Chicago-Kent Law Review

Volume 88
Issue 3 *Justice, Lawyering and Legal Education in the Digital Age*

Article 7

June 2013

Teaching Law and Digital Age Legal Practice with an AI and Law Seminar

Kevin D. Ashley

TEACHING LAW AND DIGITAL AGE LEGAL PRACTICE WITH AN AI AND LAW SEMINAR

KEVIN D. ASHLEY*

INTRODUCTION

A seminar on Artificial Intelligence ("AI") and Law can teach law students lessons about legal reasoning and legal practice in the digital age. AI and Law is a subfield of AI/computer science research that focuses on designing computer programs—computational models—that perform legal reasoning. These computational models are used in building tools to assist in legal practice and pedagogy and in studying legal reasoning in order to contribute to cognitive science and jurisprudence. Today, subject to a number of qualifications, computer programs can reason with legal rules, apply legal precedents, and even argue like a legal advocate.

In a number of law schools, seminars on AI and Law have taught students fundamental lessons about law and legal reasoning, including:

- logical and semantic ambiguities inherent in legal rules;
- the need to reason about legal rules as much as with them;
- the challenge of distinguishing hard from easy legal questions; and
- ways to argue with and about values underlying legal rules.

Researchers in AI and Law have learned these lessons the hard way. As every programmer knows, to get a program to perform a task, one needs to specify a set of steps in detail (or, with machine learning, provide a great many annotated examples). If the program is to transform certain inputs into certain outputs, such as a legal conclusion and an explanation, one needs to specify every step of the reasoning. In the process, researchers need to address all of the issues law students discover, or *should* discover, in a legal education. For example, researchers must account for students' discoveries that the meanings of

* Professor of Law and Intelligent Systems, University of Pittsburgh School of Law; Senior Scientist, Learning Research and Development Center

concepts in a legal rule often are not well specified; that it is frequently unclear what exceptions apply to a legal rule; that the rule's very logic may be ambiguous; that in interpreting how the rule should apply to problems there are useful legal argument schema beyond IRAC (Issue, Rule, Application, and Conclusion); and that some case analogies are more effective than others.

By focusing students on the problems these issues pose for scientists attempting to computationally model legal reasoning, students learn fundamental issues about law and legal reasoning in a new way. As their research papers attest, AI and Law researchers habitually "go meta"; they make processes of legal reasoning explicit that frequently are addressed only implicitly in a legal education. They specify the steps of the process and illustrate them with extended examples. Given cognitive differences between humans and computers, law students will not execute the steps in the same way as computers, but focusing on these processes sensitizes law students to the gaps and pitfalls in legal reasoning that the processes are meant to address. The result is to improve law students' ability to read legal materials carefully, draft legal documents precisely, manage legal risks rationally, and manage information effectively.

The continuing challenge in AI and Law is to connect the computational models to the tasks that attorneys, judges, and law students actually perform—and the texts they actually use—in daily practice. Increasingly, this challenge is being met. Computer programs are assisting immigration agents in the Netherlands to process clients in accordance with voluminous regulatory provisions while providing inputs to regulators about how the rules could be improved based on the performance of the program. From enormous caches of electronic documents produced in modern lawsuits, computer programs are also learning to select and cluster those documents most relevant to litigators' theories of the case based on litigators' labeling small sets of documents as relevant or not to the claims and defenses.

Today, with advances in e-discovery,[1] legal information retrieval, semantic processing of web-based information for electronic contracting, computational models of legal argument, *et cetera*, the chances are

[1] Pretrial discovery in lawsuits involves processing parties' requests for and access to materials in the hands of opponents and others to reveal facts and develop evidence for trial. *E-discovery* involves the collecting, exchanging, and analyzing of electronically stored information in pretrial discovery. Large lawsuits routinely involve millions of e-documents challenging the traditional approach of using keyword searches to retrieve documents and then manually reviewing them for relevance.

increasing that law students will use, and even depend on, such systems in their practices. Law students are increasingly computer savvy. They have been connected to the Internet from a tender age; they have witnessed IBM's Jeopardy!-winning Watson program as it knifed through millions of pages of text instantaneously and answered questions quicker and more accurately than the best human contestants. Today's law students are prepared to believe that intelligent computers can play a more central roll in legal practice. The AI and Law seminar can channel this interest, for example, by getting them to think through how a program like Watson might answer legal questions and explain its answers. The seminar encourages law students to think about processes of legal reasoning and legal practice and about how those processes employ information. As explained below, given developments in the markets for legal services and legal employment, it is useful for law students to think in terms of process engineering. The seminar also teaches students something of how the new digital documents technologies work, what they can and cannot do, how to measure performance, how to evaluate claims about the technologies, and how to be savvy consumers and users of the technologies.

This article provides a guide and examples to prepare law students for the digital age by means of an AI and Law seminar. After introducing the science of Artificial Intelligence and its application to law, the paper presents the syllabus for an up-to-date AI and Law seminar. With the Syllabus as a framework, the paper showcases some characteristic AI and Law programs, and illustrates the pedagogically important lessons that AI and Law has learned about reasoning with legal rules and cases, about legal argument and about the digital documents technologies that are becoming available, and even the norm, in legal practice.

I. INTRODUCING AI AND LAW

As noted, AI is a branch of computer science research that focuses on computationally modeling behavior commonly regarded as intelligent when performed by humans.[2] A computational model of intelligent behavior is a computer program that performs or simulates the behavior.

AI and Law researchers build computational models of legal reasoning, which are computer programs that perform or simulate legal

[2] SEMANTIC INFORMATION PROCESSING (Marvin Minsky, ed., 1968).

reasoning. If the inputs to the program are a description of a legal problem, its outputs might include a solution to the legal problem and an explanation. Legal problem-solving frequently requires reasoned explanations in addition to a decision or prediction. Indeed, the argument supporting a decision may be more important than the decision; otherwise a human user cannot judge whether to rely on the advice. The focus on explanations and arguments is a key area where AI and Law contributes to AI research.

Like computer programs generally, a computational model can be described in terms of the inputs to the program, its outputs, and the intervening steps that transform the former into the latter. The intervening set of steps (or algorithm) transforms the inputted problem description into the outputs. In specifying an AI program's input/output (I/O) behavior, three questions are especially relevant:

1. Knowledge representation and search: How are the inputs, and the information used to analyze them, represented in a problem space that can be searched systematically?
2. Inference control mechanism: What governs the search for solutions in terms of efficiency and relevance?
3. Learning: In order to improve its performance, how can the program learn from its errors and successes and from other sources of information?[3]

In the case of AI and Law programs, only a few accept problems inputted as natural language texts or reason with cases or legal rules expressed in text. More frequently, the problems are presented in a specialized representation, for instance, as answers to questions generated by an expert system[4] or as a list of dimensions factors that appear in the problem, like stereotypical patterns of facts that strengthen or weaken a side's claim.[5] Similarly, the rules in the program's knowledge base may be represented in terms of logical expressions and the cases, like the problem, may also be represented in terms of factors. In discussing some representative AI and Law programs, the focus will be on the ways in which they search systematically for solu-

[3] NEIL STILLINGS, STEVEN E. WEISLER, CHRISTOPHER H. CHASE, MARK H. FEINSTEIN, JAY L. GARFIELD, EDWINA L. RISSLAND, NEIL A. STILLINGS & STEVEN W. WEISLER, COGNITIVE SCIENCE: AN INTRODUCTION 140-42, 177, 192 (2d ed. 1995).

[4] DONALD A. WATERMAN & MARK A. PETERSON, MODELS OF LEGAL DECISIONMAKING 14 (1981) ("'Expert system[s] [are] computer programs that embody expertise and knowledge supplied by human experts and that use artificial intelligence techniques to provide inferences for [users].").

[5] Kevin D. Ashley, *Reasoning with Cases and Hypotheticals in HYPO*, 34 INT'L J. OF MAN-MACH. STUD. 753, 763-64 (1991).

tions and can learn, for example, how to detect factors in the texts of cases or to classify the texts of statutory provisions.

In theory, and often in practice, an AI or AI and Law program's problem-solving behavior can be assessed in terms of relevance and performance measures similar to those applied to human problem-solving (such as predictive accuracy, coverage, precision, recall, and explanatory adequacy of the outputs given the inputs). In addition, an AI program may be subjected to a modified Turing test in which a human judge engages in an interaction with a source, asking it questions and receiving responses. The judge, blinded as to the nature of the source, whether human or machine, tries to determine which it is.[6]

An AI and Law program's algorithm, the intervening set of steps by which it transforms an inputted legal problem description into outputs of an explained solution, may be especially interesting from a legal instructional viewpoint. In breaking the legal reasoning process into a set of steps, the researcher makes explicit some aspects that are often left implicit or unexplained in pedagogical or jurisprudential accounts. Examples of the program's I/O behavior illustrate the legal reasoning model in operation, including the steps as applied in the problem context. The examples highlight both the program's successes, where the outputs conform to the kind of intelligent behavior we expect from attorneys, and its failures, where the outputs do not so conform. A researcher uses the trace of this stepwise process to frame the question of how to improve the model's (legal) performance, extending the envelope of successes. The I/O examples illustrating the reasoning steps, assumptions, successes and failures, present law students with a window on the process of legal reasoning.

II. TEACHING AI AND LAW TO LAW STUDENTS

The steps in AI and Law programs that transform inputted legal problems into outputted explained solutions are a key reason why an AI and Law seminar has pedagogical value. They present law students with an opportunity to consider if, how, and how well human legal reasoning deals with the same issues.

In observing the decomposition of the legal reasoning process into a set of steps, students become aware of some gaps in the reasoning process that need to be—and are—"filled" perhaps by some *ad hoc* means, the details of which are left implicit, unexplained, and some-

[6] PAMELA MCCORDUCK, MACHINES WHO THINK 70 (2d ed. 2004).

times unjustified. Examples of such gaps include distinguishing hard from easy questions of law, determining what similarities and differences among cases are legally relevant, resolving conflicting rules, and making arguments from values underlying a rule. AI and Law researchers have not filled those gaps in a sense that might satisfy a legal or jurisprudential scholar, but they have focused on the problem of filling the gaps and engineered processes to address them well enough for some purposes. It is worth law students' time and effort to see how these gaps have been filled.

Even though AI and Law research involves computational models, generally non-technical law students with no experience of computer programming can still improve their understanding of law by studying the models and examples. The algorithms are described at a high level of abstraction as either flow charts or textually described sets of steps.

The examples of the algorithms in operation are intuitively accessible examples of legal reasoning from which law students can learn. As noted, the I/O examples of automated legal reasoning programs illustrate both successes and failures. The failures are more interesting; students can observe how human intelligence fills in a gap and ponder the consequences for law and for AI and Law. As a thought experiment, law students can use the trace of the stepwise process (such as the explanation of the extended example) to participate in framing the question of how to improve the program's (legal) performance, and thus design a better computational model of that kind of legal reasoning.

Of course, it requires technical expertise to understand in detail how a computational model's knowledge is represented (for example, with logical formalisms) and how the search, inference control and learning are implemented. Fortunately, law students do not need to understand the technical details in order to learn from the examples of legal reasoning as modeled.

As asserted above, a second key reason why an AI and Law seminar has pedagogical value is to help law students understand the technology of modern legal practice. Since attorneys increasingly are called upon to assess, purchase, and rely upon products that employ machine learning,[7] data mining,[8] natural language processing, and information

[7] Machine Learning is the study of computer algorithms that improve automatically through experience. TOM M. MITCHELL, MACHINE LEARNING 1-2 (1997).

[8] *Id.* at 17 ("Machine learning algorithms . . . are especially useful in . . . *data mining* problems where large databases may contain valuable implicit regularities that can be discovered

retrieval for use (for instance in e-discovery), it is valuable for non-programming law students to learn how these tools are employed and something about how they work. E-discovery tools will soon be not only permitted, but required by the professional standard of care and demanded by large corporate clients.

Although techniques such as machine learning from large data sets, data mining, or natural language processing are complex and leave legal intuitions behind, attorneys increasingly will have to select tools, use them or direct others in using them, interpret their outputs, and justify to others (such as judges) their reliability and effectiveness. For these kinds of models and software products, law students can at least learn the conceptual vocabulary, high-level descriptions of how the techniques work, the ways in which performance can be measured, what the measures signify, what the models capture and what the models leave out. The experience of thinking in terms of (1) high-level descriptions of computational algorithms; (2) examples of where the algorithms work and where they break down; and (3) the ways their performance is measured will prepare students to be informed consumers of digital tools aimed at the legal profession.

During the last twenty-five years, AI and Law seminars have been taught regularly, if not continually, in law schools at Harvard, Stanford, Northeastern, Chicago-Kent, and Pittsburgh.[9] Typically, the seminars focus students on accessible research papers that have been published in the field's major venues, such as the International Conference on Artificial Intelligence and Law (ICAIL),[10] the annual conference of Legal Knowledge and Information Systems (JURIX),[11] and the journal *Artificial Intelligence and Law*.[12] The papers illustrate the I/O examples of legal reasoning and the intervening steps. They also identify a gap that often relates to a significant legal/educational challenge, one that human reasoners easily bridge, finesse or ignore, but that computational reasoners need special techniques to address.

automatically (e.g., to analyze outcomes of medical treatments from patient databases or to learn general rules for credit worthiness from financial databases) . . . [.]") (emphasis added).

[9] *See infra* notes 13, 17, 23, 24. Kevin Ashley has taught an AI and Law Seminar at the University of Pittsburgh School of Law since 1989.

[10] INTERNATIONAL ASSOCIATION FOR ARTIFICIAL INTELLIGENCE AND LAW, http://www.iaail.org (last visited Apr. 23, 2013).

[11] JURIX, THE FOUNDATION FOR LEGAL KNOWLEDGE BASED SYSTEMS, http://www.jurix.nl (last visited Apr. 20, 2013).

[12] *See* ARTIFICIAL INTELLIGENCE & L., *available at*
www.springer.com/computer/aijournal/10506.

Sometimes, the AI and Law Seminar combines law students and computer science graduate students who study AI. This provides law students a pedagogically valuable opportunity to practice explaining the law to non-law students (and for graduate students to explain computer science to non-techies). Law students also gain experience interacting with technically- and technologically-minded persons, whom many law students may encounter in practice, either among clients or vendors.

An AI and Law Practicum can take law students a few steps further, engaging them in some aspects of building, applying, or using computational models of or for legal reasoning. AI and Law Practicums have been offered in law schools at Stanford, Northeastern, Chicago-Kent,[13] and Pittsburgh, offering law students opportunities to develop expert systems using software development tools for nonprogrammers. The practicum activities can be integrated with the AI and Law Seminar as a kind of "lab" component or follow it in the next semester. Students with or without programming skills can actually build components of legal expert systems, and in doing so, they will they grapple with bridging the gaps. Even if they fail to implement a practical bridge, they learn about the gap.

Since much of computer programming involves a design process of successive refinement, by identifying goals and sub-goals and describing a set of steps for achieving these goals at each level, law students can engage in the higher-level design tasks even without knowing a programming language. Law students who can think systematically about processes for solving legal programs can engage in a stepwise decomposition of the problem and then focus on "inventing" a technique (such as a set of steps) for solving some sub-goal. One does not need to program the steps to learn from the process of describing them and illustrating them with an example.

Actual programming in computer code happens only at the end of the process, and, in any event, it may not require special skills. Increasingly, tools are available with which non-programming law students can build solutions for real legal problems to implement their step-

[13] At Chicago-Kent, Professor Richard Wright taught a course on "Computers and Legal Reasoning" using SAGE: A Pedagogical Expert System Development Tool, a suite of computer programs of his own design, which allowed the rapid, hypertext-driven building of integrated rule bases and case bases to represent legal knowledge and model legal reasoning and, in particular, legal domains. Professor Ron Staudt teaches a "Justice and Technology Practicum" that engages students in building web-based tools, including expert systems, to support legal services advocates, pro bono volunteers, and pro se litigants.

wise expert system components. Recently, in a Georgetown Law School seminar entitled "Technology, Innovation, and Law Practice," law students built legal expert systems in a variety of domains using tools from Neota Logic that are also used by law firms to create expert systems for professional practice.[14]

Law students without programming skills can also engage in designing and applying schemes for annotating legal texts (including with argument-diagramming tools described below[15]) so that they can be used by computational models of legal reasoning or in applying machine learning to classify texts according to their legal significance. For example, students at Hofstra Law School's Law, Logic, and Technology Research Laboratory learn about legal reasoning as they participate in empirical research involving close study and annotation of evidential reasoning in legal opinions.[16]

III. AN AI AND LAW SEMINAR SYLLABUS

A Syllabus for an AI and Law Seminar that covers both key pedagogical points—learning lessons from stepwise decompositions of legal reasoning and understanding the new technology of legal practice—is presented in the Appendix and described below.

A. Part One: Computational Models of Reasoning with Legal Rules and Cases

The first section of Part One of the Syllabus, *Introduction to Computational Models of Legal Reasoning*, includes an informative survey and two early examples of computational models of legal problem-solving tasks of particular relevance to law students, who have all taken first year torts and contracts courses. The survey, by Edwina Rissland, introduces law students to Artificial Intelligence, AI's application to the legal domain, and computational models of legal reasoning.[17] The first example of such a model is Donald Waterman's expert

[14] Press Release, Neota Logic, Inc., Georgetown Law Students Challenge Tradition by Building Online Legal Advisors with Neota Logic (Apr. 24, 2012) *available at*, http://www.prweb.com/releases/neotalogic/irontechlawyer/prweb9438690.htm.

[15] *See infra* note 40.

[16] LAW, LOGIC & TECHNOLOGY RESEARCH LABORATORY, http://www.lltlab.org (last visited Apr. 23, 2013).

[17] Edwina L. Rissland, *Artificial Intelligence and Law: Stepping Stones to a Model of Legal Reasoning*, 99 YALE L. J. 1957, 1957-58 (1990). The author, University of Massachusetts Professor Emeritus of Computer Science, has regularly taught an AI and Law Seminar at Harvard Law School. In addition to assigning readings, she strongly suggested that law students (and her grad-

system that used heuristic rules (i.e., rules of thumb, based on expert experience) to advise on settling product liability claims.[18] The second sample model by Anne Gardner analyzes exam problems law students encounter in a first year contracts course.[19]

Each of these examples illustrates a computational machinery for drawing legal conclusions. Rissland's survey focuses on the Waterman and Gardner examples as well and relates them to the development of case-based models of legal reasoning described below.

Waterman's program contained three kinds of rules derived from legal authorities, litigators, and insurance claims adjustors: rules defining legal concepts like product liability, strict liability, and comparative negligence; informal rules to address indefinite legal terms and account for likely juror reactions to appearance and sympathy; and rules guiding settlement computations (for example, for computing the amount of damages for pain and suffering). The system applied the rules in a bottom-up direction; it repeatedly asked the user questions based on the rules whose conditions had not yet been satisfied and then, given answers, used the applicable rules to draw a conclusion about how much the dispute was worth for purposes of settlement. By way of explaining its conclusion, the program could output an inference tree, a tree of the rules that "fired" (that is, whose conditions were satisfied) resulting in the conclusion.

Gardner's program implemented a model of offer and acceptance as an ordered progression among four intermediate conclusions or states: (0) no relevant legal relations, (1) a pending offer, (2) the existence of a contract, and (12) the existence of a contract plus a pending proposal to modify the contract. Arcs represented the ways to move from one state to another. For instance, to move: from (0) to (1) required finding an offer, from (1) to (2) required finding an acceptance of the pending offer, and from (2) to (0) required revoking the acceptance and rejecting the offer. There were rules associated with each

uate students, of which I was one, an opportunity for which I am forever grateful) prepare a one-page summary of readings with the following major entries: 1) a brief (i.e., no more than two- or three-sentence) description of what the work is about, 2) the strengths of the approach, 3) the weaknesses of the approach, and 4) something about the approach that is relevant to some project, task, assignment, or opinion of the student. I have always found it helpful to prepare such summaries of readings, and I ask my students to do the same.

[18] WATERMAN & PETERSON, *supra* note 4. Don Waterman was an AI pioneer whose dissertation project modeled betting in Poker.

[19] ANNE VON DER LIETH GARDNER, AN ARTIFICIAL INTELLIGENCE APPROACH TO LEGAL REASONING 4 (1987). Anne Gardner is a graduate of both Stanford University's Law School and its doctoral program in Computer Science.

arc which defined the conditions and determined whether a condition had been satisfied (that is, whether the facts disclosed an offer, an acceptance, a revocation, *et cetera*).

The *Logical Models of Statutory Reasoning* section includes an early program by Marek Sergot, *et alia*, that analyzed legal questions involving British citizenship; it still represents one of the more ambitious efforts at making a statute "computable." This program, comprising 150 rules represented (manually) from the provisions of the British Nationality Act, could analyze problems involving questions of nationality and explain its answers using a trace of the logical inferences it had drawn.[20]

A practical problem arises in making large statutes computable: the formal representations of the statutory rules have a tendency to become disconnected from the statutory texts. This problem complicates explaining the system's results and maintaining a large system where the rules need to be updated as the statutory texts are amended. Trevor Bench-Capon and Franz Coenen developed an approach to coordinating the representations of statutory rules so that they are isomorphic to the statutory code's structure.[21]

Other readings in this section warn of problems when interpreting statutory texts as logical rules. One problem is the need to deal with logical ambiguity, caused by the indeterminate scopes of logical connectors in natural language texts.[22] Another problem stems from the fact that classical logic is monotonic; once a conclusion is drawn, it cannot be "taken back" even in the light of new information. Thus, it does not fit legal reasoning very well, where legal conclusions typically can be, and are, argued one way and the other.[23]

[20] Marek J. Sergot, Fariba Sadri, Robert A. Kowalski, Frank Kriwaczek, P. Hammond & H. T. Cory, *The British Nationality Act as a Logic Program*, 29 COMM. OF THE ACM 370 (1986). Marek Sergot is a Professor of Computational Logic at Imperial College, London.

[21] Trevor Bench-Capon & Franz Coenen, Exploiting Isomorphism: *Development of a KBS to Support British Coal Insurance Claims*, 1991 PROC. OF THE THIRD INT'L CONF. OF ARTIFICIAL INTELLIGENCE & L. 62.

[22] *See* Layman E. Allen & C. Rudy Engholm, *Normalized Legal Drafting and the Query Method*, 29 J. LEGAL EDUC. 380 (1977-1978). Layman Allen, a Professor Emeritus of Law at the University of Michigan, developed a series of games about logic, mathematics, and law, including WFF 'N PROOF, EQUATIONS, and THE LEGAL ARGUMENT GAME OF LEGAL RELATIONS.

[23] Donald Berman & Carole Hafner, *Obstacles to the Development of Logic-Based Models of Legal Reasoning*, *in* COMPUTER POWER AND LEGAL LANGUAGE 183-214 (Charles Walter ed., 1986). Donald Berman, a former Professor of Law and co-director of the Center for Law and Computer Science and the other co-director, Associate Professor of Computer Science Carole Hafner, regularly taught an AI and Law seminar at Northeastern University, Boston, MA.

The *Case-Based Models of Legal Reasoning* and *Models for Predicting Legal Outcomes* sections include techniques to model alternatives to purely rule-based, deductive models of legal reasoning. These include models for representing legal concepts more flexibly in terms of definitional prototypes *and* argument-driven deformations,[24] and in terms of dimensions that represent and partially order stereotypical fact patterns that strengthen or weaken a claim.[25] A third model, GREBE, employs semantic networks representing facts that judges deemed important (or *criterial*) in explaining their holdings.[26]

These programs used cases to generate arguments for and against proposed conclusions (such as HYPO's three-ply arguments[27]), a significant departure from the reasoning of the logical deductive models.

The case-based models also focused the field on the need to develop new techniques to evaluate the computational models empirically, such as a kind of Turing test of GREBE. L. Karl Branting employed a human expert to grade arguments generated by law students or by the program. The grader was blinded as to the sources of the arguments (that is, he was not warned that any argument was generated by a computer), and the program's argument was made to look like the law students' papers.[28]

The case-based models inspired factor-based approaches, CATO[29] and IBP[30], which predict outcomes of legal cases based on the strength of the pro and con arguments. This revived interest in an objective criterion for evaluating models and predictive accuracy, a recurrent thread in AI and Law since early researchers applied a nearest-neighbor approach to predicting outcomes of Canadian tax cases in-

[24] *See* L. Thorne McCarty, *An Implementation of* Eisner v. Macomber, 1995 PROC. OF THE FIFTH INT'L CONF. ON ARTIFICIAL INTELLIGENCE & L. 276. A Professor of Computer Science and Law at Rutgers University, Thorne McCarty has taught a seminar on Artificial Intelligence and Law there and as a visitor at Stanford Law School.

[25] Ashley, *supra* note 5, at 763-64.

[26] L. Karl Branting, *Building Explanations from Rules and Structured Cases*, 34 INT'L J. OF MAN-MACH. STUD. 797 (1991).

[27] Ashley, *supra* note 5, at 764.

[28] Branting, *supra* note 26.

[29] *See* Vincent Aleven, *Using Background Knowledge in Case-Based Legal Reasoning: A Computational Model and an Intelligent Learning Environment*, 150 ARTIFICIAL INTELLIGENCE 183 (2003).

[30] *See* Kevin D. Ashley & Stefanie Brüninghaus, *Computer Models for Legal Prediction*, 46 JURIMETRICS 309 (2006).

volving capital gains issues.[31] In addition to predicting, however, CATO and IBP added an ability to explain the predictions with arguments.

The section entitled *Models Integrating Cases, Statutes, Rules, Concepts, and Values* focuses on other improvements to case-based computational models. The GREBE program integrated case-based and rule-based reasoning, indexing semantic networks of criterial facts in past cases with frequently litigated statutory concepts in workmen's compensation law.[32]

Another approach to integrating cases and rules, CABARET, combined logical representations of an IRS provision governing home office deductions and dimensions with the provision's open-textured statutory terms. The dimensions indexed positive and negative case examples of the terms.[33] Notably, CABARET employed an agenda mechanism to switch between rule-based and case-based analyses, generating arguments for and against the taxpayer's position.

In 1993, an influential critique changed the course of AI and Law research on case-based computational models.[34] Donald H. Berman and Carol Hafner pointed out that the models lacked a teleological component: the underlying purposes or values served by legal statutes and rules—which are so much a focus of human legal interpretation—were missing. The criticism led to a decade of work, resulting in Bench-Capon and Sartor's landmark model of legal reasoning with cases that incorporated values in theories of value preferences induced from precedents.[35] Another approach focused on how case decisions extensionally define principles in professional ethics. It represented cases with a more generalized version of semantic networks of criterial facts in ethics decisions by "judges'" (actually engineering ethics experts).[36] In an experiment, Bruce M. McLaren empirically demonstrated the contributions to his SIROCCO program's retrieval effectiveness as cases filled in the meanings of the abstract ethical rules.

[31] *See* Ejan Mackaay & Pierre Robillard, *Predicting Judicial Decisions: The Nearest Neighbour Rule and Visual Representation of Case Patterns*, 3 DATENVERARBEITUNG IM RECHT 302 (1974).

[32] Branting, *supra* note 26.

[33] Edwina L. Rissland & David B. Skalak, *CABARET: Statutory Interpretation in a Hybrid Architecture*, 34 INT'L J. OF MAN-MACH. STUD. 839 (1991).

[34] Donald H. Berman & Carol Hafner, *Representing Teleological Structure in Case-Based Legal Reasoning: The Missing Link*, 1993 PROC. OF THE FOURTH INT'L CONF. ON ARTIFICIAL INTELLIGENCE & L. 50.

[35] Trevor Bench-Capon & Giovanni Sartor, *A Model of Legal Reasoning with Cases Incorporating Theories and Values*, 150 ARTIFICIAL INTELLIGENCE 97 (2003).

[36] *See* Bruce M. McLaren, *Extensionally Defining Principles and Cases in Ethics: An AI Model*, 150 ARTIFICIAL INTELLIGENCE 145 (2003).

B. Part Two: Defeasible Legal Reasoning with Argument Schemes

Part Two introduces the current focus of AI and Law research, the development of a unified computational framework for non-monotonic, defeasible legal reasoning using argument schema that incorporate and recast the logico-deductive, case-based, and value-based approaches to assessing legal claims described above. Non-monotonic or defeasible inference models *can* take back previously inferred consequences in light of new information, an important feature of legal reasoning.

The section on *Assessing Legal Claims with Argument Schema* presents outlines for the unified frameworks by Katie Atkinson and Trevor Bench-Capon[37] and by Thomas F. Gordon and Douglas Walton[38] This work also includes modeling evidentiary arguments about legal claims and standards of proof.

Matthias Grabmair and the author developed argument schema for case comparison with value judgments, intermediate legal concepts, and hypothetical cases as employed in U.S. Supreme Court oral arguments.[39] Vern R. Walker's model connects examples of evidentiary reasoning in cases to decision trees that model statutory requirements.[40]

As argument schema improve and extend the kinds of legal arguments that programs can model, they increase the need for better ways to represent legal fact situations and concepts. The need for flexible conceptual schema, raised by L. Thorne McCarty[41] and Rissland,[42] has become even more acute. As the author demonstrated empirically,[43] intermediate legal concepts play an important role in predicting legal

[37] Katie Atkinson & Trevor Bench-Capon, *Argumentation and Standards of Proof*, 2007 PROC. OF THE ELEVENTH INT'L CONF. ON ARTIFICIAL INTELLIGENCE & L. 107.

[38] Thomas F. Gordon & Douglas Walton, *Legal Reasoning with Argumentation Schemes*, 2009 PROC. OF THE TWELFTH INT'L CONF. ON ARTIFICIAL INTELLIGENCE & L. 137.

[39] Matthias Grabmair & Kevin D. Ashley, *Facilitating Case Comparison Using Value Judgments and Intermediate Legal Concepts*, 2011 PROC. OF THE THIRTEENTH INT'L CONF. ON ARTIFICIAL INTELLIGENCE & L. 161; Matthias Grabmair & Kevin D. Ashley, *Argumentation with Value Judgments - An Example of Hypothetical Reasoning*, 2010 PROC. OF THE 23RD ANN. CONF. ON LEGAL KNOWLEDGE & INFO. SYS. 67.

[40] Vern R. Walker, Nathaniel Carie, Courtney C. DeWitt & Eric Lesh, *A Framework for the Extraction and Modeling of Fact-Finding Reasoning from Legal Decisions: Lessons from the Vaccine/Injury Project Corpus*, 19 ARTIFICIAL INTELLIGENCE & L. 291, 296-98 (2011).

[41] *See* McCarty, supra note 24.

[42] Rissland, *supra* note 17; Rissland & Skalak, *supra* note 33, at 839-40.

[43] Kevin D. Ashley & Stefanie Brüninghaus, *A Predictive Role for Intermediate Legal Concepts*, 2003 PROC. OF THE SIXTEENTH ANN. CONF. ON LEGAL KNOWLEDGE & INFO. SYS. 153, 155.

outcomes. Changes in their meanings over time can be monitored,[44] and argument schema invite the modeling of ever more detailed argumentation about the meanings of particular intermediate legal concepts.[45]

Ontologies—systematic and explicit specifications of domain concepts used in legal rules and fact situations—help a computational model to reason flexibly with legal concepts. Ontologies also are key for someday transitioning from computational models that reason with cases represented in formalisms to cases represented simply as texts. The section on *Representing Legal Concepts and Case Knowledge in Ontologies* presents foundational work on legal ontologies;[46] some ontological requirements for supporting analogical, teleological, and hypothetical legal reasoning using argument schema;[47] and up-to-date, semi-automated means for constructing legal ontologies from legal texts.[48]

C. Part Three: Legal Information Retrieval, Information Extraction, and Text Processing

Part Three's section on *Explaining How Full-Text Legal Information Retrieval Works* introduces students to practicalities of legal information retrieval ("IR") (as explained more fully in Section V), including the use of inverted indices, the role of term frequency in assessing relevance, probabilistic models of retrieval[49] using Bayesian networks,[50]

[44] Edwina L. Rissland & M. Timur Friedman, *Detecting Change in Legal Concepts*, 1995 PROC. OF THE FIFTH INT'L CONF. ON ARTIFICIAL INTELLIGENCE & L. 127.

[45] Grabmair & Ashley, *supra* note 39, at 164-66.

[46] Trevor J.M. Bench-Capon & Peppin R.S. Visser, *Ontologies in Legal Information Systems; The Need for Explicit Specifications of Domain Conceptualisations*, 1997 PROC. OF THE SIXTH INT'L CONF. ON ARTIFICIAL INTELLIGENCE & L. 132; Joost Breuker, Andre Valente & Radboud Winkels, *Legal Ontologies in Knowledge Engineering and Information Management*, 12 ARTIFICIAL INTELLIGENCE & L. 241 (2004).

[47] *See* Kevin D. Ashley, *Ontological Requirements for Analogical, Teleological, and Hypothetical Legal Reasoning*, 2009 PROC. OF THE TWELFTH INT'L CONF. ON ARTIFICIAL INTELLIGENCE & L. 1.

[48] Enrico Francesconi, Simonetta Montemagni, Wim Peters & Daniela Tiscornia, *Integrating a Bottom-Up and Top-Down Methodology for Building Semantic Resources for the Multilingual Legal Domain, in* SEMANTIC PROCESSING OF LEGAL TEXTS: WHERE THE LANGUAGE OF LAW MEETS THE LAW OF LANGUAGE 95 (Enrico Francesconi, Simonetta Montemagni, Wim Peters & Daniela Tiscornia eds., 2010).

[49] Howard Turtle, *Text Retrieval in the Legal World*, 3 ARTIFICIAL INTELLIGENCE & L. 5, 5-6 (1995).

[50] *See* Eugene Charniak, *Bayesian Networks Without Tears*, 12 AI MAGAZINE, Winter 1991, at 50.

methods for evaluating legal IR and the applicable measures, precision and recall,[51] and methods for extracting information from legal texts.[52]

The section *Combining Legal IR and AI to Analyze Legal Claims* examines the potentially complementary advantages and disadvantages of full-text legal IR and AI and Law models. The section also explores methods of combining them, including using case-based models automatically to find legally relevant passages in case opinions in the SPIRE program,[53] using legal ontologies to improve retrieval effectiveness,[54] extracting information automatically from case law,[55] and supporting alternative legal research paradigms, such as a heuristically guided search through networks of legal information. In SCALIR, the network comprised legal cases are linked via shared terms or citations.[56] In BankXX, the network of legal knowledge connected annotated nodes representing cases as sets of factors and bundles of citations, and representing exemplars of prototypical stories and legal theories as bundles of factors.[57]

The work in the section on *Legal Information Extraction and Text Processing* connects computational models of legal reasoning and legal texts. The SMILE+IBP program classifies textually described cases in terms of applicable factors and then predicts and explains their outcomes using IBP.[58] Other programs automatically classify statutory texts in terms of major types, abstract categories, and subject matters (like "Administrative Law" or "Intellectual Property Law") or norm types (such as definition, permission, or obligation) and then extracts information such as norm features (like duty, duty bearer, action, or object of action) and regulatory functions (that is, *functional* infor-

[51] David C. Blair & M. E. Maron, *An Evaluation of Retrieval Effectiveness for a Full-Text Document-Retrieval System*, 28 COMM. OF THE ACM 289 (1985).

[52] Peter Jackson, Khalid Al-Kofahi, Alex Tyrrell & Arun Vacher, *Information Extraction from Case Law and Retrieval of Prior Cases*, 150 ARTIFICIAL INTELLIGENCE 239 (2003).

[53] Jody J. Daniels & Edwina L. Rissland, *Finding Legally Relevant Passages in Case Opinions*, 1997 PROC. OF THE SIXTH INT'L CONF. ON ARTIFICIAL INTELLIGENCE & L. 39.

[54] M. Saravanan, B. Ravindran & S. Raman, *Improving Legal Information Retrieval Using an Ontological Framework*, 17 ARTIFICIAL INTELLIGENCE & L. 101, 103 (2009).

[55] Jackson et al., *supra* note 52.

[56] Daniel E. Rose & Richard K. Belew, *A Connectionist and Symbolic Hybrid for Improving Legal Research*, 35 INT'L J. OF MAN-MACH. STUD. 1 (1991).

[57] Edwina Rissland, David Skalak & Timur Friedman, *BankXX: Supporting Legal Arguments through Heuristic Retrieval*, 4 ARTIFICIAL INTELLIGENCE & L. 1, 5-8 (1996).

[58] Kevin D. Ashley & Stefanie Brüninghaus, *Automatically Classifying Case Texts and Predicting Outcomes*, 17 ARTIFICIAL INTELLIGENCE & L. 125, 139-40 (2009).

mation about what entities and events a statute governs as well as the purpose of the statute and its constraints).[59]

In previous AI and Law work on legal information extraction and text processing, researchers could assume that the legal texts had a fairly homogeneous structure and conformed to certain constraints about content. However, those assumptions do not apply in e-discovery, in which parties to lawsuits must analyze and produce enormous numbers of documents relevant to the allegations of the complaint and answer. The work in the section *AI and Law Tools for E-Discovery: Obtaining Evidence Relevant to Lawyers' Hypotheses about Claims and Documents* addresses these challenges.[60] Given the high volumes of documents, this is a domain where information retrieval tools necessarily take the lead,[61] but where statistical techniques and knowledge-based models of legal claims and hypotheses can complement one another.[62] AI tools may play a valuable role, such as in filtering documents using social network techniques,[63] creating optimal seed sets of relevant documents for predictive coding (that is, machine learning-based classification of unseen documents[64]), and supporting attorneys in exploring documents using innovative interfaces.[65]

D. Part Four: The Future of AI and Law: Bridging Computational Models and Legal Texts

Part Four of the Syllabus turns to some future AI and Law work that promises to bridge the gap between computational models of legal reasoning and legal texts. The section *Near Term Developments in AI and Law* presents some developments one can reasonably anticipate

[59] Enrico Francesconi & Andrea Passerini, *Automatic Classification of Provisions in Legislative Texts*, 15 ARTIFICIAL INTELLIGENCE & L. 1, 2-3 (2007); Emile de Maat, Kai Krabben & Radboud Winkels, *Machine Learning Versus Knowledge Based Classification of Legal Texts*, 2010 PROC. OF THE TWENTY-THIRD ANN. CONF. LEGAL KNOWLEDGE & INFO. SYS. 87.

[60] *See* Kevin Ashley & Will Bridewell, *Emerging AI & Law Approaches to Automating Analysis and Retrieval of Electronically Stored Information in Discovery Proceedings*, 18 ARTIFICIAL INTELLIGENCE & L. 311 (2010).

[61] Douglas W. Oard, Jason R. Baron, Bruce Hedin, David D. Lewis & Stephen Tomlinson, *Evaluation of Information Retrieval for E-discovery*, 18 ARTIFICIAL INTELLIGENCE & L. 347 (2010).

[62] David D. Lewis, *Afterword: Data, Knowledge, and E-discovery*, 18 ARTIFICIAL INTELLIGENCE & L. 481 (2010).

[63] Hans Henseler, *Network-based Filtering for Large Email Collections in E-Discovery*, 18 ARTIFICIAL INTELLIGENCE & L. 413 (2010).

[64] Christopher Hogan, Robert S. Bauer & Daniel Brassil, *Automation of Legal Sensemaking in E-discovery*, 18 ARTIFICIAL INTELLIGENCE & L. 431 (2010).

[65] Caroline Privault, Jacki O'Neill, Victor Ciriza & Jean-Michel Renders, *A New Tangible User Interface for Machine Learning Document Review*, 18 ARTIFICIAL INTELLIGENCE & L. 459 (2010).

soon. The DeepQA approach of the Jeopardy™-winning IBM Watson program[66] will likely have a profound effect on how to extract relevant information automatically from legal decision texts. Using DeepQA natural language processing tools to annotate argument-schematic information in legal decision texts, a program could conceivably screen new texts that provide evidence of legal compliance.[67] Bayesian networks will be integrated with argument schema so that computational models of legal argument can account for factual, normative, moral, and empirical uncertainties[68] more systematically than in Waterman's settlement-guiding program.[69] Defeasible logical models of regulatory requirements will help ensure that business processes are designed to comply with the law[70] and to automate web-based agents in entering into electronic contracts.[71] The business process approach will continue to be extended to administrative agencies. In the INDiGO project of the Dutch immigration and naturalization service, laws and regulations are implemented in a rule engine service that immigration agents and other users call in support of case management. The rule engine provides a list of tasks for the specific case, but end users decide on ordering the tasks. Specialized tools enable the end users to provide the rule modelers and the regulators with feedback for continually improving the regulations and rules.[72]

[66] David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer & Chris Welty, *Building Watson: An Overview of the DeepQA Project*, 31 AI MAGAZINE, Fall 2010, at 59.

[67] Kevin D. Ashley & Vern R. Walker, Automated Monitoring of Legal-Rule Compliance Using DeepQA: Screening Legal Documents for Argumentation Evidence (unpublished manuscript) (on file with authors).

[68] Matthias Grabmair & Kevin D. Ashley, *A Survey of Uncertainties and their Consequences, in Probabilistic Legal Argumentation, in* BAYESIAN ARGUMENTATION 61, 61-62 (Frank Zenker ed., 2012).

[69] WATERMAN & PETERSON, *supra* note 4.

[70] Guido Governatori & Sidney Shek, *Rule Based Business Process Compliance*, 874 PROC. OF THE RULEML2012@ECAI CHALLENGE, AT THE SIXTH INT'L SYMP. ON RULES Paper No. 5 (2012), *available at* http://ceur-ws.org/Vol-874/paper17.pdf.

[71] Benjamin N. Grosof, Yannis Labrou & Hoi Y. Chan, *A Declarative Approach to Business Rules in Contracts: Courteous Logic Programs in XML*, 1999 PROC. OF THE FIRST ACM CONF. ON ELECTRONIC COM.

[72] Audrey Theunisz, INDiGO: Rules-driven Business Services; Flexibility Within the Boundaries of the Law, Presentation of the Dutch Immigration and Naturalisation Service (Oct. 5, 2010), *available at* http://www.servicetechsymposium.com/soa_archive/pdf_berlin/Audrey_Theunisz_Rules_Driven.pdf. In the absence of a published paper in English on INDiGO, this seems to be the most informative document publicly available.

E. Addenda: Intelligent Tutoring Systems and Ethics

Finally, the Syllabus includes two topics that could be addressed at the end of the seminar or integrated at an earlier stage. The section, *Intelligent Tutoring Systems in Law and Ethical Reasoning,* highlights computational models of legal reasoning that have been incorporated in systems designed to train law students in particular legal concepts,[73] and in dialectical skills such as distinguishing cases,[74] and constructing,[75] reviewing,[76] and understanding legal arguments.[77] These online instructional techniques may provide law students, even those participating in on-line legal education or MOOCs (Massive Open Online Courses), the kind of argument-making practice that students in traditional legal education receive in the classroom.

As AI programs are fielded, they raise legal issues, and AI and Law programs are no exception. The section *Legal and Ethical Issues Related to AI and Law Programs*, focuses on some of those legal issues having to do with the legal status of intelligent agents,[78] liability of intelligent agents,[79] unauthorized practice of law using legal expert systems,[80] and intellectual property in virtual environments or in contexts involving intelligent agents.[81] Law students feel comfortable taking on such topics using traditional legal research tools, but an effective legal analysis turns on a deeper understanding of how the technology works. By the end of the course students may thus be better equipped to take on these issues of legal responsibility.

For instance, a legal e-treatise and a legal expert system both contain legal knowledge, some of which may be mistaken or applied with

[73] Antoinette J. Muntjewerff, *ICT in Legal Education*, 10 GER. L. J. 669 (2009).

[74] *See* Aleven, *supra* note 29.

[75] Chad S. Carr, *Using Computer Supported Argument Visualization to Teach Legal Argumentation*, *in* VISUALIZING ARGUMENTATION: SOFTWARE TOOLS FOR COLLABORATIVE AND EDUCATIONAL SENSE-MAKING 75 (Paul A. Kirschner, Simon J. Buckingham Shum & Chad S. Carr eds., 2003).

[76] Kevin Ashley & Ilya Goldin, *Toward AI-enhanced Computer-supported Peer Review in Legal Education*, 2011 PROC. OF THE TWENTY-FOURTH ANN. CONF. ON LEGAL KNOWLEDGE & INFO. SYS. 1.

[77] Collin Lynch, Kevin D. Ashley & Mohammad H. Falakmasir, *Comparing Argument Diagrams*, 2012 PROC. OF THE TWENTY-FIFTH ANN. CONF. ON LEGAL KNOWLEDGE & INFO. SYS. 81.

[78] SAMIR CHOPRA & LAURENCE WHITE, A LEGAL THEORY FOR AUTONOMOUS ARTIFICIAL AGENTS 119 (2011).

[79] Emad A. R. Dahiyat, *Intelligent Agents and Liability: Is It a Doctrinal Problem or Merely A Problem of Explanation?*, 18 ARTIFICIAL INTELLIGENCE & L. 103 (2010).

[80] Taiwo A. Oriola, *The Use of Legal Software by Non-Lawyers and the Perils of Unauthorised Practice of Law Charges in the United States: A Review of Jayson Reynoso Decision*, 18 ARTIFICIAL INTELLIGENCE & L. 285 (2010).

[81] Woodrow Barfield, *Intellectual Property Rights in Virtual Environments: Considering the Rights of Owners, Programmers and Virtual Avatars*, 39 AKRON L. REV. 649 (2006).

injurious results. These two technologies differ, however, in the ways and extent to which they invite reliance, presumably with different implications for liability of the authors or knowledge engineers. Thus, law students can write about something they know how to do (that is, analyze legal liability), but still benefit from understanding how a legal expert system selects and applies its legal knowledge and how it explains its conclusions. Additionally, the process can help law students understand the limitations of the knowledge engineering techniques by focusing on the gaps in its knowledge. One could also imagine inviting a law student to consider using the task of analyzing legal liability of intelligent agents as the task to be modeled using AI and Law techniques.

The Syllabus thus conveys a coherent narrative about past and current work in AI and Law, its challenges and its progress. The next two sections highlight the salient lessons for law students concerning both legal reasoning and understanding the new technologies for legal practice. For convenient reference, the lessons are listed in Figure 1 and used to organize the presentation in the next two sections.

A. Lessons about legal rules

Lesson A1: Legal rules are subject to semantic ambiguity.

Lesson A2: Legal rules are subject to logical ambiguity.

Lesson A3: Statutory structure affects the meaning of legal rules.

Lesson A4: Legal rules are subject to unstated conditions.

B. Lessons about reasoning with legal cases

Lesson B1: Distinguishing hard from easy cases of law is a key process.

Lesson B2: Analogizing and distinguishing cases are specifiable processes.

Lesson B3: Reasoning with cases and values is a kind of theory construction.

Lesson B4: Legal advocates pose hypothetical cases to test legal rules.

C. Lessons about legal argument

Lesson C1: Advocates employ legal-domain-specific argument schemes.

Lesson C2: One may attack a legal argument by attacking its assumptions or identifying exceptions.

Lesson C3: Legal arguments involve complex relationships among logic, rhetoric, uncertainty, and narrative.

D. Lessons about legal Digital Documents Technologies

Lesson D1: Legal Digital Documents Technologies are based on process models and algorithms.

Lesson D2: Processes underlying legal Digital Documents Technologies involve different kinds of texts as inputs.

Lesson D3: Digital Documents Technologies find texts relevant to legal problems.

Lesson D4: Digital Documents Technologies and their process models need empirical methodologies for testing.

Lesson D5: Finding texts relevant to legal problems is different from applying relevant texts to solve legal problems.

Lesson D6: Computational models of legal reasoning can be a bridge between legal texts and legal problem solving.

Figure 1: Lessons law students can learn from the materials in the AI and Law Seminar Syllabus

IV. AI AND LAW LESSONS ABOUT LEGAL RULES, CASES, AND ARGUMENTS

In uniquely concrete ways, AI and Law addresses lessons that law students need to learn about legal rules, about reasoning with cases, and about legal arguments generally. Opportunities to drive home these lessons arise at various points in the Syllabus. These are opportunities for students to experience "Aha!" moments—small apotheoses that can help ideas fall into place in law students' minds. This section identifies those moments in the Syllabus and provides some context.

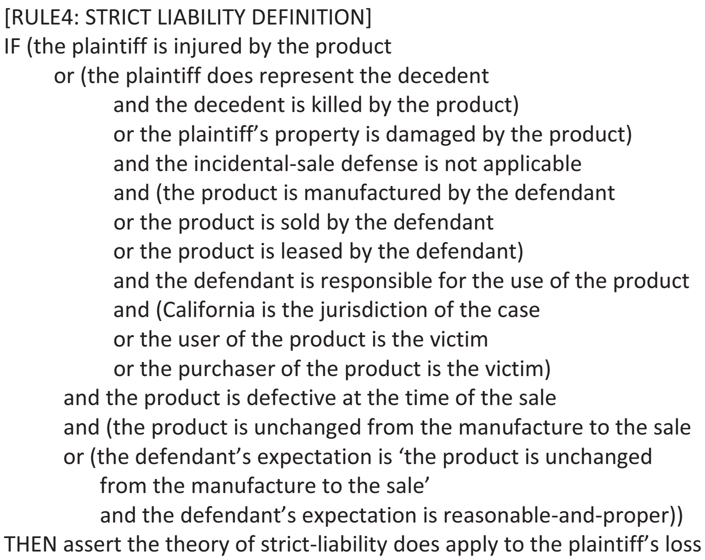
A. Lessons About Legal Rules

Lesson A1: Legal rules are subject to semantic ambiguity. By sometime in their first semester, law students have probably learned that the meanings of concepts in a legal rule often are not well specified. By the time they take an AI and Law seminar this is no longer news, but it may yet be edifying for students to observe the practical ramifications for building computer programs that reason with legal rules.

A statute can be modeled as a logic program[82] or using heuristic rules, but when the rules run out, resort must be made to something else: expert queries, arguments from cases, or arguments from the structure and purpose of the statute.

[82] *See* Sergot et al., *supra* note 20.

Waterman faced this problem early on when using rules to define product liability, strict liability, and comparative negligence in his program to guide settlement decision-making. Characteristically, his rule in Figure 2 defining "strict liability" employs terms and concepts that are defined in other rules, such as, "responsible for use of product" and "incidental sale," each of which are defined in still other rules.[83] However, some of the legal predicates important in this product liability context, such as "reasonable and proper" or "foreseeable," are *not* otherwise defined in a legal rule. Some of the legal concepts—including "emergency," "improper description," "property," "injury," "defective," and "careless,"—whether defined in rules or not, are ill-defined. Applying them to specific fact situations is an oft-litigated matter of interpretation and legally plausible arguments may often be made that the concept does or does not apply.



```
[RULE4: STRICT LIABILITY DEFINITION]
IF (the plaintiff is injured by the product
     or (the plaintiff does represent the decedent
           and the decedent is killed by the product)
           or the plaintiff's property is damaged by the product)
           and the incidental-sale defense is not applicable
           and (the product is manufactured by the defendant
           or the product is sold by the defendant
           or the product is leased by the defendant)
           and the defendant is responsible for the use of the product
           and (California is the jurisdiction of the case
           or the user of the product is the victim
           or the purchaser of the product is the victim)
      and the product is defective at the time of the sale
      and (the product is unchanged from the manufacture to the sale
      or (the defendant's expectation is 'the product is unchanged
          from the manufacture to the sale'
          and the defendant's expectation is reasonable-and-proper))
THEN assert the theory of strict-liability does apply to the plaintiff's loss
```

Figure 2: Waterman's Rule Defining Strict Liability

Human legal analysis does not come to a standstill when encountering a term whose meaning is underspecified, and system designers and knowledge engineers need to model how humans proceed. While law students may understand the concept of "semantic ambiguity," it is

[83] WATERMAN & PETERSON, supra note 4, at 16, 38.

less clear how deeply they have thought about the methods human lawyers use to deal with semantic ambiguity, the kinds of arguments they use, what those arguments assume, and how those arguments can be attacked.

Waterman listed four computational techniques with which a legal expert system could deal with ill-defined legal concepts:

1. Using heuristic rules (gleaned from legal experts or insurance adjusters) that attempt to capture how the term was used in the past;
2. Presenting examples and letting the user decide if the term is satisfied in the current fact situation;
3. Enabling the system to compare the current fact situation with examples in order to determine if the term is satisfied; and
4. Modifying the system's rules in a process of successive refinement in order to capture the concept's meaning.[84]

Much subsequent work in AI and Law has addressed these computational techniques. For instance, legal expert systems use the first and second techniques to guide users in practical decision-making. Researchers have focused on the third technique, in particular, by enabling a system to reason analogically with cases and make arguments that the concept should apply or not apply to a new situation. This is the subject of the work discussed in Section III of the Syllabus. Some work has addressed the fourth technique, where the successive refinement comes in the form of annotating judicial explanations of decisions. The annotations highlight portions of the explanations that can be matched to analyze new fact situations.[85]

Lesson A2: Legal rules are subject to logical ambiguity. While semantic ambiguity surprises few law students, far fewer are aware of the problem of logical ambiguity in legal rules. Most seem surprised to learn that even the *logic* of a legal rule may be ambiguous because the scopes of logical connectors are not clearly delimited in natural language and because of the complexity of statutory structure. In formal or mathematical logic, parentheses delimit scopes of logical or mathematical operators, but this is not the case in statutory and other natural language texts. Layman Allen provided many examples demonstrating that even simple statutory rules have multiple logical interpretations, of which the legislature was probably mostly unaware.

[84] *Id.* at 26.

[85] *See, e.g.*, Branting, *supra* note 26; McLaren, *supra* note 36; Ashley & Walker, *supra* note 67.

This is in sharp contrast to semantic ambiguities that may have been included intentionally in order to facilitate a legislative compromise. For instance, a Louisiana statute provided, "No person shall engage in or institute a local telephone call . . . of an anonymous nature and therein use obscene, profane, vulgar, lewd, lascivious or indecent language, suggestions or proposals of an obscene nature and threats of any kind whatsoever."[86] To be in violation, however, must the telephone call include obscene language *and* threats, or is either sufficient? In a criminal law context, of course, the answer can mean an acquittal. Similarly, students learn that logical ambiguity may be crucial in the interpretation of insurance policies and contracts. Given that one cannot assume that today's law students appreciate the need for close reading and drafting of legal rules, this is an important lesson.

Allen lays out a stepwise procedure for normalizing a statute—that is, for systematically enumerating the alternative logical possibilities—so that one can select among the alternative logical interpretations. He emphasizes, however, that the choice is a matter for the legislature, ideally in the drafting process. When developing a legal expert system comprising statutory rules, knowledge engineers also need to select among the alternatives. They try to select the alternative that the legislature probably meant, but their selection is not authoritative. It is simply the knowledge engineer's interpretation of the statute.[87] Alternatively, the rules in a legal expert system could be heuristic rules embodying an expert's view on a field of statutory law. The expert may have re-conceptualized or re-characterized statutory terms in other ways that resolve syntactic (and semantic) ambiguities in light of the expert's experience and that suffice for the purpose of the system.

Allen's work also focuses students on the utility of normalization and outline indentation in providing a flow-chart-like aid to understanding a complex statute. He illustrates this with a complex tax provision, IRC Sec. 354 governing exchanges of stock and securities in certain reorganizations, which, like other provisions of the U.S. Internal Revenue Code, is "just plain awful from a syntactic viewpoint."[88] He then compares it with a normalized version that provides a flow chart

[86] Allen & Engholm, *supra* note 22, at 384 (quoting State v. Hill, 157 So.2d 462, 462 (1963)).

[87] *Id.* at 396 ("[W]hen a law has already been enacted, any form of the statute other than the original enactment represents an interpretation. It is important that any representation which purports to correspond to a statute say neither more nor less than the statute itself does.").

[88] *Id.* at 388.

through the provision's logic.[89] It illustrates alternate paths through the statutory rule's conditions to a conclusion and makes it much easier to see if there *is* a path to a conclusion. Students begin to appreciate the potential for a more systematic way of drafting and presenting complex legal provisions as an aid to their understanding of statutes and as a goal for drafting contractual language.

Lesson A3: Statutory structure affects the meaning of legal rules. Allen's tax code example also raises the problem of charting logical paths when a statutory provision incorporates other provisions by reference (which may involve yet other exceptions) or worse, when other provisions provide exceptions that are not explicitly cross-referenced. This lesson also leads students to consider the significance in statutory interpretation of a provision's position in the statute or statutory code relative to other provisions. The hierarchical structure of a statute or code carries information that can help to resolve an otherwise ill-defined term, as occurs regularly in civil law interpretation.

AI and Law researchers have not made much progress on modeling such phenomena.[90] However, they encounter these problems frequently and usually with frustration. In work on automatically extracting information from statutory texts using machine learning—for instance, where it is reasonable to assume that a provision's outline structure of parts and subparts provides semantically valuable information—it may even be unclear to a competent human reader whether an unlabeled paragraph at the end of a provision is a subpart of the preceding subsection or an independent subpart.

Lesson A4: Legal rules are subject to unstated conditions. Just as a legal rule may have unreferenced exceptions, it also may have unstated conditions, for instance, that it is not unconstitutional, that it is not preempted, or that it can be applied without contravening legal principles. Allen raises the question of how one knows which unstated conditions to include, when such a condition applies and how to resolve whether the rule applies. These are tough questions for both law students and knowledge engineers. While it might be thought that so-called canons of construction resolve such questions, Allen points to

[89] *Id.* at 393, fig. 4.

[90] *But see* Bench-Capon & Coenen, *supra* note 21; Tom Routen, *Hierarchically Organised Formalisations*, 1989 PROC. OF THE SECOND INT'L CONF. ON ARTIFICIAL INTELLIGENCE & L. 242.

Karl Llewellyn's convincing characterization of the canons as makeweights.[91]

In *Obstacles to the Development of Logic-Based Models of Legal Reasoning*, Berman and Hafner argued persuasively that, even if there is agreement on facts and the applicable law, adversaries can still generate reasonable pro and con arguments, and courts come to different conclusions. This legal indeterminacy is caused by conflicts inherent in the unstated conditions concerning whether a legal rule really should apply, whether applying it counters underlying principles, what the rule's open-textured concepts mean in a given context, and how to resolve inconsistent canons of construction and inconsistent precedents. As a result of this legal indeterminacy, they argued, deductive logic-oriented models of statutory law cannot succeed.[92]

Berman and Hafner cite *Riggs v. Palmer*[93] among other examples of legal indeterminacy. *Riggs* posed the quandary as to whether a grandson who murdered his grandfather would inherit under his grandfather's will. The explicit conditions of the rule that "any heir to property of a deceased person may claim that property from the estate of the decedent" are satisfied. But applying the rule would violate the principle that "any legal rule will not apply to the benefit of a party who performed a felonious act offering serious harm to others, which brought about conditions under which the rule would apply to his benefit." Such a principle is a rule about rules and poses technical problems. The logical system would need to support quantification of a rule over all rules.

More importantly, Berman and Hafner concluded, the phenomenon of legal indeterminacy leads to a more profound problem: logic-based computational models are inappropriate for modeling how lawyers reason. The technical problem is that, in classical logic, it is impossible to validly prove (or argue for) a proposition and its opposite. Their examples illustrate, however, that even when advocates start with the same premises (that is, axioms, rules, and accepted facts), they still generate legally reasonable, but contradictory arguments.[94]

[91] Layman Allen & Charles Saxon, *Some Problems in Designing Expert Systems to Aid Legal Reasoning*, 1987 PROC. OF THE FIRST INT'L CONF. ON ARTIFICIAL INTELLIGENCE & L. 94, 105. For example, "A statute cannot go beyond its text." v. "To effect its purpose a statute may be implemented beyond its text." "If language is plain and unambiguous it must be given effect." v. "Not when literal interpretation would lead to absurd or mischievous consequences or thwart manifest purpose."

[92] Berman & Hafner, *supra* note 23.

[93] 22 N.E. 188 (N.Y. 1889).

[94] Berman & Hafner, *supra* note 23, at 191.

Indeed, legal instructors regularly demand that law students produce arguments and counterarguments in analyzing exam problems and demand that students recommend resolutions.

Reading Berman and Hafner's account, it would be hard for law students not to reflect on how they, as human reasoners, deal with legal indeterminacy. Given these problems of semantic and syntactic ambiguity, implicit exceptions, and unstated conditions, it is clear that attorneys reason *about* legal rules as opposed to simply reasoning *with* legal rules. Attorneys employ arguments in order to convince themselves and others what a concept means, whether an exception applies, or whether an unstated condition is satisfied. The lesson for law students, and for AI and Law researchers, is that reasoning about legal rules involves the need to consider arguments for and against applying the rule.

As developed in the Syllabus, AI and Law researchers have sought to model some of those types of arguments (including reasoning with cases, values, and principles) using argument schema with conditions and exceptions, leading to the remaining pedagogical lessons.[95]

B. Lessons About Reasoning with Legal Cases

Lesson B1: Distinguishing hard from easy cases of law is a key process. There are so many potential conflicts in applying legal rules that students might well wonder how human reasoners distinguish hard from easy cases of law.

If hard questions are those where, even though attorneys agree on the facts, there are conflicting reasonable arguments about the outcome or the explanation, are there any easy questions where legal experts would agree on both the outcome and explanation of the result? Surely some legal decisions raise no disputable point of law for a side; legal practitioners routinely make dozens of legal decisions!

In a first-year legal writing or legal process course, law students may have encountered this issue in the form of the jurisprudential debate between H.L.A. Hart and Lon Fuller,[96] but it is an issue that law students confront repeatedly, if not reflectively, at the end of every semester. Law students must, of course, decide which issues to develop at length on an exam and which to ignore, a decision that involves as-

[95] Alternatively, AI and Law researchers have sought to find applications where these limitations on logic will not have negative practical consequences.

[96] *See, e.g.,* Berman & Hafner, *supra* note 23, at 185-86.

sessing the opportunities presented by the strengths and weaknesses crafted into the instructor's scenario in light of the legal rules and decisions covered in the course. Indeed, assessing students' ability to make such decisions effectively, flexibly and under pressure is one of the reasons why instructors assign essay-type exam problems in law school. Thus, helping law students to reflect on how these decisions are made can be a timely and useful lesson.

In her contracts-test-taking program, Anne Gardner was the first to offer an algorithmic heuristic method for distinguishing hard from easy questions. The inputs to her (unnamed) program are representations of law student contracts examination problems dealing with offer and acceptance. The outputs are analyses of the problem characterizing the events as offers or acceptances and identifying the legal issues involved. Gardner took the position that there must be easy cases. At least, we treat questions as easy all the time. If there were no easy cases, everything would be open to interpretation and answers would forever elude us. Crucially from the viewpoint of system designers, she pointed out that the method for determining whether a case is easy must be easy, too. Her heuristic method for distinguishing hard from easy cases is presented in Figure 3 as amplified in a characterization by Rissland.[97]

```
For every predicate in the rule,
        If common sense knowledge rules provide an answer, then CSK-Answer is True, else False.
        If problem matches positive examples of predicate, then Pos-Examples is True, else False.
        If problem matches negative examples of predicate, then Neg-Examples is True, else False.

If ¬ CSK-Answer
        If ¬ (Pos-Examples or Neg-Examples) --> Question is HARD
        If (only one of Pos-Examples / Neg-Examples) --> Question is EASY
        If (Pos-Examples and Neg-Examples) --> Question is HARD

If CSK-Answer
        If ¬ (Pos-Examples or Neg-Examples) --> Question is EASY
        If (only one of Pos-Examples / Neg-Examples)
                If (agrees only-one w/ CSK-Answer) --> Question easy, else Question is HARD
        If (Pos-Examples and Neg-Examples) --> Question is HARD
```

Figure 3: Gardner's Heuristic Method for Distinguishing Hard from Easy Cases

It is instructive for law students to ponder Gardner's method as they think about how attorneys decide which predicates to argue

[97] Rissland, *supra* note 17, at 1970, n.62; Gardner, *supra* note 19, at 54-55, 160-61.

about, and, indeed, how they (the students) decide which exam issues to develop. The strategy in Gardner's program can be expressed algorithmically as a set of steps and provides a model that students can adapt into a more realistic strategy. For one thing, students will probably require the algorithm to involve case comparisons.

In Gardner's program, the examples were not full-blown cases, but at best snippets of cases capturing paradigm, standard, or extreme values for variables like "immediately." However, simple direct matches of positive and negative examples are not sufficient. A way is needed to compare the problem to case examples more robustly. The algorithm would be a more useful model of students' exam-taking strategies, if it took into account comparing a problem scenario's facts with those of cases where courts decided that the legal predicate did or did not apply. The decision of whether the problem is a hard or easy case depends on the strength of the analogies to other cases. If the positive case example is very close on the facts but the negative one is not, it may not be a hard question.

Lesson B2: Analogizing and distinguishing cases are specifiable processes. Considering an algorithm for distinguishing hard from easy questions of law leads naturally to considering the design of algorithms for comparing cases. In short, it leads students to contemplate the point of drawing legal analogies, how one defines "relevant similarity" and "relevant difference", and how one assesses the strength of the analogies. Gardner's examples clearly did not contain enough factual information to compare how analogous cases are, to assess how their similarities might justify treating a problem scenario in the same way as a case, or to assess how their differences might justify not doing so.

Law students all know that they need to analogize and distinguish cases. Much Socratic classroom discussion involves students distinguishing among the cases in the course text on a particular issue. In course essay exams, students are called upon to analogize or distinguish those cases and the problem scenario. It is less clear that law students understand what analogies and distinctions are and why they matter in legal analysis.

As summarized in Table 1, the Syllabus provides numerous examples of the ways in which AI and Law researchers have attempted to model legally relevant similarities and differences. They focus on identifying legally meaningful patterns of fact, ranging from:

- generalized fact descriptions whose presence or absence is deemed by an expert to bear on an issue, to

- stereotypical patterns of facts that, according to experts, and as confirmed in cases, typically strengthen a side's claim or argument on an issue, to varying degrees, to
- excerpts of the critical facts from a judge's explanation justifying his or her decision of a particular issue.

Researchers index these patterns by claims, elements of claims or other issues, and underlying values or principles. The indexes are then associated with reasons. As a result, computer programs can identify legally relevant similarities and differences, use them to analogize and distinguish cases, and explain the significance of the analogies and distinctions to the decision of a problem. These tasks focus on the manner in which underlying values play a role in relevance.

ble 1: Computational Measures of Relevance

Relevance Representation	**Definition**	**Domain / Example**	**Relevance Measure**	**Source**
Descriptor	Binary fact descriptions relevant to legal outcome (46 descriptors for tax law, capital gains)	Qualified for capital gains tax / at time of purchase, private party had other intention than to resell at a profit	k-nearest neighbor	Mackaay & Robillard , *supra* note 31.
Dimension	Stereotypical pattern of facts that strengthen / weaken: a side's claim (13 trade secret dimensions)	Trade secret misappropriation / Disclosures-to-outsiders	Overlap of sets of dimensions in scenario and cases	Ashley, *supra* note 5.
Dimension per ssue (i.e., claim element or rule erm)	Stereotypical pattern of facts that strengthen / weaken: side's argument re issue (14 dimensions for tax law, home office deduction)	Home office deduction/principal-place of business:relative-home-work-time	Overlap of sets of dimensions per issue in scenario and cases	Rissland & Skalak, *supra* note 33.
Factor	Binary version of dimension (26 trade secret factors)	Trade secret misappropriation / Disclosures-to-outsiders (T or F)	Overlap of sets of factors in scenario and cases	Aleven, *supra* note 29.
Abstract factor	Reason why factor matters (13 trade secret abstract factors)	Trade secret misappropriation / Efforts to maintain secrecy	Shared reason	Aleven, *supra* note 29.
Factor per issue	Binary version of dimension associated with issue	Trade secret misappropriation / Info-trade-secret / maintain-secrecy	Overlap of sets of factors per issue in scenario and cases	Ashley & Brüninghaus, *Computer Models*, *supra* note 30.
Criterial explanation per issue	Significant excerpt from explanation of why judge decided issue (represented as semantic network)	The employee (Vaughn) had intense hunger, which impedes his duties as truck driver and which was decreased by having food at a restaurant.	Ratio of case criterial facts matched in scenario	Branting, *supra* note 26.
Factors plus values	Legal value served by factor (4 factors and 3 values for property law in wild animals)	Property in wild game / plaintiff-pursuing-his-livelihood safeguards socially desirable economic activity.	Best theory of preferences among competing factors and competing values	Bench-Capon & Sartor, *supra* note 35.
Criterial explanations per principle	Critical excerpt from explanation of why judge decided principle applied (or not)	Owner's attorney instructs Engineer A to withhold information regarding apartment building.	Best structural mapping using heuristic cost function	McLaren, *supra* note 36.

It is instructive for law students to compare the representational techniques in Table 1 in terms of the extent to which they capture or omit legally relevant considerations. For instance, descriptors, dimensions and factors serve as a kind of checklist of important fact patterns. These lists support reasoning about a problem in terms of fact patterns that are *not* present, and they are associated explicitly with reasons or values that serve to explain why they matter. On the other hand, these lists are fixed; they need to be updated as new patterns emerge. Criterial explanations are dynamic in the sense that, as a human enters the cases and their explanations into the database, the individual can annotate the explanatory snippets that the judge regarded as critical. The explanatory snippets can then be associated with or indexed by the reasons or values to which they relate. In processing a new case, the explanatory snippets can be matched to the new case's facts, taking into account that these facts may be expressed in different terms.

One way or the other, legally important fact patterns can be associated with underlying reasons, values, and principles. This association is important, because, as Berman and Hafner drove home,[98] the sometimes competing values to which the facts give rise are a crucial focus of legal arguments on how to decide a problem scenario and a major focus of jurisprudential efforts to justify legal analogy as an interpretive phenomenon. How to represent the weights or significance of the patterns and values in the analysis of a scenario, and how to resolve competing values, has proven controversial. While representing weights numerically facilitates computational processing, researchers generally observe that explanations and arguments directed to legal practitioners and judges do not treat weights numerically. It is also agreed that the weights of values cannot be modeled using fixed hierarchies; some more context-sensitive means of setting the weights is needed. Finally, it is agreed that the weights vary over time as society's values change. For instance, over the relatively short period in which the Internet has become pervasive, the value of privacy across societies and generations within a society has increasingly been in flux.

Lesson B3: Reasoning with cases and values is a kind of theory construction. In their model of reasoning with cases and values, Bench-Capon and Sartor characterize the model "as a process of constructing and using a theory."[99] In this kind of theory construction, the outcomes

[98] Berman & Hafner, *supra* note 34.

[99] Bench-Capon & Sartor, *supra* note 35, at 98. This important insight stems from McCarty, *supra* note 24, at 285: "The task for a lawyer or a judge in a 'hard case' is to construct a theory of

of past cases reveal preferences between sets of factors present in those cases, and those, in turn, reveal preferences between sets of values. The theory has explanatory power. The induced preferences between sets of values explain preferences between sets of factors, and these explain the outcome in past cases. The theory can also be applied to determine and explain the outcome of new cases.

It is an elegant theoretical model, but not without problems. First, more than one theory can be induced; one has to assess the competing theories in terms of their coherence operationalized in terms of their explanatory power (that is, the number of cases the theory explains), consistency, and simplicity. However, the latter two criteria are not very well understood. Second, from a jurisprudential viewpoint, it does not appear that judges do or should use preferences among competing values in past cases to decide new problems. Assessing proposed outcomes of a problem in terms of values is an ethical decision of its own. A judge needs to consider how the values apply, given the problem's particular facts. Even if a judge were to use value preferences in past cases as a guide, one would expect that the judge would still need to compare the problem to the cases in detail to be sure that applying that preference in the new circumstance is appropriate. For this kind of comparison, a finer grained representation of the facts would surely be necessary.

In any event, from an instructional viewpoint, the Bench-Capon and Sartor piece focuses law students on the question of what a legal theory of a case looks like, how it relates to precedents and values, and the role of underlying values in defining relevant similarities for purposes of analogizing and distinguishing cases. More generally, the question is, how *do* legal practitioners take values and principles into account in reasoning about how to decide problems?

Lesson B4: Legal advocates pose hypothetical cases to test legal rules. One way legal practitioners take values and principles into account is by posing hypotheticals. A hypothetical is a made-up scenario designed to test certain properties of a proposed rule for deciding a case. Law students who have read, listened to, or watched any oral arguments before the United States Supreme Court will have encountered hypotheticals. To the consternation of legal advocates, the justices are famous for posing hypotheticals. And, of course, students have likely encountered hypotheticals posed by an instructor in Socratic

the disputed legal rules that produces the desired legal result, and then to persuade the relevant audience that this theory is preferable to any theories offered by an opponent."

classroom discussions. Students, however, may not have adequately considered the rhetorical point of posing hypotheticals in a legal argument: What constitutes an effective hypothetical, what force does a hypothetical have, and how can one respond to such an argument?

AI and Law researchers have made progress in computationally modeling hypothetical reasoning.[100] For law students, this work can help to clarify the nature of the adversarial process in which an advocate tries to convince a court how to decide a case. The advocate is really trying to convince the court that the advocate's proposed rule or test for deciding the case at bar is a good rule to adopt both in the instant case and for future cases. On one view, the advocate is arguing that the effect on applicable values of deciding the case in accord with the proposed rule is preferable to the effect of not doing so or of applying some other rule.

Thus, the adversarial process unfolds as follows: The advocate proposes a rule or test to decide the case at bar in favor of his or her client. The judge challenges the proposed test by posing a hypothetical situation where the result under the test is debatable. The advocate responds to the hypothetical by arguing that the result under the proposed test is justifiable, either by modifying the proposed test so that the result is justifiable or by distinguishing or analogizing the hypothetical.[101]

If the judge is concerned that the proposed rule may be too broad, he or she may construct a hypothetical to which the rule applies, but where the effects of applying the rule to the hypothetical are detrimental to a value underlying the rule. The advocate may respond by arguing that:

- the rule does not apply to the hypothetical, which is distinguishable, or
- the rule applies but the effect on the value is not as severe as the judge suggests, or
- the rule applies and is detrimental to the value, but that another value related to the rule is promoted, or
- the rule applies and is detrimental to the value and that the rule should be changed so that it no longer applies to the hypothetical.

[100] *See, e.g.,* Ashley, *supra* note 5; Kevin D. Ashley, *Teaching a Process Model of Legal Argument with Hypotheticals*, 17 ARTIFICIAL INTELLIGENCE & L. 321 (2009); Grabmair & Ashley, *supra* note 39.

[101] Ashley, *Teaching a Process Model*, *supra* note 100, at 326-27; Grabmair & Ashley, *supra* note 39, at 164.

A complementary scheme of arguments applies where the judge is concerned that the proposed rule may be too narrow.

By considering the models of hypothetical reasoning and its application in Supreme Court arguments, law students can come to understand this important legal and rhetorical strategy.

C. Lessons About Legal Argument

Lesson C1: Advocates employ legal domain-specific argument schemes. As the previous lessons suggest, one way to characterize the progress of AI and Law research is as a continuing search to identify and model legal domain-specific argument schemes, including schemes for arguing from rules, from cases, and with values. An argument scheme corresponds to a typical framework in the domain for making an inference sanctioned by the argument; it corresponds to a kind of *prima facie* reason for believing the argument's conclusion.[102] Some of these argument schemes connect with the computational models of legal reasoning that have been developed thus far. Others are new; researchers are developing argument schemes for arguing from evidentiary facts to rule-driven legal conclusions.[103]

From their Legal Writing course, law students are already familiar with suggested formats or schemes for proving a conclusion of law in a written brief. For instance:

> 1. State your conclusion.
>
> 2. State the primary rule that supports the conclusion.
>
> 3. Prove and explain the rule through citation to authority, description of how the authority stands for the rule, discussion of subsidiary rules, analyses of policy, and counter-analyses.
>
> 4. Apply the rule's elements to the facts with the aid of subsidiary rules, supporting authority, policy considerations, and counter-analyses.
>
> 5. If steps 1 through 4 are complicated, sum up by restating your conclusion.[104]

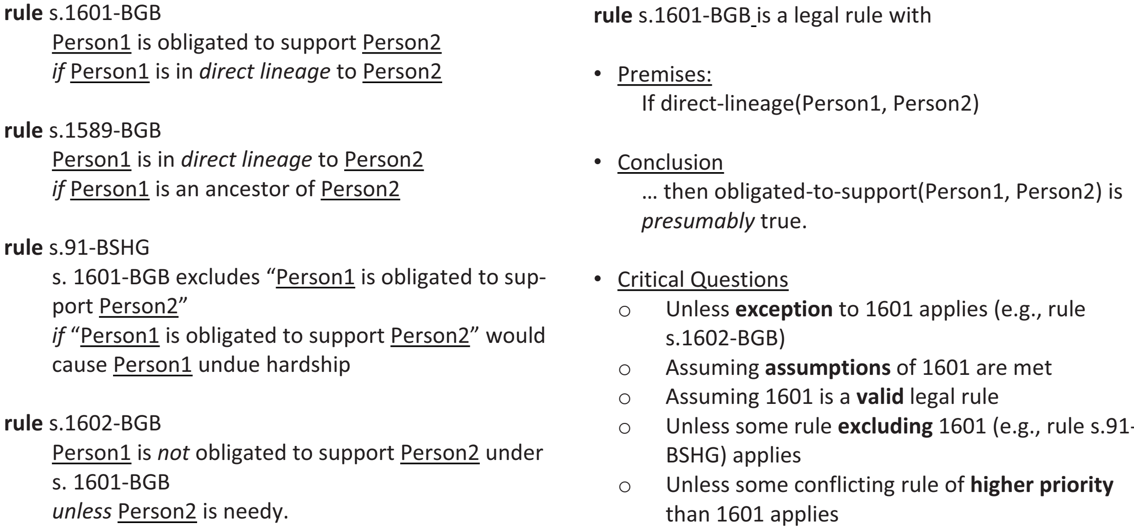
At least some of those steps, however, can be unpacked into additional schemes. For instance, one can imagine unpacking Step 4 into, and AI and Law researchers have implemented, additional argument schema to support or counter the assertion that a proposed rule

[102] HENRY PRAKKEN, ARTIFICIAL INTELLIGENCE AND LAW, LOGIC AND ARGUMENT SCHEMES, IN ARGUING ON THE TOULMIN MODEL 236 (David Hitchcock & Bart Verheij eds., 2006).

[103] *See, e.g.,* Walker et al., *supra* note 40.

[104] RICHARD K. NEUMANN, LEGAL REASONING AND LEGAL WRITING: STRUCTURE, STRATEGY AND STYLE 93-94 (6th ed. 2009).

should apply by analogizing a case in support of applying a rule's elements to the facts or by countering such an analogy.[105]



rule s.1601-BGB
Person1 is obligated to support Person2
if Person1 is in *direct lineage* to Person2

rule s.1589-BGB
Person1 is in *direct lineage* to Person2
if Person1 is an ancestor of Person2

rule s.91-BSHG
s. 1601-BGB excludes "Person1 is obligated to support Person2"
if "Person1 is obligated to support Person2" would cause Person1 undue hardship

rule s.1602-BGB
Person1 is *not* obligated to support Person2 under s. 1601-BGB
unless Person2 is needy.

rule s.1601-BGB is a legal rule with

- Premises:
  If direct-lineage(Person1, Person2)
- Conclusion
  … then obligated-to-support(Person1, Person2) is *presumably* true.
- Critical Questions
  - Unless **exception** to 1601 applies (e.g., rule s.1602-BGB)
  - Assuming **assumptions** of 1601 are met
  - Assuming 1601 is a **valid** legal rule
  - Unless some rule **excluding** 1601 (e.g., rule s.91-BSHG) applies
  - Unless some conflicting rule of **higher priority** than 1601 applies

Figure 4: "Family law" rule set and representation of rule as defeasible

Lesson C2: One may attack a legal argument by attacking its assumptions or identifying exceptions. An argument scheme for interpreting how legal rules should apply to problems, for instance, can focus law students on questioning the rule's assumptions or considering exceptions. Currently in AI and Law research, reasoning with a legal rule would be treated as defeasible and subject to an argument scheme for reasoning with defeasible rules.

Defeasible reasoning through argument can be illustrated in an argument-scheme-driven example of statutory reasoning using Tom Gordon's argument diagramming program, Carneades.[106] For instance, the left half of Figure 4 shows four legal rules involving family law

[105] *See, e.g.,* Grabmair & Ashley, *supra* note 39, at 164-66.

[106] Gordon & Walton, *supra* note 38; Thomas F. Gordon, *Analyzing Open Source License Compatibility Issues with Carneades*, 2011 PROC. OF THE THIRTEENTH INT'L CONF. ON ARTIFICIAL INTELLIGENCE & L. 50, 51; Thomas F. Gordon, *Constructing Legal Arguments with Rules in the Legal Knowledge Interchange Format (LKIF)*, *in* COMPUTABLE MODELS OF THE LAW 162, 168-69 (Pompeu Casanovas, Giovanni Sartor, Núria Casellas & Rossella Rubino eds., 2008).

based on the German Civil Code. Rule s.1601-BGB declares that person1 is obligated to support person2 if the former is in "direct lineage" with the latter, a concept defined in Rule s.1589-BGB.

Employing a scheme for arguing with defeasible rules is necessary because a classical logical system could not successfully handle the rules on the left of Figure 4. If the goal is to determine if Harry is obligated to support Sally and one learns that Harry is an ancestor of Sally, a classical logical system has no difficulty in chaining backward from Rule s.1601-BGB to Rule s.1589-BGB, inferring from Harry's being Sally's ancestor that Harry is in "direct lineage" to Sally and thus has the obligation to support her. Suppose one then learns, however, that being obligated to support Sally would cause Harry undue hardship. As a matter of deductive reasoning, rule s.91-BSHG implies that Harry's obligation to support Sally is excluded. The system would then have proven two inconsistent consequences; Harry is both obligated and not obligated to support Sally. Berman and Hafner identified this problem.[107] Classical logical deduction is monotonic; once one proves a proposition, one cannot take it back just because one learns of new information. This ability to prove a proposition *and* its opposite means the system is inconsistent and can prove anything. CRASH!

And yet, as explained by Berman and Hafner,[108] that kind of thing happens all the time in law! If researchers want a computer to draw inferences from legal rules in a realistic manner, classical logical deduction cannot be used. One has to use something else.

In the field of AI and Law, the current answer to "what else is there?" is a computational model of argument with appropriate argument schemes to enable defeasible reasoning. What is different about the representation of a legal rule as defeasible is shown in the right half of Figure 4. Rule s.1601-BGB is defeasible because the statute contains two exceptions to the obligation to support. First, under Rule s.91-BSHG person1 is not obligated to support when it would cause person1 "undue hardship." Second, Rule s.1602-BGB provides for another exception if person2 is not "needy."

Thus, Rule s.1601-BGB contains premises and a conclusion, but the conclusion is only *presumably* (i.e., defeasibly) true, signaling that there are some facts the system might learn that could defeat the conclusion, even though the premises were true. These defeasibility condi-

[107] Berman & Hafner, *supra* note 23.

[108] *Id.*

tions are represented as "Critical Questions," which identify assumptions and exceptions that affect the rule's applicability: Are the specified underlying assumptions met? Does a specified exception apply? Is the rule valid? Does some exclusionary rule apply? Does some conflicting rule of higher priority apply?

There is little agreement across the field about how to express such questions or what to call the different types of defeasibility conditions, but researchers generally agree that these defeasibility conditions need to be specified because, as directed by an argument scheme, the program will test them.

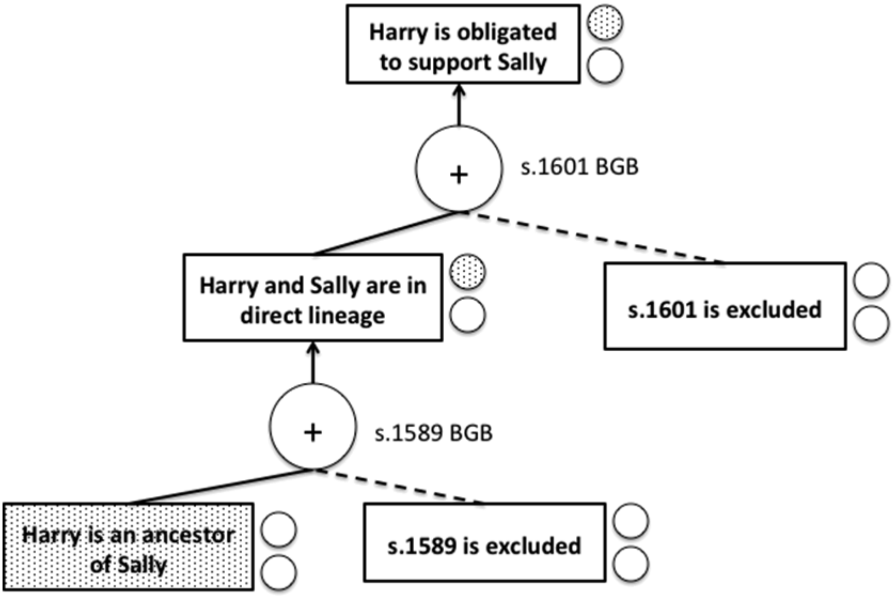

Harry is obligated to support Sally
+
s.1601 BGB
Harry and Sally are in direct lineage
s.1601 is excluded
+
s.1589 BGB
Harry is an ancestor of Sally
s.1589 is excluded

Figure 5: Carneades argument diagram with defeasible inference rules

A system like Tom Gordon's Carneades can reason with the defeasible inference rules and construct arguments. Figure 5 shows a manually prepared version of an argument diagram that Carneades can generate.109 Statement nodes are boxes and argument nodes are circles (with a "+" indicating support and a dashed line indicating an exception to the argument's applicability). If one specifies a goal of

[109] Gordon & Walton, *supra* note 38; Gordon, *Analyzing Open Source License Compatibility Issues*, *supra* note 106, at 54-55; Gordon, *Constructing Legal Arguments, supra* note 106, at 179.

showing that Harry is obligated to support Sally, the Carneades rules engine reasons backwards from that goal to find defeasible inference rules to support that goal: rules s.1601 and s.1589. As the arguments are constructed and edited, they are visualized in an argument map or diagram. A shaded statement node indicates that the statement is assumed to be true (by the relevant audience.) A shaded upper small circle next to a node indicates that the argument is acceptable (according to the relevant proof standard). (If the lower circle were shaded, the statement's complement is acceptable.)

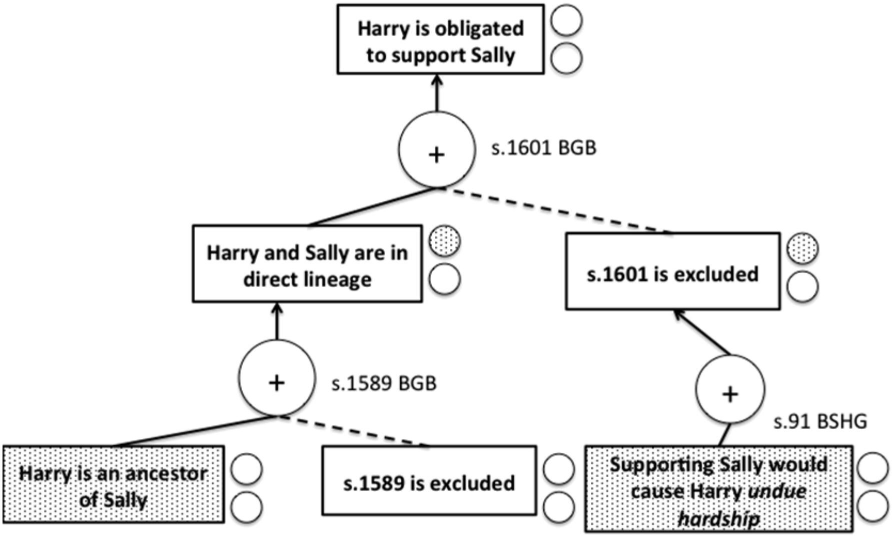

Harry is obligated to support Sally
+
s.1601 BGB
Harry and Sally are in direct lineage
s.1601 is excluded
+
s.1589 BGB
+
s.91 BSHG
Harry is an ancestor of Sally
s.1589 is excluded
Supporting Sally would cause Harry *undue hardship*

Figure 6: Rule's defeasible conclusion prevented by argument based on another rule.

If one specifies a goal of defeating that argument, the program searches through the critical questions associated with the rules to find applicable exceptions, exclusions, or failures of assumptions that would prevent the argument's presumed conclusion. Thus, having learned that being obligated to support Sally would cause Harry undue hardship, the program would find such an exclusion. The rule s.91 BSHG applies and prevents the argument's conclusion, as indicated in Figure 6.

Suppose that being obligated to support Sally would cause Harry undue hardship is *not* a given fact. Instead, let us assume that, although "undue hardship" is not defined by a legal rule, courts have found for or against undue hardship in cases involving certain factors:

Plaintiff(P) Factors (pro finding of *undue hardship*):	*Defendant(D) Factors* (con finding of *undue hardship*):
PF1. has-already-provided-much-support	DF1. expected-duration-of-support-is-short
PF2. never-had-parent-child-relationship	DF2. has-not-provided-care
PF3. would-cause-irreparable-harm-to-family	

In particular, suppose there are three cases:

Mueller: P wins *undue hardship* issue where {PF2}

Schmidt: D wins *undue hardship* issue where {PF2, DF1}

Bauer: P wins *undue hardship* issue where {PF2, DF1, PF3}

Finally, let us assume that: Harry and Sally never had a parent-child relationship, PF2, and that Sally needs support for only a short time period, DF1.

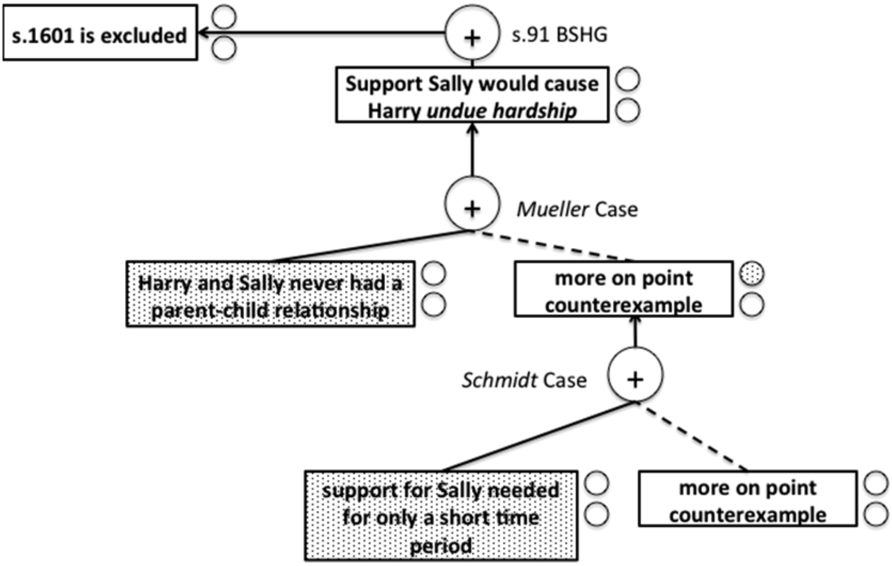

s.1601 is excluded
+
s.91 BSHG
Support Sally would cause Harry *undue hardship*
+
Mueller Case
Harry and Sally never had a parent-child relationship
more on point counterexample
Schmidt Case
+
support for Sally needed for only a short time period
more on point counterexample

Figure 7: Rule's defeasible conclusion prevented by argument based on *Mueller* case, which, in turn is trumped by argument based on *Schmidt* Case.

In this situation, Carneades could construct arguments for and against Harry having an obligation to support Sally based on *case-based* inferences, as shown in Figure 7.[110] The *Mueller* case, which shares PF2 with the problem, supports an argument that s.91 BSHG causes the obligation to be excluded under s.1601 BGB. The *Schmidt* case, however, is more on point (because the set of factors shared between *Mueller* and the problem is a subset of those shared between *Schmidt* and the problem, namely PF2 and DF1) and had the opposite result. In the terminology of HYPO[111] the *Schmidt* case trumps *Mueller*.

Lesson C3: Legal arguments involve complex relationships among logic, rhetoric, uncertainty, and narrative. Although the family law example is simple, law students might still be impressed with Carneades' ability to generate and integrate legal arguments of multiple types into a visual format that is readily interpretable. Law students can see an argument structure that is more complex than simply applying legal rules deductively.

The demonstration also begs the question how attorneys factor uncertainty and rhetorical considerations into their assessments of legal arguments. Legal argument is a technique for dealing with uncertainty (such as in legal planning), but uncertainty also affects legal argument. This uncertainty includes uncertainty about: evidence and the plausibility of factual assertions, the normative application of the law to facts, moral assessment of the alternative results, and the likelihood of acceptance of arguments.

The question then arises if and how to factor uncertainty into argument diagrams. A number of articles in the Syllabus address dealing with uncertainties in a Carneades-like argument model. One way is to implement burdens of persuasion or burdens of proof in the argument diagrams such as "preponderance of the evidence" and "beyond a reasonable doubt." These burdens of proof assist, for example, by associating a quantitative measure of evidential certainty with the arcs connecting arguments to conclusions and imposing an appropriate quantitative standard.[112]

In addition, the Carneades argument models could be augmented with an ability to reason with uncertainties (using Bayesian belief networks); this change would allow an advocate to explore the conse-

[110] See sources cited *supra* notes 38 and 109.

[111] Ashley, *supra* note 5, at 784.

[112] Gordon & Walton, *supra* note 38, at 137-38; Atkinson and Bench-Capon, *supra* note 37.

quences of various assumptions about evidentiary uncertainties.[113] For instance, should she argue the facts or the law? In order to assess how best to improve her chances of a favorable decision, an advocate could modify the assumed probability distributions representing the jurys' beliefs in various evidential facts or modify the weights representing the jury's belief in the plausibility of various arguments. The same work explores the role of narrative in assessing evidential uncertainties in the stories that attorneys attempt to fit to the evidence. Different narratives lend themselves to different legal claims. Evidence that seems plausible given one story may be irrelevant to or contradicted by another story. While the computational techniques for modeling these aspects of legal argument are in their infancy, law students can benefit from thinking about the underlying phenomena.

This completes the discussion of how an AI and Law seminar can be used to teach law students some classic lessons about law. The material can also teach lessons that help prepare law students for legal practice in the digital age.

V. Connecting Computational Models of Legal Reasoning with the Digital Technologies in Legal Practice

Law students need to understand the "information revolution" affecting the field of law and how it impacts their prospects for legal employment, the kinds of legal employment available, and their responsibilities as legal practitioners in a digital age. A distinguished academic observer of legal practice trends and their effects on the law school curriculum has recently observed:

> The complexity of a . . . [law school] curriculum is further compounded by an information revolution that is changing the mix of current and future jobs. Because of the emphasis on *process* and technology now taking hold within the legal industry, the practical technical skills and domain knowledge [now taught] may be inadequate for a large proportion of law students graduating in the year 2015.[114] . . . [Students] . . . are unprepared to learn that law is becoming less about jury trials and courtroom advocacy and more about *process engineering*, predictive coding, and the collaborative and technical skills those *processes* entail.[115]

The technology to which Henderson refers mainly involves automating the document review process so that it can deal with the ex-

[113] Grabmair & Ashley, *supra* note 68.

[114] William D. Henderson, *A Blueprint for Change*, 40 PEPP. L. REV. 461, 501 (2013).

[115] *Id.* at 505-06 (emphasis added).

traordinarily large numbers of electronic documents involved in modern litigation using machine learning techniques, also known as predictive coding. Programs are learning to select electronic documents from enormous caches produced in modern lawsuits and to cluster them in terms of their relevance, based on litigators' labeling of small sets of documents as relevant or not to the claims and defenses.

> "[P]redictive coding [. . .] [i]n essence, . . . is machine *algorithms* partially replacing humans altogether in the search for relevant information."[116] This development is driven by the size and diversity of the e-discovery caches.
>
> Because of the massive explosion of digital data, where so much of our daily lives are encoded into emails, text messages, internal knowledge management platforms designed to replace email, and digitized voice mail, the scope of discovery in civil or white collar litigation has become prohibitively expensive using traditional methods of review.[117]

The emphasis on "processes" is especially important for law students facing the legal employment market. Paraphrasing Richard Susskind's *The End of Lawyers?*,[118] Henderson states,

> These changes [from legal work that is bespoke (or customized) to standardized, systematized, packaged, and, ultimately, commoditized] are made possible by identifying recursive patterns in legal forms and judicial opinions, which enables the use of *process* and technology to routinize and scale very cheap and very high quality solutions to the myriad of legal needs.[119]
>
> [F]ormerly labor-intensive work that has traditionally been performed by entry-level United States law school graduates . . . is now being done by Indian law graduates [working for Legal *Process* Outsourcers (LPOs),] who are learning how to design and operate *processes* that extract useful information from large masses of digital text. Not only are the Indian law graduates getting the employment, they are learning valuable skills that are entirely—*entirely*—absent from U.S. law schools.[120]

Attorneys, law firms, judges, and regulatory agencies increasingly will need to assess, purchase, rely upon, and justify products and services that employ machine learning, data mining, natural language processing, and information retrieval for use in e-discovery and other legal information management processes. For ease of reference, these techniques will be referred to collectively as "Digital Documents Tech-

[116] *Id.* at 487 (emphasis added).

[117] *Id.*

[118] RICHARD SUSSKIND, THE END OF LAWYERS? 6 (2010).

[119] Henderson, *supra* note 114, at 479 (emphasis added).

[120] *Id.* at 487(emphasis added).

nologies". Although driven by the pressures of e-discovery, Digital Documents Technologies are not just for litigation. They will be important in all areas of corporate business and legal practice that involve the need to develop processes for managing corporate documents, for example, tax, mergers and acquisitions, and regulatory compliance.

As developed below, the AI and Law seminar can help law students understand Digital Documents Technologies by exposing them to information processes, algorithms, and process engineering. Indeed, the seminar may frequently be their *first* systematic exposure to these concepts.

It should be made clear from the outset, however, that the computational models of legal reasoning discussed in Parts 1, 2, and 3 of the Syllabus are different from those at the heart of current Digital Documents Technologies. Digital Documents Technologies have their roots in full-text legal information retrieval, advances in computational linguistics (such as shallow text parsing and named entity recognition) and statistical natural language processing, machine learning and data mining. The ramifications of these different roots and a description of how the approaches might ultimately intertwine are described below.

The relevant point here is, simply, that most of the AI and Law work in the Syllabus, used to teach law students lessons about legal rules, cases, and arguments, does *not* address the same tasks and processes as the Digital Documents Technologies. Indeed, a stark contrast exists between the ways in which Digital Documents Technologies and students process legal knowledge, a contrast that pits the efficiency of the machine learning, data mining, natural language processing and information retrieval models against the expressiveness of the AI and Law models. Legal information retrieval, for instance, deals with vast numbers of documents and natural language queries, with which Westlaw and Lexis are remarkably effective. However, those programs do not process the texts in any depth; Westlaw and Lexis cannot even tell who won a case or issue, much less how to use the text in an argument about a legal issue. Moreover, these programs cannot filter documents that have/do not have specified features, as opposed to phrases.

Despite these technological differences, the AI and Law seminar can help law students understand Digital Documents Technologies by teaching them the following tools:

1. a conceptual vocabulary for describing the Technologies in terms of computational models, algorithms, relevance measures, and experimental evaluations;
2. high-level descriptions of the task specifications, computational models, and algorithms underlying the Technologies;
3. the ways in, and extent to, which the performance of these Technologies can be measured experimentally, and what the measures signify;
4. examples of what the computational models underlying the Technologies capture and what the models leave out; and
5. the legal processes into which the Technologies are embedded, including hardware, software, human technical intermediaries, and human legal practitioners.

In sum, participating in the AI and Law seminar gives law students an experience of thinking "computationally" about the processes of legal practice and about the processing of legal information. Although today's law students grew up with the Internet and are intimately familiar with interfaces into the World Wide Web, they probably have not engaged much in thinking systematically about information processes in legal practice. These students most likely have little background in programming, and have thought neither about developing algorithms to implement specifications nor about using successive refinement to develop sub-algorithms for each step. However, by the end of the seminar, these students will have done both, first with the legal reasoning algorithms and then with algorithms for e-discovery.

This is a sensible progression for law students unfamiliar with how full-text legal information retrieval works. Techniques such as Bayesian belief networks for full-text retrieval, machine learning from large data sets, data mining, or natural language processing are complex and may leave legal intuitions behind. It will be helpful for students to gain an introduction to thinking computationally with algorithms that model reasoning processes with which they have become familiar by the end of the first year of law school.

The progression also makes sense in another way. As argued below, the computational models of legal reasoning ultimately *will* lead the way for improving the intelligent behavior of the Digital Documents Technologies. At least some law students taking the AI and Law seminar may be well positioned to design those techniques themselves or to support others in designing them.

Even if the law students become only consumers or users of the Digital Documents Technologies, rather than designers or improvers of them, it is valuable for non-programming law students to learn something about how these tools work. Corporate and governmental clients will soon demand e-discovery tools, and these tools will not only be permitted, but required by the professional standard of care. The process of accepting these tools into the profession has been, and will continue to be, performed by attorneys—former law students—selecting the tools, using them, directing others how to use them, interpreting their outputs and justifying their use to others (such as judges) by explaining how they work and how reliable and effective they are. The AI and Law Seminar prepares future attorneys for these tasks by teaching them some valuable lessons about legal Digital Documents Technologies.

VI. Lessons About Legal Digital Documents Technologies

This section highlights some of the lessons about Digital Documents Technologies students can learn from the AI and Law seminar.

Lesson D1: Legal Digital Documents Technologies are based on process models and algorithms. As noted above, the move from customized to commoditized legal services means that law students can benefit from an understanding of the concept of a "process involving legal knowledge."

The AI and Law seminar exposes law students to the notion of such processes by example. All of the computational models discussed in the Syllabus focus on processing legal knowledge. This is true of the AI and Law reasoning models as well as the Digital Documents Technologies models.

A number of the AI and Law models are especially well-developed representations of processes involving legal knowledge and algorithmic implementations of the processes or parts of them. For instance, the CABARET program applies an agenda mechanism to switch control between rule-based and case-based reasoning as the need arises.[121] Students can inspect the output of the process to compare CABARET's control mechanism with that of human problem solvers, who, probably, switch less frequently and are more focused on identifying issues and resolving them. The IBP+SMILE program reifies a process of predicting legal outcomes with past cases. The IBP component breaks a problem into issues, and for each issue, poses a hypothesis about who should win based on cases indexed by factors pertaining to that issue. Then the program tests the hypothesis for counterexamples that it

[121] Rissland & Skalak, *supra* note 33, at 853.

cannot explain away.[122] The SMILE component employs machine learning to learn how to identify the factors in textual descriptions of cases. In the combined program, an algorithm employs SMILE to process a problem input as text and outputs a list of applicable factors to IBP, which then predicts an outcome.[123]

Other Syllabus research papers provide good examples of processes involving machine learning and automatically extracting information from texts, including statutory texts,[124] case texts,[125] and texts involved in pre-trial discovery.[126] The work by Francesconi and colleagues illustrates a nice integration of bottom-up and top-down processing to focus human annotators on statistically interesting features that may deserve manual inclusion in an ontology.[127]

An existing Digital Documents Technology for pretrial e-discovery[128] implements an iterative process for identifying documents relevant to a lawsuit. The authors focus on senior litigators' "sensemaking": the "process of collecting, organizing and creating representations of complex information sets [i.e, the e-documents], all centered around some problem they need to understand [i.e., how they relate to the litigation]."[129] The goal is to tease out the litigators' relevance hypotheses: more-or-less abstract descriptions of subject matter that, if found in a document, would make that document relevant to the law suit.[130] The paper describes an iterative user modeling process to elicit these relevance hypotheses. It is a computer-mediated process of successive refinement in which the litigator communicates the information needed, sample documents are retrieved, and the litigator confirms whether they are responsive. If they are not responsive, the litigator refines the hypothesis. This process yields a model of the litigator's objectives, the legal and other concepts relevant to the case and their level of specificity, and the variety of ways of expressing the concept. This information, in effect, operationalizes the litigator's hypothe-

[122] Ashley & Brüninghaus, *Computer Models*, *supra* note 30, at 347.

[123] Ashley & Brüninghaus, *Automatically Classifying Case Texts*, *supra* note 58, at 139-40.

[124] Francesconi & Passerini, *supra* note 59; de Maat et al., *supra* note 59.

[125] Jackson et al., *supra* note 52.

[126] Privault et al., *supra* note 65.

[127] Francesconi et al., *supra* note 48.

[128] Hogan et al., *supra* note 64, at 434-35.

[129] Robert Bauer, Teresa Jade, Bruce Hedin & Christopher Hogan, *Automated Legal Sensemaking: The Centrality of Relevance and Intentionality*, 2008 PROC. OF THE SECOND INT'L WORKSHOP ON SUPPORTING SEARCH & SENSEMAKING FOR ELECTRONICALLY STORED INFO. IN DISCOVERY SESSION 2, Paper No. 5, 2, *available at* http://eprints.ucl.ac.uk/9131/1/9131.pdf.

[130] Hogan et al., *supra* note 64, at 446-48, 455.

sis and is then used to retrieve more documents relevant to that hypothesis. As machine learning develops a probabilistic model of concepts of interest, tools can help litigators visualize how closely new documents relate to the conceptual clusters of relevant documents.[131]

Over the years, in writing papers for the AI and Law Seminar at the University of Pittsburgh School of Law, students have developed their own representations of processes involving legal knowledge. Some representative titles include:

- "Artificial Intelligence Applications in Prior Art Searches: An Overview and Forecast for the Patent Literature Search Industry"
- "AI Improvements to Gene Patent Application Processing"
- "For the Truth of the Story Asserted: Using AI to Assist Juries in Coherent Reconstruction of Facts at Trial"
- "Adapting AI and Law Models to Predict Litigation Outcomes by Normalizing Against a Null-Rule"

Lesson D2: Processes underlying legal Digital Documents Technologies involve different kinds of texts as inputs. The processes underlying the Digital Documents Technologies involve texts as inputs; students in the AI and Law Seminar can thus learn about the challenges of computationally dealing with texts.

Most of the computational models of legal reasoning cited in the Syllabus do *not* accept texts as inputs. Instead, a human reads the texts of a problem scenario or of the cases to be stored in the program's knowledge base, extracts the information called for by the program and manually represents the problem or cases in the appropriate formats. Thus, the inputs are, in a sense, manually-processed texts. The main exception is the IBP+SMILE program described above.

In the processes and domains of the Digital Documents Technologies, too many documents exist to process each one manually. Legal IR providers like Lexis and Westlaw acquire thousands of cases per day. The case texts are submitted electronically by the courts, and indexed automatically in an inverted index, in which a given document is indexed by every "major" word (that is, with stop words like "the," "a," "an," *et cetera*, filtered out of the indexing terms). Thus, legal IR programs search millions of documents in response to a query; the AI and Law models search at most hundreds (but some, like SPIRE,[132] do connect to full-text legal IR sources, thus expanding their searches).

[131] Privault et al., *supra* note 65.

[132] Daniels & Rissland, *supra* note 53.

The models also deal with very different kinds of texts. Some of the legal text processing research deals with more structured texts whose functions in a process of legal administration are known (such as statutory texts or the texts of a court decision). In the e-discovery context, by comparison, the texts are extremely heterogeneous documents produced in litigation. They include not only corporate memoranda and agreements, but also the full range of internet-based communications, including emails. The emails may be from managers to and from employees, or from employees to other employees, customers and suppliers, or to employees' families and friends, *et cetera*. In fact, for emails, a social network analysis of who is communicating with whom over what time frame may provide valuable information for selecting relevant texts for further analysis.[133]

Lesson D3: Digital Documents Technologies find texts relevant to legal problems. Digital Documents Technologies tend to focus on a process of *finding* information relevant to legal problems. This process affects the way in which such a system assesses relevance, the way the system is evaluated, and the measures used to assess its performance. Law students will find that IR relevance, evaluation, and measures are defined succinctly in the Syllabus materials on e-discovery:

> *Relevance*: In IR tasks, "[a] correctly returned document (broadly conceived as any container of information) is considered relevant if the user would wish to see it, and not relevant otherwise. . . . "
>
> *Evaluation:* "An important consequence of relevance being an opinion (rather than an objectively determinable fact) is that retrieval *effectiveness* is a principal focus for evaluation. . . . [T]he vast majority of IR evaluation is concerned principally with just one aspect of relevance: topicality. . . . [T]he most widely used definition of *topical relevance* by IR researchers is substantive treatment of the desired topic by any part of the document."
>
> *Measures:* "The effectiveness of a retrieval approach is then measured by its ability to retrieve, for each topic, those documents which have positive assessments for that topic. Assuming binary (i.e., relevant vs. non-relevant) assessments, two measures of effectiveness are very commonly reported. *Recall* is the proportion of the extant relevant documents that were retrieved by the system, while *precision* is the proportion of retrieved documents which were in fact relevant. Together they reflect a user-centered view of the fundamental tradeoff between false positives and false negatives."[134]

Topicality in a legal context is a moving target. Although the Westlaw key numbering system provides a rich list of topics spanning

[133] Henseler, *supra* note 63.

[134] Oard et al., *supra* note 61, at 360, 362, 363 (emphasis in original).

legal subject matter, attorneys often seek more detailed topics. Attorneys seek to capture factual patterns in the targeted cases that are relevant to the problem situation (that is, the source case) they are re-researching. In e-discovery, no such list exists. The topics of relevance are the abstract descriptions known primarily to the litigator; indeed, as described above, providers of Digital Documents Technology need to develop special processes to induce the topics based on litigators' determinations of the relevance of documents.

The Syllabus materials provide law students with a readable account of how full text legal information retrieval systems like Westlaw and Lexis use probabilities based on frequencies of terms in documents and in the corpus to model the topic a user deems important (that is, the user's information need). Moreover, these materials explain how information retrieval systems estimate documents' relevance to the user's information need, achieving high precision in a context in which recall usually is of somewhat secondary importance.[135] This system works well in legal research where an exhaustive search for every relevant case is rarely necessary; a handful of good cases are enough to develop an understanding of the issue and to formulate an argument on how to decide it.[136]

In e-discovery, however, retrieving "any and all" documents pertaining to a topic is the typical goal.[137] That is, there is a premium on recall. This fact is unfortunate for two reasons. First, research shows that attorneys tend to overestimate their rates of recall.[138]

> That [seminal] study established a gap between the perception on the part of lawyers that using their specific queries they would retrieve on the order of 75% of the relevant evidence to be found in a collection of 40,000 documents gathered for litigation purposes, whereas the researchers were able to show that only about 20% of relevant documents had in fact been found.[139]

Second, "[i]t is now well understood that as data sets get larger, high-precision searches generally become somewhat easier, but ''indeterminacy multiplies making it increasingly difficult to conduct successful specific or exhaustive searches.''[140]

[135] Turtle, *supra* note 49, at 7-9.

[136] Oard et al., *supra* note 61, at 353.

[137] *Id.* at 351.

[138] Blair & Maron, *supra* note 51, at 293-94.

[139] Oard et al., *supra* note 61, at 348-49.

[140] *Id.* at 349 (citation omitted).

These factors have consequences for legal practitioners, who as noted above, are responsible for understanding, selecting, interpreting, and justifying e-discovery processes and tools, a daunting task given the welter of vendors' alternative products and conflicting claims and the dearth of objective evaluations.

Lesson D4: Digital Documents Technologies and their process models need empirical methodologies for testing. For law students who increasingly face the need to deal with models of legal information proprocesses, either as consumers or designers, the AI and Law Seminar teaches an empirical methodology for testing process models. The history of progress in AI and Law has been marked by testing the models empirically or in thought experiments, focusing on examples where the model breaks down and then adapting it. This history includes the invention of experimental techniques for evaluating the models.[141] The progression of work from Berman's and Hafner's first call for representing the teleological underpinnings of case-based reasoning,[142] to CATO's Factor Hierarchy,[143] to hypothesis-testing in IBP,[144] to theory-induction,[145] to argument schemes,[146] to the value judgment model,[147] is an instance of example-driven incremental and continuing refinement of a legal information process model. Another example is the iterative refinement in models of hypothetical reasoning.[148]

The Syllabus provides a treatment of the challenges of and methods for objectively evaluating e-discovery IR tools.[149] It introduces students to the TREC Legal Track, in which teams from academia and industry vie annually to field and evaluate the most effective e-discovery IR tools applied to publicly available datasets. Publicly available datasets include the collections of emails and other documents in the Tobacco litigation and the Enron scandal. The requirements of the TREC competition, namely the increased numbers of e-documents and the need to compare entries in terms of three aspects of system effectiveness for realistic discovery, have driven the TREC organizers to

[141] *See, e.g.,* Branting, *supra* note 26; Ashley & Brüninghaus, *Computer Models, supra* note 30; McLaren, *supra* note 36.

[142] Berman & Hafner, *supra* note 34.

[143] *See* Aleven, *supra* note 29, at 191-93.

[144] *See* Ashley & Brüninghaus, *Computer Models, supra* note 30.

[145] Bench-Capon & Sartor, *supra* note 35.

[146] Gordon & Walton, *supra* note 38; Atkinson & Bench-Capon, *supra* note 37.

[147] Grabmair & Ashley, *supra* note 39.

[148] *See supra* note 100.

[149] Oard et al., *supra* note 61.

extend the experimental techniques of the Blair and Maron study[150] into large scale evaluations employing many law students, paralegals, and lawyers as volunteer relevance assessors.[151]

As potential consumers of this technology, it is useful for law students to learn that the e-discovery task is not monolithic. While there are three aspects to the TREC competition, all focus on "responsive review: . . . in which a party . . . is served . . . with a request for production of documents and, . . . must find and produce . . . any and all documents that are responsive to the request."[152] The specific three aspects focused on are:

1. Interactive task (i.e., retrieving all and only the documents consistent with a topic authority's definition of what is relevant);[153]
2. Ad Hoc task (i.e., a single pass, first-pass automatic search);[154] and
3. Relevance Feedback task (i.e., a second pass search based on human feedback on first-pass results).[155]

As the OARD article makes clear, the techniques for evaluating the three tasks differ, and there are other tasks for which the evaluation techniques have yet to be invented.[156]

Armed with a conceptual understanding of IR relevance, evaluation, and measures as they are applied to e-discovery and an empirical and example-based methodology for testing claims about process models, students will be well-prepared to critically question vendors' claims regarding e-discovery tools, to use the tools effectively in their own e-discovery processes, and to justify the tools to clients and judges.

Lesson D5: Finding texts relevant to legal problems is different from applying relevant texts to solve legal problems. Having studied some Digital Documents Technologies, as well as some computational mod-

[150] Blair & Maron, *supra* note 51.

[151] Oard et al., *supra* note 61, at 367.

[152] *Id.* at 370.

[153] *Id.*

[154] *Id.* at 373.

[155] *Id.* at 375.

[156] *Id.* at 370. "Early in the lawsuit, an attorney may conduct exploratory searches of a document collection in order to test hypotheses and to build a theory of the case; further along, an attorney may search for documents relevant to the activities of a particular individual in order to prepare for the deposition of a witness; as trial approaches, an attorney may search through the set of generally relevant documents in order to find the small subset that he or she would like to enter as exhibits in trial; and so on."

els of legal reasoning, students will be in a good position to consider how the two approaches might be integrated.

Why might such integration be worthwhile? If the history of the legal services market has indeed progressed from customization through standardization, systematization, and packaging, to commoditization, what, one may ask, comes next? As efficient as it may be to produce, commoditized legal work is no longer customized to solve a client's particular problem. While it may be too expensive to employ human attorneys to adapt the commoditized product to solve the client's problem, computational techniques may yet perform such "post-commoditization" customization efficiently.

As noted above, the Digital Documents Technologies tend to focus on a process of *finding* information relevant to legal problems. By contrast, the computational models of legal reasoning tend to focus on a process of *applying* relevant information to solve legal problems. As a result, the two approaches have complementary advantages and disadvantages.

The measure of relevance in information retrieval (that is, the statistical probability that a document substantively treats the desired topic) is remarkably effective and requires no special document representation; the documents are texts in an inverted index. On the other hand, only a human can use the documents so retrieved to solve a legal problem by, say, incorporating the information contained within the document into a legal argument. The computational models of legal reasoning, on the other hand, employ relevance measures, like those in Table 1, and can use the retrieved information to solve a legal problem by generating an argument. One downside, however, is that these relevance measures can only work if the cases are specially represented in terms of dimensions, factors, critical facts, *et cetera*. And, with the exception of efforts like SPIRE[157] or SMILE+IBP[158], a program cannot fill out the special representations automatically.

Nevertheless, one begins to see programs that are bridging the gap between Digital Documents Technologies and computational models of legal reasoning.

Lesson D6: Computational models of legal reasoning can be a bridge between legal texts and legal problem solving. Some existing and current work in AI and Law employs computational models of legal rea-

[157] Daniels & Rissland, *supra* note 53.

[158] *See* Ashley & Brüninghaus, *Automatically Classifying Case Texts, supra* note 58.

soning and argument as a bridge between legal texts and the answers and arguments humans seek.

The SPIRE program, for example, demonstrated how to use a factor-based computational model of a bankruptcy law issue (like whether a bankruptcy plan has been submitted in good faith) to seed queries into a full-text legal information retrieval system. The program used the Hypo-style computational model to find the most relevant cases for solving a legal problem and used these cases to seed a query; in effect, instructing the full-text IR system to retrieve more similar texts.[159] In experiments, SPIRE found new relevant cases very similar to the inputted problems (for example, involving the same kind of legal stories). A SPIRE user could also indicate the particular features of interest, and the program would automatically highlight the parts of the text of the retrieved cases that correspond to that feature.

Other examples involve applying pattern matching[160] and machine learning techniques to categorize statutory provision texts in terms of their broad regulatory functions (e.g., as a definition, liability, prohibition, duty, permission, or penalty). These techniques can then be used to extract typical features associated with each function.[161] For instance, the provision, "A controller intending to process personal data falling within the scope of application of this Act shall have to notify the Guarantor thereof," is classified as "duty" with the associated features: Bearer (of the duty) = "controller", Action = "notification", Counterpart = "Guarantor", and Object = "process personal data". The aim of learning classifiers and features of statutory provisions in this work is to extract legal rules directly from legislative texts. Ultimately, a program could reason with formalized rules so extracted. As of yet, no reported work has succeeded in extracting legal rules from the statutory texts for this purpose. However, in the meantime, the information extracted could be mapped into comprehensive conceptual indices (like legal thesauri, dictionaries, or ontologies) and used to support more focused conceptual queries for all legal rules relevant to particular business compliance issues, for example.

Other work involves annotating legal decision texts in a way that connects them to trees of rule-based inferences so that a program can learn to reason about previously unseen texts. For instance, efforts are aimed at integrating NLP techniques employed in the IBM Watson

[159] *Id.* at 131-32.

[160] de Maat et al., *supra* note 59.

[161] Francesconi et al., *supra* note 48.

question-answering project[162] with defeasible logic models that represent statutory and regulatory requirements as trees of rule conditions linked to chains of reasoning in the legal decisions that connect the evidentiary input to the findings of fact.[163] A proposition of interest (for example, that a statutory provision has been satisfied) is thus decomposed into an inference tree consisting of legal propositions and related factual assertions whose expressions can be annotated in the natural language texts of legal decisions. This use increases the granularity of the legal concepts to the point where they can be linked closely to common sense reasoning about facts and used to facilitate more effective searching, extraction, and reporting. For example, although high-level legal concepts (such as "entitled to compensation") would pose a vague retrieval task, lower-level concepts into which they are decomposed (such as "medical condition," or "time of onset") would be more amenable to automated search and reasoning about new texts.

Given a sufficient number of annotated decisions, machine learning and natural language processing can be applied so that a program can learn to identify the annotated patterns in new texts it has not yet seen, just as IBM's Watson program did in a question answering context. As noted above, in at least one law school lab setting, law students are engaged in the process of annotating statutory and evidentiary arguments in legal decisions.[164] In so doing, these students are learning a comprehensive model of legal reasoning and encountering *all* of the lessons recited above—lessons about legal rules, cases, arguments, Digital Documents Technologies, and the process engineering that connects them.

Apart from Digital Documents Technologies, AI and Law process models are being developed in a variety of areas. The INDiGO project and work by Governatori[165] are based on meticulous representation of regulatory, administrative, and business processes and careful consideration of how AI expert systems can be integrated into those processes to assist humans. In the INDiGO project, for example, the processes in focus include not only applying regulations, but also improving them in light of experience. An expert system assists immigration agents in the Netherlands with processing clients in accordance with voluminous regulatory provisions. At the same time, the system collects feed-

[162] Ferrucci et al., *supra* note 66, at 67.

[163] Ashley & Walker, *supra* note 67.

[164] *See* Law, Logic & Technology Research Laboratory, *supra* note 16.

[165] Governatori & Shek, *supra* note 70; Theunisz, *supra* note 72.

back and data about problems arising under the regulations to pass along to regulators and rule modelers concerning the aspects of the rules that need to be improved. Legal process engineering can be expected to grow, offering opportunities to law students who understand its potential.

CONCLUSION

This article has presented a sample syllabus for an AI and Law Seminar to teach law and legal practice in a digital age. The discussion has drawn out four sets of pedagogical lessons implicit in the material. The AI and Law Seminar not only teaches students valuable pedagogical lessons about legal rules, cases and arguments, but also introduces them to process models and algorithms for legal practice in a digital age. In this respect, the Seminar makes a timely contribution to the law students' education and addresses the current exigency facing law schools in uncertain times for the legal profession.

AI and Law models of legal reasoning and legal text processing add value to real world applications, including e-discovery, visualizing legal arguments, predicting outcomes, making settlement decisions, and legal expert systems. Law students *can* contribute to this technology as informed users and also as annotators, designers, and inventors of new models.

With the benefit of these lessons, law students would be ideally prepared to apply them in some practical context. For instance, an AI and Law practicum in the subsequent semester could serve as a kind of laboratory course.[166] Law students could build their own legal expert system using the tools described above or explore how such tools could be extended to accommodate missing features of legal reasoning. These missing features include: reasoning with cases and dealing with uncertainty;[167] experimenting with applying predictive coding to publicly available e-discovery datasets;[168] participating in the Trec Legal Track as volunteer relevance assessors;[169] and participating in diagramming arguments and annotating legal documents according to a computational model of legal argument so that machine learning tools can be applied.[170]

[166] *See supra* Section III.

[167] *See supra* note 113 and accompanying text.

[168] *See supra* note 149 and accompanying text.

[169] *Id.*

[170] *See supra* note 163 and accompanying text.

Some of those law students may even contribute new computational models and new approaches to knowledge representation as AI and Law researchers turn to the challenges of modeling legal planning and creative legal problem solving.

APPENDIX

Sample Syllabus for AI and Law Seminar

Part One: Computational Models of Reasoning with Legal Rules and Cases

I. Introduction to Computational Models of Legal Reasoning

- Edwina L. Rissland, Artificial Intelligence and Law: Stepping Stones to a Model of Legal Reasoning, 99 YALE L. J. 1957 (1990).
- DONALD A. WATERMAN & MARK A. PETERSON, MODELS OF LEGAL DECISIONMAKING (1981).
- ANNE VON DER LIETH GARDNER, AN ARTIFICIAL INTELLIGENCE APPROACH TO LEGAL REASONING (1987).

II. Logical Models of Statutory Reasoning

- Marek J. Sergot, Fariba Sadri, Robert A. Kowalski, Frank Kriwaczek, P. Hammond & H. T. Cory, *The British Nationality Act as a Logic Program,* 29 COMM. OF THE ACM 370 (1986).
- Trevor Bench-Capon & Franz Coenen, *Exploiting Isomorphism: Development of a KBS to Support British Coal Insurance Claims*, 1991 PROC. OF THE THIRD INT'L CONF. ON ARTIFICIAL INTELLIGENCE & L. 62.
- Layman E. Allen & C. Rudy Engholm, *Normalized Legal Drafting and the Query Method,* 29 J. LEGAL EDUC. 380 (1977-1978).
- Donald Berman & Carole Hafner, *Obstacles to the Development of Logic-Based Models of Legal Reasoning, in* COMPUTER POWER AND LEGAL LANGUAGE 183-214 (Charles Walter ed., 1986).

III. Case-Based Models of Legal Reasoning

- L. Thorne McCarty, *An Implementation of* Eisner v. Macomber, 1995 PROC. OF THE FIFTH INT'L CONF. ON ARTIFICIAL INTELLIGENCE & L. 276.
- Kevin D. Ashley, *Reasoning with Cases and Hypotheticals in HYPO,* 34 INT'L J. OF MAN-MACH. STUD. 753 (1991).

- L. Karl Branting, *Building Explanations from Rules and Structured Cases*, 34 INT'L J. OF MAN-MACH. STUD. 797 (1991).

IV. Models for Predicting Legal Outcomes

- Vincent Aleven, *Using Background Knowledge in Case-Based Legal Reasoning: A Computational Model and an Intelligent Learning Environment*, 150 ARTIFICIAL INTELLIGENCE 183 (2003).
- Kevin D. Ashley & Stefanie Brüninghaus, *Computer Models for Legal Prediction*, 46 JURIMETRICS 309 (2006).

V. Models Integrating Cases, Statutes, Rules, Concepts, and Values

- Edwina L. Rissland & David B. Skalak, *CABARET: Statutory Interpretation in a Hybrid Architecture*, 34 INT'L J. OF MAN-MACH. STUD. 839 (1991).
- Donald H. Berman & Carol Hafner, *Representing Teleological Structure in Case-Based Legal Reasoning: The Missing Link*, 1993 PROC. OF THE FOURTH INT'L CONF. ON ARTIFICIAL INTELLIGENCE & L. 50.
- Trevor Bench-Capon & Giovanni Sartor, *A Model of Legal Reasoning with Cases Incorporating Theories and Values*, 150 ARTIFICIAL INTELLIGENCE 97 (2003).
- Bruce M. McLaren, *Extensionally Defining Principles and Cases in Ethics: An AI Model*, 150 ARTIFICIAL INTELLIGENCE 145 (2003).

Part Two: Defeasible Legal Reasoning with Argument Schemes

VI. Assessing Legal Claims with Argument Schema

- Thomas F. Gordon & Douglas Walton, *Legal Reasoning with Argumentation Schemes*, 2009 PROC. OF THE TWELFTH INT'L CONF. ON ARTIFICIAL INTELLIGENCE & L. 137.
- Katie Atkinson & Trevor Bench-Capon, *Argumentation and Standards of Proof*, 2007 PROC. OF THE ELEVENTH INT'L CONF. ON ARTIFICIAL INTELLIGENCE & L. 107.
- Matthias Grabmair & Kevin D. Ashley, *Facilitating Case Comparison Using Value Judgments and Intermediate Legal Concepts*, 2011 PROC. OF THE THIRTEENTH INT'L CONF. ON ARTIFICIAL INTELLIGENCE & L. 161.
- Vern R. Walker, Nathaniel Carie, Courtney C. DeWitt & Eric Lesh, *A Framework for the Extraction and Modeling of Fact-Finding Reasoning from Legal Decisions: Lessons from the Vaccine/Injury Project Corpus*, 19 ARTIFICIAL INTELLIGENCE & L. 291 (2011).

VII. Representing Legal Concepts and Case Knowledge in Ontologies

- Kevin D. Ashley & Stefanie Brüninghaus, *A Predictive Role for Intermediate Legal Concepts*, PROC. OF THE SIXTEENTH ANN. CONF. ON LEGAL KNOWLEDGE & INFO. SYS. 153 (2003).
- Kevin D. Ashley, *Ontological Requirements for Analogical, Teleological, and Hypothetical Legal Reasoning*, 2009 PROC. OF THE TWELFTH INT'L CONF. ON ARTIFICIAL INTELLIGENCE & L. 1.
- Trevor J.M. Bench-Capon & Peppin R.S. Visser, *Ontologies in Legal Information Systems; The Need for Explicit Specifications of Domain Conceptualisations*, 1997 PROC. OF THE SIXTH INT'L CONF. ON ARTIFICIAL INTELLIGENCE & L. 132.
- Joost Breuker, Andre Valente & Radboud Winkels, *Legal Ontologies in Knowledge Engineering and Information Management*, 12 ARTIFICIAL INTELLIGENCE & L. 241 (2004).
- Enrico Francesconi, Simonetta Montemagni, Wim Peters & Daniela Tiscornia, *Integrating a Bottom-Up and Top-Down Methodology for Building Semantic Resources for the Multilingual Legal Domain*, *in* SEMANTIC PROCESSING OF LEGAL TEXTS: WHERE THE LANGUAGE OF LAW MEETS THE LAW OF LANGUAGE 95 (Enrico Francesconi, Simonetta Montemagni, Wim Peters & Daniela Tiscornia eds., 2010).

Part Three: Legal Information Retrieval, Information Extraction, and Text Processing

VIII. How Full-Text Legal Information Retrieval Works

- Howard Turtle, *Text Retrieval in the Legal World*, 3 ARTIFICIAL INTELLIGENCE & L. 5 (1995).
- David C. Blair & M. E. Maron, *An Evaluation of Retrieval Effectiveness for a Full-Text Document-Retrieval System*, 28 COMM. OF THE ACM 289 (1985).
- Eugene Charniak, *Bayesian Networks Without Tears*, 12 AI MAGAZINE, Winter 1991, at 50.

IX. Combining Legal IR and AI to Analyze Legal Claims

- Jody J. Daniels & Edwina L. Rissland, *Finding Legally Relevant Passages in Case Opinions*, 1997 PROC. OF THE SIXTH INT'L CONF. ON ARTIFICIAL INTELLIGENCE & L. 39.
- M. Saravanan, B. Ravindran & S. Raman, *Improving Legal Information Retrieval Using an Ontological Framework*, 17 ARTIFICIAL INTELLIGENCE & L. 101 (2009).
- Peter Jackson, Khalid Al-Kofahi, Alex Tyrrell & Arun Vachher, *Information Extraction from Case Law and Retrieval of Prior Cases*, 150 ARTIFICIAL INTELLIGENCE 239 (2003).

- Edwina L. Rissland & M. Timur Friedman, *Detecting Change in Legal Concepts*, 1995 PROC. OF THE FIFTH INT'L CONF. ON ARTIFICIAL INTELLIGENCE & L. 127.
- Daniel E. Rose & Richard K. Belew, *A Connectionist and Symbolic Hybrid for Improving Legal Research*, 35 INT'L J. OF MAN-MACH. STUD. 1 (1991).
- Edwina Rissland, David Skalak & Timur Friedman, *BankXX: Supporting Legal Arguments through Heuristic Retrieval*, 4 ARTIFICIAL INTELLIGENCE & L. 1 (1996).

X. Legal Information Extraction and Text Processing

- Kevin D. Ashley & Stefanie Brüninghaus, *Automatically Classifying Case Texts and Predicting Outcomes*, 17 ARTIFICIAL INTELLIGENCE & L. 125 (2009).
- Enrico Francesconi & Andrea Passerini, *Automatic Classification of Provisions in Legislative Texts*, 15 ARTIFICIAL INTELLIGENCE & L. 1 (2007).
- Emile de Maat, Kai Krabben & Radboud Winkels, *Machine Learning Versus Knowledge Based Classification of Legal Texts*, 2010 PROC. OF THE TWENTY-THIRD ANN. CONF. ON LEGAL KNOWLEDGE & INFO. SYS. 87.

XI. AI and Law Tools for E-Discovery: Obtaining Evidence Relevant to Lawyers' Hypotheses about Claims and Documents

- Kevin Ashley & Will Bridewell, Emerging AI & Law Approaches to Automating Analysis and Retrieval of Electronically Stored Information in Discovery Proceedings, 18 ARTIFICIAL INTELLIGENCE & L. 311 (2010).
- Douglas W. Oard, Jason R. Baron, Bruce Hedin, David D. Lewis & Stephen Tomlinson, Evaluation of information Retrieval for E-discovery, 18 ARTIFICIAL INTELLIGENCE & L. 347 (2010).
- Hans Henseler, Network-based Filtering for Large Email Collections in E-Discovery, 18 ARTIFICIAL INTELLIGENCE & L. 413 (2010).
- Christopher Hogan, Robert S. Bauer & Daniel Brassil, Automation of Legal Sensemaking in E-discovery, 18 ARTIFICIAL INTELLIGENCE & L. 431 (2010).
- Caroline Privault, Jacki O'Neill, Victor Ciriza & Jean-Michel Renders, A New Tangible User Interface for Machine Learning Document Review, 18 ARTIFICIAL INTELLIGENCE & L. 459 (2010).
- David D. Lewis, Afterword: Data, Knowledge, and E-discovery, 18 ARTIFICIAL INTELLIGENCE & L. 481 (2010).

Part Four: The Future of AI and Law: Bridging Computational Models and Legal Texts

XII. Near Term Developments in AI and Law

- David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer & Chris Welty, *Building Watson: An Overview of the DeepQA Project*, 31 AI MAGAZINE, Fall 2010, at 59.
- Kevin D. Ashley & Vern R. Walker, Automated Monitoring of Legal-Rule Compliance Using DeepQA: Screening Legal Documents for Argumentation Evidence, (unpublished manuscript) (on file with authors).
- Guido Governatori & Sidney Shek, *Rule Based Business Process Compliance*, 874 PROC. OF THE RULEML2012@ECAI CHALLENGE, AT THE SIXTH INT'L SYMP. ON RULES Paper No. 5 (2012), *available at* http://ceur-ws.org/Vol-874/paper17.pdf.
- Matthias Grabmair & Kevin D. Ashley, *A Survey of Uncertainties and their Consequences*, *in* PROBABILISTIC LEGAL ARGUMENTATION IN BAYESIAN ARGUMENTATION 61 (Frank Zenker ed., 2012).
- Audrey Theunisz, INDiGO: Rules-driven Business Services; Flexibility Within the Boundaries of the Law, Presentation of the Dutch Immigration and Naturalisation Service (Oct. 5, 2010), *available at*http://www.servicetechsymposium.com/soa_archive/pdf_berlin/Audrey_Theunisz_Rules_Driven.pdf. (In the absence of a published paper in English on INDiGO, this seems to be the most informative document publicly available.)
- Benjamin N. Grosof, Yannis Labrou & Hoi Y. Chan, *A Declarative Approach to Business Rules in Contracts: Courteous Logic Programs in XML*, 1999 PROC. OF THE FIRST ACM CONF. ON ELECTRONIC COM.

Addenda

XIII. Intelligent Tutoring Systems in Law and Ethical Reasoning

- Vincent Aleven, Using Background Knowledge in Case-Based Legal Reasoning: A Computational Model and an Intelligent Learning Environment, 150 ARTIFICIAL INTELLIGENCE 183 (2003).
- Chad S. Carr, Using Computer Supported Argument Visualization to Teach Legal Argumentation, in VISUALIZING ARGUMENTATION: SOFTWARE TOOLS FOR COLLABORATIVE AND EDUCATIONAL SENSE-MAKING 75 (Paul A. Kirschner, Simon J. Buckingham Shum & Chad S. Carr, eds., 2003).

- Kevin D. Ashley, Teaching a Process Model of Legal Argument with Hypotheticals, 17 ARTIFICIAL INTELLIGENCE & L. 321 (2009).
- Antoinette J. Muntjewerff, ICT in Legal Education, 10 GER. L. J. 669 (2009).
- Kevin Ashley & Ilya Goldin, Toward AI-enhanced Computer-supported Peer Review in Legal Education, 2011 PROC. OF THE TWENTY-FOURTH ANN. CONF. ON LEGAL KNOWLEDGE & INFO. SYS. 1.
- Collin Lynch, Kevin D. Ashley & Mohammad H. Falakmasir, Comparing Argument Diagrams, 2012 PROC. OF THE TWENTY-FIFTH ANN. CONF. ON LEGAL KNOWLEDGE & INFO. SYS. 81.

XIV. Legal and Ethical Issues Related to AI and Law Programs

- Samir Chopra & Laurence White, A Legal Theory For Autonomous Artificial Agents 119 (2011).
- Taiwo A. Oriola, *The Use of Legal Software by Non-Lawyers and the Perils of Unauthorised Practice of Law Charges in the United States: A Review of Jayson Reynoso Decision*, 18 Artificial Intelligence & L. 285 (2010).
- Emad A. R. Dahiyat, *Intelligent Agents and Liability: Is It a Doctrinal Problem or Merely A Problem of Explanation?*, 18 Artificial Intelligence & L. 103 (2010).
- Woodrow Barfield, *Intellectual Property Rights in Virtual Environments: Considering the Rights of Owners, Programmers and Virtual Avatars*, 39 Akron L. Rev. 649 (2006).