Cambridge University Press
978-1-107-17150-3 — Artificial Intelligence and Legal Analytics
Kevin D. Ashley
Excerpt
More Information

1

Introducing AI & Law and Its Role in Future Legal Practice

1.1. INTRODUCTION

Artificial Intelligence and Law (AI & Law), a research field since the 1980s with roots in the previous decades, is about to experience a revolution. Teams of researchers in question answering (QA), information extraction (IE), and argument mining from text planted the seeds of this revolution with programs like IBM's Watson and Debater and the open-source information management architectures on which these programs are based. From these seeds, new applications for the legal domain are sure to grow. Indeed, they are growing now. This book explains how.

Programs like Watson and Debater will not perform legal reasoning. They may be able to answer legal questions in a superficial sense, but they cannot explain their answers or make legal arguments. The open-source text analysis tools on which they are based, however, will make a profound difference in the development of new legal applications. They will identify argument-related information in legal texts that can transform legal information retrieval into a new kind of conceptual information retrieval: argument retrieval (AR).

Computational models developed by AI & Law researchers will perform the legal reasoning. The newly extracted argument-related information will connect the computational models of legal reasoning (CMLRs) and argument directly with legal texts. The models can generate arguments for and against particular outcomes in problems input as texts, predict a problem's outcome, and explain their predictions with reasons that legal professionals will recognize and can evaluate for themselves. The result will be a new kind of legal app, one that enables cognitive computing, a kind of collaborative activity between humans and computers in which each performs the kinds of intelligent activities that they can do best.

This chapter introduces the subject of AI & Law and explains the role it will play in light of the new technologies for analyzing legal texts. It explains how these technologies enable new tools for legal practice using computational models of legal reasoning and argumentation developed by AI & Law researchers.

Some questions addressed in this chapter include: What is the subject of Artificial Intelligence and Law? What is a CMLR? What are the new technologies for automated QA, IE, and argument mining from texts? What roles will AI & Law CMLRs and argument play given these new technologies? What are conceptual information retrieval and cognitive computing, and what kind of legal app will support them?

1.2. AI & LAW AND THE PROMISE OF TEXT ANALYTICS

The goal of much of the research in AI & Law has been to develop CMLRs that can make legal arguments and use them to predict outcomes of legal disputes. A *CMLR* is a computer program that implements a process evidencing attributes of human legal reasoning. The process may involve analyzing a situation and answering a legal question, predicting an outcome, or making a legal argument. A subset of CMLRs implements a process of legal argumentation as part of their reasoning. These are called computational models of legal argument (CMLAs).

CMLRs and CMLAs break down a complex human intellectual task, such as estimating the settlement value of a product liability suit or analyzing an offer and acceptance problem in a first-year contracts course, into a set of computational steps or *algorithm*. The models specify how a problem is input and the type of legal result to output. In between, the model builders have constructed a computational mechanism to apply domain knowledge to perform the steps and transform the inputs to outputs.

In developing these models, researchers address such questions as how to represent what a legal rule means so that a computer program can decide whether it applies to a situation, how to distinguish "hard" from "easy" legal issues, and the roles that cases and values play in interpreting legal rules. Their answers to these questions are not philosophical but scientific; their computer programs not only model legal reasoning tasks but also actually perform them; and the researchers conduct experiments to evaluate how well their programs perform.

While AI & Law researchers have made great strides, a knowledge representation bottleneck has impeded their progress toward contributing to legal practice. So far, the substantive legal knowledge employed by their computational models has had to be extracted *manually* from legal sources, that is, from the cases, statutes, regulations, contracts, and other texts that legal professionals actually use. That is, human experts have had to read the legal texts and represent relevant parts of their content in a form the computational models could use. An inability to automatically connect their CMLRs directly to legal texts has limited the researchers' ability to apply their programs in real-world legal information retrieval, prediction, and decision-making.

Recent developments in computerized QA, IE from text, and argument mining promise to change that. "A Question-answering system searches a large text collection and finds a short phrase or sentence that precisely answers a user's question" (Prager et al., 2000). "Information extraction is the problem of summarizing the essential details particular to a given document" (Freitag, 2000). Argument mining involves automatically identifying argumentative structures within document texts, for instance, premises and conclusion, and relationships between pairs of arguments (ACL-AMW, 2016). All three technologies usually rely, at least in part, on applying machine learning (ML) to assist programs in processing semantic information in the texts.

A more general term for these techniques, *text analytics* or text mining, "refers to the discovery of knowledge that can be found in text archives . . . [It] describes a set of linguistic, statistical, and machine learning techniques that model and structure the information content of textual sources for business intelligence, exploratory data analysis, research, or investigation" (Hu and Liu, 2012, pp. 387–8). When the texts to be analyzed are legal, we may refer to "legal text analytics" or more simply "legal analytics," the "deriving of substantively meaningful insight from some sort of legal data," including legal textual data (Katz and Bommarito, 2014, p. 3).

The text analytic techniques may open the knowledge acquisition bottleneck that has long hampered progress in fielding intelligent legal applications. Instead of relying solely on manual techniques to represent what legal texts mean in ways that programs can use, researchers can automate the knowledge representation process.

As a result, some CMLRs and CMLAs may soon be linked with text analysis tools to enable the construction of a new generation of legal applications and some novel legal practice tools. Specifically, CMLRs and CMLAs developed in the AI & Law field will employ information extracted automatically from legal texts such as case decisions and statutes to assist humans in answering legal questions, predicting case outcomes, providing explanations, and making arguments for and against legal conclusions more effectively than existing technologies can.

In a complementary way, the AI & Law programs can provide answers to questions that are likely on the minds of technologists in commercial laboratories and start-ups: Now that we are able to extract semantic information automatically from legal texts, what can computer programs do with it? And, exactly what kind of information should be extracted from statutes, regulations, and cases? The CMLRs demonstrate how the new text processing tools can accommodate, adapt, and use the structures of legal knowledge to assist humans in performing practical legal tasks.

Some CMLRs and CMLAs could help advanced AI programs make intelligent use of legal sources. Certainly, the extracted information will be used to improve legal information retrieval, helping to point legal professionals more quickly to relevant information, but what more can be done? Can computers reason with the legal information extracted from texts? Can they help users to pose and test legal hypotheses, make legal arguments, or predict outcomes of legal disputes?

The answers appear to be "Yes!" but a considerable amount of research remains to be done before the new legal applications can demonstrate their full potential.

Indeed, that is what this book is about: how best to perform that research. This book will also assist practitioners and others in contributing to this research and in applying the resulting legal apps. This includes commercial firms interested in developing new products and services based on these models and public agencies wishing to modernize their workflows.

1.3. NEW PARADIGMS FOR INTELLIGENT TECHNOLOGY IN LEGAL PRACTICE

The technology of legal practice is changing rapidly. Predictive coding is transforming discovery in litigation. Start-ups like Ravel (Ravel Law, 2015a), Lex Machina (Surdeanu et al., 2011), and the Watson-based Ross (Ross Intelligence, 2015) (see Sections 4.7 and 12.2) are garnering attention and enlisting law firm subscribers. These and other developments in text analytics offer new process models and tools for delivering legal services, promising greater efficiency and, possibly, greater public accessibility.

These changes present challenges and opportunities for young attorneys and computer scientists, but it has not been easy to predict the future of legal practice. Declines in hiring by law firms have led to reductions in the number of law school applicants. Prospective applicants weigh the chances of gainful employment against the size of their student loans and look elsewhere. There is uncertainty about what law-related tasks the technology can perform. After citing press, academic, and commercial predictions of "the imminent and widespread displacement of lawyers by computers," Remus and Levy argue persuasively that the predictions "fail to engage with technical details . . . critical for understanding the kinds of lawyering tasks that computers can and cannot perform. For example, why document review in discovery practice is more amenable to automation than in corporate due diligence work, and why the automation of ... sports stories does not suggest the imminent automation of legal brief-writing" (Remus and Levy, 2015, p. 2).[1]

It is also unclear what law students need to learn about technology. Law firms have long called for law schools to graduate "practice-ready" students but even firms seem confused about the kinds of technology the firms will require, whether to develop the technology in house or rely on external suppliers, and the skills and knowledge that would best prepare law students for evaluating and using the new technologies.

William Henderson, a law professor at Indiana University's Maurer School of Law, has argued that legal *processing engineering* has changed law practice and will

[1] While I agree that these predictions of displacing attorneys are overblown, Remus and Levy have largely overlooked the AI & Law research reported in this book, research that will enable AR and cognitive computing to assist attorneys in legal practice.

continue to do so, necessitating that law schools teach students process engineering skills.

> 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 . . . [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. (Henderson, 2013, pp. 505f)

Process engineering (or "reengineering") has been defined in the business and information management literature as a "change process,"

> the aim of [which] is quick and substantial gains in organizational performance by redesigning the core business process, [addressing] a need to speed up the process, reduce needed resources, improve productivity and efficiency, and improve competitiveness. (Attaran, 2004, p. 585)

Information Technology (IT) has been called "the most effective enabling technology" for such business process reengineering, establishing "easy communication, improving the process performance," and helping "the reengineering effort by modeling, optimizing and assessing its consequences" (Attaran, 2004, p. 595).

Henderson emphasizes the role process engineering has played in the evolution of legal work, a concept he draws from Richard Susskind's *The End of Lawyers?*, according to which legal work is evolving from bespoke (or customized) to standardized, systematized, packaged, and, ultimately, to a commoditized format:

> These changes [from legal work that is bespoke to . . . 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. [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. (Henderson, 2013, pp. 479, 487)

In focusing on the use of process and technology to design cost-efficient methods to deliver legal solutions, Henderson agrees with Susskind that *commoditization* is the culmination of this evolution of legal work.

> A legal commodity . . . is an electronic or online legal package or offering that is . . . made available for direct use by the end user, often on a DIY [Do It Yourself] basis. [T]he word "commodity" in a legal context [refers] to IT-based systems and services . . . [that are] undifferentiated in the marketplace (undifferentiated in

> the minds of the recipients and not the providers of the service). For any given commodity, there may be very similar competitor products. (Susskind, 2010, p. 31ff)

In other words, the result of legal commoditization is a software service or product that anyone can purchase, download, and use to solve legal problems without hiring an attorney, or, in current parlance, a kind of computerized legal application, a "legal app."

1.3.1. *Former Paradigm: Legal Expert Systems*

The two concepts, process engineering and commoditization, raise interesting questions. If process engineering of legal services is rethinking how to deliver "very cheap and very high quality" solutions, who or what will be responsible for tailoring those solutions to a client's particular problem? If, as Susskind mentions, commoditization means "Do It Yourself," does that mean the client is on its own? In other words, what kind of support does the legal app provide? In particular, can the legal app perform some level of customization?

Not so long ago, the paradigm computational model for designing a legal app would have been a legal expert system. As Susskind, the developer of a pioneering legal expert system, defined them,

> "expert systems" are computer applications that contain representations of knowledge and expertise . . . which they can apply – much as human beings do – in solving problems, offering advice, and undertaking a variety of other tasks. In law, the idea is to use computer technology to make scarce expertise and knowledge more widely available and easily accessible. (Susskind, 2010, p. 120f)

Typically, legal expert systems deal with narrow areas of law but have enough "knowledge and expertise" in the narrow domain to ask a client user pertinent questions about his/her problem, to customize its answer based on the user's responses, and to explain its reasons. Their "expertise" comprises *heuristics* that skilled practitioners use in applying legal rules to specific facts. These heuristics are "rules of thumb," frequently useful but not guaranteed to lead to a correct result (Waterman and Peterson, 1981).

The rules are represented in a declarative language specifying their conditions and conclusion. They are derived through a largely manual knowledge acquisition process: manually questioning human experts, presenting them with problem scenarios, inviting them to resolve the problems, and asking them what rules the experts applied in analyzing the problem and generating a solution (Waterman and Peterson, 1981).

Waterman's Product Liability Expert System

Don Waterman's legal expert system (let's call it W-LES) is a classic example from the 1980s of a CMLR that performed limited but automatic legal reasoning around a practical problem. It provided advice on settlement decisions of product liability

[RULE3.1: DEFINITION OF LOSS]
IF the type of the plaintiff's loss is "injury"
THEN assert the plaintiff is injured by the product.

[RULE3.2: DEFINITION OF LOSS]
IF the type of the plaintiff's loss is "decedent"
THEN assert the plaintiff does represent the decedent and the decedent is killed by the product.

[RULE3.3: DEFINITION OF LOSS]
IF the type of the plaintiff's loss is "property-damage"
THEN assert the plaintiff 's property is damaged by the product.

[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 1.1. Heuristic rules defining loss and strict liability (Waterman and Peterson, 1981)

disputes (Waterman and Peterson, 1981). The inputs to W-LES were descriptions of disputes involving product liability. As outputs, W-LES recommended settlement values and explained its analyses.

The recommendations of W-LES whether to settle a legal dispute and for how much were based on heuristic rules, including claims adjusters' rules for calculating damages and "formalized statements of the California legal doctrine for product liability as stated in statutes, court opinions, and legal treatises" (Waterman and Peterson, 1981, p. 15). Figure 1.1 illustrates the program's heuristic rules defining three kinds of losses and the claim of strict liability.

W-LES mechanically processed a fact situation by applying these heuristic rules in a kind of *forward chaining*. Its inference engine cycled through the rules, testing if any could "fire," that is, if a rule's conditions were satisfied by the facts in the database representing the current problem. If so, the applicable rule did fire and its deduced consequences were added to the database. The inference engine repeatedly cycled through its rules until no more rules could apply.

Ideally, by the end of the process, the rules whose conclusions represented a solution to the problem have "fired" successfully, yielding a prediction and an assessment (or in other legal expert systems, a selection and completion of a relevant legal form). The explanation of the result consists of an "audit trail" or trace back through the rules that fired and the satisfied conditions that led to their firing (Waterman and Peterson, 1981).

Other expert systems applied rules through *backward chaining*. The inference engine begins with a set of desired goals, picks one, and cycles through its database of rules (and facts) in search of a rule whose conclusion is the desired goal. Then, it adds that rule's conditions to the set of desired goals and repeats the cycle until all of the goals are satisfied or there are no more rules (or facts) with which to satisfy remaining goals (Sowizral and Kipps, 1985, p. 3).

Waterman faced three design constraints in developing legal expert systems: legal rules vary across jurisdictions; legal rules employ ill-defined legal concepts; and inferences in the proof are uncertain.

First, different states' legal rules of product liability differ, for instance, in whether the rule of contributory or comparative negligence applies. If contributory negligence applies, the plaintiff's negligence eliminates liability. If comparative negligence, the plaintiff's negligence proportionately reduces the plaintiff's recovery. Waterman addressed this problem by representing multiple states' rules and allowing users to specify which rules to apply in order to demonstrate the differences in outcome.

Second, the legal rules employed some legal concepts without defining them (i.e., "imprecise terms" in Waterman's parlance), such as "reasonable and proper" or "foreseeable" (Waterman and Peterson, 1981, p. 18). Waterman considered a number of possible solutions. These included providing more "rules that describe how an imprecise term was used previously in particular contexts," displaying "brief descriptions of instances of prior use of the imprecise term" and letting the user decide, comparing "prior cases in which the term applied, and provid[ing] a numeric rating that indicates the certainty that the rule . . . applies . . . In the end, he settled on having the system ask the user if the term applied" (Waterman and Peterson, 1981, p. 26).

Third, litigators are uncertain about proving factual issues and applicable legal doctrine. Waterman's suggestions included incorporating the uncertainties as additional premises within each rule or treating uncertainties as a separate rule to be applied after other rules have been considered. Users would "consider a case independently of . . . uncertainty, reach a tentative conclusion, and then adjust that conclusion by some probabilistic factor that represents their overall uncertainty about the case" (Waterman and Peterson, 1981, p. 26).

Modern Legal Expert Systems

Although no longer the paradigm, legal expert systems are still widespread in use in a number of contexts.

Neota Logic provides tools for law firms, law departments, and law school students to construct expert systems. Its website offers examples of computerized advisors concerning questions involving, for instance, the FCPA, bankruptcy risks in cross-border transactions, and the Family and Medical Leave Act (Neota Logic, 2016) (see Section 2.5.1).

CALI, the Center for Computer-Assisted Legal Instruction, and IIT Chicago-Kent College of Law's Center for Access to Justice & Technology, overseen by Professor Ron Staudt, provide a web-based tool to author expert systems. Using the tool, non-programmers with legal skills can create expert systems called A2J Guided Interviews® that lead self-represented litigants through a legal process resulting in a document to be filed in court (A2J, 2012).

As discussed in Section 2.5, firms employ management systems with expert-systems-style business rules to monitor whether their processes comply with relevant regulations.

While still widely used, legal expert systems may not be the paradigm "killer app" for the legal domain. There are at least three reasons for this. First, the techniques developed to enable expert systems to deal with uncertain and incomplete information tend to be *ad hoc* and unreliable. Second, the manual process of acquiring rules is cumbersome, time-consuming, and expensive, a knowledge acquisition bottleneck that has limited the utility of expert systems in law and many other fields (Hoekstra, 2010). Third, text analytics cannot solve this particular knowledge acquisition bottleneck. While the new text analytics can extract certain kinds of semantic legal information from text, they are not yet able to extract expert systems rules.

From time to time, we will return to expert systems, their promise, and their limitations in this book; suffice it to say here that if the legal app is to customize solutions to the particularities of the user's problem, it may be necessary to find some other paradigms.

1.3.2. *Alternative Paradigms: Argument Retrieval and Cognitive Computing*

Unlike expert systems, the two alternative paradigms, AR and cognitive computing, do not purport to solve users' legal problems on their own. Instead, computer programs extract semantic information from legal texts and use it to help humans solve their legal problems.

Conceptual information retrieval, of course, is not new. AI has long sought to identify and extract semantic elements from text such as concepts and their relationships. As defined by Sowa, "concepts represent any entity, action, or state that can be described in language, and conceptual relations show the roles that each entity plays" (Sowa, 1984, p. 8). Similarly, it has long been a goal of AI to make information retrieval smarter by using the extracted semantic information to draw inferences about the retrieved texts. Roger Schank employed the term, "conceptual information retrieval" in 1981 to describe:

> a system to deal with the organization and retrieval of facts in relatively unconstrained domains (for example, . . ., scientific abstracts). First, the system should be able to automatically understand natural-language text – both input to the database

> and queries to the system . . . in such a way that the conceptual content or meaning of an item can be used for retrieval rather than simply its key words . . . If categories are specified by concepts, and if the natural-language analyzer parses text into a conceptual representation, then inferences can be made from the conceptual representations (or meanings) of new items to decide which categories they belong in. (Schank et al., 1981, pp. 98, 102)

Nor is conceptual legal information retrieval new. Pioneering efforts to achieve conceptual retrieval in the legal domain were undertaken by Hafner (1978) and Bing (1987). As discussed in Sections 7.7 and 11.2, modern legal IR services take into account the substantive legal concepts and topics of interest that users intend to target. Other recent work has focused on extending conceptual information retrieval systems so that they return legal information conceptually related not just to the query but to the problem to which the user intends to apply the targeted information (see Winkels et al., 2000).

Today, *conceptual legal information retrieval* can be defined as automatically retrieving relevant textual legal information based on matching concepts and their roles in the documents with the concepts and roles required to solve the user's legal problem. As the definition makes clear, conceptual legal information retrieval is different from ordinary legal IR. It focuses on modeling human users' needs for the information they seek in order to solve a problem, for instance in the legal argument a user seeks to make, and on the concepts and their roles in that problem-solving process.

Even focusing conceptual legal IR on helping users construct viable arguments in support of a claim or counter an opponent's best arguments is not new. Dick and Hirst (1991) explored manually representing cases in terms of schematic argument structures to support lawyers' "information seeking . . . to build an argument to answer the problem at hand." At that time, however, the authors could only assume "that in due course, . . . both language analysis and language generation by machine will be possible."

Their assumption has finally come true. For years, robust means for extracting such conceptual, argument-related information from natural language texts for purposes of conceptual legal information retrieval were not available. Today, however, language analysis tools that can *automatically* identify argument-related information in case texts are finally available, and with them a new paradigm is born: robust conceptual legal IR based on argument-related information, or AR as it is referred to in Section 10.5.

Cognitive computing is a second new paradigm for system development. Despite its name, cognitive computing is *not* about developing AI systems that "think" or perform cognitive tasks the way humans do. The operative unit of cognitive computing is neither the computer nor the human but rather the collaborating team of computer *and* human problem-solver(s).