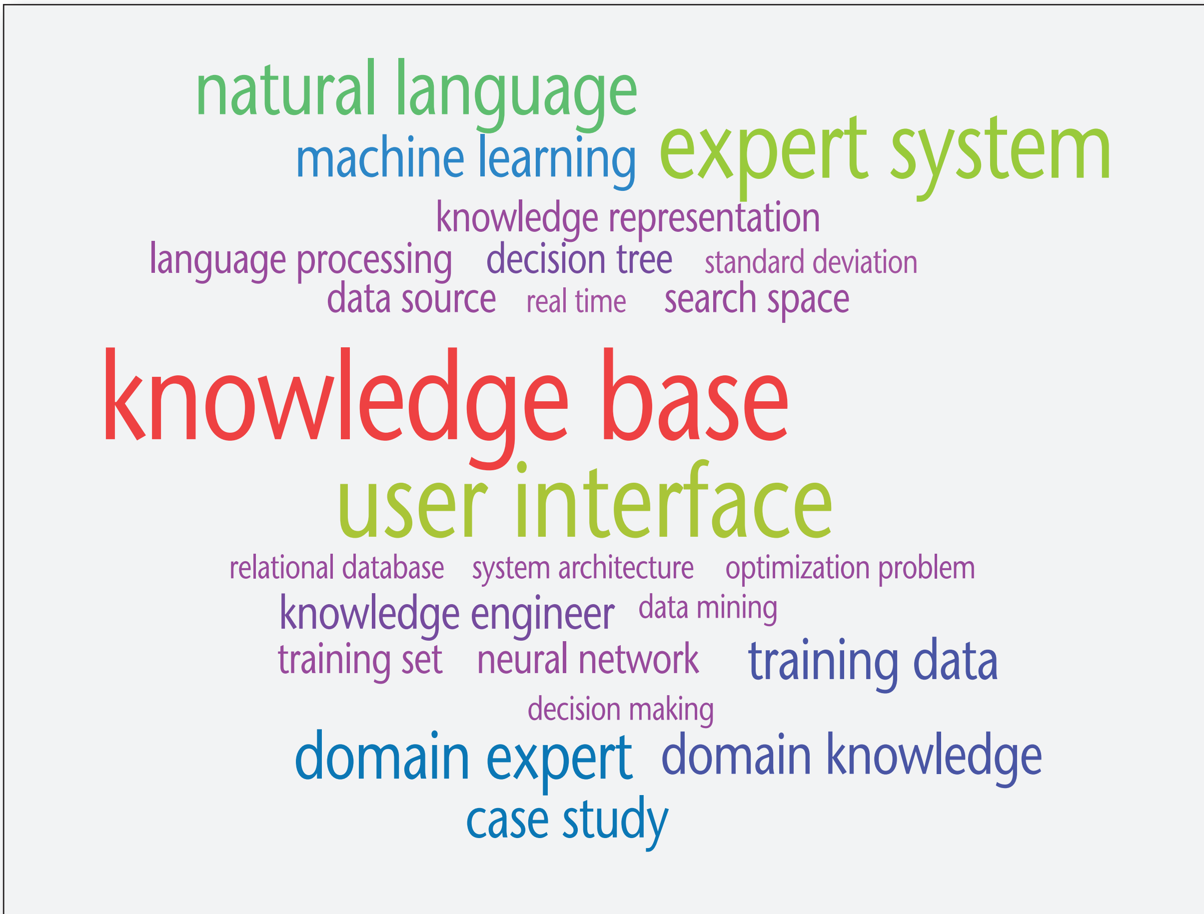

Figure 6. Top Concepts Mentioned in IAAI Papers, 1989–2016.

Integrated Communications Officer (INCO) expert system (Muratore et al. 1989). The system monitored space shuttle telemetry data and advised flight controllers on fault detection and diagnosis. It provided fault identification and diagnosis before the traditional INCO console could update the parameters of the faulty unit.

The system made use of the C Language Integrated Production System (CLIPS) Expert System Shell, now available as open source software.[2] It was also the first of more than 30 NASA applications reported to date at IAAI and, as is well known, AI systems later flew in space — and navigated autonomously on the moon and Mars.

The next two applications are intelligent assistants that played in center ring from first deployment. Both have been presented twice at IAAI — covering initial deployment and a 10-year update.

1989 and 1999: Ford Motor Company Direct Labor Management System

The Direct Labor Management System is integrated into Ford's Global Study Process Allocation System (GSPAS) (O'Brien et al. 1989, Rychtyckyj 1999). Its purpose is the automatic generation of work instructions for vehicle assembly, with associated times. It does so by analyzing high-level structured English descriptions. The system is also able to make accurate estimates of direct versus indirect labor time and to plan for mix/volume changes and line balancing. It has become an integral part of Ford's assembly process planning business.

The natural language component in this application was one of the few from the early days. The system was implemented on the NIKL/KL-ONE (Woods and Schmolze 1992) knowledge representation model — one of the first such applications.

Over the years, the system has undergone several knowledge base upgrades and ports to different platforms to keep the system viable and up to date through various organizational and business practice changes. It was later called the Global Study Process Allocation System.

1995: The FinCEN Artificial Intelligence System and 1999: The NASD Regulation Advanced-Detection System (ADS)

Created to identify potential money laundering from reports of large cash transactions, the FinCEN Artificial Intelligence System (FAIS) used link diagrams to support detection of money laundering (Senator et al. 1995). ADS: NASD Regulation Advanced-Detection System (Kirkland et al. 1999) used temporal sequences to support detection of securities fraud. Their different domains of use dictated different knowledge representations.

FAIS links and evaluates reports of large cash transactions. To give an idea of the money involved, fincen.gov reported suspicious transactions totaling approximately $28 billion in October 2015. The FAIS key idea is "connecting the dots" — thus link diagrams, now commonplace in social network analysis, was an appropriate choice. The appropriate representation choice in FAIS enabled a reporting app based on the original detection system. This was an unanticipated bonus.

ADS monitors trades and quotations in the Nasdaq Stock Market, to identify suspicious patterns and practices. In this application, temporal sequences are key — not so much links as in FAIS — so a representation that supports them was a good choice.

2005 and 2014: Engineering Works Scheduling for Hong Kong's Rail Network

The Hong Kong rail network moves 5 million passengers a day through the city's rapid transit subway, airport express, and commuter rail lines. The AI application streamlines the planning, scheduling, and rescheduling process and provides automatic detection of potential conflicts as work requests are entered; verification that no conflicts exist in any approved work schedules before execution; generation and optimization of weekly operational schedules; automatic update to repair schedules after changes; and generation of quarterly schedules for planning (Chun et al. 2005, Chun and Suen 2014).

To be successful, the system must coordinate with the staff members who carry out the scheduled work. To this end, the developers found that the system must be able to explain the schedules it creates. As a result, they veered away from the original genetic algorithms approach toward heuristic search. A recent report about this system appeared in *New Scientist* (Hodson 2014).

1994 and 2004: Plastics Color Formulation Tool

Since 1994, GE Plastics (later SABIC) has employed a case-based reasoning (CBR) tool that determines color formulas that match requested colors (Cheetham 2004). FormTool has saved millions of dollars in productivity and material (that is, colorant) costs. It is the basis for the online color-selection service called ColorXpress Services.

Determining the colorants and loading levels that can be added so the plastic matches a given color is a difficult problem for multiple reasons. For example, there is no accurate method to predict the color produced when a set of colorants is added to plastic. Unlike paint, where light primarily reflects off the surface, in plastics a significant percentage of light penetrates the surface and reacts with the internal structure to produce a color that depends on both the internal structure and the lighting conditions (natural sunlight versus fluorescent lighting).

The AI system used case-based reasoning to replace programs that used prohibitively expensive exhaustive search to determine the colorant-loading proportions for a color formula that matches a customer's desired color.

1995: Scheduling of Port of Singapore Authority

This expert system (Weng et al. 1995) is responsible for assisting with planning and management of all operations of the Port of Singapore Authority. With hundreds of vessels calling at Singapore every day, a fast and efficient allocation of marine resources to assist the vessels in navigating in the port waters is essential. Manual planning using pen and paper was erroneous, uncoordinated, and slow in coping with the rapid increase in the vessel traffic. Included in the purview of the application is scheduling the movement of vessels through channels to terminals, deploying pilots to tugs and launches, allocating berths and anchorages to ships, and planning stowage of containers.

To generate accurate, executable deployment schedules, the automated scheduler requires real-time feedback from the resources on their job status, any estimated delays, and end times of their jobs. This is achieved by integrating the system with the port's mobile radio data terminal system.

2006: Expressive Commerce and Its Application to Sourcing

This application has produced one of the largest ROI figures of any system thus far reported at IAAI. Originally CombineNet, later renamed SciQuest, it improves procurement decisions for spend categories that are typically beyond the capabilities of traditional eSourcing software. Even in the early days of 2006, it had already handled $35 billion in auctions and delivered $4.4 billion in savings to customers through lower sourcing costs (Sandholm 2007).

The challenge in developing an expressive commerce system is handling the combinatorial explosion of possible allocations of businesses to suppliers. Their key development is a sophisticated tree search algorithm. Much has been written about this algorithm (refer to Sandholm [2007] for a list of articles), though some of its details are kept proprietary.

2014: CiteSeerX

CiteSeerX (Wu et al. 2014) is a database and search engine for more than 4 million research articles from various disciplines. Starting in 1997 as CiteSeer, the service was the first to extract and index citations from documents automatically. Today, it is also capable of extracting metadata from individual paragraphs and sentences as well as tables and figures. The metadata and the original documents are made freely available for researchers who work on advanced information-retrieval algorithms.

The CiteSeerX service is accessed 2 million times per day and an average of 10 articles are downloaded per second. The size of the document database (after deduplication) has grown significantly over the years, from 500,000 in 2008 (when CiteSeerX debuted) to nearly 3 million in 2013. Today, between 50,000 and 100,000 PDFs are analyzed per day.

CiteSeerX's implementation makes use of several AI components, including document classification, duplicate detection, metadata extraction, author name disambiguation, and search indexing. The researchers' 25 years of experience is documented (Wu et al. 2014) and serves as a showcase of the variety of AI techniques available, for example, rule engines, neural networks, probabilistic graphical models, and the importance of choosing the Appropriate technique.

Lessons Learned

As the story of developing, implementing, and upgrading applications has unfolded, so has the conventional wisdom about building successful AI applications.

The Power Is in the Knowledge ... But Manual Knowledge Acquisition Is Hard

The first lesson learned by builders of early AI systems is that "the power is in the knowledge." By 1989, thanks to the pioneering efforts of Ed Feigenbaum, Bruce Buchanan, and many others, we understood that domain-specific knowledge (chemistry, medicine, and others) and task-specific knowledge (turbine maintenance, plant scheduling, and others) are more important for high performance and accurate reasoning than general problem-solving approaches.

But manual knowledge acquisition is hard and takes a long time. In the past, we called this the "knowledge acquisition bottleneck." Furthermore, ongoing knowledge base maintenance and curation are essential. Knowledge is perishable — everything changes over the lifetime of an application: the domain evolves, new use cases arise, new experts arrive with different knowledge, new data sets become available, the technology advances, and so on. If system builders are engaged in manual knowledge acquisition, then they will also need an army of people to keep the knowledge base up to date. Therefore, they need a lot of revenue (hence, a lot of users) to support that effort. Due to the difficulty of knowledge acquisition and maintenance, many systems fell by the wayside, even if they were excellent at one time. For example, the field of medicine is too large and changes too rapidly for manual knowledge acquisition (Myers, Pople, and Miller 1982).

Today, knowledge comes in a variety of forms. Early knowledge systems made use of symbolic rules that encoded experts' knowledge because such rules are compact, they are representationally adequate for many tasks and domains, and low-powered machines (by today's standards) are sufficient to perform the required inference procedures. However, rule-based expert systems are not effective for "big data" problems such as visual object recognition (for example, faces) and speech recognition. Progress in neural networks and deep learning, in particular, probabilistic graphical models and other machine-learning (ML) techniques, has greatly expanded the reach of AI systems. Yet, systems that use machine learning still use knowledge. Rather than expert-defined rules, ML systems make use of knowledge in several forms: training data including procedures for their acquisition and preprocessing, feature selection, model selection, and various parameters found by experimentation. It has been said, "there is no such thing as a free lunch," and in AI and ML, there is no such thing as free knowledge. There is no escape from the need to maintain and curate knowledge and data, even if some aspects are automated by machine learning.

Knowledge Representation Matters

The structure of knowledge in the system has a large impact on the system's reasoning capabilities and performance. The pioneers taught us that selecting the appropriate representation has a big impact, for five reasons: adequacy, efficiency, flexibility, maintainability, and explainability.

Adequacy

As John McCarthy and his colleagues stated, a system cannot reason about what it cannot represent (McCarthy 1960, 1981). Davis, Schrobe, and Szolovits (1993) referred to a knowledge representation as a "surrogate" for real-world entities. An adequate surrogate has a "correspondence" with real-world entities and these correspondences have high "fidelity," that is, they closely match the relevant characteristics of the real-world entities.

Efficiency

Every programmer knows that representations, or data structures, can have an impact on efficiency. For example, linked lists do not support quick random access while arrays do not support quick addition or deletion of elements. Similar trade-offs characterize knowledge representations. In general, the more simplistic the representation, the more efficient the reasoning algorithms. For example, reasoning over propositional logical is often quite efficient, while few tools exist that are capable of efficiently and reliably reasoning over first-order logic with types (Sutcliffe and Pelletier 2016). Likewise, nearest neighbor classification procedures require virtually no training while neural networks typically require significant training. During inference, however, a naïve nearest neighbor algorithm (for example, linear lookup) will likely be significantly slower than a neural network.

Flexibility

A good knowledge representation will support growth in the knowledge base that was not anticipated during initial knowledge acquisition. Additionally, long-running systems must be capable of evolving over time as the customer changes and the context in which the system was initially deployed migrates to other contexts and use cases. A good knowledge representation will be able to represent new knowledge concepts and adapt existing ones without significant updates to the representation or reasoning algorithms. The American Authorizer's Assistant, the FinCEN system, and the ADS systems are all good examples of this lesson in action.

In summarizing 10 years of work on Mycin, Buchanan and Shortliffe (1984) attributed the success of the program to flexibility — in the rule-based representation that allowed rapid modification and in the reasoning that allowed reaching reasonable conclusions with imperfect or missing information.

Maintainability

It is helpful if the knowledge representation can be understood and modified by subject matter experts, who may not be (and typically are not) experts in computer programming and knowledge representation. Ideally, software engineers will not need to be called in whenever the knowledge base needs an update. As John McCarthy noted, declarative representations are more learnable and maintainable than procedural ones (McCarthy 1960). The reason is that declarative representations are better separated from internal conventions of the reasoning algorithms, thus allowing subject matter experts to focus on the knowledge being represented.

Today, it is particularly important for subject matter experts to be able to interpret and modify the results of machine-learning systems that are driven primarily by raw data and empirical validation. They may, for example, suggest that there are critical data sets missing that would improve the analysis, and avoid simply handing the problem over to data scientists.

Explainability

In organizations, the AI system encodes and represents the decision criteria of the management. Thus, when the AI system suggests a decision, it should be able to explain that decision to the user so that the user (and the management) can "own it and be able to defend it" in terms of the organization's decision criteria. Explanation may be unnecessary if the algorithm makes money, for example, Wall Street trading. But in many other contexts, without an explanation system, organizational and user acceptance of AI applications is more challenging. This has been understood since SHRLDU in the Blocks World (Winograd 1972), and Mycin, in the knowledge-intensive world of medicine (Buchanan and Shortliffe 1984).

Explainability is more problematic in the case of noninterpretable models such as neural networks, in which it is not at all clear exactly what knowledge has been stored as a result of training.

Lately, there has been some discussion in the press about "algorithmic accountability" (Diakopoulos 2013, Lohr 2015), and several companies are pursuing explanation as a differentiator, for example, Watson Paths and the Narrative Science extension for the Qlik visual analytics tool (Hammond 2015).

Separate the Knowledge Base and the Inference Engine

As a corollary of the maintainability of declarative representations, the pioneers also taught us that it is a good idea to separate the knowledge base and the inference engine because a separate knowledge base is easier to change, update, debug, and explain. Recognizing the importance of separation, from a knowledge representation and a knowledge delivery perspective, many people have devoted their time to the development of expert system shells (for example, M1, S1, ART and CLIPS), knowledge representation languages (for example, KL-ONE and OWL), ontology editors (for example, Protégé), and general-purpose machine-learning models.

Successful Applications Incorporate a Variety of Techniques

Successful AI applications incorporate a wide range of techniques, strategies, and knowledge, embodying rules, objects, ontologies, statistics, and signal processing to name a few. Self-driving cars are an obvious example. Their capabilities include modeling, simulation, sensing, motion planning, object recognition, obstacle avoidance, machine learning, error recovery, and so on. The learnings have been reported multiple times at AAAI (Montemerlo et al. 2006, Thrun 2006) and IAAI (Urmson et al. 2009).

Modern text-analytics systems also illustrate the point. For example, the i2k Connect platform uses a variety of knowledge and AI techniques to perform document reading and enrichment. It uses ontologies to represent domain-specific knowledge about, for example, the oil and gas industry, the field of artificial intelligence, and topics related to supply-chain management and health care. Document text and metadata are extracted using machine-learning methods. Visual and language rules are used to extract the document's title and summary. Documents are then analyzed with a variety of rules in order to identify the domain-specific topics that the document is about. Multiple technologies from AI and elsewhere are needed for this processing pipeline.

A combination of various kinds of knowledge and techniques should be expected in any large-scale AI application that is required to integrate multiple sources and types of information. The architecture of such an AI application should make such integration feasible by, for example, separating different processing tasks into distinct modules and supporting a common interface for communication among the components. The Robot Operating System[3] is a paradigmatic example of such an architecture. Different robots may have vastly different components and purposes, yet ROS offers high-level abstractions that enable various sensors, actuators, and algorithms to communicate using a common language.

AI Applications Must Integrate into Existing Work Flows

Perhaps the most important lesson learned by AI system builders is that success depends on integrating into existing workflows — the human context of actual use. It is rare to replace an existing work flow completely. Thus, the application must play nicely with the other tools that people use. Put another way, ease of use delivered by the human interface is the "license to operate." Unless designers get that part right, people may not ever see the AI power under the hood; they will have already walked away.

As AI systems began to function well enough that they were able to play in the center ring, so to speak, risk mitigation, project management, and budgetary control became more important. The systems were no longer in a "research" or "proof of concept" phase. In other words, standard IT rules — and consumer mobile app acceptance rules — apply. Many AI practitioners have made these points in the context of AI applications in particular. But the rules are valid for all applications of information technology.

In the early days, we talked as if AI systems had a big box of AI — the important stuff — and a small box of all that other messy IT stuff. We quickly learned that in real-world systems, it was mostly the other way around. The AI was a piece of the puzzle, and sometimes not a very big piece.

Consider the Dipmeter Advisor (Smith and Baker 1983), started at Schlumberger in the early 1980s and based on the knowledge of the legendary oil finder, Al Gilreath, shown in figure 7. The Dipmeter Advisor demonstrated the challenges of infrastructure: getting the data from the field systems was a bigger problem than originally anticipated; and the challenges of technology transfer: nontraditional hardware (D-Machines) and software (Interlisp-D) became major stumbling blocks, though without these technologies Schlumberger would have had no system at all.

The amount of effort that had to be devoted to the non-AI components was dominant. The user interface accounted for almost half the code. The rule engine and knowledge base accounted for 30 percent. Of course, lines of code do not necessarily tell the whole story, but the numbers are consistent with the development effort expended. Much of the coding effort went into the interactive graphics system, not the AI. For some clients, interactive graphics was the most important element.

Security and privacy have become increasingly crucial over time, and the application's performance characteristics in the deployed setting must meet industry or consumer expectations.

Additionally, change management is unavoidable (Hiatt 2006). But the amount of change management required is inversely proportional to the power of the new technology. It is also directly proportional to the amount of change in existing work flows required to adopt it.

Convincing people to make substantial changes to their existing work flows to take advantage of a new technology that isn't much better than the old technology requires a great deal of change management effort. On the other hand, convincing people to make small changes to their existing work flows to take advantage of new technology that is an order of magnitude better than the old technology requires only modest change management effort.

As Mehmet Goker put it in a private communication to the authors: "Applications with a small and flexible core that solve a real-world problem have the biggest impact and are the easiest to put into the workplace."

To summarize, in any large organization, standard IT rules apply and the AI application should fit into the broader IT infrastructure to ensure successful adoption. Management, end user, and IT support and participation are essential. Budget approval will be challenging without business unit management support, deployment into a company's existing infrastructure is not possible without support from the IT organization, and adoption is unlikely without continuous end-user participation in system development.

In the real world of applications, our experience also suggests that the dichotomy suggested by Markoff (2015) between artificial intelligence and intelligence augmentation or amplification does not exist. They are two ends of a spectrum that meet in most applications. The successful systems enable people to do what people do best and use computers to do what computers do best.

A Way Around the Knowledge Acquisition Bottleneck

Machine learning offers a way around the knowledge acquisition bottleneck ... but success depends on human insight folded into the methods, like the choice of features.

One thing has not changed over the history of IAAI. It is still very hard to build, curate, and maintain large knowledge bases by hand. The manual knowledge-acquisition bottleneck is still firmly in place.

Aside: This is a special case of a larger point. Manual information governance is not sustainable. Very few

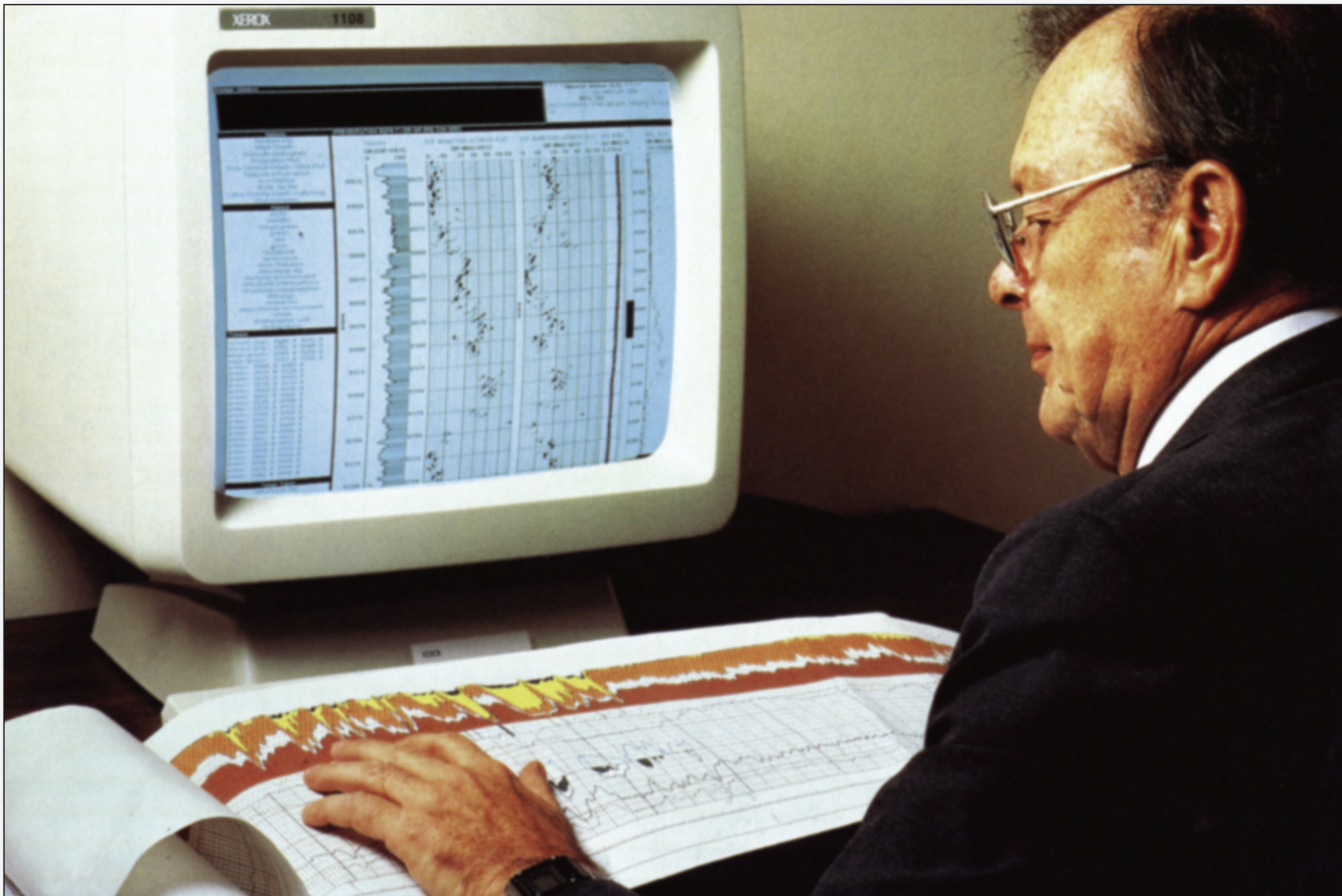

Figure 7. The Dipmeter Advisor System.

> humans have the passion and consistency to tag and manage their own unstructured data ... look at your own hard drive or your organization's file shares if you doubt it. This is one of the main reasons why so much unstructured corporate data is "lost in the cloud." It may be there, but you are likely to struggle to find it if you didn't write it yourself. More than half of employees in companies surveyed worldwide express deep dissatisfaction with the findability of corporate information.[4] In contrast to Internet content, today it is rare to see search engine optimization applied to intranet content.

But now armed with billions of crowdsourced examples from the web, we have learned that data-driven, statistical methods are "unreasonably effective" in several domains. The statistics bring the ability to deal with noise and to cover problems where humans either have difficulty explaining how they do it, or where they don't do it very well in the first place.

The bottom line is that machine learning is a way around the knowledge-acquisition bottleneck in a surprisingly broad number of domains, but two caveats are worth considering:

> Howie Shrobe made an observation that rings true. "... when you look closer at successful statistical approaches, a lot of the success is in the choice of features to attend to or other similar ways of conveying human insight to the technique ..." (private communication). Indeed, mitigating this problem is a focus of some research on deep learning algorithms — to learn feature representations from unlabeled data.[5]

> There is a very long tail on the types of problems encountered in the world. Developers will not have millions of examples for all of them. In those cases, some kind of reasoning is essential; for example, from basic principles captured via case-based reasoning or encoded in a rule-based system.

Apps Can Be Built with Components That Reason from Different Starting Points.

In the early days of expert systems running on machines with relatively little processing power and memory, the standard starting point for delivering domain and task-specific knowledge can be characterized by labels like *slow, cognition, search, top-down, model-driven.*

Today, armed with the compute power, data, and machine-learning algorithms now available to us, we are much better equipped to build apps that reason from a starting point characterized by labels like *fast, recognition, look-up, bottom-up, data-driven*.

For example, *Fast versus Slow*. The focus of Daniel Kahneman's 2011 book is a dichotomy between these two modes of thought: "fast, instinctive, and emotional" and "slow, more deliberative, and more logical" (Kahneman 2011).

Alternatively, Herb Simon put it this way: "The situation has provided a cue; this cue has given the expert access to information stored in memory, and the information provides the answer. Intuition is nothing more and nothing less than recognition." (Simon 1992). *Fast* corresponds to *recognition*. *Slow* corresponds to *cognition* or *search*. In this regard, compare the recognition approach of the human chess master to the search approach of Deep Blue (Campbell, Hoane, and Hsu 1999). (Because of this, a typical grandmaster does six orders of magnitude less search per move than Deep Blue did.)

Another example, well known to American football fans, is that of the Manning brothers, Peyton and Eli. It has been widely reported that their father Archie started the boys learning football and quarterbacking at the earliest possible age. This maximized the time they had to store millions of the small chunks of recognition knowledge, later buttressed by countless hours spent studying game film.

Rod Brooks championed what he called a new approach to artificial intelligence and robot design — which can be called "bottom-up" — as an alternative to the "top-down" model-driven approach of the pioneers (Brooks 1991).

Today, some authors seem to see a conflict between "data-driven" (new think) systems and "model-driven" (old think) systems as if the "good" applications today are all data driven and work well, in contrast with the "bad" model-driven applications of the old days that didn't work well.

Many AI apps have combined reasoning from opposite starting points, going way back to the early days. The Hearsay II speech-understanding system combined top-down and bottom-up processing (Erman et al. 1980). Mycin used backward and forward rule chaining (Buchanan and Shortliffe 1984). And the Dipmeter Advisor was both data driven, converting raw signals to patterns, and model driven, using rules to classify stratigraphic and tectonic features from the patterns (Smith and Baker 1983). Overall accuracy depended on the contributions of all the components — data driven and model driven.

We also don't accept the criticism that the early AI community was too focused on model-driven approaches when it should have been focused on data-driven approaches. We believe the pioneers were doing the best they could with the machines and data available to them. They were forced into cognitive approaches in some cases (for example, vision) because they had to do something to finesse the need for orders of magnitude more processing power, storage, and sensors than were available to them in the day.

The good news these days is that all the components are substantially more powerful, thanks to the computing and data revolutions. We are not restricted to either a "fast" or a "slow" starting point. We can have both.

That said, it is important for developers to give due consideration to the new possibilities offered by the substantial increases in processor speed and memory available today — and to not implicitly be stuck in the "slow" thinking mind set of the early days. Going forward, there is the possibility of storing massively larger knowledge bases that are composed of small chunks of very specific domain and task knowledge, retrieved by fast recognition processes (more of what Simon was referring to).

Thus, a knowledge base for a domain would have powerful rules (as in the past, thousands of them) plus these small chunks of very specific experiential knowledge (millions of them). With modern sensors, the small chunks may be very easy to capture. Certainly, there will be things missing that might have been implied by rules (that is, not everything possible is actually observed and remembered as a chunk). But overall, knowledge acquisition will have become far easier to do and cheaper. These "hybrid" knowledge base architectures will dominate in applications. This also seems like a fruitful avenue for reconsidering older models of human cognition. (The authors thank Ed Feigenbaum for this observation.)

Checklist for Tomorrow's Application Builders

Our examination of nearly 30 IAAI conferences, our personal experiences, and stories related to us by colleagues and friends, lead to the checklist in table 3. We briefly explain each entry in the following.

As will be apparent to experienced application developers, much of this advice mirrors general software engineering best practice. But some of the points are even more important for AI systems. We invite your feedback and your own lessons learned.

Select Problems with a Solid Business Case

Successful IT applications in general start with a focus on the business case and the customer — not the technology. This is particularly true for AI applications. In the early days of AI applications, the mind share of the developers tended more heavily to the technology (the knowledge-representation methods and the reasoning machinery) than it did to the customer need. In retrospect, this was to be expected. The early implementers were almost always AI researchers, infringing on an IT community that was by and large skeptical of the hype and the baggage that came along with the technology — nonstandard hardware and software, methods that were not understood by the

■ Select problems with a solid business case.
■ Minimize changes required in existing work flows.
■ Identify domain- or task-specific knowledge and data for the problem.
■ Select appropriate knowledge representations and data sources.
■ Develop knowledge and data acquisition and maintenance plans.
■ Select appropriate reasoning/learning strategies.
■ Develop a set of test cases and performance metrics.
■ Add safeguards and opt-out capabilities.
■ Test with real data from users or operating environment.

Table 3. Checklist for Builders of AI Applications.

community, the need to bring in outside experts.

Over the years, AI application developers have made a major mind shift. We have learned the hard way that success starts with solving problems important to the customer.

One caveat: Although public interest in AI is on the rise, do not add an AI component to an application just for the sake of it. AI can introduce complexity, and systems should always only be as complicated as is necessary to model the domain and task. Again, focus on customer over technology.

Minimize Changes Required in Existing Work Flows

Think about the integration of AI with other tools and parts of the larger system. It is rare to completely replace an existing work flow. Thus, it is prudent to build new systems so that they can slot into the approaches already used by the customers as much as possible. Few new AI systems solve stand-alone problems that require no user interaction. Most are used as "intelligent assistants" and the amount of change management required to succeed in adoption is directly proportional to the magnitude of the changes required in existing work flows. Ease of use is the "license to operate."

Identify Domain- or Task-Specific Knowledge and Data for the Problem

The history of successful AI applications shows that "the power is in the knowledge," both expert-provided knowledge and knowledge extracted out of data with the appropriate preprocessing, feature selection, learning techniques, and parameter tuning. Devote up front the effort needed to acquire enough of the knowledge and data so that you will better understand how to design the rest of the system to best match the domain and task.

Select Appropriate Knowledge Representations and Data Sources

Depending on the nature of the domain- and task-specific knowledge, choose a knowledge representation that most closely models the world while still supporting efficient reasoning strategies. Prefer declarative knowledge since it is easier to understand, explain and change than procedural knowledge. For machine-learning approaches, select high-quality data sources (for example, data with expert-verified ground truth) when feasible or develop a strategy for learning and reasoning with noisy data sources.

Develop Knowledge and Data Acquisition and Maintenance Plans

Consider knowledge/data acquisition and maintenance to be an ongoing process. Make the process iterative: repeatedly evaluate if the knowledge and data are appropriate for the reasoning/learning strategies and domain/task and refine accordingly.

Select Appropriate Reasoning and Learning Strategies

Most large AI systems will require various kinds of reasoning and learning strategies for various subproblems. Design a system architecture that supports decoupling of these disparate components so that refinements in one component will not require drastic changes in other components.

Depending on the constraints dictated by the domain and task, select an approach and components that are data driven or model driven, or use a combination.

Develop a Set of Test Cases and Performance Metrics

Due to the complexity of most AI systems, testing and performance evaluation are critical.

The word *performance* encompasses a variety of concerns, including run-time speed and use of resources plus adequacy of the knowledge and reasoning components. Run-time speed and use of resources are standard computational concerns that must be addressed in any system that is to be deployed and scaled. They are not specific to AI applications — but adequacy of the knowledge and reasoning components is specific to AI applications.

Sometimes the desired behavior of the system is clear, as was the case when IBM was developing Watson to win the *Jeopardy!* game show against human contestants (Ferrucci et al. 2010). Keep track of the performance of every revision and consider a policy (as IBM did with Watson) that rejects any revisions that do not push performance closer to the goal.

When the goal criteria are not as clear, make extensive use of regression tests to ensure that solved cases are never broken in the future. Sometimes, even regression tests are too precise as multiple different outcomes may be equally good. A good technique in these situations is to build machinery to automatically identify any changes in the system's output after each code or knowledge base revision. Knowing what changed after an update is a valuable first step in identifying if development is on the right track.

Add Safeguards and Opt-Out Capabilities

AI has been known, on occasion, to produce odd and unpredictable results due to complex reasoning systems, large data sets, and large knowledge bases. Hence, special care should be taken to verify data produced by AI subsystems. In addition, there is a premium on testing carefully for machine-learning systems that do not have transparent reasoning processes.

This advice should be heeded more diligently for builders of AI applications that make use of human input and applications that are responsible for making decisions for users. For that matter, such applications should provide an opt-out capability that lets the user complete an action without AI assistance. An AI system is even stronger when it can explain its decisions and can help users make sense of the AI's assistance and better decide if they prefer to continue making use of it.

Test with Real Data from Users or Operating Environment

At i2k Connect, we have learned that there is a long tail of the kinds of documents humans (and computers) produce and that may be fed into our document enrichment service. During the early development effort, we focused on straightforward cases such as research articles in PDF form and Microsoft Word documents made up mostly of text. However, real data from real users can be drastically different and highly variable. For example, we learned that our system did not properly handle large text files produced by computer software (such as log files or data dumps), and needed extra logic to examine each file before deciding what kinds of processing would be appropriate. In other examples, roboticists know that robots must be tested in the real world and not just simulations, and developers of personal assistants, chatbots, search engines, and other tools know that humans are an unpredictable source of a wide range of inputs.

Conclusion

For AI to benefit humankind it must be deployed; for successful deployment, good AI ideas must be integrated into the human context of actual use and into the IT context of organizations. In this article, we have tried to summarize what has been learned about building, maintaining, and extending AI applications. We have boiled it down into a simple checklist for the developers of today and tomorrow.

Going forward, we can expect the landscape of AI applications to continue to diversify and expand. The revolutions will continue all around us, in computers and data, as well as sensing.

So, it follows that apps will continue to get more powerful, more knowledgeable, and cover a broader array of domains and tasks.

It also follows that apps will be increasingly data driven, guided by human knowledge. And they will have a lot more data available, as the Internet of Things takes off.

Finally, intelligent assistants will be even more proficient at improving quality of life. The partnership between human and machine is going to be stronger and closer. How will this improve quality of life? Jobs tend to be more satisfying when we humans are able to focus on the *real* work we set out to do, not distracted by the low-level clutter that most people are forced to deal with today, because computers aren't powerful enough, or because no attempt has yet been made to automate the jobs people don't want to do. Intelligent assistants will deal with the clutter of low-level tasks, or tasks that require extended concentration, consistency, scale, and so on.

As an example, we see big opportunities with unstructured data. It will no longer be lost in the cloud — whether the corporate cloud or the Internet cloud. We will have the tools to find it and unlock its connections. We will also have the tools to extract the essential information from the cluttered real-time data streams that overwhelm us today.

As the developers of today and tomorrow address the new opportunities, the history of IAAI conferences offers lessons in how to build successful deployed AI applications. We have attempted to distill these lessons to increase the chances of future success. In these concluding remarks, we have just a few final bits of advice.

It is prudent for AI researchers to pay attention to what is being learned through engineering practice — deployed applications — as was hoped for from the beginning of IAAI. And it is prudent for practitioners to take advantage of opportunities to learn from research, as was hoped for by colocating the AAAI and IAAI conferences, and by adding the Emerging Applications track to the IAAI conference in 1997.

It is also wise to pay attention to what is happening in the rest of the computing, data, and sensing world. Factors external to AI are likely to have the largest impact on what matters, or what is possible, or where opportunities lie. The biggest impact on how we are able to build applications today has come from revolutions that were not of our own making. Watch for signals from the periphery.

And finally, to quote Neil Jacobstein, "AI expands the range of the possible." So keep doing it!

Acknowledgments

In preparing this article, we have drawn

heavily from AI applications observations made by others. We encourage readers to look back at Feigenbaum, McCorduck, and Nii (1988); Feigenbaum (1993); Shrobe (1996); Shrobe (2000;) and Jacobstein (2007).

Thanks to the following people for generously sharing their time and their ideas on high-impact applications and for their observations on what (we thought) we knew then and what (we think) we know now: Bruce Buchanan, Andy Chun, Edward Feigenbaum, Markus Fromherz, Ashok Goel, Mehmet Goker, Haym Hirsh, Neil Jacobstein, Phil Klahr, Alain Rappaport, Nestor Rychtyckyj, Eric Schoen, Ted Senator, Howard Shrobe, David Stracuzzi, Ramasamy Uthurusamy, and Peter Yeh.

Thanks especially to Bruce Buchanan and Ed Feigenbaum for years of guidance … and patience.

Thanks also to Jon Glick, AITopics pioneer, and to Raj Reddy, godfather of IAAI.

Finally, thanks to our families for their foundational contributions.

This article is based on the IAAI-16 Robert S. Engelmore Memorial Lecture given by the first author at AAAI/IAAI 2016 in honor of Bob Engelmore's extraordinary service to AAAI and his contributions to applied AI.[6]

Notes

1. See Dennis Bode, The Ivy Bridge Test: Intel Core i7-3770K and all i5 models (Hardware LUXX), available at www.hardwareluxx.com/index.php/reviews/hardware/cpu/21569-ivy-bridge-test-intel-core-i7-3770k-and-all-i5-models.html?start=13.

2. For more information about CLIPS see www.clipsrules.net/?q=AboutCLIPS.

3. The Enterprise Search and Findability Survey 2014, by Carl Björnfors and Mattias Ellison, is available at www2.findwise.com/findabilitysurvey2014.

4. Robot Operating System (ROS) is available at www.ros.org.

5. deeplearning.stanford.edu/wiki/index.php/UFLDL_Tutorial.

6. Available at www.reidgsmith.com/2016-02-15_Engelmore_Lecture.pdf

References

Brooks, R. A. 1991. Intelligence Without Representation. *Artificial Intelligence* 47(1–3): 139–159.

Buchanan, B. G., and Shortliffe, E. H. 1984. *Rule Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project* (The Addison-Wesley Series in Artificial Intelligence). Boston, MA: Addison-Wesley Longman Publishing Co., Inc.

Campbell, M.; Hoane, J.; and Hsu, F. 1999. Search Control Methods in Deep Blue. In *Search Techniques for Problem Solving Under Uncertainty and Incomplete Information: Papers from the 1999 AAAI Spring Symposium.* Technical Report SS-99-07. Menlo Park, CA: AAAI Press.

Cheetham, W. E. 2004. Tenth Anniversary of the Plastics Color Formulation Tool. In *Proceedings of the Sixteenth Conference on Innovative Applications of Artificial Intelligence,* 770–776. Menlo Park, CA: AAAI Press.

Cheyer, A. 2014. Siri: Back to the Future. Invited presentation at the Twenty-Sixth IAAI Conference / Twenty-Eighth AAAI Conference, Montreal, Quebec, Canada, July.

Chun, A. H. W., and Suen, T. Y. Y. 2014. Engineering Works Scheduling for Hong Kong's Rail Network. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence and the Twenty-Sixth Annual Conference on Innovative Applications of Artificial Intelligence,* 2890–2897. Palo Alto, CA: AAAI Press.

Chun, A. H. W.; Yeung, D. W. M.; Lam, G. P. S.; Lai, D.; Keefe, R.; Lam, J.; and Chan, H. 2005. Scheduling Engineering Works for the MTR Corporation in Hong Kong. In *Proceedings of Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference,* 1467–1474. Menlo Park, CA: AAAI Press.

Clancy, P.; Gerald, H.; and Arnold, S. 1989. An Expert System for Legal Consultation. In *Proceedings of the First Annual Conference on Innovative Applications of Artificial Intelligence,* 125–135. Menlo Park, CA: AAAI Press.

Davis, R.; Shrobe, H.; and Szolovits, P. 1993. What Is a Knowledge Representation? *AI Magazine* 14(1): 17–33.

Diakopoulos, N. 2013. Rage Against the Algorithms: How Can We Know the Biases of a Piece of Software? By Reverse Engineering It, of Course. *The Atlantic*. 3 October. Available at www.theatlantic.com/technology/archive/2013/10/rage-against-the-algorithms/280255

Dzierzanowski, J. M.; Chrisman, K. R.; MacKinnon, G. J.; and Klahr, P. 1989. The Authorizer's Assistant: A Knowledge-Based Credit Authorization System for American Express. In *Proceedings of the First Conference on Innovative Applications of AI,* 168–172. Menlo Park, CA: AAAI Press.

Eckroth, J.; Dong, L.; Smith, R. G.; and Buchanan, B. G. 2012. NewsFinder: Automating an Artificial Intelligence News Service. *AI Magazine* 33(2): 43-54.

Erman, L.; Hayes-Roth, F.; Lesser, V. R.; and Reddy, D. R. 1980. The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty. *ACM Computing Surveys* 12(2): 213–253. doi.org/10.1145/356810.356816

Feigenbaum, E.; McCorduck, P.; and Nii, P. 1988. *The Rise of the Expert Company.* New York: Times Books, Random House.

Feigenbaum, E. A. 1993. Tiger in a Cage: The Applications of Knowledge-Based Systems. Joint invited presentation at the 11th National Conference on Artificial Intelligence and the Fifth Conference on Innovative Applications of Artificial Intelligence, Washington, DC, July.

Ferrucci, D.; Brown, E.; Chu-Carroll, J.; Fan, J.; Gondek, D.; Kalyanpur, A. A.; Lally, A.; Murdock, J. W.; Nyberg, E.; Prager, J.; Schlaefer, N.; and Welty, C. 2010. Building Watson: An Overview of the DeepQA Project. *AI Magazine* 31(3): 59–79.

Fox, M. S.; Lowenfeld, S.; and Kleinosky, P. 1983. Techniques for Sensor-Based Diagnosis. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence,* 159–163. Los Altos, CA: William Kaufmann, Inc.

Frantzi, K.; Ananiadou, S.; and Mima, H. 2000. Automatic Recognition of Multi-Word Terms: The C-Value/NC-Value Method. *International Journal on Digital Libraries* 3(2): 115–130.

Hammond, K. 2015. How Artificial Intelligence Explains Analytics. *Computerworld,* June 19. Available at www.computerworld.com/article/2936142/emerging-technology/ how-artificial-intelligence-explains-analytics.html.

Halevy, A.; Norvig, P.; and Pereira, F. 2009. The Unreasonable Effectiveness of Data. *Intelligent Systems* 24(2): 8–12.

Hiatt, J. 2006. *ADKAR: A Model for Change in Business, Government and Our Community.* Fort Collins, CO: Prosci.

Hodson, H. 2014. The AI Boss That Deploys Hong Kong's Subway Engineers. *Technology News,* 2 July. Available at www.newscientist.com/article/mg22329764.000-the-ai-boss-that-deploys-hong-kongs-subway-engineers.

Jacobstein, N. 2007. Innovative Applications of Early Stage AI. Presented at the 2007 Singularity Summit, San Francisco, CA, September. Available at foresight.org/cms/ events_library/267.

Kahneman, D. 2011. *Thinking, Fast and Slow.* New York: Farrar, Straus and Giroux.

Kirkland, J. D.; Senator, T. E.; Hayden, J. J.; Dybala, T.; Goldberg, H. G.; and Shyr, P. 1999. The NASD Regulation Advanced-Detection System (ADS). *AI Magazine* 20(1): 55–67.

Lohr, S. 2015. If Algorithms Know All, How Much Should Humans Help? *New York Times,* April 6. Available at www.nytimes.com/2015/04/07/upshot/if-algorithms-know-all-how-much-should-humans-help.html.

Markoff, J. 2015. *Machines of Loving Grace: The Quest for Common Ground Between Humans and Robots*. New York: Ecco, an imprint of HarperCollins Publishers.

McCarthy, J. 1981. Epistemological Problems of Artificial Intelligence. In *Readings in Artificial Intelligence,* ed. B. Webber and N. Nilsson, 459–465. Los Altos, CA: Tioga.

McCarthy, J. 1960. Programs with Common Sense. In *Proceedings of the Symposium on the Mechanization of Thought Processes,* National Physical Laboratory 1: 77–84. Teddington, UK: Her Majesty's Stationery Office. Available at aitopics.org/publication/programs-common-sense.

Miller, R. A.; Pople, Jr, H. E.; and Myers, J. D. 1982. Internist-I, An Experimental Computer-Based Diagnostic Consultant for General Internal Medicine. *New England Journal of Medicine* 307(8): 468–476.

Minsky, M. 1961. Steps Toward Artificial Intelligence. Proceedings of the IRE, 49(1), 8–30. Reprinted in *Computers & Thought,* ed. E. A. Feigenbaum and J. Feldman, 406–450. Cambridge, MA: The MIT Press.

Montemerlo, M.; Thrun, S.; Dahlkamp, H.; Stavens, D.; and Strohband, S. 2006. Winning the DARPA Grand Challenge with an AI Robot. In *Proceedings of the National Conference on Artificial Intelligence* 21(1): 982–987. Menlo Park, CA: AAAI Press.

Muratore, J. F.; Heindel, T. A.; Murphy, T. B.; Rasmussen, A. N.; and McFarland, R. Z. 1989. Applications of Artificial Intelligence to Space Shuttle Mission Control. In *Proceedings of the Conference on Innovative Applications of Artificial Intelligence,* 15–22. Menlo Park, CA: AAAI Press.

Myers, J. D.; Pople, H. E.; and Miller, R. A. 1982. CADUCEUS: A Computerized Diagnostic Consultation System in Internal Medicine. In *Proceedings, The Sixth Annual Symposium on Computer Applications in Medical Care,* ed. B. I. Blum, 44–47. Los Alamitos, CA: IEEE Computer Society.

O'Brien, J.; Brice, H.; Hatfield, S.; Johnson, W.; Woodhead, R. 1989. The Ford Motor Company Direct Labor Management System. In *Innovative Applications of Artificial Intelligence,* 331–346. Cambridge, MA: AAAI Press / The MIT Press.

Preshing, J. 2012. A Look Back at Single-Threaded CPU Performance. Preshing on Programming Blog, February 8. Available at preshing.com/20120208/a-look-back-at-single-threaded-cpu-performance/.

Rychtyckyj, N. 1999. DLMS: Ten Years of AI for Vehicle Assembly Process Planning. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence,* 821–828. Menlo Park, CA: AAAI Press.

Sandholm, T. 2007. Expressive Commerce and Its Application to Sourcing: How We Conducted $35 Billion of Generalized Combinatorial Auctions. *AI Magazine* 28(3): 45–58.

Schorr, H., and Rappaport, A. T., eds. 1989. *Innovative Applications of Artificial Intelligence*. Cambridge, MA: AAAI Press / The MIT Press.

Senator, T. E.; Goldberg, H. G.; Wooton, J.; Cottini, M. A.; Khan, A. U.; Klinger, C. D.; and Wong, R. W. 1995. The FinCEN Artificial Intelligence System: Identifying Potential Money Laundering from Reports of Large Cash Transactions. In *Proceedings of the 7th Annual Conference on Innovative Applications of Artificial Intelligence,* 156–170. Menlo Park, CA: AAAI Press

Shrobe, H. E. 1996. The Innovative Applications of Artificial Intelligence Conference: Past and Future. *AI Magazine* 17(4): 15–20.

Shrobe, H. E. 2000. What Does the Future Hold? *AI Magazine* 21(4): 41–57.

Simon, H. 1992. What Is an "Explanation" of Behavior? *Psychological Science* 3(3): 150–161.

Smith, R. G., and Baker, J. D. 1983. The Dipmeter Advisor System: A Case Study in Commercial Expert System Development. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*-Volume 1, 122–129. Los Altos, CA: William Kaufmann, Inc.

Sutcliffe, G., and Pelletier, F. J. 2016. Hoping for the Truth — A Survey of the TPTP Logics. In *Proceedings of the Twenty-Ninth International Flairs Conference,* 110–115. Palo Alto, CA: AAAI Press.

Thibodeau, P. 2008. Computer Science Graduating Class of 2007 Smallest This Decade. *Computerworld* Online. March 5. Available at www.computerworld.com/article/2537436/it-careers/computer-science-graduating-class-of-2007-smallest-this-decade.html.

Thompson, E. D.; Frolich, E.; Bellows, J. C.; Bassford, B. E.; Skiko, E. J.; and Fox, M. S. 2015. Process Diagnosis System (PDS) — A 30 Year History. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence and the 27th Annual Conference on Innovative Applications of Artificial Intelligence,* 3928–3933. Palo Alto, CA: AAAI Press.

Thrun, S. 2006. A Personal Account of the Development of Stanley, the Robot That Won the DARPA Grand Challenge. *AI Magazine* 27(4): 69–82.

Urmson, C.; Baker, C.; Dolan, J.; Rybski, P.; Salesky, B.; Whittaker, W.; and Darms, M. 2009. Autonomous Driving in Traffic: Boss and the Urban Challenge. *AI Magazine* 30(2): 17–28.

Weng, G. K.; Seng, G. K.; Whye, L. C.; Hwa, T. B.; and Kiang, T. E. 1995. Scheduling of Port of Singapore Authority: A Total Approach. In *Proceedings of the Seventh Conference on Innovative Applications of Artificial Intelligence,* 62–69. Menlo Park, CA: AAAI Press.

Winograd, T. 1972. Understanding Natural Language. *Cognitive Psychology* 3(1): 1–191.

Woods, W. A., and Schmolze, J. G. 1992. The KL-ONE Family. *Computers & Mathematics with Applications* 23(2–5): 133–177.

Wu, J.; Williams, K.; Chen, H.-H.; Khabsa, M.; Caragea, C.; Ororbia, A.; Jordan, D.; and Giles, C. L. 2014. CiteSeerX: AI in a Digital Library Search Engine. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence and the Twenty-Sixth Annual Conference on Innovative Applications of Artificial Intelligence,* 2930–2937. Palo Alto, CA: AAAI Press

Zilis, S. 2015. The Current State of Machine Intelligence 2.0. *O'Reilly Media on Our Radar,* December 10. Available at www.oreilly.com/ideas/the-current-state-of-machine-intelligence-2-0.

Zweben, S., and Bizot, B. 2015. Taulbee Survey. *Computing Research News* 27(5): 2–51.

Reid G. Smith is cofounder and CEO of i2k Connect. Formerly, he was vice president of research and knowledge management at Schlumberger, enterprise content management director at Marathon Oil, and senior vice president at Medstory. He holds a Ph.D. in electrical engineering from Stanford University and is a Fellow of AAAI. He is coeditor of AITopics.

Joshua Eckroth is an assistant professor at Stetson University and chief architect of i2k Connect. His research interests lie in abductive reasoning and belief revision as well as computer science pedagogy. Eckroth holds a Ph.D. in computer science from the Ohio State University. He is coeditor of AITopics.