University of Cincinnati Law Review

Volume 90 | Issue 3 Article 5

March 2022

The Limits of Law and AI

Ryan McCarl

THE LIMITS OF LAW AND AI

*Ryan McCarl**

TABLE OF CONTENTS

I. INTRODUCTION 923
II. WHAT AI IS AND WHAT IT CAN DO 924
A. *What is AI?* 925
B. *Symbolic (classical) AI* 927
C. *Machine learning and "deep learning"* 928
1. Supervised machine learning 929
2. Unsupervised machine learning 931
3. Neural networks and the "deep learning" revolution 932
D. *Natural language processing* 933
E. *Problem definition in machine learning* 937
III. WHAT TO EXPECT OF LAW AND AI 937
A. *Obstacles to building AI-driven legal applications* 939
1. The natural language understanding problem 939
2. The vagueness problem 939
3. The problem of eliciting, representing, and organizing facts 941
4. The explainability problem 941
5. The task-specialization problem 941
B. *Legal reasoning systems: a dead end?* 942
C. *Other applications of AI in law* 943
1. Legal research, information retrieval, and e-discovery 943
2. Expert systems for assisted legal decision-making 947
IV. CONCLUSION 949

I. INTRODUCTION

Many predict that artificial intelligence technologies will transform the economy, and some point to the legal profession as one of the fields ripe to be transformed. For example, Richard Susskind, president of the

* Founding Partner, Rushing McCarl LLP; former PULSE Fellow in Artificial Intelligence Law and Policy, UCLA School of Law. The author would like to thank Ted Parson and Richard Re for their helpful guidance in shaping this article.

Society for Computers and Law, stated that "[AI] will disrupt not just the world of practising lawyers but also our common perception of the legal process."[1] Susskind envisions:

> an online service that contains vast stores of structured and unstructured legal materials (primary and secondary sources, and legal analysis), that can understand legal problems spoken to it in natural language, that can analyse and classify the fact pattern inherent in these problems, that can draw conclusions and offer legal advice, and that can even express this guidance in some computer-simulated voice.[2]

Many scholars have dedicated their careers to pursuing similar visions. For thirty years, scholars in the field of "law and AI"[3] have explored the extent to which tasks performed by lawyers and judges can be made computable or computer-assisted. In particular, numerous scholars have argued that legal reasoning is amenable to computation. This Article contends that while artificial intelligence-based software is likely to improve legal research and make lawyers more efficient, it is unlikely to replace traditional legal work or otherwise transform the practice of law.

II. WHAT AI IS AND WHAT IT CAN DO

Artificial intelligence is not an easy subject to understand or research. The inner workings of neural networks—currently the dominant AI technology—are largely inaccessible to people without a background in mathematics and computer science.[4] AI is an increasingly pervasive and important part of modern life, so we unquestionably need to research it outside the realm of computer science. But there is a danger that scholarship not rooted in a technical understanding of AI may be too speculative to be useful.

Some AI experts believe in the possibility of "artificial general intelligence" (or "general AI"), referring to AI with generalizable powers of perception and reasoning that rival those of humans.[5] However, it is more useful to think about AI as a set of extant, albeit fast-developing, technologies centered on machine learning. In this Article, I focus on AI

1. RICHARD SUSSKIND, TOMORROW'S LAWYERS: AN INTRODUCTION TO YOUR FUTURE 55 (2d ed. 2017).

2. *Id.*

3. For an overview of the field, *see* KEVIN D. ASHLEY, ARTIFICIAL INTELLIGENCE AND LEGAL ANALYTICS: NEW TOOLS FOR LAW PRACTICE IN THE DIGITAL AGE (2017). The field's most prominent journal is *Artificial Intelligence and Law*, and its major conferences include the International Conference on Artificial Intelligence and Law (ICAIL) and the JURIX International Conference on Legal Knowledge and Information Systems.

4. For a nontechnical overview of neural networks in this article. *See infra* Part II.C.3.

5. *See generally, e.g.*, Sam S. Adams et al., MAPPING THE LANDSCAPE OF HUMAN-LEVEL ARTIFICIAL GENERAL INTELLIGENCE, AI MAGAZINE (Spring 2012).

as a technology rather than as an idea. In other words, this Article focuses on AI as it exists now and is likely to exist in the medium-term future.

First, I provide an overview of what AI is and what it can do. By reaching a rough understanding of the current and foreseeable state of AI technologies, we can better evaluate how AI is likely or unlikely to transform the practice of law.

In the sections that follow, Part A explains the contours of the term "artificial intelligence." Part B briefly discusses *symbolic* (or *classical*) AI. Part C explains the various branches of machine learning. Finally, Part D discusses natural language processing, which is the branch of AI that involves processing and understanding human language (speech and text) and uses both symbolic and machine-learning methods.

A. What is AI?

While there is no generally accepted definition of AI, AI traditionally focuses on the construction of *rational agents*: computer agents (or software programs) that work to achieve the best expected outcome according to a performance measure.[6] AI agents often perform one or more of the following activities[7]:

- Search (exploring possibilities, such as sequences of moves in a chess game or routes in a wayfinding application, that would help the agent reach its goal)
- Planning and scheduling (placing steps toward the agent's goal in an optimal order, given constraints)
- Perception (observing the agent's environment and using environmental cues to update the agent's internal state)
- Knowledge representation (encoding external knowledge in a computer-usable form so the agent can use it to reason and make decisions)

This Article uses the following, perhaps overinclusive, definition of AI: AI software performs functions associated with the human mind. These functions commonly include perception, pattern recognition, classification, reasoning, and language processing. AI software also commonly possesses certain other characteristics: it is often autonomous (that is, able to control itself), goal-directed, and capable of learning and self-improvement. Further, AI software sometimes runs on specialized

6. *See* STUART J. RUSSELL & PETER NORVIG, ARTIFICIAL INTELLIGENCE: A MODERN APPROACH 3–4 (4th ed. 2021) (characterizing the foundations of AI as computer programs that are expected to achieve the best outcome in as they "operate autonomously, perceive their environment, persist over a prolonged period, adapt to change, and create and pursue goals.").

7. The activities listed here are a broad, non-exclusive summary by the author of the main categories in which AI functions today. Russell and Norvig's text provides the classic overview of AI functions. *See id.*

hardware, such as a robot or self-driving car, that allows the software to learn from and interact with the external physical environment.

This Article does not equate AI with machine learning. "Machine learning is a subfield of AI that studies the ability to improve performance based on experience."[8] Though machine learning currently dominates AI research and practice, conflating the two overlooks systems that have traditionally been considered AI but do not improve themselves through experience.[9]

Furthermore there is an important distinction between narrow and general AI. The term *narrow AI* refers to AI systems that perform specific tasks in particular domains, such as systems that play chess or diagnose blood infections.[10] *General AI*, however, refers to software that can learn to perform virtually any task across a variety of domains.[11] This Article focuses exclusively on narrow AI because general AI does not yet exist and is not likely to exist in the near future. Though some AI experts believe that general AI is possible, few seem to believe that we are on the cusp of developing such technology.[12] For this Article's purposes—evaluating how AI is likely (or not) to transform legal practice in the medium-term future—it is more useful to think of AI in terms of software that can perform specific tasks.

Finally, to contextualize the meaning of AI, it is helpful to ask: what sort of tasks can AI systems perform? AI systems and technologies in use today perform the following tasks:

- Image recognition and description (including facial recognition, medical image diagnosis, and automatic caption generation)
- Robotics (including autonomous vehicles and autonomous weapons)
- Medicine and public health (including drug discovery, drug interaction prediction, and infectious disease modeling)
- Recommendation systems and personalization (including product and content recommendations, personalized news, and ad targeting)
- Natural language processing (including speech recognition, virtual assistants, machine translation, text generation, and sentiment

8. *Id.* at 1 n.1.

9. Akif Celepcikay & Yetkin Yildrim, *Artificial Intelligence and Machine Learning Applications in Education*, 2 EURASIAN J. HIGHER ED. 1, 2 (2021) ("For example, symbolic logic, expert systems, and knowledge graphs are AI but they are not machine learning.").

10. *See* MELANIE MITCHELL, ARTIFICIAL INTELLIGENCE: A GUIDE FOR THINKING HUMANS 46 (2019).

11. Mitchell describes "general AI" as "the AI that we see in movies, that can do most everything we humans can do, and possibly much more." *Id.*

12. For a range of expert opinions on the matter, *see* MARTIN FORD, ARCHITECTS OF INTELLIGENCE: THE TRUTH ABOUT AI FROM THE PEOPLE BUILDING IT (2018) (containing interviews with over a dozen of "the world's most prominent AI research scientists and entrepreneurs," as well as "deep learning agnostics"). *See also* JOHN BROCKMAN, POSSIBLE MINDS: TWENTY-FIVE WAYS OF LOOKING AT AI (2020) (collecting essays from leaders in the AI field).

analysis)

- Search and information retrieval (including document- and image-based search queries and question answering systems)
- Logistics and planning (including wayfinding and resource-deployment programs)
- Predictive analytics (including credit and insurance decisions, sales forecasting, and securities trading)[13]

B. Symbolic (classical) AI

Symbolic or *classical* AI—sometimes called "Good Old-Fashioned AI (GOFAI)"—refers to AI software in which concepts are represented as "symbols" and then manipulated by a software program.[14] Symbolic AI is characterized by practices such as grouping and structuring concepts into data structures or knowledge graphs that computationally describe their relationships.[15] Symbolic AI programs manipulate these data structures with if-then rules and other algorithms. Many symbolic AI programs are so-called *expert systems* that attempt to encode the knowledge of domain experts (e.g., cardiologists) and build data structures and rule trees that use that encoded knowledge to make decisions simulating those an expert would make.[16]

Though symbolic AI has largely fallen out of favor as the field turned its focus to machine learning, the subfield accomplished many achievements. Here are a few examples:

- Chess programs using rule trees and other symbolic AI techniques were able to defeat grandmasters.[17]
- The U.S. military's DART (Dynamic Analysis and Replanning Tool) program assisted with logistics planning and force deployment in the

13. *See* RUSSELL & NORVIG, *supra* note 6, at 27-30.

14. MARGARET A. BODEN, ARTIFICIAL INTELLIGENCE: A VERY SHORT INTRODUCTION 5 (2018). *See also* Giuseppe Futia & Antonio Vetrò, *On the Integration of Knowledge Graphs into Deep Learning Models for a More Comprehensible AI—Three Challenges for Future Research*, 11 INFORMATION 122, 125-26 (2020) (contrasting GOFAI with the "connectionist approach" that relies on networking principles and optimization techniques); Robert Hoehndorf & Núria Queralt-Rosinach, *Data Science and symbolic AI: Synergies, challenges and opportunities*, 1 DATA SCI. 27, 29 (2017) (distinguishing data science and symbolic AI).

15. Hoehndorf & Queralt-Rosinach, *supra* note 7, at 29.

16. In expert systems, "[p]roblem solving expertise is encoded, usually as a set of rules, and then some mechanical inference process manipulates this 'knowledge base' to solve problems in the domain by simulating the methods which the original expert might have used himself." Marek Sergot, *A Brief Introduction to Logic Programming and Its Applications in Law* 35, *in* COMPUTER POWER AND LEGAL LANGUAGE: THE USE OF COMPUTATIONAL LINGUISTICS, ARTIFICIAL INTELLIGENCE, AND EXPERT SYSTEMS IN THE LAW (Charles Walter ed., 1988).

17. *See, e.g.*, KARSTEN MÜLLER & JONATHAN SCHAEFFER, MAN VS. MACHINE: CHALLENGING HUMAN SUPREMACY IN CHESS (2018); Troy D. Kelley & Lyle N. Long, *Deep Blue Cannot Play Checkers: The Need for Generalized Intelligence for Mobile Robots*, J. Robotics, 2010.

Gulf War, helping the government save millions of dollars.[18]
- The system MYCIN used around 600 expert-defined rules to identify blood diseases.[19]

None of these programs relied on the machine learning technology that underpins most current AI applications and research, however.

Most existing legal AI systems are expert systems, which use symbolic AI techniques such as if-then rules.[20] However, the AI research and engineering community has largely moved on from symbolic AI to machine learning. Symbolic AI programs tend to be brittle since they operate according to hand-programmed rules that process drastically simplified representations of reality. Additionally, evaluating each rule's validity and programming them into a system is an expensive, time-consuming, and error-prone task that requires extensive input from domain experts. Even if these resource limitations can be overcome, it is often impossible to distill expert knowledge into robust, workable rules, especially in domains such as law that are nondeterministic and dependent on the exercise of reasoned judgment.

This Article ultimately explains how these obstacles have hampered efforts to develop AI programs in the legal sector.

C. Machine Learning and "Deep Learning"

The term *machine learning* refers to software capable of improving itself automatically as it learns from data.[21] Most contemporary machine learning tools can be thought of as predictive models or prediction-making machines. By observing and detecting patterns in large quantities of data, these machines learn to make predictions about unseen data.[22]

18. *See* U.S. DEPT. OF COMMERCE, CRITICAL TECHNOLOGY ASSESSMENT OF THE U.S. ARTIFICIAL INTELLIGENCE SECTOR xi (1994) (recapping how DART "solved the logistical nightmare of moving the U.S. military assets to the Saudi Desert. The application was developed to schedule the transportation of all U.S. personnel and materials such as vehicles, food, and ammunition from Europe to Saudi Arabia. This one application alone reportedly more than offset all the money the Advanced Research Projects Agency had funneled into Al research in the last 30 years.").

19. *See* MITCHELL, *supra* note 10, at 40–41.

20. For example, Kevin Ashley's influential HYPO program is essentially an expert system that combined a database of encoded legal cases with the domain knowledge of a lawyer familiar with trade secrets law. *See generally* KEVIN D. ASHLEY, MODELING LEGAL ARGUMENT: REASONING WITH CASES AND HYPOTHETICALS (1990). HYPO is discussed in Part III.B.

21. *See, e.g.*, Anastassia Lauterbach, *Introduction to Artificial Intelligence and Machine Learning* 33, *in* THE LAW OF ARTIFICIAL INTELLIGENCE AND SMART MACHINES (Theodore F. Claypoole ed., 2019).

22. Another important type of machine-learning program relies on *reinforcement learning*. *See, e.g.*, MITCHELL, *supra* note 10, at 133–44. Reinforcement learning agents learn by trial and error which actions bring them closer to their goals. *Id.* at 139-140. Examples of reinforcement learning include a robot teaching itself to walk, or software teaching itself to play Atari games. Reinforcement learning is not discussed in this article because it is not yet clear if it could be fruitfully used to solve problems in the legal domain.

Machine learning tools are trained on a set of examples, each of which is typically paired with a label. For example, a spam-detection tool would be trained on a set of emails, each of which has been assigned a label of "spam" or "not-spam." Most machine learning tools use *supervised* learning, in which the training examples are paired with human-assigned labels. Other tools use *unsupervised* learning, in which the software attempts to detect its own patterns in the data without recourse to human-assigned labels.[23]

Below, Subsection 1 discusses supervised machine learning, while Subsection 2 discusses unsupervised machine learning. Subsection 3 delves into two machine-learning subfields that can involve either supervised or unsupervised learning: deep learning and natural language processing.

The purpose of these discussions is twofold. First, I anticipate that these ideas will be largely unfamiliar to many of this Article's readers. For those readers, I hope to provide a mostly nontechnical and math-free overview of these core AI technologies. Second, to understand how AI might be fruitfully applied in the legal domain, it is important to have an approximate understanding of how the various AI technologies work. With such knowledge, we can ground discussions about possible legal applications of AI in the context of near-term technological possibilities rather than long-term speculation.

1. Supervised Machine Learning

Most modern machine learning consists of *supervised learning*.[24] In supervised learning, software learns from a set of data samples in which each sample (input) is associated with a label (output).[25] Through exposure to many sample-label pairs, the software learns a complex mathematical function that shows a relationship between the samples and labels. This process is called *training* the machine-learning model. A trained model can then use the function it has learned to guess the appropriate label for new, unseen samples for which it lacks human-assigned labels.[26]

This is best illustrated by example. What follows is a description of two standard examples of supervised machine learning: a spam classifier and a house-price estimator.

23. For an overview of the various types of machine learning, *see* ANKUR A. PATEL, HANDS-ON UNSUPERVISED LEARNING USING PYTHON: HOW TO BUILD APPLIED MACHINE LEARNING SOLUTIONS FROM UNLABELED DATA 3–26 (2019).

24. Aidan Wilson, *A Brief Introduction to Supervised Learning*, TOWARDS DATA SCIENCE (Sept. 2, 2019), https://towardsdatascience.com/a-brief-introduction-to-supervised-learning-54a3e3932590.

25. PATEL, *supra* note 23, at 3-4.

26. *Id.* at 4.

i. Spam classifier

Many of the emails transmitted on the Internet are unwanted spam. Every email provider uses a system to automatically detect whether incoming mail is spam so that spam emails can be directed to a spam folder and ignored. Modern email providers use machine learning to perform this task.

First, consider a non-machine-learning approach to spam detection. We could design a program that uses various explicit rules to detect spam. For example, we could check to see whether an incoming email uses more than four exclamation points and, if so, mark it as spam. By studying examples of spam emails, we could devise hundreds or thousands of similar rules. Such manually crafted rules are the hallmark of symbolic AI systems.

Alternatively, we can use machine learning to automatically detect and weigh the features (variables) associated with spam emails. As it learns these features, the machine learning software develops a complex mathematical model mapping the features to a classification of spam or non-spam. The trained software can then use this model to make a prediction about the legitimacy of incoming emails that it has not yet seen.

Our spam-detection system is an example of a *binary classification* program: its goal is to classify unseen samples as one of two options, spam or non-spam. We can also imagine a *multiclass classification* program, such as a program that classifies photos as displaying a dog, cat, horse, or none-of-the-above.

Many current applications of machine learning involve supervised-learning systems performing classification tasks, so the spam detector example should be the first thing that comes to mind when you think about AI and machine learning. Here are some other examples of classification problems:

- Given a newspaper article, predict its topic from a set of topics such as *sports* or *politics*.
- Given a movie review, predict its sentiment as *positive* or *negative*.
- Given a handwritten digit, predict whether the number written is 0, 1, 2, etc.

ii. House Price Predictors

Classification problems such as spam detection involve assigning data samples to various "buckets" (output labels) such as *spam* and *non-spam*. This is essentially a qualitative task, although the machine learning classifier could output either a binary answer (e.g., 1 for spam, 0 for non-spam) or a set of probabilities (e.g., 0.7 for spam and 0.3 for non-spam, representing a prediction with 70% confidence that the email is spam).

Predicting house prices in a district involves a different, quantitative task: predicting a *continuous* output variable rather than a *discrete* output variable. Prices are continuous, theoretically ranging from $0.01 to infinity. The problem of predicting a continuous output variable given a set of input variables is called a *regression* problem. Supervised machine learning tasks involve either classification or regression, and many problems can be modeled as either.

A house-price predictor works as follows. The machine learning model is trained with samples consisting of sets of variable-value pairs such as the following. Each sample represents one housing district:

District 1	
Longitude	-122.23
Latitude	37.86
Housing median age	41
Total rooms	880
Total bedrooms	129

The machine learning model is trained by processing thousands of samples. Each sample is paired with a label representing the variable predicted—in this case, the median house price in each district. As shown in the table above, each sample has a number of *features* (variables) that may serve as clues to the target output. The model learns the relationship between these features and the median price. Once trained, the model can take an unlabeled sample, such as information about a district, and generate a prediction of that median house price.

Other examples of regression problems include:

- Given the features of a car, estimate its price.
- Given a person's demographic information, estimate the person's income.
- Given a college applicant's GPA and standardized test scores, estimate the student's probability of being admitted to a top university.
- Given historical weather data, estimate tomorrow's temperature.

Note that in each of these examples, the output value is continuous rather than discrete.

2. Unsupervised Machine Learning

In the spam-detection and house-price-estimator examples above, the models were trained using *labeled data*. Each input sample was paired with a target output (or label), and the machine-learning model learned the relationship between the samples' features and the label. In the real world, most data is unlabeled, and the problem of assigning labels to tens of thousands of data samples is often laborious, error-prone, and expensive.

Unsupervised learning algorithms attempt to find relationships between features in a set of unlabeled input samples. Such algorithms essentially recognize patterns in the input data without having any particular goal (target output) in mind.

Here are some examples of unsupervised learning problems:

- Given a set of credit-card transactions with associated features such as price, customer address, and transaction address, detect outlier transactions that should be flagged as potentially fraudulent.
- Given a set of social media users, group the users into separate clusters based on what ads they are likely to click on. Then, target a new ad to all users in a particular cluster.

Unsupervised learning is growing in importance.[27] Further, it continues to increase its contributions to AI through the advent of *semisupervised* learning. In semisupervised learning, unsupervised learning is used to generate the labeled data needed for a supervised learning task. Consider the following example: Given a set of Amazon shoppers, group the shoppers into separate clusters based on what they are likely to buy. After examining the results, note that one of the clusters tends to buy cat litter and scratching posts. Then, manually assign the label "cat owner" to each of these shoppers. Amazon now has a set of labeled data that it can use to predict whether a user is a cat owner, and it can use this information to target ads to these users in the future.

3. Neural Networks and the Deep Learning Revolution

The most important development in the past decade of AI research has been the onset of *deep learning*, which refers to multilayered neural networks. Neural networks derive patterns from input data that allow the network to map similar but unseen data to certain outputs.

Neural networks operate by passing each sample of input data through various *layers*, each of which transforms the data in a way that makes it easier to associate the sample with an appropriate output.[28] When the transformed sample emerges from a pipeline of internal (*hidden*) layers, the final (*output*) layer associates the transformed sample with a predicted

27. *See, e.g.*, Ana Mezic, *Why Unsupervised Machine Learning is the Future of Cybersecurity*, TECHNATIVE (Sept. 9, 2021), https://technative.io/why-unsupervised-machine-learning-is-the-future-of-cybersecurity/; Thorsten Wuest et al., *Machine learning in manufacturing: advantages, challenges, and applications*, 2016 PROD. & MFG. RSCH 23, at 33 (discussing the growing importance of unsupervised learning and the research opportunities in the field).

28. For an overview, *see* MITCHELL, *supra* note 10, at 70–80.

value. If the predicted value does not match the sample's true (human-assigned) value, the model subtly adjusts itself to reduce similar mistakes in the future. Once trained, a successful model can assign appropriate labels to unseen data.

Most neural networks are engaged in supervised learning, and the most common supervised learning task is, as previously discussed, classification. Consider a neural network designed to identify images of handwritten digits. Each input to the model (i.e., each *sample*) would be an image of the digit to be classified. Before the images are fed to the model, they must be converted to numerical matrices of equal size. For example, a greyscale image could be represented by a 255 x 255 grid in which each slot contains a numeric value representing the darkness of a particular pixel. Each pixel's darkness can be thought of as a feature of the image. The neural network uses these features as clues bearing some relation to the image's output label.

After passing to the neural network, the image undergoes a transformation in which each layer's units multiply feature values by randomized weights that accentuate or downplay the features. At the end of the network pipeline, the image's transformed data is used to predict the output label. For example, the model might output an array of ten numbers, with each number representing the chance that the image is a particular digit. If slot one of the output array represents the digit "zero," the output array [0 0 1 0 0 0 0 0 0 0] might signify a guess that the handwritten digit is the number two. Alternatively, the output array might show probabilities rather than "yes/no" answers. Thus, the array [0 0 .5 0 0 0 0 .5 0 0 0] might signify a 50% probability that the digit is the number two and a 50% probability that the digit is the number seven.

Suppose that on its first pass through the image data, the network incorrectly classifies a handwritten five as the digit eight. At the output stage, the network would compare its guess of eight to the true label of five, determine that it erred, and slightly adjust each layer's weights to reduce the likelihood of repeating that error. Thus, the model learns to make better predictions in the future.

D. Natural Language Processing

Computers fundamentally work by processing numbers, not words. Since the onset of AI research, AI inventors have worked to create machines that can bridge the human-computer communication gap and work with natural human language. This branch of AI is called *natural language processing* ("NLP"). NLP's fundamental status in AI is illustrated by the famous "Turing test," derived from Alan Turing's proposal that a machine is intelligent if it can answer questions so well as to fool a human questioner into believing that the machine is human.[29]

Traditional symbolic-AI approaches to NLP work by processing input language according to fixed rules.[30] At the simplest level, a *regular expression* can parse human text into different components which can then be used as variables in algorithms. For example, the regular expression `([A-Z][a-z]+)` could be used to search a document for character strings consisting of one capital letter followed by one or more lowercase letters. This could serve as a primitive way to search for proper nouns. Of course, such a method is deeply flawed, as it would treat every capitalized sentence-starting word as a proper noun, and it would not recognize multiple-word proper nouns such as "Grand Canyon" as constituting a single proper noun as opposed to two separate proper nouns. Regardless, this example illustrates how a computer using symbolic AI techniques can learn to parse text strings into components with which the computer can reason.

As a somewhat more sophisticated example, I developed a simple symbolic-AI NLP program that can solve basic estates-in-land problems in property law. First-year property students learn to parse deeds into present and future property interests. For example, a deed granting land "to Jenny and her heirs for so long as the land is used as a school, then to Bob" would be parsed as follows:

- Jenny has a fee simple determinable; and
- Bob has an executory interest that becomes possessory when the land is no longer used as a school.

The language in deeds is often predictable, making such problems amenable to computation. Here is an example of the program in action:

Input:

Richard to Ted so long as the land is used for educational purposes; and if it is not used for educational purposes, then to Ryan.

Output:

Grant: Richard to Ted so long as the land is used for educational purposes; and if it is not used for educational purposes, then to Ryan.

Analysis: Richard granted a fee simple subject to an executory limitation to Ted. Ryan holds a future interest which will become

29. *See* MITCHELL, *supra* note 10, at 49–52 (2019).

30. For a comprehensive overview of NLP concepts that predates the deep learning revolution, *see generally* DANIEL S. JURAFSKY & JAMES H. MARTIN, SPEECH AND LANGUAGE PROCESSING: AN INTRODUCTION TO NATURAL LANGUAGE PROCESSING, COMPUTATIONAL LINGUISTICS, AND SPEECH RECOGNITION (2000).

possessory if the property is not used for educational purposes.

Grantor: Richard

Grantee: Ted

Negative condition: The property must be used for educational purposes.

Future interest holder: Ryan

The program would work by using advanced regular expressions to search for grammatical patterns in the deed language. It parses the deed language into components that fill variables such as *grantor*, *grantee*, *life estate holder*, and *condition*. It then uses these variables to create and fill templates such as: `{grantor} granted a fee simple subject to an executory limitation to {grantee}.`

Symbolic AI techniques, such as regular-expression-based grammar parsers, can produce satisfactory results for many tasks. However, since language is complex and unpredictable, and requires so much background knowledge to understand, there is a ceiling on programmers' ability to process language with parsing algorithms, database lookups, and if-then rules. Such tools tend to exhibit the symbolic-AI shortcoming of *brittleness*. They are also often not portable across different speech domains; for example, a program designed to process news articles is likely unsuccessful at parsing medical literature or children's speech, and a program designed to parse English is helpless when confronted with Chinese.

The NLP community has therefore joined the rest of the AI community in reducing the role of many symbolic AI techniques in favor of deep learning (multilayered neural network) technologies. This application led to unprecedented successes on many NLP tasks. For example, *speech recognition* is far better than ever before: digital assistants such as Amazon's Alexa and Apple's Siri can decode most human speech into text strings, which prompt the software to perform certain tasks, such as launching applications, sending messages, and looking up answers to basic factual questions (e.g., "What's the weather like today?"; "Who won the Super Bowl?"; etc.). AI expert Melanie Mitchell considers near-human-level speech recognition to be the greatest achievement of AI to date.[31]

Advances in speech recognition have been made possible by programs

31. MITCHELL, *supra* note 10, at 180.

that combine deep neural networks, each of which specializes in a narrow task. For instance, one neural network might map input sounds to human phonemes, another might map those phonemes to words, and yet another might map groups of words into phrases.[32]

Another significant achievement of deep neural networks for NLP is machine translation. Contemporary machine translation systems such as Google Translate are far better than their older, symbolic-AI equivalents. Before the advent of deep-learning translation methods, translation systems worked by parsing an input sentence into phrases, then probabilistically matching those phrases to equivalent phrases in the target language.[33] Now, translations are typically performed with *encoder* and *decoder* neural networks. The encoder network learns a mathematical representation of the input text, and the decoder network learns to map the encoder network's representation back to output text in a different language.[34]

Another notable advance in NLP is the development of image-captioning systems that attempt to describe an image's content. Many of these systems use techniques similar to those used in deep-learning machine translation approaches, except the encoder network maps an input *image* to a mathematical representation instead of mapping an input *text* to a mathematical representation.[35]

Yet another recent advance is *text-generation*.[36] Text-generation systems such as Google's Smart Reply can perform functions such as helping the user complete sentences in emails. With Smart Reply, for example, as you begin to type "To Whom" at the top of an email, the words "it May Concern" may appear in grey after your cursor, and you can then press the tab key to accept the suggestion and include it in your email. Text-generation systems work by learning to perform the following task: given a sequence of text, predict what text will likely come next. These systems may exhibit a type of creativity by introducing randomness to their predictions.

These significant advances in natural language processing should not obscure the limits of what deep neural networks can do, however. Neural networks create highly complex mathematical functions that map input data to output data. However, neural networks cannot understand

32. *Id.*

33. *Id.* at 199.

34. *Id.* at 199–201.

35. *See generally*, *e.g.*, I. Hrga and M. Ivašić-Kos, *Deep Image Captioning: An Overview*, at 995-1000 in 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 995 (2019).

36. *See* John Seabrook, *The Next Word: Where Will Predictive Text Take Us?*, NEW YORKER (Oct. 14, 2019), https://www.newyorker.com/magazine/2019/10/14/can-a-machine-learn-to-write-for-the-new-yorker.

language in a human sense because they lack background knowledge about the world, and human communication often relies on implicit messages that a computer cannot detect.[37] Computers are unlikely to overcome these obstacles anytime soon.[38]

E. Problem Definition in Machine Learning

It is tempting to marvel at the accomplishments of machine learning systems and imagine that they can solve any kind of problem. However, machine learning is not appropriate for every task, so it is important to consider whether a problem can be defined in such a way as to be amenable to a machine learning approach. For example, when creating a program to solve a particular problem, one must consider what its inputs and outputs would be.

When thinking about whether a problem is amenable to a *supervised* machine learning approach, consider the following:

- Can you think of the task in terms of mapping inputs (such as pictures) to outputs (such as "dogs" or "cats")?
- Do you have thousands (or more) of input samples that are already labeled with the relevant output so that you can train the system?

Conversely, when thinking about whether a problem is amenable to an unsupervised machine learning approach, consider the following:

- Given a set of data samples, can you think of some reason to look for hidden patterns in the data (especially to separate the samples into groups or to detect anomalies)?

Engaging in problem specification of this sort is an essential step to determine the short- and medium-term practical applications of AI.

III. WHAT TO EXPECT OF LAW AND AI

Law is not particularly amenable to computation. As Kevin Ashley has acknowledged, "legal rules tend to be incomplete, logically and semantically ambiguous, sometimes inconsistent, and frequently hard to find or even to identify. [Additionally,] the law is adversarial; legal problems frequently have no one right answer."[39] Despite these difficulties, AI techniques can assist with many specialized legal tasks.

37. *See, e.g.*, Raymond J. Moody and Gerald DeJong, *Learning schemata for natural language processing* 681-87, INT'L JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, (1985), https://www.ijcai.org/Proceedings/85-1/Papers/131.pdf.

38. *See* MITCHELL, *supra* note 10, at 180.

39. ASHLEY, *supra* note 20, at 2.

Section A of this Part first describes obstacles to developing legal AI systems. Section B delves specifically into efforts to develop legal reasoning systems and explains how and why such efforts have fallen short. Section C describes promising applications of AI concepts to law, including advanced legal information retrieval systems and collaborative expert systems that can assist, rather than replace, a legal decisionmaker.

A. Obstacles to Building AI-Driven Legal Applications

Problems amenable to computer automation usually match fixed, yes-or-no inputs with certain outputs in a predictable way. Some legal issues are amenable to this sort of problem structuring. For example, traffic laws are typically "bright-line rules" that motorists either do or do not violate, leaving zero room for judicial discretion. Perhaps unsurprisingly, we see enforcement mechanisms such as red-light cameras and speed cameras that operate as fully automated law enforcement systems. These systems are monolithic programs that replace, within a limited domain, the role previously played by a police officer, court clerk, and judge.

However, most legal problems are not as simple as speeding tickets. For instance, the principle underlying red-light cameras cannot be extended to reckless driving because what constitutes "recklessness" is a fact-intensive inquiry that calls for the reasoned application of a vague legal standard. Even if a program could be developed to make a prediction about whether a judge would rule that certain driving activities are "reckless," due process considerations would limit the utility of such a program.[40] A reckless driving conviction has significant consequences for the driver, and I would not trust an automated system to dole out those consequences without having a deep understanding of how the system works. We need reason to believe that the system's decisions accurately reflect decisions that would be made by a human judge reviewing the same facts and law. A workable system that meets that criterion is prohibitively difficult to conceive, plan, execute, and test.

The following subsections briefly discuss five obstacles to applying AI in the legal domain: the problems of natural language understanding, vagueness, fact representation, explainability, and task generalization.

1. The Natural Language Understanding Problem

Legal reasoning frequently involves interpreting texts. Computers can easily identify statistical relationships among words and phrases, but they

40. *See, e.g.*, Han-Wei Liu et al., *Beyond State v Loomis: Artificial intelligence, government algorithmization and accountability*, 27 INT'L J. L. INFO. TECH. 122 (2019); Danielle Keats Citron and Frank Pasquale, *The scored society: Due process for automated predictions*, 2014 WASH. L. REV. 89.

cannot read or understand text in a way that approximates what lawyers do when they read a judicial opinion. Given two judicial opinions, a program might detect similarities and differences between the opinions, noticing that the opinions cite some of the same cases, are of similar length, or were written by the same judge. However, the program will not have any sense of which similarities and differences are relevant.[41] Identifying relevant similarities and differences between cases is a critical aspect of legal reasoning in a common-law system.

Improvements in natural language processing have led to impressive feats, such as the development of question-answering systems and virtual assistants, but no computer can "read" in the sense that even a novice human reader can.[42] Human readers bring to their task background knowledge, "common sense," and an ability to make inferences that go beyond the information explicitly stated in the text.[43] The obstacles to making these skills computable are unlikely to be overcome anytime soon. Any AI-related predictions that assume away such problems amount to speculation.

Legal reasoning programs designed to date operate on manually written representations of legal texts rather than on the texts themselves. Given the fluidity of legal concepts and legal reasoning, however, it is difficult or impossible to codify the contents of most legal cases or fact situations in a reproducible way—that is, in such a way that a number of legal experts would look at the same judicial opinion or fact situation and reliably encode its relevant information in an identical fashion.

2. The Vagueness Problem

Law is pervaded by vagueness. A concept or term is vague if its boundaries are not easily defined, so it is often difficult or impossible to objectively determine whether the term encompasses a particular fact situation.[44] For example, while the term "reasonable" is pervasive in law, it lacks a fixed meaning that can be computationally encoded. Whether a defendant's conduct is reasonable in a negligence case depends on an unknown, and potentially unknowable, number of variables, including the

---

41. *See* MITCHELL, *supra* note 10, at 105 (noting that a machine learning program learns what *it* observes in the data rather than what a human might observe, meaning that it will learn statistical associations in data even if a human would judge them to be irrelevant to the task).

42. *See id.* at 222–28.

43. *Id.* at 224.

44. *See generally* Ryan McCarl, *Incoherent and Indefensible: An Interdisciplinary Critique of the Supreme Court's "Void-for-Vagueness" Doctrine*, 42 HASTINGS CONST. L. Q. 73, 82–88 (2014) (discussing linguistic vagueness and the law). Logician Willard V.O. Quine wrote that many words are "best depicted as forming not a neatly bounded class but a distribution about a central *norm*." WILLARD V.O. QUINE, WORD AND OBJECT 77 (2d. ed. 2013) (emphasis in original).

nuances of what actions the defendant took and failed to take as well as the factfinder's view of what a reasonable person would do in similar circumstances. The same goes for concepts such as "good faith."

Because vagueness is so common in the law,[45] it is difficult or impossible to robustly model many legal rules and principles that can operate on computerized representations of legal cases or fact situations. These problems are exacerbated by the fact that legal rules change over time[46] and are subject to unwritten exceptions.

Attempts to encode a legal rule into an algorithm entail calcifying an essentially dynamic concept. In other words, efforts to make legal reasoning computable involve reducing vague concepts and rules into constants, variables, and formulas that can be computationally manipulated. Theoretically, such efforts can help clarify how legal rules operate. But if the goal is to build operational systems that play a role in deciding legal disputes, we should be wary of indulging our taste for "false clarity"[47] by artificially simplifying the messy complexity of the law.

3. The Problem of Eliciting, Representing, and Organizing Facts

Relatedly, much of the law is driven by the discovery, characterization, and organization of facts, and there are inherent difficulties in making any fact-centric legal process computable. For example, in client or witness interviewing, "[i]t is the responsibility of the human lawyer to elicit *all* the relevant facts . . . [and] to decide what is important among all the information supplied by the client."[48] It is difficult to conceive of a computer program that could perform these functions. There is also the more fundamental problem of *knowledge representation*, a strand of AI that considers how to represent facts to a computer in the first place to enable the computer to perform logical operations with those facts.

A related problem is that AI programs lack "common sense": an understanding of basic concepts such as the fact that a physical entity can only be in one place at a time. Rules such as that must usually be manually encoded. Expensive, decades-long efforts to encode such rules have been mostly fruitless.[49]

45. This problem is often discussed in terms of the "open-textured" nature of legal concepts. Legal concepts "cannot be defined by necessary and sufficient conditions which are universally valid over their domain of application." Kevin D. Ashley & Edwina L. Rissland, *Law, learning, and representation*, 150 ARTIFICIAL INTELLIGENCE 17, 18 (2003) (discussing H.L.A. Hart, *Positivism and the Separation of Laws and Morals*, 71 HARV. L. REV. 593 (1983)).

46. *See id.* at 26.

47. KEES VAN DEEMTER, NOT EXACTLY: IN PRAISE OF VAGUENESS 6 (2010).

48. URI J. SCHILD, EXPERT SYSTEMS AND CASE LAW 117 (1992) (emphasis in original).

49. *See, e.g.*, MITCHELL, *supra* note 10, at 248–50 (describing the history of similar efforts).

The need to create appropriate data structures to store facts and the relationships between facts is a major obstacle to the development of useful expert systems in any domain, particularly in law. And "[e]ven if we assume an enormously large knowledge-base there is a certain problem no system can cope with: We could never guarantee that yet another fact, so far absent and irrelevant to all previous cases[,] may not take an all-important significance in the case at hand."[50] In other words, because judicial precedents in the common-law system consist of legal decisions inextricably intertwined with the facts that gave rise to the dispute, there is no guarantee that the correct rule of decision for every new case can be identified in the corpus of past decisions.

4. The Explainability Problem

Deep learning systems and neural networks generally cannot provide reasons for their decisions. These systems are sometimes referred to as "black boxes" because while we can observe their inputs and outputs, we cannot meaningfully observe their internal decision-making processes or make sense of the predictive models they construct. The term *explainability problem* refers to the fact that even the most advanced deep-learning systems cannot presently explain how they reach decisions or make predictions.[51]

The explainability problem poses a major obstacle to the development of AI technologies that can enhance or substitute for the work of lawyers and judges. A fundamental aspect of the legal system is the need to justify conclusions with reasons. Judges, for example, are ordinarily expected to write reasoned opinions in support of any significant decision. Reasoned opinions promote confidence that judicial decisions are principled and nonarbitrary. AI technologies cannot perform this all-important task.

5. The Task-Specialization Problem

Though there are judging-related tasks that are amenable to computation, one should keep in mind the difference between "narrow" and "general" AI discussed earlier in this Article.[52] For the foreseeable future, all AI programs will be essentially specialist, designed to perform narrow tasks within a restricted domain. The act of presiding over disputes as a judge, however, involves a wide variety of tasks: deciding whether evidence is admissible, deciding what to do about breaches of

50. SCHILD, *supra* note 48 at 141.

51. *See, e.g.,* Pantelis Linardatos et al., *Explainable AI: A Review of Machine Learning Interpretability Methods*, 23 ENTROPY 18 (2021).

52. *See supra* notes 10-11 and accompanying text.

procedure such as late filings, resolving an ambiguity in a statute or contract, and so on. We currently lack computer systems that can perform *any* of these judicial tasks on their own, and a computer program asked to substitute for a judge might need to perform all of them.

B. Legal Reasoning Systems: A Dead End?

For several decades, researchers have attempted to build legal reasoning software that automates (wholly or partly) the process of applying legal rules to new fact situations. This research has aimed to design systems that, given a user's description of a fact situation, can select relevant judicial precedents and apply those precedents to (or distinguish them from) the fact situation.[53]

In 1990, Kevin Ashley published *Modeling Legal Argument*, an important work in the field of law and AI. In that book, Ashley introduced HYPO, a legal reasoning program that Ashley described as follows:

> Hypo is a case-based reasoning program. It employs actual legal cases in its database to analyze problem disputes. Given a description of a legal dispute, the program compares the problem to relevant cases, selects the most analogous cases, and cites them in arguments. It draws simple factual analogies between the problem and precedents, distinguishes precedents, cites counterexamples, and poses hypothetical variations of the problem to spur an attorney to focus on important additional facts that would strengthen or weaken the arguments. In short, Hypo symbolically compares and contrasts the problem situation and cases in its case database.[54]

While HYPO is undoubtedly impressive, it had limitations that made it impractical for attorneys. These limitations remain relevant because they reflect still-unsolved problems with building legal reasoning software. For example, these limitations include:

- **Small case database in a limited domain.** HYPO's database contained only thirty cases relating to a single issue in tort law: whether a defendant misappropriated a trade secret.
- **Manual inputting and encoding of cases.** The cases in HYPO's database consisted not of plain-text judicial opinions but rather sets of variable-value pairs manually encoded by a lawyer or law student who had analyzed the judicial opinion and extracted salient facts. Any new cases had to be inputted manually for HYPO to consider them; there was no mechanism for automatically processing new opinions. HYPO was therefore blind to changes in the law, and any legal researcher who

53. *See* Kevin Ashley, *Case-Based Reasoning* 24, *in* INFORMATION TECHNOLOGY AND LAWYERS (Arno R. Lodder & Anja Oskamp eds. 2006).

54. ASHLEY, *supra* note 20, at 25.

used HYPO would have to supplement its work by performing manual searches for relevant cases outside its database—with a particular eye toward recently issued opinions.

- **Naïve method of weighing precedents.** HYPO used simple counting to determine whether a precedent was analogous to the case at hand.[55] For example, a case sharing two analogous facts was deemed to be more on point than a case sharing only one analogous fact.[56]

My point is not to criticize HYPO but rather show that its limitations illustrate unsolved and perhaps unsolvable problems in the field of Law and AI. Similar limitations pervade every attempt to create a program that can engage in legal reasoning.[57] For example, whereas new cases can be automatically added to legal information retrieval systems such as Lexis and Westlaw, these cases must be manually analyzed and coded before being entered one-by-one into a legal reasoning system.[58] That is so because programs cannot perform legal reasoning with raw text; they can only reason with (i.e., perform logical operations over) concepts that have been made meaningful to the computer through manual programming.

C. Other Applications of AI in Law

1. Legal Research, Information Retrieval, and e-Discovery

Despite the limitations regarding AI's application to the law discussed above, it is reasonable to expect that AI technologies will in some ways transform legal practice in the areas of legal information retrieval and e-discovery. These changes will be enabled by advances in natural language processing, search technologies, and recommendation engines.

i. Advances in Legal Research and Information Retrieval

Information retrieval (IR) technologies have improved dramatically since the early days of Lexis and Westlaw. To understand why, it helps to think about cases and other legal materials as sets of features (variables), and search queries as attempts to retrieve materials that contain certain

55. *See* SCHILD, *supra* note 48 at 131 (criticizing HYPO on this point and noting that "[a] case with just *one* fact in common with the case at hand may sometimes be more convincing than a case with many such common dimensions").

56. HYPO also used simple counting in its algorithm for deciding whether a precedent was "stronger" or "weaker" than the case at hand. For example, if a party disclosed a trade secret to two outsiders, that sharing was deemed to be meaningfully worse than if the party had disclosed the secret to only one outsider. In practice, however, it is easy to conceive of a situation in which one improper disclosure would be more legally significant than two improper disclosures.

57. These programs are discussed in ASHLEY, *supra* note 3 at 73–106.

58. *See id*.

features and not others.

Consider the task of searching a legal IR system for cases involving negligence claims in which the defendant, a motorist, ran a red light and caused an accident. A basic method of searching for relevant cases would be entering a search query such as `vehicle "red light" negligence` and allowing the system to then search its case database for cases that mention those three terms.[59] The system may return only cases that contain all three terms or use a more generous algorithm to retrieve cases containing only one or two of the terms. After determining which cases to retrieve, the system can rank the cases based on some measure, such as the number of times each case mentions the three search terms.

There are various ways to build on this elementary system. For example, the system could use *query expansion*, automatically expanding the search query to include similar or related terms such as "stop light" and "traffic signal" for "red light," or "car" and "automobile" for "vehicle."[60] The system could also use TF-IDF searching to retrieve documents that not only mention the terms but mention the terms *proportionally more often than other cases in the database*.[61]

Our example search query—`vehicle negligence "red light"`—can be understood as a feature vector in which each term corresponds to a feature that exists or does not exist in the IR database's case documents. Recent advances allow IR systems to interpret both search queries and a database's cases as more complex and meaningful feature vectors for purposes of matching the query to target cases. For example, instead of checking whether a case includes the term "negligence," it is possible for a system to make an educated guess as to whether the case deals with the *concept* of negligence.[62] Then, the relevant search feature would not be the term "negligence," which is nothing more than a string of characters, but the idea of a *negligence case*—that is, a tort case in which one of the plaintiff's claims is negligence.

Legal IR developers have also come up with increasingly creative and

59. By entering "red light" in quotation marks, we instruct the search engine to treat the phrase as a single term.

60. *See* ASHLEY, *supra* note 3, at 227.

61. TF-IDF is short for "term frequency / inverse document frequency," a simple mathematical measure of the relative importance of a term in a document as compared with other documents in a corpus. The measure indicates a term's "frequency in the document discounted by its frequency in the corpus." *Id.* at 237. TF-IDF allows a search engine to consider how rare a search term is. For example, if a corpus of judicial opinions contained only negligence cases, the search term "negligence" would contain no information to help distinguish one opinion from another. A search engine using TF-IDF could effectively assign that term little weight in the search query.

62. *See id.*

complex ways to map relationships among the cases in a database; if a search query identifies one case, the IR engine can also retrieve cases related to that initial case. One way to accomplish this is through identifying whether cases share common concepts, as discussed above. Another is to use citation networks that consider, for example, what cases an opinion cites or is cited by and how often those related cases themselves are cited by other cases. Note that citation networks can be thought of as recommendation mechanisms in which one case recommends another as potentially relevant based on their common citations.

Three other IR technologies deserve mention:

(1) **Document vectorization**.[63] For purposes of an IR program that retrieves judicial opinions, an opinion in the database can be thought of as a collection of features such as terms or concepts that are either present in or absent from each case. The IR system uses features in the search query to identify cases having similar features.

However, there is another, more abstract way to represent a case or other text. The text can be computationally represented as a point in a multidimensional space. The point's location is ultimately determined by the presence or absence of features in a feature vector, but the feature vector is transformed into a single point and cannot necessarily be reverse engineered based on the point's location. Though the details of this idea are beyond this Article's scope, suffice it to say that similar texts (i.e., texts with similar feature vectors) are geometrically closer to each other than texts that share little in common.

The idea of representing an entire document as a point has given rise to what I consider the most useful and intriguing development in legal reasoning in recent years: the ability to submit entire documents as search queries. For example, instead of submitting the query `vehicle negligence "red light,"` a litigator could submit an opposition brief or trial court opinion from the litigator's current case.[64] That document would then be transformed into a feature vector, represented as a point in a multidimensional feature space, and the search would return documents (e.g., cases) that are closest in space to the query document.

(2) **Relevance feedback**. *Relevance feedback* techniques in IR systems allow users to indicate whether retrieved results are relevant. Users can do so either explicitly or implicitly by, for example, clicking

63. *See, e.g.*, Anita Kumari Singh & Shashi Mogalla, *Vectorization of text documents for identifying unifiable news articles*, 10 INT'L J. ADV. COMP. SCI. APPL. 305, 305–08 (2019).

64. I have tried two commercial IR systems that offer document-based searching: CaseText's CARA AI and Westlaw's "Quick Check."

on the document and spending screen time reading it. These user actions provide additional information that can either bolster a previous search query or serve as a new search query to retrieve documents similar to the document marked as relevant.

One can imagine building machine learning systems that turn users' judgments of relevance into labeled data to predict relevant documents for future queries. For example, a machine learning program might be trained on data consisting of a search query paired with a document marked as relevant to that search query. In this example, the query is the sample or training instance, while the relevant document is the "label" associated with the training instance. Software trained in this way might be able to map future queries to documents likely to be relevant. Because commercial IR programs do not generally disclose how their proprietary systems work, it is unclear whether such a mechanism is currently being used.

(3) **Text summarization.** Another potential area for advances in AI is in text summarization, particularly in multi-document summarization, such as an algorithm that extracts rules from a set of judicial opinions. Because legal propositions are often followed by citations, it is often possible to distinguish rules from other statements in a case automatically. A multi-document text summarization program might be able to identify and extract similar rule statements that recur in a batch of cases.

ii. Advances in e-Discovery

Electronic discovery, or "e-discovery," is a notoriously time-consuming and expensive task.[65] In complex and high-stakes litigation, attorneys and paralegals might be forced to sift through hundreds of thousands of documents in search of relevant information that could affect the case's outcome. This task is becoming increasingly manageable, however, because of the advent of tools that use modern NLP, search, and recommendation-engine technologies to help users identify related documents and classify documents as either relevant or not-relevant.

Recommendation engines are algorithms that recommend content to users based on the users' explicit or implicit preferences.[66] For example, Amazon recommends products to users based on what products they have browsed or clicked on in the past. Facebook recommends stories to users based on what content they have "liked" or clicked on in the past. Modern

65. *See, e.g.*, David Degnan, *Accounting for the Costs of Electronic Discovery*, 12 MINN. J.L. SCI. & TECH. 151 (2011).

66. *See, e.g.*, Prem Melville and Vikas Sindhwani, *Recommender Systems*, 1 ENCYCLOPEDIA OF MACHINE LEARNING 829–38 (2010).

recommendation engines often rely on a combination of supervised and unsupervised machine learning to group users or content into clusters.[67] In the e-discovery context, a user's decision to mark a certain document as "relevant" can be used to point the user to other potentially relevant documents that have something in common with the first document. Additionally, by marking certain documents as relevant, the user may be training a classification algorithm that can become increasingly good at predicting the relevance of unseen documents. Documents that share common features with documents marked as relevant are more likely to be relevant.

The novel search technologies discussed above also have applications in the e-discovery context. For example, a lawyer might search the bank of discovery responses for documents similar to an original document. Using document vectorization, the original document can be used as the search query, and the system can return documents that have similar features.

2. Expert Systems for Assisted Legal Decision-Making

Not every aspect of the law is equally pervaded by vagueness and indeterminacy. In some domains, such as tax and immigration law, the principal reason for complexity lies in the number of potentially applicable rules and their arrangement in a thicket of statutes and regulations. There is a role for AI programs that could help lawyers navigate such complexity.

At the outset, we should make clear that computer programs substituting for human judgment by making binding legal decisions is not a desirable goal.[68] Law is a high-stakes affair, with rights and livelihoods in the balance. There is little reason to believe that even an advanced AI system could perform many of the functions involved in resolving a dispute; a computer cannot weigh a witness's credibility. Additionally, a computer can consider only the factors that it has been programmed or trained to consider. If a dispute involves some novel fact that, in the eyes of a human judge, might justify a departure from precedent or an exception to a rule, a computer might be blind to that novel fact because it has not learned to look for it and has no mechanism by which to weigh its significance.

67. *See, e.g.*, Christina Christakou, et al., *A Movie Recommender System Based on Semi-Supervised Clustering* 2, INT'L CONFERENCE ON COMPUTATIONAL INTELLIGENCE FOR MODELLING, CONTROL AND AUTOMATION AND INT'L CONFERENCE ON INTELLIGENT AGENTS, WEB TECHNOLOGIES AND INTERNET COMMERCE (2005).

68. *See, e.g.*, Sergot, *supra* note 16, at 37 ("Most legal applications are so sensitive, however, that it would never be acceptable to let a machine make legal decisions, whether it can explain its conclusions or not.").

However, one thing that we can fairly expect of AI in the legal profession is that AI-enhanced software programs can be used instrumentally as tools to help people decide disputes or navigate legal complexity.

One example is Intuit's TurboTax. The Internal Revenue Code is so forbidding that many lawyers refuse to handle tax issues, instead referring those matters to specialists (many of whom have obtained LLMs in tax law in addition to their JD degrees). TurboTax has nevertheless helped hundreds of thousands of Americans file their taxes on their own without resorting to accountants or tax attorneys. It does so by guiding the user through a complex flowchart of questions. If the user answers "yes" to certain questions, new lines of relevant questions are triggered. At the end of the process, embedded legal rules are applied to the user's answers to conclude how much tax the user owes, and any necessary tax forms are filled out on the user's behalf. The program can then file the taxes electronically. In tech-speak, the software succeeds by reducing the "friction" involved in doing one's taxes.

Though expert systems are usually understood as failed experiments of the past, TurboTax can be understood as a modern expert system. In addition to making people's lives more convenient, it helps democratize legal knowledge by making it accessible.

Crucially, expert systems like TurboTax tend to delegate computationally-difficult problems to the user. For example, if TurboTax is unsure how to classify certain income, the software is not likely to guess; instead, it is likely to ask the user to classify the income while presenting the user with a pop-up explaining the applicable rules.

One can imagine building other expert systems on TurboTax's model, especially where the requisite legal knowledge can be easily expressed as a set of flowcharts and if-then rules. For example, TurboTax-like software might be able to help a married couple apportion property during a divorce by guiding them through a set of questions whose answers tend to show whether certain assets are community or separate property. In that way, expert systems can help make legal help more accessible and affordable.

Such software can also be designed for lawyers and judges rather than for the general public, such as an expert system designed to help a judge navigate the federal sentencing guidelines.[69] In a bank robbery case, the system could ask questions such as:

- What statute was violated?
- [If the criminal statute targets theft, ask:] Was money taken from a

69. To understand what applying the sentencing guidelines entails, *see* UNITED STATES SENTENCING COMMISSION, FEDERAL SENTENCING: THE BASICS (Nov. 2018).

"financial institution?" (Here, the software could link to a pop-up with the statutory or regulatory definition of "financial institution" and to any applicable caselaw).

- [If the criminal statute targets theft, ask:] How much money was stolen?
- [If the criminal statute targets robbery or other use of force, ask:] Did the defendant brandish a "dangerous weapon"? (Again, the software could link to information that defines that term, but the software need not try to infer the answer on its own).

Answering such questions can lead the program to recommend a sentencing range and, if using a template system, develop the beginnings of a presentence report. The program would not give too much power to the algorithm because the human judge would be required to assist the algorithm at each juncture in which human interpretation is advisable. The software need not attempt to resolve any ambiguity or exercise any discretion on its own.

A variation on such expert systems is software that asks the user to supply relevant information and then uses that information to generate documents with templates. Some legal tasks such as business formation, simple contracts, and wills can at times be relatively simple, and these tasks can be assisted by a template program. For example, a will-generation program might ask the user whether she wishes to be an organ donor. If the user answers "yes," the program could insert a boilerplate organ donation provision into the will.

IV. CONCLUSION

Advances in machine learning and even "deep learning" do not remedy the basic obstacles to making law computable discussed in this Article. Machine learning does not involve reasoning; rather, it involves pattern detection. Machine learning software typically consists of prediction-making models that refine themselves over time through exposure to new data. To point to machine learning or deep learning as a potential solution to the problem of making law computable, one must be able to describe the task one wants the machine-learning model to perform, identify the program's inputs and outputs, and supply the training data from which the software will build a predictive model. Additionally, machine learning systems cannot engage in reasoning or understand natural-language texts like human readers can.

Although there may be fruitful applications of machine learning for legal predictive analytics and classification tasks, such as e-discovery document review, an important role remains for traditional symbolic AI techniques in legal applications of AI. Modern information retrieval systems allow users to search for information in creative new ways, including uploading entire documents as search queries. Modern expert systems such as TurboTax show that computer programs can work collaboratively with a user to help a user solve complicated legal problems, even if the computer itself cannot exercise legal judgment in difficult questions involving vague legal standards.