The Yale Law Journal
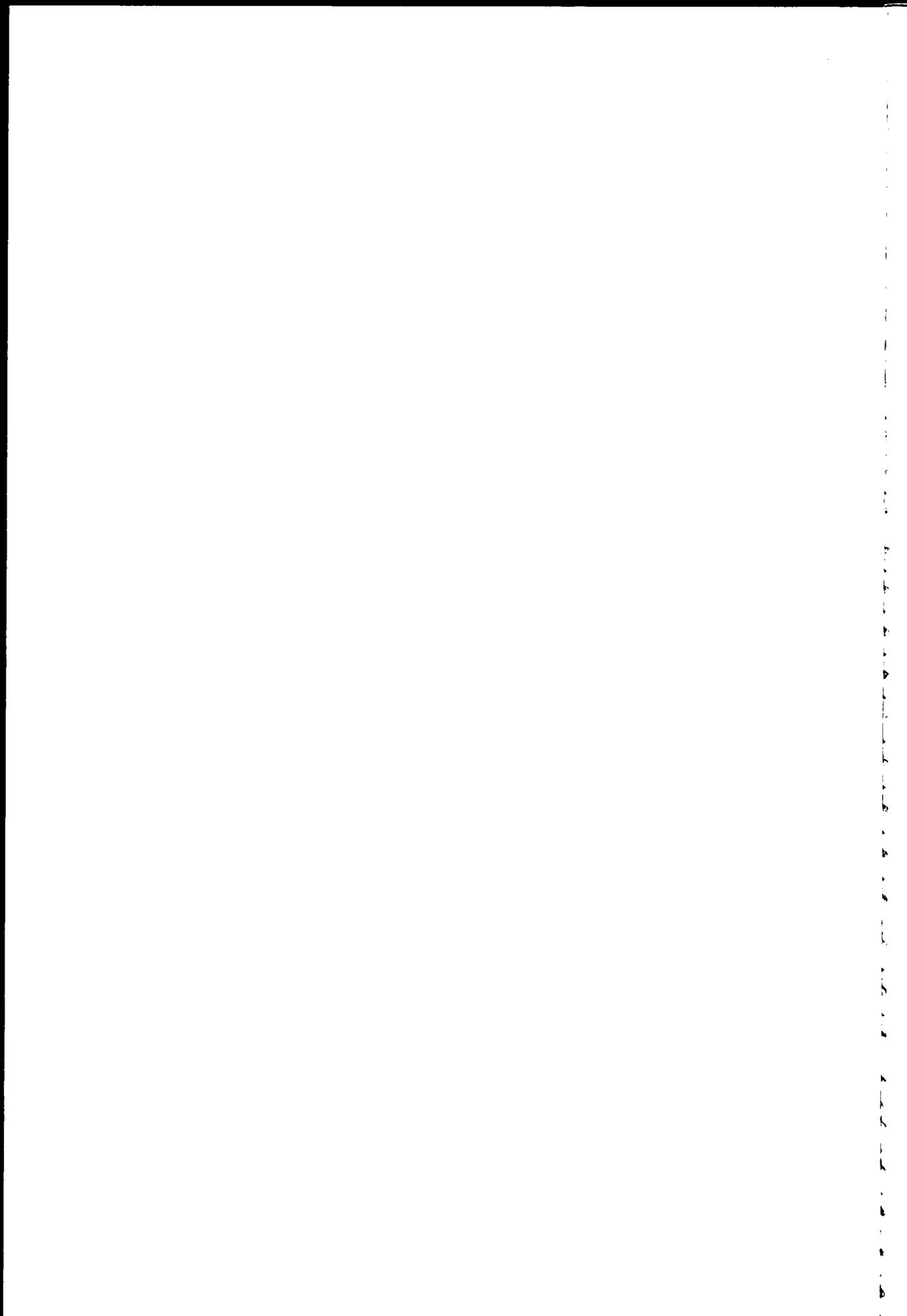
Artificial Intelligence and
Law: Stepping Stones to a
Model of Legal Reasoning

by

Edwina L. Rissland

99 YALE L.J. 1957

Artificial Intelligence and Law: Stepping Stones to a Model of Legal Reasoning

Edwina L. Rissland†

This Comment discusses developments in the twenty-year-old interdisciplinary field of Artificial Intelligence (AI) and law. This field is important for both AI and law because it is directed at improving our understanding and modeling of legal reasoning.

The AI and law projects discussed here are landmarks in this field.[1] A unifying theme of the projects is the goal to understand and model legal argument, a keystone of an overarching goal to understand and model legal reasoning. These goals require that we know first how to represent several types of knowledge, such as cases, rules, and arguments; second, how to reason with them, such as to manipulate precedents, to apply and make inferences with rules, and to tailor arguments to facts; and third, how to use them ultimately in a computer program that can perform tasks in legal reasoning and argumentation, such as analogizing favorable cases and distinguishing contrary ones, anticipating parries in adversarial argument, and creating artful hypotheticals.

The projects constitute a coherent set of studies about key topics in AI and law: (1) reasoning with rules; (2) handling open-textured legal concepts; (3) reasoning with cases and hypotheticals; (4) integrating reasoning with rules and reasoning with cases; and (5) representing legal knowledge. The limitations of techniques for handling the first topic are addressed by the second, and the third topic addresses critical issues not covered at all by work on the first. The fourth integrates work on the first three, and the fifth provides underpinnings needed for all of them. The projects are some of the major accomplishments in AI and law, especially

---

1. For one of the earliest discussions of AI and law, see Buchanan & Headrick, *Some Speculation About Artificial Intelligence and Legal Reasoning*, 23 STAN. L. REV. 40 (1970). *See also* Gardner, *Law Applications*, in THE ENCYCLOPEDIA OF ARTIFICIAL INTELLIGENCE 456 (S. Shapiro ed. 1989) (providing general overview of AI and law). For a representative collection of current work, see THE SECOND INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND LAW: PROCEEDINGS OF THE CONFERENCE (1989) [hereinafter ICAIL-89].

those seeking to understand and model legal argument, and they exemplify and illustrate the progress, concerns, and style of its research.

I. BACKGROUND ON ARTIFICIAL INTELLIGENCE

A. *What is Artificial Intelligence?*

AI is the study of cognitive processes using the conceptual frameworks and tools of computer science.[2] As a distinct subfield of computer science, AI had its beginnings in the mid-fifties.[3] In 1968 Marvin Minsky, one of the founders of AI, said it well: AI is "the science of making machines do things that would require intelligence if done by man."[4] Thus, all manner of intelligent behavior is in the realm of AI, including playing chess,[5] solving calculus problems,[6] making mathematical discoveries,[7] understanding short stories,[8] learning new concepts,[9] interpreting visual scenes,[10] diagnosing diseases,[11] and reasoning by analogy.[12] Any discussion of AI must

---

2. For additional background, see E. CHARNIAK & D. McDERMOTT, INTRODUCTION TO ARTIFICIAL INTELLIGENCE (1985); 1-4 HANDBOOK OF ARTIFICIAL INTELLIGENCE (A. Barr, P. Cohen & E. Feigenbaum eds. 1981-89); G. LUGER & W. STUBBLEFIELD, ARTIFICIAL INTELLIGENCE AND THE DESIGN OF EXPERT SYSTEMS (1989); THE ENCYCLOPEDIA OF ARTIFICIAL INTELLIGENCE, *supra* note 1; P. WINSTON, ARTIFICIAL INTELLIGENCE (2d ed. 1984).

3. A summer conference held at Dartmouth College in 1956 is perhaps the most convenient landmark to denote its beginning. *See* P. McCORDUCK, MACHINES WHO THINK 93-114 (1979) (describing history surrounding the Dartmouth Conference and its lasting effects in AI community).

4. SEMANTIC INFORMATION PROCESSING v (M. Minsky ed. 1968).

5. *See, e.g.*, Newborn & Kopec, *Results of The Nineteenth ACM North American Computer Chess Championship*, 32 COMM. ACM 1225 (1989) (describing several games).

6. Research on symbolic mathematics began with James Slagle's SAINT, a program which solved 52 out of 54 problems selected from MIT freshman calculus final examinations. *See* Slagle, *A Heuristic Program That Solves Symbolic Integration Problems in Freshman Calculus*, in COMPUTERS AND THOUGHT 191 (E. Feigenbaum & J. Feldman eds. 1963). The next project was Joel Moses's program SIN. *See* Moses, *Symbolic Integration: The Stormy Decade*, 14 COMM. ACM 548 (1971). Further research on symbolic mathematics has led to the development of systems such as MIT's MACSYMA, "a large, interactive computer system designed to assist mathematicians, scientists, and engineers in solving mathematical problems." 2 HANDBOOK OF ARTIFICIAL INTELLIGENCE, *supra* note 2, at 143. Commercial versions of such programs are now available for personal computers. Wayner, *Symbolic Math on the Mac*, BYTE, Jan. 1989, at 239.

7. A program that modeled the process of discovery in mathematics was Doug Lenat's AM. Using a knowledge base containing fundamental concepts and rules of thumb, AM was able to "discover" such mathematical concepts as "divisor" and "prime." R. DAVIS & D. LENAT, KNOWLEDGE-BASED SYSTEMS IN ARTIFICIAL INTELLIGENCE 1-225 (1982); *AM*, in 3 HANDBOOK OF ARTIFICIAL INTELLIGENCE, *supra* note 2, at 438.

8. *See, e.g.*, R. SCHANK & R. ABELSON, SCRIPTS, PLANS, GOALS, AND UNDERSTANDING: AN INQUIRY INTO HUMAN KNOWLEDGE STRUCTURES (1977) (discussing theory for story understanding); INSIDE COMPUTER UNDERSTANDING: FIVE PROGRAMS PLUS MINIATURES (R. Schank & C. Riesbeck eds. 1981). For general collections of work on natural language processing, see READINGS IN NATURAL LANGUAGE PROCESSING (B. Grosz, K. Jones & B. Webber eds. 1986); STRATEGIES FOR NATURAL LANGUAGE PROCESSING (W. Lehnert & M. Ringle eds. 1982).

9. *See, e.g.*, 1-2 MACHINE LEARNING: AN ARTIFICIAL INTELLIGENCE APPROACH (R. Michalski, J. Carbonell & T. Mitchell eds. 1983, 1986).

10. *See, e.g.*, READINGS IN COMPUTER VISION: ISSUES, PROBLEMS, PRINCIPLES, AND PARADIGMS (M. Fischler & O. Firschein eds. 1987).

11. *See, e.g.*, RULE-BASED EXPERT SYSTEMS, THE MYCIN EXPERIMENTS OF THE STANFORD HEURISTIC PROGRAMMING PROJECT (B. Buchanan & E. Shortliffe eds. 1984) [hereinafter Buchanan & Shortliffe] (thorough discussion of MYCIN diagnostic system for bacterial infections).

12. *See, e.g.*, Gentner, *Structure-Mapping: A Theoretical Framework for Analogy*, 7 COGNITIVE

note that tasks involving "common sense" reasoning or perception, such as language understanding, are by far the most difficult for AI.[13] More technical tasks, like solving calculus problems or playing chess, are usually much easier. That is because the latter can be framed in well-defined terms[14] and come from totally black-and-white domains, while the former cannot and do not. What distinguishes the AI approach from other studies of cognition and knowledge, such as psychology or philosophy, is its insistence on grounding the analysis in computational terms—preferably in a successfully running computer program that embodies the analysis.[15]

AI is pursued for at least two reasons: to understand the workings of human intelligence[16] and to create useful computer programs and computers that can perform intelligently. Most workers in the field of AI pursue these goals simultaneously. For instance, in the course of designing a computer program for commercial purposes like making credit card approval decisions, the program designer needs to examine how experienced people make such decisions, since they are usually the best, and often the only, source of information about how the job is done.[17] Likewise, in engaging in AI for the sake of understanding or modeling cognition, it is far

---

Sci. 155 (1983). Reasoning by analogy was the subject of one of the earliest programs in AI. Evans, *A Program for the Solution of a Class of Geometric-Analogy Intelligence-Test Questions*, in SEMANTIC INFORMATION PROCESSING *supra* note 4, at 271.

13. *See, e.g.*, E. CHARNIAK, TOWARD A MODEL OF CHILDREN'S STORY COMPREHENSION 271-74 (MIT AI Lab Technical Report No. 266, 1972) (even understanding "simple" children's stories can be quite messy).

14. For instance, the rules of chess completely define the game. Furthermore, in the case of games, solutions can be found using high-powered, specialized "search" techniques. This is how highly successful chess programs, such as Deep Thought, work. Deep Thought can examine 720,000 chess positions per second. *See* Newborn, & Kopec, *supra* note 5, at 1225. However, it is quite a different matter whether such programs work in the same way human chess experts do and whether they can shed light upon human thought processes. *See* Chase & Simon, *The Mind's Eye in Chess*, in VISUAL INFORMATION PROCESSING 215, 278 (W. Chase ed. 1973).

15. *See* D. DENNETT, BRAINSTORMS: PHILOSOPHICAL ESSAYS ON MIND AND PSYCHOLOGY 112 (1978); MIND DESIGN (J. Haugeland ed. 1981).

16. For two interesting theories of human cognition, see M. MINSKY, THE SOCIETY OF MIND (1986); A. NEWELL, UNIFIED THEORY OF COGNITION (forthcoming 1990).

17. Even if psychological validity is not usually paramount, it is often helpful. The MYCIN Project illustrates this point. The goal of the MYCIN Project was to build a system that could diagnose bacterial blood infections at an expert level. Although the goal was not to model closely the diagnostic behavior of expert physicians, observations of medical experts were critical during the early phases of the project stage, when the AI researchers (known as "knowledge engineers") gathered, structured, and encoded the experts' medical knowledge for use by the program. Later, having the program operate in a comprehensible manner was critical for debugging and refining it. *See generally* Buchanan & Shortliffe, *supra* note 11. It is usually the case that if there is no point of contact between the program's processing style and the human's, the program behavior appears inscrutable, impeding its development. Some similarity between the program's and the experts' processing also enhances one's belief in the correctness of the output of the program; sometimes this is so because it is easier for the program to explain its own reasoning in the user's terms. With respect to the issue of capturing the style of expert reasoning, a chess playing program like Deep Thought is an extreme case of a high performance program where there is no claim to cognitive validity. *See* Newborn & Kopec, *supra* note 5. There was no attempt to make Deep Thought think like a grand master. *See* Leithauser, *Kasparov Beats Deep Thought*, N.Y. Times, Jan. 14, 1990, § 6 (Magazine), at 33, 74 (discussion of grand master Kasparov's thoughts on ramifications of some of these issues).

more satisfying to exhibit a running program: to some degree success is a working computational model.[18]

In the context of law, these twin rationales translate into the twin goals of understanding certain key aspects of legal reasoning and building computational tools useful for legal practice, teaching, or research. An example of the former is the development of an AI model for reasoning based upon the doctrine of precedent. The process of developing an AI model causes one to learn about legal reasoning. Modeling involves elucidating key ingredients of precedent-based reasoning, such as making assessments of the relevance of precedents to new situations, distinguishing contrary cases, and drawing connections between relevant cases; then describing them in detail and building a program to execute them.

An example of the second, more applications-oriented goal, is construction of a set of computational tools (a lawyer's workbench, so to speak) to assist in the preparation of a brief. These may include functions to assist in gathering relevant precedents from data bases, sorting them according to their doctrinal approaches, and "Shepardizing" them. Building a practical system, like one to assist with writing a brief, requires developing analytical models. Typically, satisfaction with an analytical model increases if it offers insights leading to the practical advances.

The desire to develop a model of legal reasoning is not new. Certainly key aspects of legal reasoning, such as the analysis of precedent, have been the subject of many discussions.[19] However, for the most part, previous studies have not provided the level of detail required of an AI model; that is, they have not provided enough detail to indicate how they could be implemented as a computer program.[20] In AI, one is forced to be detailed. For instance, law has been described as "reasoning by example."[21] This may be an appropriate level of description for some purposes, but for AI, it leaves too many questions unanswered. To take advantage of invaluable

---

18. Actually building a program is quite different from speculating about it. Programming makes abundantly clear the weaknesses or difficulties of the model. *Cf.* McDermott, *Artificial Intelligence Meets Natural Stupidity,* in MIND DESIGN, *supra* note 15, at 143, 156-59 (highlighting risk of theorizing about implementation without actually implementing).

19. *See, e.g.,* B. CARDOZO, THE NATURE OF THE JUDICIAL PROCESS (1921); E. LEVI, AN IN-TRODUCTION TO LEGAL REASONING (1949); K. LLEWELLYN, THE BRAMBLE BUSH (1930); K. LLEWELLYN, THE CASE LAW SYSTEM IN AMERICA (1989); Radin, *Case Law and Stare Decisis: Concerning Präjudizienrecht in Amerika,* 33 COLUM. L. REV. 199 (1933).

20. Although this was obviously not their purpose, some discussions have come remarkably close. For instance, some of the analyses done by Llewellyn and Radin capture the spirit of the sort of description desired in AI. *See, e.g.,* K. LLEWELLYN, *supra* note 19, at 50 n.1 (describing doctrine of precedent, particularly concerning broad and narrow reading of rule of case); Radin, *supra* note 19, at 206-09 (describing concept evolution in law). Radin's description is uncannily similar to an al-gorithm, called the "candidate elimination algorithm," used in machine learning. *See* 3 HANDBOOK OF ARTIFICIAL INTELLIGENCE, *supra* note 2, at 385-91. For a comparison of this algorithm and Radin's analysis, see Rissland & Collins, *The Law as Learning System,* in PROCEEDINGS OF THE FOURTH ANNUAL CONFERENCE OF THE COGNITIVE SCIENCE SOCIETY 500, 501 (1986) (examining evolution of concept of "inherently dangerous" from Radin).

21. E. LEVI, *supra* note 19, at 1.

insights offered by legal scholars about legal reasoning, an AI researcher needs to specify both how the reasoning is to happen and what information and methods are required.

The AI approach forces one to be relentlessly analytic and specific. It advocates that one use ideas and methods of computer science to develop conceptual and computational frameworks.

B. *AI and Law: A Fruitful Synergy*

The law offers abundant opportunities for developing analytic and computational AI models. Law also has unique characteristics that make it a particularly challenging field for AI:

1. Legal reasoning is multi-modal, rich and varied: it includes reasoning with cases, rules, statutes and principles;

2. Case law has an explicit style and standard of reasoning and justification: stare decisis.

3. Specialized legal knowledge, such as cases and statutory rules, is well-documented and available from many sources, including case reporters, treatises, restatements, statutes, commercial summaries, and scholarly commentaries.

4. The law is self-aware and self-critical, and has an established tradition of examining its processes and assumptions. There is lively debate between proponents of competing jurisprudential schools.

5. The character of answers in the law is different from those in many other disciplines: answers are much more a matter of degree than clear-cut yes-or-no and they can change over time.

6. The knowledge used in legal reasoning is diverse, ranging from common sense to specialized legal knowledge, and it varies greatly in structure, character, and use.

These observations suggest the possibility of fruitful synergy between law and AI, and have implications for AI approaches. That the law is multi-modal means that an AI program will need to know about several modes of reasoning and how to use them in concert. That the law has an explicit, accepted style of reasoning gives the researcher a definite reasoning style to incorporate into a model and to use to analyze legal reasoning. However, some of the modes of reasoning, such as reasoning with cases, are very different from those used most widely in AI, and AI researchers have become interested in these only recently.

That the law is well-documented, self-aware, and self-critical makes it more accessible to AI researchers. If there were no repositories of legal knowledge and no tradition of trying to describe and criticize the goals and methods of legal reasoning, an AI researcher would have to start from scratch in trying to understand legal reasoning and to elucidate the knowledge involved. By no means is this to say that all the spadework has been done—the philosophical descriptions of legal reasoning are vague for AI

purposes, and there is a lot more to legal knowledge than the "book knowledge" of traditional legal materials. Still, an AI study of the law has been provided with a good beginning, an epistemological leg up.

That what counts as an "answer" in the law is not clearcut is also different from other disciplines. In law there is usually no unique right answer; rather there are reasonable alternative answers, more a matter of degree than of extremes. The answers are highly contextual, depend on goals and points of view, and change as the law evolves. Even the rule-based aspects of legal reasoning cannot be modeled with purely deductive methods. This also means that, unfortunately, there is never the comfort of a *quod erat demonstrandum* at the end of a reasoning episode to sanction it as sound and beyond reproach, as there is in mathematics. From the legal point of view, this is no real deficit—it's a feature and not a bug, to use the computer scientists' phrase—since it allows law to accommodate to changes in circumstance and purpose. But, computationally, the nature of legal answers adds complexity and difficulty as well as richness and flexibility.

These observations all suggest that the law is an exceedingly challenging domain for AI. Research in AI and law will impel AI in new directions and thus benefit AI. In turn, law will also benefit from AI, both analytically and practically. As an analytical medium, AI forces meticulous attention to details and precise testing of theoretical ideas. This, in turn, facilitates the unmasking of flaws, furthers the understanding of assumptions, and leads to proposals for refinements. AI focuses a spotlight on issues of knowledge and process to a degree not found in non-computational approaches, which often assume that some key task, like analogical reasoning, will be accomplished magically without a hint as to how,[22] or with too many critical details left underspecified.[23] The practi-

---

22. The philosopher Dan Dennett calls this the problem of a "loan of intelligence" or the hidden homunculus. *See* D. DENNETT, *supra* note 15, at 12. In an AI model, some process, somewhere, must actually do the work or else as Dennett puts it, the theory is "in the red." *Id.* A great danger is in "designing a system whose stipulated capacities are miraculous." *Id.* at 112; *see also* McDermott, *supra* note 18, at 143.

23. Some of Dworkin's models are intriguing in this regard, such as his model of hard cases. *See, e.g.*, R. DWORKIN, TAKING RIGHTS SERIOUSLY (1977). He argues that there may not really be any "hard" cases since one can use a set of relatively weighted principles to resolve certain "hard" questions, which arise because the on point cases are in conflict or there is a tie in the principles. The principles and weights are generated from a collection of relevant precedents. Dworkin omits the details of how to decide which precedents to use and how to induce principles from them. He is not necessarily wrong, but it would be instructive to extract a more detailed description of how his model works. By declining to instruct further on how to develop the weighting system, Dworkin has simply moved the problem of analysis back one step. Regarding the assignment of relative weights, he has walked headlong into the "credit assignment" quagmire, well known to workers in machine learning, where the problem is to assign credit or blame for the overall success or failure of a problem solution to an individual step or aspect of it. For instance, is the credit for a win or the blame for a loss in a game of checkers to be given to the penultimate move of a game, the first move, or some intermediate move or moves? How can one tell which features or principles "caused" a case to be resolved for one party or the other? *See* Minsky, *Steps Toward Artificial Intelligence*, in COMPUTERS AND THOUGHT, *supra* note 6, at 432 (1963); Samuel, *Some Studies in Machine Learning Using the Game of Check-*

cal benefits to law from AI are intelligent computational tools. The relationship between AI and law is truly synergistic.

C. *Some Desiderata for AI and Law Programs*

Given the special characteristics of law described above, we can enumerate several goals that we would like an ideal AI and law program to achieve. It should be able, among other things, to:

1. Reason with cases (both real and hypothetical) and analogies;
2. Reason with rules;
3. Combine several modes of reasoning;
4. Handle ill-defined and open-textured concepts;
5. Formulate arguments and explanations;
6. Handle exceptions to and conflicts among items of knowledge, like rules;
7. Accommodate changes in the base of legal knowledge, particularly legal concepts, and handle non-monotonicity, that is, changes in which previous truths no longer hold as more becomes known;[24]
8. Model common sense knowledge;
9. Model knowledge of intent and belief;
10. Perform some aspects of natural language understanding.

We are a long way from such an ideal. There is, however, activity on all of these important fronts, and impressive progress on a few.

In fact, one can today speculate about which of these desiderata can be expected now, in the near future, someday, or probably never. Presently, AI is actively pushing back the boundaries on case-based reasoning (goal 1), has well-understood methodologies in rule-based reasoning (goal 2), and is exploring multi-paradigm reasoning (goal 3). Reasoning with open-textured predicates (goal 4) has had an admirable first cut, but it will require further contributions from other AI specialities like case-based reasoning (CBR)[25] and machine learning.[26] Major contributions

---

ers, in 3 IBM J. Res. & Dev. 211 (1959).

24. For instance, the overturning of established doctrine with a new precedent.

25. "Case-based reasoning" is the process by which a program uses previous cases (e.g., precedents, experiences, problem-solving episodes, detailed solutions, plans) to solve a new case. There are two kinds of CBR: (1) "problem-solving CBR," in which the emphasis is on generating a solution to a new problem (case) by reusing and modifying old solutions; and (2) "interpretive CBR," in which the emphasis is on generating an interpretation of a new case by comparing and analogizing it with interpretations made in past cases. For both varieties, the key issues are the structure and content of case memory, indexing mechanisms for retrieving cases from memory, and metrics for assessing similarity or relevancy of retrieved cases. The sort of precedent-based reasoning done in the law is a paradigm for the second type of CBR. *See generally* Proceedings: Workshop on Case-Based Reasoning (J. Kolodner ed. 1989) (good sampling of current research); Kolodner, Rissland & Waltz, *Case-Based Reasoning from DARPA: Machine Learning Program Plan*, in Proceedings: Case-Based Reasoning Workshop 1 (1989) (sponsored by Defense Advanced Research Projects Agency; sets forth major classes of CBR, ingredients of CBR, and outstanding research issues).

26. "Machine learning" is the ability of a program to change itself, particularly its knowledge, so that it can perform better.

have been made to precedent-based argumentation,[27] and progress is being made on explanation (goal 5). Thus one can be optimistic that someday AI will be able to deal with many of the key goals for an AI and law program in at least a rudimentary manner. I would go so far as to say that the first five desiderata are attainable now or in the near future.

The next two desiderata—handling exceptions and conflict (goal 6) and accommodating to change in the base of knowledge (goal 7)—presently are being addressed broadly in AI, particularly in machine learning. It is reasonable to expect AI programs to be able to handle gradual change soon, but not abrupt or rapid change.[28] Non-monotonicity is being vigorously explored by an active cohort of researchers, primarily using approaches grounded in sophisticated logics. Thus, these two desiderata might be met someday.

As for the last three desiderata, I am much less optimistic. Modeling common sense reasoning, knowledge of intent and belief, and natural language capabilities are by far the hardest tasks. With regard to language, some capabilities like interactive dialogue and understanding of short summaries can be achieved in narrowly circumscribed domains; fuller capabilities are far off.[29] Widely applicable general "solutions" are very distant. Some, like the understanding of written appellate opinions—the ultimate fantasy—I expect never to see.[30]

II. SIGNIFICANT STEPS TOWARD A MODEL OF LEGAL ARGUMENT

This Section discusses several landmark projects in AI and law that provide significant steps toward the long range goal of understanding and modeling legal reasoning, particularly legal argument. They address: (1) reasoning with rules: (2) reasoning with open-textured concepts; (3) reasoning with cases, hypotheticals, and precedent-citing argumentation; (4) mixed paradigm reasoning with rules and cases; and (5) deep models of legal knowledge.

---

27. *See infra* Section II.C.

28. Changes of the type involved in "normal science," to use Kuhn's term, are amenable to techniques from machine learning. In contrast, the types of change involved in a Kuhnian "paradigm shift" are probably not, at least in the sense of the system on its own recognizing the need for a shift and carrying it out through the creation of novel concepts. *See* T. KUHN, THE STRUCTURE OF SCIENTIFIC REVOLUTIONS 10, 66 (2d ed. 1970); *see also* E. LEVI, *supra* note 19, at 8, 9 (describing change in law).

29. *See generally*, Lehnert, *Knowledge-Based Natural Language Understanding*, in EXPLORING ARTIFICIAL INTELLIGENCE: SURVEY TALKS FROM THE NATIONAL CONFERENCES ON ARTIFICIAL INTELLIGENCE 83 (1988).

30. This is not so much because of any special difficulties with opinions, but because understanding text is just plain hard. *See supra* notes 13, 29. Legal opinions use specialized vocabulary and conventions, so they might be harder because these considerations require extra processing; on the other hand, the extra constraints these considerations impose might make legal opinions more amenable to computational analysis.

A. *Reasoning with Rules*

One of the earliest steps toward a model of legal reasoning was the use of expert systems[31] to model certain rule-based aspects of law.[32] This step reflects the development of AI: Rule-based expert systems were the first type of AI system to become widely available and employed beyond the AI research community.[33] Furthermore, their underlying computational mechanisms are conceptually clear and they have many computational strengths. While from the legal standpoint there is a variety of opinions as to the validity, usefulness, and status of rules, and there are acknowledged difficulties in representing them,[34] it is still quite natural to take some body of legal rules and embed them in a standard rule-based computational framework.

In the rule-based approach, a rule is encoded in a simple, stylized if-then format: If certain conditions are known to hold, then take the stated action or draw the stated conclusion.[35] Rule-based systems work by chaining these rules together.[36]

---

31. An "expert system" is a special-purpose computer program, which can be said to be expert in a narrow problem area. Typically, such a program uses rules to represent its knowledge and to reason. *See, e.g.,* P. HARMON & D. KING, EXPERT SYSTEMS (1985); D. WATERMAN, A GUIDE TO EXPERT SYSTEMS (1986); B. BUCHANAN & R. SMITH, *Fundamentals of Expert Systems,* in 4 HANDBOOK OF ARTIFICIAL INTELLIGENCE, *supra* note 2, at 149.

32. An even earlier effort related to rule-based aspects of law was Layman Allen's work on "normalization." His emphasis was on eliminating syntactic ambiguity in statutes and legal documents rather than on using computational programs to reason with them. *See* Allen, *Symbolic Logic: A Razor-Edged Tool for Drafting and Interpreting Legal Documents,* 66 YALE L.J. 833 (1957). Of course, if certain ambiguities, for instance those about the scope of logical connectives and the meaning of words like "if," "except," and "unless," were eliminated from legal sources, encoding legal rules for use by an expert system would be easier and less open to debate. Allen & Saxon, *Some Problems in Designing Expert Systems to Aid Legal Reasoning,* in THE SECOND INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND LAW: PROCEEDINGS OF THE CONFERENCE 94 (1987) [hereinafter ICAIL-87] (discussing 48 alternative interpretations of structure of proposed limitations on exclusionary rule).

33. For example, the expert system DENDRAL has been widely used by organic chemists, *see* R. LINDSAY, B. BUCHANAN, E. FEIGENBAUM & J. LEDERBERG, APPLICATIONS OF ARTIFICIAL INTELLIGENCE FOR ORGANIC CHEMISTRY: THE DENDRAL PROJECT (1980); sources cited *supra* note 31.

34. Besides "syntactic" difficulties, *see* Allen, *supra* note 32, there are "semantic" difficulties such as the presence of conflicting rules, imprecise terms, and incompleteness. *See* Berman & Hafner, *Obstacles to the Development of Logic-Based Models of Legal Reasoning,* in COMPUTER POWER AND LEGAL LANGUAGE 183 (C. Walter ed. 1988) (discussing difficulties with logic-based approaches); Berman, *Cutting Legal Loops,* in ICAIL-89, *supra* note 1, at 251 (discussing definitional circularity, recursion, and self-referencing in statutes).

35. For instance, Rule 1, concerning "responsibility for use of the product," from the case settlement system of Waterman and Peterson states, "IF the use of [the product] at the time of the plaintiff's loss is foreseeable and (that use is reasonable-and-proper or that use is an emergency or (there is a description by the defendant of that use and that description is improper) or there is not a description by the defendant of that use) THEN assert the defendant is responsible for the use of the product." D. WATERMAN & M. PETERSON, MODELS OF LEGAL DECISION MAKING 37 (Institute for Civil Justice of The Rand Corporation Memo R-2717-ICJ 1981) (parentheses used to denote scope and distribution of logical connectives; thus, here there are two principal antecedents, second of which can be satisfied in alternative ways).

36. The systems can work either "forward" by reasoning from facts to a desired conclusion supported by them, or "backward" from a desired conclusion to find facts supporting it. Forward chaining simply is the repeated application of the logical inference rule *modus ponens*: If one has a rule "If A then B" and the fact A, conclude the fact B. Alternatively, in backward chaining, to establish the

The rule-based approach is particularly useful because in many domains much of an expert's knowledge is amenable to expression in if-then rules, many of which are "heuristic."[37] Heuristics are typically an expert's individual synthesis of past problem solving, and they capture methods for making educated hunches. Expert systems provide a straightforward way to harness heuristic expertise, expressed as rules.

The first uses of the expert systems approach in law were those of Donald Waterman and Mark Peterson at the RAND Corporation's Center for Civil Justice.[38] In their systems, rules were used to encode statements of doctrine as well as legal expertise, cast as rules of thumb.[39] Their work, done while expert systems approaches were beginning to be widely applied, is representative of both what can be accomplished in law with the expert systems methodology and also of the difficulties inherent in such an approach. Waterman and Peterson investigated how the expert systems approach fared as a practical tool in legal applications, and demonstrated it as a methodology for modeling legal expertise.[40]

The problem area of one of their systems, called LDS (for Legal Decision-making System), was assessing the "worth" of a products liability case for settlement purposes. The program, using rule-based models for strict liability, comparative negligence, and calculation of damages, was able to compute the worth of a case, to capture chains of inferences supporting a conclusion like negligence, and to model certain aspects of settlement negotiation. The project demonstrated the applicability of rule-based techniques for legal applications.

Perhaps their system's greatest weaknesses were that it glossed over difficulties inherent in reasoning with imprecise terms and it underplayed the adversarial nature of legal reasoning. For instance, LDS asked the user whether the use of the product was "foreseeable."[41] This is a notoriously subtle question, whose answer is open to interpretation and is entirely contextual. One can argue that coming to a conclusion about the foreseeability of use is at the heart of reasoning in the negligence area,

---

fact B, one looks for such a rule and verifies that one has information to satisfy its precondition, A; if A were not satisfied, one then would look for rules establishing A, and so on until the necessary factual basis were reached and the desired conclusion logically supported. See W. SALMON, LOGIC (2d ed. 1973); sources cited supra note 31.

37. A "heuristic" is a rule of thumb, a bit of encapsulated wisdom. Much problem-solving behavior is guided by heuristic rules of thumb, such as "If it is fourth down and long yardage is required, then punt," or "If you need to get your citations in correct law journal form, then ask a law journal editor." Heuristics are methods that past experience has shown to be "good" or "wise" things to do; they do not necessarily guarantee a solution, as might algorithms or mathematical theorems, and occasionally they might even produce wrong answers or lead in counter-productive directions. The word "heuristic" stems from the Greek for invention or discovery. The mathematician George Polya discussed heuristic reasoning and the use of heuristics in mathematical reasoning and problem solving. See G. POLYA, HOW TO SOLVE IT (1973).

38. See generally D. WATERMAN & M. PETERSON, supra note 35.
39. See, e.g., supra note 35 (example of one of their rules).
40. See D. WATERMAN & M. PETERSON, supra note 35, at 13-15.
41. Id. at 45.

and that by asking the user to make such an interpretive call, the system has given away the whole game. LDS also illustrates another general problem with the rule-based approach: it does not give enough prominence to the adversarial nature of legal reasoning, where the opposing sides seek to establish different, and often contradictory, conclusions. Perhaps more fundamentally, the rule-based approach assumes that the set of rules has no inherent difficulties, like ambiguities, gaps, and conflicts. To make a rule-based system work, the programmer must usually eliminate these problems and make the rules appear more consistent and complete than they are.

Although oversimplified, the system built by Waterman and Peterson was a landmark in unknown territory. It is unfair to condemn it for failing to solve all the hard problems of modeling legal reasoning. To their credit, Waterman and Peterson confront these criticisms head-on in their own critiques. For instance, they address both the difficulty in handling imprecise legal predicates and the need for subtler reasoning with such concepts.[42]

Today, expert systems are one of the most important approaches used and there are many additional examples of current applications. Two recent efforts in Britain are particularly noteworthy. The first is the Latent Damage System by Richard Susskind and Phillip Capper, which addresses a problem area concerning the 1986 British Latent Damage Act.[43] Their system addresses the complex legal issues relating to time periods within which claimants may start proceedings when the damage or suffered loss was "latent."[44] Besides providing a useful application in an intricate body of law, this work is also interesting in the way in which Susskind explains the project in terms of analytic jurisprudence.[45]

The second effort is a group of projects of the Logic Programming Group at Imperial College, University of London, one of which modeled the recent 1981 British Nationality Act.[46] A major goal of this project was to test the suitability of rule-based approaches for representing large, complicated statutes. The process of modeling the statute with rules helped the programmers uncover problems specific to the Act, such as undefined legal predicates and loopholes in the Act, as well as a variety of general problems in the rule-based approach concerning the formalization of negation, the use of defaults and counterfactuals, the representation of com-

---

42. *See id.* at 26. Waterman, a pioneer in machine learning, understood how learning issues were also critical to expert systems. He was well ahead of his time and his early death in 1987 was a great loss.

43. Latent Damage Act, 1986.

44. P. CAPPER & R. SUSSKIND, LATENT DAMAGE LAW—THE EXPERT SYSTEM (1988). The book contains diskettes for installing and running the system on a machine like the IBM PC or XT.

45. *See* R. SUSSKIND, EXPERT SYSTEMS IN LAW (1987).

46. British Nationality Act, 1981. *See* Sergot, Sadri, Kowalski, Kriwaczck, Hammond & Cory, *The British Nationality Act as a Logic Program*, 29 COMM. ACM 370 (1986) [hereinafter Sergot].

mon sense knowledge, and the use of administrative discretion in statutes, each of which poses technically interesting problems for AI.[47]

Although rule-based techniques are but one component in a full model of legal argument, they can be satisfyingly powerful from the perspective of building useful applications. There are abundant examples of applications, many motivated by a need for practical, labor-saving tools in the trenches, where expert resources are terribly strained.[48] Given the availability of inexpensive expert systems development tools and rule-based programming languages, such application systems will continue to proliferate; anyone having access to such tools can write one.[49] In summary, rule-based techniques have demonstrated utility for frequently performed analyses on stereotypical cases in stable, well-developed bodies of law.[50]

B. *Reasoning with Open-Textured Concepts*

Whether there are run-of-the-mill cases amenable to straightforward rule-based analysis is a question with a long jurisprudential history. This brings us to understanding the limitations of rule-based reasoning and the special problems inherent in legal concepts. Anne Gardner's work both elucidates the limitations of rule-based approaches and also points the way toward the use of case-based techniques.[51] Gardner investigated the adequacy of rule-based approaches through the framework of "hard/easy questions."[52] In the hard/easy paradigm, a question is considered "easy" if the legal experts agree as to its analysis or outcome; otherwise, it is "hard." Hard questions typically arise in conjunction with "open-textured" legal concepts.[53] To implement this distinction in a program,

---

47. *See, e.g.*, Sergot, *supra* note 46, at 379 (defining "negation as failure" as the conclusion that something is false if all known ways of showing it true fail);*id.* at 382 ("counterfactual conditionals" as in the statutory phrase "became a British citizen by descent or would have done so but for . . . ."); *id.* at 382 (discretion as in the phrase "If . . . the Secretary of State sees fit"); Berman, *supra* note 34; Berman & Hafner, *supra* note 34.

48. *See, e.g.*, Grady & Patil, *An Expert System for Screening Employee Pension Plans for the Internal Revenue Service*, in ICAIL-87, *supra* note 32, at 137 (describing Internal Revenue Service project to process employee pension plans); Pethe, Rippey & Kale, *A Specialized Expert System for Judicial Decision Support*, in ICAIL-89, *supra* note 1, at 190-94 (system for processing claims under The Federal Black Lung Benefits Act); Weiner, *CACE: Computer-Assisted Case Evaluation in the Brooklyn District Attorney's Office*, in ICAIL-89, *supra* note 1, at 215-23 (describing system for post-arrest/pre-trial processing of drug busts by police).

49. *See, e.g.*, Wiehl, *Computers Assuming New Roles at Law Firms*, N.Y. Times, Jan. 20, 1989, at B4, col. 3; Blodgett, *Artificial Intelligence Comes of Age*, 73 A.B.A. J. 68 (1987).

50. For example, Paul Brest has commented that constitutional law would be "wildly unsuited for an expert system because the principles are vague . . . . Expert systems are best for areas of law that are rule-bound." Blodgett, *supra* note 49, at 70 (quoting Brest).

51. *See generally* A. GARDNER, AN ARTIFICIAL INTELLIGENCE APPROACH TO LEGAL REASONING (1987). Gardner holds a J.D. from Stanford and practiced law before returning to Stanford to obtain a Ph.D. in computer science.

52. For a discussion of this framework, see Hart, *Positivism and the Separation of Law and Morals*, 71 HARV. L. REV. 593 (1958), *reprinted in* H.L.A. HART, ESSAYS IN JURISPRUDENCE AND PHILOSOPHY (1983); R. DWORKIN, *supra* note 23, at 81.

53. For a discussion of "open textured" legal concepts, see H.L.A. HART, THE CONCEPT OF LAW 121-32 (1961).

several specifics must be provided, such as who are the experts to be used, what counts as a disagreement among them, and how to tell if a disagreement exists. From a practical standpoint, if one could sift the easy from the hard, one could employ rule-based techniques, for instance, to solve the easy questions, and attack the hard questions with other methods.[54]

Gardner built a computational model for the hard/easy paradigm and for reasoning with open-textured legal concepts. The problem area of her program was classic offer-and-acceptance law, and its task was to analyze issue-spotter questions, taken from sources such as bar exams and Gilbert's Law Summaries. The program analyzed what the legal questions were, spotted which were hard, and answered the easy ones. Gardner required that her program be computationally reasonable; it should not take an unduly hard computation to decide which questions are easy and hard, or else nothing much would have been gained, at least computationally, if not theoretically.

In Gardner's model, hard questions arise from problems with rules in two ways: either the rules relevant to the domain, such as those from the Restatement of Contracts,[55] are incomplete, circular, or contradictory; or the legal predicates used in the rules are indeterminate and their interpretation cannot be resolved.[56] When there is a mixture in the disposition of cases used to resolve these problems, Gardner's program categorizes the problem as hard. Typically it is not possible to handle indeterminate concepts because, as Gardner puts it, the rules "run out;" that is, a needed rule's premise uses an undefined term.[57]

Gardner's approach to these problems was to give her program a rich body of knowledge and several powerful heuristics. Her program's knowledge included (1) Restatement-like rules for the doctrine of offer and acceptance,[58] (2) a "network" to represent various states in which parties

---

54. This is of course not the only way to model the hard/easy problem; for instance, case-based techniques could be used for both types of questions. However, given that rule-based methods are well-understood, Gardner's approach is quite natural.

55. RESTATEMENT (SECOND) OF CONTRACTS (1981).

56. Note that a typical rule-based approach to handling difficulties with the rule set is to resolve conflicts by hand before encoding the rules in the program. Of course, one cannot always eliminate conflicts in the rule set because even the experts disagree. For example, Gardner's program has pairs of conflicting rules about a rejection that revokes a previous acceptance of an offer, and about simultaneous acceptance and proposal to modify the contract. See A. GARDNER, supra note 51, at 134. Gardner deliberately leaves conflicting rules in her system because she wants her model to be able to handle the fact that legal experts, and thus their rules, can be in conflict. See id. at 3-4. A typical rule-based way to finesse problems with open-textured legal predicates is either simply to ask the user to make the judgment or engage in what might be called "definitional backchaining:" write rules whose pre-conditions specify whether a legal predicate obtains, and then write rules defining what those pre-conditions mean, and so on. See Sergot, supra note 46, at 378 (implementing this approach); D. WATERMAN & M. PETERSON, supra note 35, at 45 (suggesting this approach).

57. A. GARDNER, supra note 51, at 33-34 (discussing problem of inadequacy of rules). This brings the traditional rule-based technique (of backward chaining) to a complete standstill before an adequate definitional basis is reached and application of the ambiguous term can be resolved.

58. For example, her program uses about twenty rules covering contract doctrines such as offer and acceptance. See id. at 133.

could be in offer and acceptance situations, and transitions between these states,[59] (3) relevant common sense knowledge,[60] and (4) prototypical fact patterns (exemplars) for certain key concepts.[61] The exemplars serve as simplified precedents in that their classifications have already been determined and are used to interpret new cases. For convenience in this discussion, the rules, network, and common sense knowledge will be called as a group the "non-exemplar" knowledge.

Gardner's program tries to answer a question by first using the non-exemplar knowledge, and then, if that fails, by using the exemplars, which function as prototypical clear cases. If an answer can be derived with non-exemplar knowledge, the validity of the answer is checked against relevant exemplars.[62] Thus, her program uses exemplars both in a primary way to derive an answer, and in a secondary way to check the validity of an answer. In any case, if there is a mix in the disposition of relevant exemplars, the question is considered hard.[63]

Gardner took the skeleton of an idea from legal philosophy and provided it with computational flesh. Obviously, hers is not the only model that one can propose for the hard/easy paradigm, but it is a computationally disciplined one. By examining the model's computational underpinnings and assumptions, one can learn something about the hard/easy distinction.[64] Her work raises interesting questions about the use and content

---

59. For example, there can be an offer pending, and an acceptance will enable one to make a transition to the state of being in a contractual relationship. *See id.* at 123-25; *id.* at § 6.2.

60. For instance, that a telegram is a kind of document, which in turn is a kind of inanimate object, which in turn is a kind of physical object. *See id.* at 90-91. *See generally id.* at 85-117 (very detailed presentation of representation issues).

61. For example, her program has two prototypical fact patterns for the legal predicate "produces a manifestation with content:" making an utterance with some content, and sending a document with some content. *See id.* at 156-57.

62. The determination of hard/easy cases is made as follows. If a problem such as the application of a legal predicate to the current facts can be (tentatively) answered using the program's non-exemplar domain knowledge, primarily rule-like in character, and if there are no opposing case exemplars, then the question is deemed easy and its answer is the one derived. If the case exemplars used to check the tentative answer all point the opposite way, then the question is also considered easy but the answer is that supported by the case exemplars. If, on the other hand, there is a mixture in the disposition of the case exemplars, then the question is flagged as hard and the program does not attempt to provide an answer. If a (tentative) answer cannot be derived with the domain rules because, for example, a predicate cannot be resolved, but all the relevant cases point the same way and thus can be used to posit and support an answer, then the question is considered easy and the answer is that indicated by the case exemplars. On the other hand, if an answer cannot be derived using the domain rules and there is a mixture in the cases, then the question is also deemed hard. *See id.* at 54-55, 160-61 (abbreviated descriptions of her program's algorithm).

63. One might ask, why not go to the cases right off? My interpretation is that the reasoning involving the other knowledge, the rules, the transition network, and the common sense knowledge, provides a means by which to index relevant cases. Furthermore, even in the easy cases, one would need to derive the answer by some means, and in her model this is attempted with the rule-guided reasoning, which would need to be done at some point anyway. Alternatively, one could omit looking at the rules altogether and just consider cases. Of course, one would need to say which cases to consider and what exactly to do with them. *See infra* Section II.C (discussing work of Ashley).

64. For instance, too many questions might be categorized as "hard" in this model since there is very often a mixture in the disposition of the cases. Perhaps one needs to be more discriminating in the use of cases.

of cases, and about which cases to include in a decision maker's case base.[65]

While Gardner's work certainly addresses some of the shortcomings of purely rule-based models, her program still falls short with regard to reasoning with cases. Some of the core features of precedent-based argumentation, such as determining relevancy and drawing analogies, are beyond the scope of her program; these issues are the focus of the next set of projects to be explored here. Nonetheless, Gardner's research provides an exceptionally important step toward the goal of understanding and modeling legal argumentation.

C. Precedent-Based Reasoning with Cases and Hypotheticals

The next landmark AI project was Kevin Ashley's, which was done in this author's research group at the University of Massachusetts.[66] Ashley developed a program called HYPO to model certain aspects of case-based reasoning as exemplified by appellate-style argumentation with precedents. The problem area of this project was trade secret law. The task was to produce elements of precedent-based argument. The HYPO project focused solely on reasoning with cases, and was the first project—not only in AI and law, but also in AI in general—to attack squarely the problem of reasoning with cases and hypotheticals in a precedent-based manner.

HYPO performs as follows: Given a fact situation, HYPO analyzes it according to its model of trade secret law and then retrieves relevant cases from its knowledge base of cases. It then determines which relevant cases are most on point, or potentially so, for whose point of view, and from which analytic approach. HYPO then generates the skeleton of an argument. In such an argument snippet, HYPO first argues for side one (plaintiff or defendant) by making a legal point and citing its best, most on point cases; then it argues for side two by responding with a counterpoint, citing a most on point case supporting side two's point of view or also by distinguishing the current facts from side one's cases; and finally, HYPO argues again for side one with a rebuttal of side two's position, which may include distinguishing side two's cases and strengthening the

65. For instance, in Gardner's program cases are prototypical fact patterns rather than actual cases. What about using real fact patterns instead, or in addition? Which ones? How should the program's memory of cases be organized? Are certain cases to be preferred to others?

66. See generally K. ASHLEY, MODELING LEGAL ARGUMENT: REASONING WITH CASES AND HYPOTHETICALS (forthcoming 1990) (originally completed as Ph.D. dissertation, Department of Computer and Information Science of the University of Massachusetts, Amherst, February, 1988, under this author's direction); Ashley & Rissland, A Case-Based Approach To Modeling Legal Expertise, IEEE EXPERT, Fall 1988, at 70–77 (short overview and example). Ashley holds a J.D. from Harvard Law School and practiced as a litigator with White & Case in New York before returning to study computer science at the University of Massachusetts.

relationship of side one's analysis to the current facts.[67] At various points in the argument, HYPO may generate and employ hypotheticals—for instance, to refute a claim when no actual counter-example case exists in HYPO's knowledge base of cases.[68]

HYPO is able to: assess relevancy of cases; decide for each side which relevant cases are most on point and which of these are best to cite in argument; analogize and distinguish cases; generate and reason with hypotheticals; cite various types of counterexamples to a point; and construct the skeleton of a case-citing argument. HYPO does not in any way attempt to bring in policy-level concerns or argumentation; rather, it sticks to arguing with cases, on their facts, in a technical way. HYPO also does not include other aspects of legal reasoning, such as reasoning with rules.

A key feature of HYPO is a kind of case index, called a "dimension," which it uses to retrieve and analyze cases. Dimensions represent important legal factors. They encode the knowledge that the presence of certain facts enables a case to be addressed from a certain point of view.[69] A dimension enables HYPO to retrieve a set of cases that support the same analytic approach and to compare and assess the relative strength of the cases within this group.[70]

HYPO uses dimensions to define concepts like "relevant," "most on point" and "best" cases.[71] In HYPO, a case is considered "relevant" if it shares at least one dimension with the fact situation. HYPO ranks cases according to how on point they are by examining the overlap between the set of dimensions present in the fact situation and the sets of dimensions present in the cases. If the overlapping set of dimensions of one case, B, is contained within that of another case, A, then A is defined as more on point than B, the rationale being that A has more lines of analysis in common with the fact situation than does B.[72]

---

67. For several examples, see K. ASHLEY, *supra* note 66; Ashley & Rissland, *supra* note 66, at 74-76.

68. K. ASHLEY, *supra* note 66.

69. For instance, in HYPO's domain of trade secret misappropriation law, knowing facts about the relative costs incurred by the plaintiff and defendant in developing their putative secrets into a product enables one to make arguments about the gain of competitive advantage, and knowing facts about the disclosures made by the plaintiff enables one to make arguments about how well the plaintiff kept his knowledge secret. *See, e.g.*, Gilburne & Johnston, *Trade Secret Protection for Software Generally and in the Mass Market*, 3 COMPUTER L.J. 211 (1982). For further explanation and examples, see K. ASHLEY, *supra* note 66; Rissland, Valcarce & Ashley, *Explaining and Arguing with Examples*, in PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE 288, 291-93 (1984).

70. In HYPO, for instance, the greater the disparity between the development costs of plaintiff and defendant, the easier it is to argue that, all other things being equal, the defendant gained an unfair competitive advantage through misappropriation of plaintiff's trade secret. Thus, for example, since there is a case in which the plaintiff prevails with a ratio of two to one for plaintiff's to defendant's costs, *see* Telex Corp. v. IBM Corp., 367 F. Supp. 258 (N.D. Okla. 1973), then a new case with a ratio of four to one would represent an even stronger position for the plaintiff in HYPO.

71. *See* K. ASHLEY, *supra* note 66.

72. So, for instance, suppose the dimensions that apply to the current fact situation (CFS) are W, X, Y, and Z. Suppose Case A shares X, Y, and Z with CFS, and Case B shares just X and Y. Then

By then taking into account which cases support or cut against a party, their relative strength along shared dimensions, and the dimensions not shared, HYPO is able to analogize, distinguish, and otherwise manipulate cases to construct snippets of precedent-citing argument. For instance, to distinguish a case that the opposing party, say the plaintiff, has cited as controlling the outcome of the current fact situation, HYPO looks at the cited case for the existence of dimensions unshared with the current situation. HYPO then argues, for the defendant, that the presence or absence in the current factual setting of these dimensions diminishes the applicability of the cited case; in other words, these are differences that really make a difference.[73]

To summarize, the HYPO project models some of the key ingredients of reasoning with precedents. The model provides computational definitions of relevant cases, on point cases, and best cases to cite.[74] Because of the precision of HYPO's model, a researcher can examine specific details and assumptions.[75] By providing an analysis and computational model for reasoning with cases, perhaps the most vital aspect of legal reasoning, Ashley's work is a giant step toward the goal of understanding legal argumentation.

An earlier effort addressing some of the same concerns as HYPO was the work of Thorne McCarty on legal argument.[76] One of McCarty's longstanding goals has been the construction of knowledge representation mechanisms with which to model certain aspects of legal argumentation. For instance, McCarty has proposed a mechanism called "prototypes and deformations" to model certain aspects of appellate argument. He has used it to examine the sort of situation in which one incrementally steps from a desirable precedent to the current case, through a sequence of intermediate real and hypothetical cases, in order either to construct an argument by analogy for the result one desires, or else to show the inconsistency of the result argued for by one's opponent.[77] His model requires a

---

Case A is more on point than Case B because the set that B shares with CFS is a subset of the set that A shares. Suppose there is a third relevant case, Case C, sharing dimensions W and X with CFS; Case C is neither more nor less on point than A or B. Both Case A and Case C are most on point cases since each shares maximally with respect to a subset of dimensions.

73. For instance, if the case cited by the plaintiff is very strong on a pro-plaintiff dimension not present in the current case, the defendant can argue that that particular dimension is responsible for the pro-plaintiff outcome. If, in addition, the defendant can point to a different case, the same in all respects to the plaintiff's cited case except for that one dimension, in which the defendant prevailed, then the assignment of credit for the outcome in the plaintiff's case to the missing dimension is all the more convincing.

74. For instance, the best cases for a side to cite are defined as those most on point cases for that side that share at least one applicable dimension favoring that side.

75. For instance, an alternative way to measure relevancy is to minimize the sets of dimensions not shared. One could then swap this alternative definition for relevancy in the HYPO model and observe the repercussions.

76. *See generally* McCarty, *Reflections on TAXMAN: An Experiment in Artificial Intelligence and Legal Reasoning*, 90 HARV. L. REV. 837 (1977).

77. *See* McCarty & Sridharan, *The Representation of an Evolving System of Legal Concepts: II*

great deal of attention to details concerning the representation of cases and their constituent parts.[78] Thus, although some of McCarty's research is clearly tied to that on cases, hypotheticals, and argument, the bulk of his work is more aptly grouped with efforts on developing "deep models" of legal knowledge, which concentrate more on developing highly detailed models of knowledge than the other projects discussed thus far, and will be discussed briefly later.[79]

One of the shared concerns of McCarty and this author is the use of hypotheticals in legal arguments and development of computational methods to generate them.[80] For example, HYPO uses hypotheticals to fulfill the need for counter-examples to refute another's position when real cases cannot be found or to show the sensitivity of a given fact to the interpretation of a case.[81] It uses its internal mechanisms, like dimensions, to generate hypotheticals, for instance by adding, deleting, or changing a fact affecting a dimension.[82]

The conceptual framework provided by such AI models can also be used to analyze expert use of hypotheticals in Socratic law school dialogues, advocacy, judical opinion, and appellate oral argument.[83] Improvements of advocacy and pedagogical skills are two potential benefits from these analyses. Another is increased understanding of the development of legal concepts and theories. For instance, a good hypothetical provides a way to test the validity of a theory: it provides a *gedanken* experiment. This is especially valuable when there are very few real cases. Hypotheticals play a special role in case-based reasoning, and increased understanding of them would contribute to our goal of modeling legal reasoning.

Another contribution to understanding the case-based aspects of legal reasoning would be to understand how new arguments could be generated by modification of old arguments, without starting "from scratch" from a statement and analysis of the facts, proceeding through an analysis of rele-

---

Prototypes and Deformations, PROCEEDINGS OF THE SEVENTH INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE 246 (1981) (discussion of prototypes and deformations); L. MC-CARTY & N. SRIDHARAN, A COMPUTATIONAL THEORY OF LEGAL ARGUMENT (Laboratory for Computer Science Research, Rutgers University, New Brunswick, N.J. Technical Report No. LRP-TR-13, 1981) (presenting example from Justice Pitney's opinion and Justice Brandeis' dissent in landmark tax case Eisner v. Macomber, 252 U.S. 189 (1920)).

78. McCarty & Sridharan, *supra* note 77.

79. *See infra* Section II.E.

80. *See, e.g.*, McCarty & Sridharan, *supra* note 77; L. MCCARTY & N. SRIDHARAN, *supra* note 77; Rissland, *Examples in the Legal Domain: Hypotheticals in Contract Law*, in PROCEEDINGS OF THE FOURTH ANNUAL CONFERENCE OF THE COGNITIVE SCIENCE SOCIETY 96 (1982).

81. *See* K. ASHLEY, *supra* note 66; Rissland & Ashley, *Hypotheticals as Heuristic Device*, PROCEEDINGS OF THE FIFTH NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE 289 (1986).

82. For instance, one can make a trade secret misappropriation case incrementally weaker or stronger by variation of a key fact, like the relative competitive advantage gained by the defendant or the number of disclosures made by the plaintiff. *See, e.g., supra* notes 69-70.

83. *See, e.g.*, Rissland, *Argument Moves and Hypotheticals*, in COMPUTING POWER AND LEGAL REASONING 129 (C. Walter ed. 1985) (examples of hypotheticals in Socratic teaching); Rissland & Ashley, *supra* note 81, at 290-91 (examples in context of advocacy); McCarty & Sridharan, *supra* note 77 (examples in context of judicial opinion).

vant precedents, and culminating in an argument, as HYPO describes. Of course, a model to describe this alternative approach would require representation and manipulation of a knowledge base of arguments. This is the subject of the on-going research.[84] Another step would be to explore more fully the relationship between relevance and analogy.[85]

However, just as Waterman and Peterson artificially confined themselves to reasoning with rules, others, like Ashley, considered only reasoning with cases. From the long range perspective of developing an all-encompassing model for legal reasoning, both approaches leave something out, since many areas of the law are neither purely case-based nor purely rule-based. Both types of reasoning are clearly needed in statutory areas, where, despite the drafters' best efforts, there are always legal terms and rules whose meaning and scope are not totally fixed and for which one must look to cases for interpretation. The next major step toward a model of legal reasoning is development of approaches to combine reasoning with cases and reasoning with rules.

D. *Reasoning with Rules and Cases*

Reasoning in a statutory area like tax law is typical of mixed paradigm reasoning. For instance, consider the subsection of tax law governing the deductibility of expenses for a so-called home office. The tax code includes requirements that the office be used regularly and exclusively, and that it is the principal place of business or a place where one meets or deals with clients.[86] However, nowhere in the tax code are the requirements for what constitutes "regular" or "exclusive" use, etc., set forth. All such terms have been the subject of numerous cases,[87] and to determine whether they apply to a new situation, one must typically examine precedents and create precedent-based arguments or subarguments to justify the interpretation. Of course, the argumentation must be tied back to the requirements and structure of the statute.

Methods for accomplishing hybrid rule-based and case-based reasoning go beyond those needed for pure case-based and or rule-based reasoning in isolation. For instance, suppose that one had a rule with three preconditions for the awarding of a certain benefit: "In order to receive bene-

---

84. *See, e.g.*, Branting, *Representing and Reusing Explanations of Legal Precedents*, in ICAIL-89, *supra* note 1, at 103 (developing model of how past arguments can be modified for re-use in new case).

85. *See* K. BELLAIRS, CONTEXTUAL RELEVANCE IN ANALOGICAL REASONING: A MODEL OF LEGAL ARGUMENT (submitted to the University of Minnesota, 1989) (Ph.D. thesis) (how relevance can be defined in terms of analogy). This work continues the examination begun by Ashley of what it means for one case to be relevant to another. It draws on work such as Gentner's, *supra* note 12, which views analogy as a mapping between structured objects like legal concepts and cases.

86. I.R.C. § 280A(c)(1) (1988).

87. For example, cases examining "exclusive use" include Frankel v. Commissioner, 82 T.C. 318 (1984), Baie v. Commissioner, 74 T.C. 105 (1980), and Chauls v. Commissioner, 41 T.C.M. (CCH) 234 (1980).

fit B, one must satisfy requirements R1, R2, and R3." Suppose that R1 and R2 are clearly satisfied. To argue for receiving the benefit, one has several strategies to consider. For instance, one can try to find on point precedents addressing the interpretation of antecedent R3, and then argue that, in fact, the facts actually do satisfy R3 and thus the rule. Alternatively, one can try to find on point precedents in which only the first two requirements were met and the benefit was awarded nonetheless, and then argue that the first two are sufficient by themselves. These are examples of "near miss" strategies. To carry them out requires not only the use of both the rule and the cases, but also that both types of reasoning be done in concert. It is not good enough to confine one's attention to the cases; the case-based aspects must speak to the rule. Similarly, one gets nowhere using the rules alone, since the rules "run out." Figuring out how to coordinate the reasoning activities of the rule-based and case-based aspects is a question of "control."[88]

Gardner's research suggests one model for integrating rule-based and case-based reasoning: use cases when reasoning with rules comes to an impasse, and also use cases to validate the conclusions reached using rules.[89] In this approach, case-based reasoning is subjugated to rule-based reasoning. Of course, selecting the dominant mode of reasoning should really depend on the context or circumstances: sometimes the rules should drive the reasoning, and sometimes the cases.

A different approach for an AI program would be to have independent processes—one for case-based tasks, one for rule-based tasks—where each works from its own point of view independently. Later, an executive process would integrate the results. Metaphorically, this is like sending off associates to do specific tasks, such as library work on case law or statutory law, and having the senior associate integrate the information and the reasoning.

For a computational model, one must spell out when and how the processes interact. In particular, how should the associate processes communicate their results with one another? Should they wait until they are done with their individual tasks before communicating? Should they share intermediate results and insights?

One AI model for integrating the work of the associate processes would be the following: each process could have access to a common blackboard, on which it could write anything interesting or useful it learns, and from which it could read any piece of information it found to be interesting or useful. In this model, the processes reason largely independently of each other, and yet they can capitalize opportunistically on each other's results.

---

88. "Control" refers to issues concerning how a program is organized, how its parts interact, and, in general, how it decides what to do next.

89. *See supra* Section II.B.

This is in fact the approach taken in so called "blackboard" systems.[90] Of course, many more details must be specified before an actual AI and law program could be built using this framework.[91]

In the same spirit, but slightly simpler, is the so-called "agenda-based" model in which, at any given moment, an individual process is in control as it tries to accomplish its assigned task.[92] As the process performs its assignment, it makes note of other tasks it would find necessary, interesting, or useful to be accomplished by another process or itself. As a task is being worked on, other tasks are proposed for consideration.[93] Specifically, the tasks spawned are added to an overall "wish list" of things to do. Since computational resources are limited, this list (the agenda) is ordered according to some standard, such as interest, novelty, or importance. The task with the highest ranking is selected from the agenda and is worked on next. Heuristic rules, like strategies for dealing with near misses, are used to specify what sort of tasks should be proposed and how they should be ordered.[94]

Blackboard and agenda-based systems behave in a dynamic, opportunistic way, sometimes working from one approach, sometimes from another, and deciding which task is best to do as it proceeds. The methods that a system uses to manage the posting and execution of proposed tasks largely define its overall style of reasoning.[95]

Research on hybrid reasoners is currently underway at several sites. Two using an agenda-based approach are the PROLEXS Project at the Vrije Universiteit in Amsterdam in the area of Dutch landlord-tenant law,[96] and the CABARET (for CAse-BAsed REasoning Tool) project of this author's group at the University of Massachusetts in statutory areas,

---

90. *See* Nii, *Blackboard Systems*, in 4 HANDBOOK OF AI, *supra* note 2, at 3. The blackboard model was first used in the 1970's in the HEARSAY II project for understanding continuous speech (this has nothing to do with the hearsay rule in evidence). *See, e.g.*, Erman, Hayes-Roth, Lesser & Reddy, *The HEARSAY-II Speech-Understanding System: Integrating Knowledge To Resolve Uncertainty*, 12 COMPUTING SURVEYS 213 (1980).

91. For example, one would need to determine which "knowledge sources," as the rule-based and case-based associates would be called, have access to what information on the blackboard. For instances, can some access more information than others? What is the relative import of their results?

92. The program AM used an agenda. *See generally* R. DAVIS & D. LENAT, *supra* note 7.

93. For instance, in our benefits rule example, the first task of applying the benefit rule would spawn several near-miss tasks concerning the third antecedent, R3, such as finding cases to show R3 is satisfied or not needed.

94. The overall behavior of agenda-based systems is to work in a "best first" manner. Since the decisions as to what counts as "best" are often based on evaluative heuristics, and at any given moment the tasks on the agenda represent alternatives, agenda-based systems perform "heuristic best first search."

95. The management scheme can be used to bias the system to behave in a certain way. For instance, one can bias the system always to look at rule-based tasks in preference to case-based ones.

96. *See, e.g.*, Oskamp, *Knowledge Representation and Legal Expert Systems*, in ADVANCED TOPICS OF LAW AND INFORMATION TECHNOLOGY 195 (G. Vandenberghe ed. 1989); Oskamp, Walker, Schrickx & van den Berg, *PROLEXS Divide and Rule: A Legal Application*, in ICAIL-89, *supra* note 1, at 54.

such as the home office deduction.[97] There is a rising tide of interest in such systems throughout AI.

CABARET contains full-fledged rule-based and case-based reasoners, each with its own dedicated "monitoring" process, which harvests observations and describes them in terms understandable by the agenda-based controller. The controller, in turn, applies heuristics to these observations to propose and order tasks.[98] For instance, the rule-based monitor would post the observation that a near-miss situation had obtained with regard to antecedent R3 of our benefits example, and this would trigger near-miss heuristics. CABARET currently uses approximately thirty heuristics, grouped into ten or so categories, such as ways to begin reasoning, to check on reasoning, to respond to failures in reasoning, to respond to near misses, to broaden rules and concepts, and to narrow rules and concepts.[99] These were developed through observations of experts as well as from legal sources.[100]

E. *Deep Models of Legal Knowledge*

The goal of research on deep models is to allow an AI program to reason about ingredients of its own legal and general knowledge by representing this knowledge in great detail.[101] Types of knowledge used in deep models include knowledge about temporal and spatial relations, quantity, common sense taxonomies, relations among individuals and beliefs. This topic is often addressed under the rubric of "deep models" and is very much related to fundamental problems concerning common sense reasoning. The motivating assumption is that the richer the representation, both in scope and detail, the better the reasoning. The need to use deeper models of knowledge will ultimately forge a link between the other research

---

97. *See supra* text accompanying notes 85-87; Rissland & Skalak, *Interpreting Statutory Predicates,* in ICAIL-89, *supra* note 1, at 46; Rissland & Skalak, *Combining Case-Based and Rule-Based Reasoning: A Heuristic Approach,* in 1 PROCEEDINGS OF THE ELEVENTH INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE 524 (1989) [hereinafter Rissland & Skalak, *A Heuristic Approach*].

98. For some of the computational details, see Rissland & Skalak, *A Heuristic Approach, supra* note 97.

99. *Id.* at 526.

100. *See, e.g.,* W. TWINING & D. MIERS, HOW TO DO THINGS WITH RULES (2d ed. 1982) (discussing various problems, examples, and approaches for reasoning with statutes and rules).

101. By contrast, in most of the systems discussed in this Comment, perhaps with the exception of Gardner's system, there is no method by which the program may reason about such knowledge explicitly. While questions of time and number come up, for instance, in HYPO's reasoning about disclosure events, *see supra* notes 69, 82, HYPO cannot reason explicitly about time or numeracy as topics in their own right. However, note that since HYPO knows that 10,000 disclosures is a far worse number of disclosures from the plaintiff's point of view than two would be, in a sense HYPO implicitly knows the difference between big and small numbers. One might even say that since, with respect to the dimensions about disclosures, fact situations with the same number of disclosures are treated as being the same (all other things aside), HYPO could be said to know what 2 or 10,000 means. However, numbers and their absolute magnitudes are not topics HYPO can reason about in an explicit way. If HYPO were redesigned so that it used a deep model containing information about numbers it would be able to do so.

discussed in this Comment and that research seeking to develop rich representation schemes for legal knowledge.

Work on representation of legal knowledge is exemplified by the research projects of Thorne McCarty[102] and Tom Gordon.[103] Deep models stand in contrast to situations where one simply uses representation items to encode domain knowledge, but cannot reason further about their meaning. If one were to employ deep knowledge in a system like HYPO, for instance, one would represent in much more detail information about such things as disclosure events, companies, time periods of employment and product development, obligations of employers and employees, and products. For instance, in a deep model an AI researcher could reify a relation such as employer-employee at a deeper level, perhaps in terms of relations of permission or obligation, so that the program could reason about the employer-employee relation as a topic in itself.[104]

III. CONCLUSIONS

Obviously, I cannot do justice to the number and variety of projects in AI and law in this Comment. However, I hope my discussion gives some idea of the richness and variety of current projects.

I foresee the continuation of work on rule-based, case-based, mixed paradigm, hypothetical reasoning, and argument, and predict it will be extended by results concerning deep models of representation, sophisticated control regimes, and machine learning. These lines of research eventually will merge and culminate in an extensive theory of legal argumentation. In time, more and more of the desiderata for AI models of legal reasoning will be incorporated into running AI systems. One can expect some research systems, like HYPO, to have an impact on practical systems, just as Waterman and Peterson's did.[105] What was first a research idea will find its way into practical systems.[106]

---

102. *See, e.g.*, McCarty, *Permissions and Obligations*, 1 PROCEEDINGS OF THE EIGHTH INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE 287 (1983) (using "permissions" and "obligations" to represent certain relations between parties in case); *cf.* Hohfeld, *Some Fundamental Legal Conceptions as Applied in Legal Reasoning*, 23 YALE L.J. 16 (1913) (developing similar relationship-type structures). The need for representing such deontic relationships in his scheme required McCarty to develop a great deal of logical apparatus, which has been the focus of his work for many years. Of late, McCarty has been coming back to some of his primary concerns: development of a comprehensive representational theory for legal knowledge and ultimately, legal argument. *See, e.g.*, McCarty, *A Language for Legal Discourse I. Basic Features*, in ICAIL-89, *supra* note 1, at 180.

103. *See* Gordon, *Issue Spotting in a System for Searching Interpretation Spaces*, in ICAIL-89, *supra* note 1, at 157 (1989); Gordon, *OBLOG-2: A Hybrid Knowledge Representation System for Defeasible Reasoning*, in ICAIL-87, *supra* note 32.

104. Hohfeld, *supra* note 102.

105. *See supra* Section II.A.

106. For instance, I predict that work on case-based reasoning will lead to practical tools for creating and managing case data bases of individual practitioners and firms, which can then be used in preparation of new cases. A beneficial side-effect of such CBR tools, and of course, of traditional expert systems, will be the capturing and preservation of a firm's "institutional memory" and its use to leverage new or inexperienced attorneys in the areas of the firm's expertise to higher levels of

I have tried to show how AI and law researchers are pursuing their twin goals of analytic and practical advances, and how past and ongoing research can be viewed as a coherent attempt to model legal reasoning, particularly argumentation. Even though we may be a long way from some vision of the ideal legal reasoning AI program, and in fact may never be able to achieve certain aspects of human expertise, we can already accomplish very interesting and useful projects. We can use the fine lens of AI to explicate the process of legal reasoning, for instance the creation of case-based arguments; to shed light on questions in legal philosophy, such as the nature of open-textured predicates; and to provide practical, even socially beneficial, applications, such as expert systems in certain administrative areas. The insistence on using computational methods in the AI approach provides a useful discipline for considering longstanding issues in legal reasoning.

Although I have not discussed it here, this body of research can also be fruitfully applied to the law school curriculum. For instance, we can provide our law students with environments in which to examine their own legal knowledge.[107] The conceptual framework of AI can also provide a way to describe our own expertise, such as posing of artful hypotheticals, and show students how they may also possibly acquire it. AI provides a set of tools, based on detailed models of representation and process.

By this discussion, I hope both to encourage those wishing to attempt AI-style projects, and also to reassure those unsure of whether we should. For instance, some might be concerned that the use of AI models will somehow trivialize legal reasoning by making it seem too simple, undermine the importance of lawyers and judges by relegating them to the role of mere users of systems which do all of the interesting reasoning, or dehumanize us by describing intelligent behavior in well-defined terms. I think AI research shows just the opposite: The more we understand human reasoning, the more we marvel at its richness and flexibility, the more questions we ask as we try to understand its workings, and the more we require of a computer program exhibiting intelligence.

Legal reasoning is complex. Our current AI models, albeit too simple, are but steps to more subtle and complete models, and at each step we understand more. There will always be a need for human lawyers and judges. The goal is to assist, not to replace. Demystification of some of the

---

performance, at the very least by keeping them from asking "obvious" questions and making "silly" mistakes.

107.   For example, one can use currently available commercial expert systems shells to allow students to build their own small applications and then to experiment with them. The very exercise of developing a rule base forces students to develop a theory of the application area and to think about general issues regarding the validity and appropriateness of using rule-based approaches. For instance, developing rule sets for consumer tort law or offer-and-acceptance law requires understanding of the specifics of the law as well as general issues about rules and legal predicates and problematic aspects of them. *See supra* Sections II.A-B.

most precious qualities of human intelligence is not to be feared; understanding does not destroy or diminish that which is understood. We should not be afraid to know how we know.

Of course, use of AI techniques, whether in basic research or straightforward applications, requires detailed knowledge of their underlying assumptions, simplifications, strengths, and weaknesses. Successful projects require not only thorough knowledge of both AI and the law, but also a willingness to try new approaches without knowing exactly where they will lead, or whether or not they will achieve exactly the desired result. In short, work in AI and law is no different from work in other fields: you cannot get anywhere without trying; and one invariably learns from trying.

In seeking to understand and model legal reasoning, AI will be challenged and enriched. By engaging in AI endeavors, the law will be challenged and enriched too; it will better understand its own modes of reasonings, including the knowledge and assumptions underlying them, and it will benefit from practical computational tools and models. The relationship between AI and law is a true synergy, the shared specialty of AI and law adding value to both.