Duquesne Law Review

Volume 58
Number 1 *Artificial Intelligence: Thinking About Law, Law Practice, and Legal Education*

Article 4

2020

The Automation of Legal Reasoning: Customized AI Techniques for the Patent Field

Dean Alderucci

The Automation of Legal Reasoning: Customized AI Techniques for the Patent Field

*Dean Alderucci**

ABSTRACT

As Artificial Intelligence and Machine Learning continue to transform numerous aspects of our everyday lives, their role in the legal profession is growing in prominence. A subfield of AI with particular applicability to legal analysis is Natural Language Processing (NLP). NLP deals with computational techniques for processing human languages such as English, making it a natural tool for processing the text of statutes, regulations, judicial decisions, contracts, and other legal instruments. Paradoxically, although state-of-the-art Machine Learning and NLP algorithms are able to learn and act upon patterns too complex for humans to perceive, they nevertheless perform poorly on many cognitive tasks that humans routinely perform effortlessly. This profoundly limits the ability of AI to assist in many forms of legal analysis and legal decision making.

This article offers two theses. First, notwithstanding impressive progress on NLP tasks in recent years, the state-of-the-art in NLP will remain unable to perform legal analysis for some time. Second, lawyers, legal scholars, and other domain experts can play an integral role in designing AI software that can partially automate legal analysis, overcoming some of the limitations in NLP capabilities.

* Dean Alderucci is the Director of Research for the Center for AI & Patent Analysis at Carnegie Mellon University. He was previously Chief IP counsel for a global financial services firm, and before that he was Chief Counsel for a business incubator. New York University School of Law, LL.M.; Columbia University, M.S. in Computer Science; Columbia University, M.S. in Operations Research.

TABLE OF CONTENTS

INTRODUCTION

As Artificial Intelligence and Machine Learning continue to transform numerous aspects of our everyday lives, their role in the legal profession is growing in prominence. AI can be used for tasks such as locating relevant documents in discovery and predicting the outcome of pending litigation. AI also has the potential to assist in legal analysis by helping the legal expert assess the consequences of applying relevant law to a particular fact pattern. Software capable of performing thorough legal analysis could provide numerous benefits to lawyers, legal scholars, judges, and others in the legal field. Such software could, for example, evaluate hundreds of contracts for particular weaknesses, generate a concise summary of a set of cases, answer questions about whether a particular fact pattern would violate a law, and accurately find cases that support or contradict a legal position.

A subfield of AI with particular applicability to legal analysis is Natural Language Processing (NLP). NLP deals with computational techniques for processing human languages such as English, making it a natural tool for processing the text of statutes, regulations, judicial decisions, contracts, and other legal instruments. Paradoxically, although state-of-the-art Machine Learning and NLP algorithms are able to learn and act upon patterns too complex for humans to perceive, they nevertheless perform poorly on many cognitive tasks that humans routinely perform effortlessly. This profoundly limits the ability of AI to assist in many forms of legal analysis and legal decision making.

This article offers two theses. First, notwithstanding impressive progress on NLP tasks in recent years, the state-of-the-art in NLP will remain unable to perform legal analysis for some time. Second, lawyers, legal scholars, and other domain experts can play an integral role in designing AI software that can partially automate legal analysis, overcoming some of the limitations in NLP capabilities.

This article provides a detailed but accessible explanation of the limitations of AI in legal analysis, as well as a path for overcoming these limitations. Part I briefly introduces Machine Learning and NLP techniques for a lay audience. This Part also explains common NLP "shortcuts" that have allowed AI software to make seemingly-intelligent analyses of text without actually performing anything we might consider to be understanding or reasoning. Unfortunately, these shortcuts do not scale well to the kinds of inferences and tasks that legal analysis requires. The usual candidates for improved Machine Learning performance, more training data and faster computers, also cannot help with these shortcomings.

Part II provides a non-technical explanation of why exactly many legal analysis tasks exceed the capabilities of the state of the art in AI. In summary, the most prominent Machine Learning and NLP algorithms cannot penetrate many types of "common sense" reasoning that are essential to the vast majority of legal analysis tasks. Moreover, such reasoning often relies on underlying knowledge that is not present in the text being processed and not otherwise accessible to the computer. This type of reasoning is especially prominent in various legal domains, making analysis in those areas difficult to automate with AI technology.

Nevertheless, understanding the capabilities and shortcomings of NLP illuminates a way to improve AI systems for legal analysis.

As described in Part III, the typical legal analysis task is composed of "subsidiary tasks," such as identifying the elements of a legal issue, evaluating potentially relevant facts and other pieces of information, and drawing simple inferences from those facts. The nature of the subsidiary tasks depends on the legal questions at issue. For example, determining whether a contract has formed involves different subsidiary tasks than determining whether the elements of a case of negligence are present.

Such subsidiary tasks are simpler than the complete legal analysis and can be feasible for AI processing, thereby allowing the process of legal analysis to be partially automated. AI software that processed subsidiary tasks would assist a human decision maker in more quickly locating relevant information and making legal determinations than if the human were to work unaided. The human would then remain responsible for combining the results of the subsidiary tasks in an appropriate manner to form the legal conclusion. This might involve weighing all factors of some analysis under a totality of the circumstances or some other manner of combining that relies on significant human judgment and cognition.

Moreover, this proposal requires that lawyers and other legal experts be part of the design of the AI software; only people with the requisite domain knowledge can identify the subsidiary tasks and potentially relevant information for a particular type of legal analysis, as well as define how exactly that information might be extracted from the text of documents. This shows that lawyers are a critical part of the solution, and lawyers with basic knowledge of AI and NLP systems can make significant contributions to his area.

AI software that performs subsidiary tasks for a legal analysis has additional potential beyond assisting the legal decision maker. By identifying information that is potentially relevant to a particular legal analysis task, the AI software is essentially identifying training data. Specifically, the potentially relevant information are inputs, and the legal decision ultimately rendered is the desired output for those inputs. A collection of inputs and corresponding desired outputs defines training data to teach software what outputs to produce for given inputs. Superior NLP systems might eventually use this training data to learn to perform decision making that more closely approximates the legal analysis performed by humans. When the legal expert identifies subsidiary tasks for a particular type of legal analysis, the expert is providing information on how to divide a complex legal reasoning task into tractable steps. In essence, the human instructs the AI software how to perform portions of the legal reasoning task.

Finally, Part IV outlines the proposed method by which legal experts and NLP system designers can collaborate to design AI software that can partially automate different types of legal analysis. Part IV also provides illustrative examples of this process for a legal analysis task in the field of patent law. This includes corresponding subsidiary tasks that AI software can perform to assist in patent-specific forms of legal analysis. The patent field is especially appropriate for this type of NLP-assisted legal analysis because much of patent analysis involves interpreting the text of the patent itself. Therefore, software can extract a significant amount of the information required in patent analysis from a readily-available document. Moreover, drafting techniques of attorneys who write patents can be reverse engineered to better extract from patents information that is useful to different types of legal analysis.

I. AI, MACHINE LEARNING, AND NLP

A. *AI Fundamentals*

AI involves computer software that appears to behave with human intelligence.[1] Machine Learning is the subfield of AI in which software employs statistical analysis of data to learn how to perform a task, such as categorizing a document or predicting the outcome of pending litigation.[2] A common method of developing Machine Learning systems involves providing the software with verified examples of some phenomenon, such as a set of emails that a human has classified as spam.[3] The software analyzes these examples to learn their characteristics and thereby learns to predict when a future example (e.g., a new email message) shares these characteristics.[4] In other words, the software learns from numerous examples, known as "training data" or a "training set," how to distinguish spam from non-spam emails.[5] In this manner, Machine Learning is an inductive technique; the software develops a model of the world induced from observation, rather than from general rules.[6]

1. Kevin D. Ashley, *Automatically Extracting Meaning from Legal Texts: Opportunities and Challenges*, 35 GA. ST. U. L. REV. 1117, 1117 (2019).
2. *Id.* at 1118; *see also* Harry Surden, *Machine Learning and Law*, 89 WASH. L. REV. 87, 89 (2014) (explaining that Machine Learning systems "are able to learn from experience and thus improve their performance over time").
3. Surden, *supra* note 2, at 93.
4. *Id.*
5. *Id.*
6. *Id.* at 91 n.21.

This Machine Learning approach, known as "supervised learning," is in contrast to a manual approach in which a programmer specifies a set of rules that instruct the computer exactly how to recognize the pattern of interest (e.g., which features of a new email suggest that it is probably spam).[7] In general, the more complex the problem the more training data is required for the software to learn the necessary patterns.[8] Machine Learning is the backbone of the technology used in impressive advancements such as self-driving cars[9] and facial recognition[10] systems.

Natural Language Processing (NLP) is a subfield of Artificial Intelligence that overlaps significantly with Machine Learning. NLP deals with computer processing and manipulating of "natural" languages, such as English or Spanish.[11] NLP can involve spoken language (speech) or written language (text). This article primarily deals with the latter since text is arguably more significant to the practice of law than speech; statutes, case law, briefs, contracts, and other important legal documents are composed of text, so text encodes a substantial amount of the information that is relevant to law and legal practice.

Among the different subfields of Artificial Intelligence, a defining characteristic of NLP is the use of some knowledge of the natural language being processed.[12] That knowledge can be deeply profound or extremely shallow, so this definition admits many very simple programs into the universe of NLP. For example, merely counting the words in a document can be considered an NLP task because that requires knowledge of what a word is.[13] In English it is relatively simple to identify a word as a sequence of alphabetic characters ending with a boundary character such as a space or punctuation mark.[14]

---

7. *Id.* at 93.

8. *See, e.g.*, Xiangxin Zhu et al., *Do We Need More Training Data?*, 119 INT'L J. COMPUTER VISION 1 (2015); Alon Halevy et al., *The Unreasonable Effectiveness of Data*, IEEE INTELLIGENT SYS., March/April 2009, at 8, 8-12 (2009).

9. *See generally* Brandon W. Jackson, *Artificial Intelligence and the Fog of Innovation: A Deep-Dive on Governance and the Liability of Autonomous Systems*, 35 SANTA CLARA HIGH TECH. L.J. 35 (2019); Harry Surden & Mary-Anne Williams, *Technological Opacity, Predictability, and Self-Driving Cars*, 38 CARDOZO L. REV. 121 (2016).

10. *See generally* Sharon Nakar & Dov Greenbaum, *Now You See Me. Now You Still Do: Facial Recognition Technology and the Growing Lack of Privacy*, 23 B.U. J. SCI. & TECH. L. 88 (2017).

11. DANIEL JURAFSKY & JAMES H. MARTIN, SPEECH AND LANGUAGE PROCESSING 30-31 (2nd ed. 2008).

12. *Id.*

13. *Id.*

14. *Id.* It is somewhat more difficult to identify English word boundaries. For example, "rock 'n roll" can be considered a single word, even though it contains spaces as well as non-

NLP techniques form the backbone of a wide variety of available products including virtual assistants, such as Apple's Siri, that are programmed to understand simple commands spoken by the user.[15] NLP decodes the user's speech, which might instruct the virtual assistant to set a timer or play particular music.[16] If the user requests an answer to a query then NLP enables the virtual assistant to generate a spoken answer.[17]

Text contains information that human readers typically identify and interpret with ease. Text is a type of "unstructured" data because text is not clearly arranged for interpretation by a computer.[18] This is in contrast to data that is "structured" because it contains information in well-specified locations defined by some predetermined organizational scheme.[19] For example, a spreadsheet of client names, addresses, and amounts owed is structured because one can easily translate a desired type of information, such as the amount owed by a client named "Jane Smith," into an exact location where that information can be found. In reality, there can be different degrees of structure in data, and even the text of natural language exhibits some structure imposed by grammar and explicit indicia such as section headings.[20] Indeed, NLP techniques can sometimes exploit aspects of this linguistic structure to extract information that is buried within the text.

B. NLP Shortcuts that Approximate Reasoning

AI software cannot understand legal texts at a human level of performance.[21] Nevertheless, some NLP techniques can provide impressive results without utilizing anything at all like the comprehension and reasoning that a human might perform. Using what Harry Surden has called "approximating intelligence by proxy,"[22] software can mimic the decisions that would have been produced by

---

alphabetic characters. *Id.* at 24. In languages such as Chinese, Japanese, and Thai it is even more difficult to define the boundaries of words. *Id.* at 25.

15. *See, e.g.*, Peng Lai Li, *Natural Language Processing*, 1 GEO. L. TECH. REV. 98, 103 (2016).

16. Robert D. Lang & Lenore E. Benessere, *Alexa, Siri, Bixby, Google's Assistant, and Cortana Testifying in Court*, 74 J. MO. B. 20, 20-21 (2018).

17. *Id.*

18. CHRISTOPHER D. MANNING ET AL., AN INTRODUCTION TO INFORMATION RETRIEVAL 1 (2009).

19. *Id.*

20. *Id.* at 1-2.

21. Ashley, *supra* note 1, at 1120.

22. Surden, *supra* note 2, at 97-98 (explaining that non-cognitive software can produce seemingly intelligent results on complex tasks even though the software does not use human-level cognition).

a human who employs high-level human cognitive processes. Two very common approximation techniques in NLP are "statistical techniques" and "selectional restrictions," both of which are described immediately below.

Statistical techniques lie at the heart of machine learning and other data-driven algorithms and have been responsible for the successes of the last several decades in such NLP tasks as automatic document summarization and automated question-answering.[23] In the realm of NLP, one statistical technique is to determine which words tend to occur together in the same sentence, in the same document, or within some predefined number of words of each other.[24] For example, software could analyze a set of documents to determine that the words "apple" and "peel" appear together in sentences much more often than the words "apple" and "digitize" do. This simple counting process allows the software to determine correlations among pairs of words. It is trivial for software to process thousands of documents and count word co-occurrences for all words in those documents. Publicly available general-purpose datasets, such as the Wikitext-103 corpus[25] (over 100 million words) and the One Billion Word Benchmark,[26] provide access to copious amounts of training data for various NLP applications. In the legal domain, the Caselaw Access Project provides access to all official, book-published United States case law, starting from 1658.[27] This corpus includes about 12 billion words from over 6 million United States legal cases.[28]

It is important to note that software can easily calculate how often pairs of words co-occur without having any knowledge whatsoever of the meaning of any of those words, much less the meaning of a complete sentence formed by a sequence of words. A manual analogy of the co-occurrence process would be a human reading a document in a language she does not understand and recording how

---

23. Hector J. Levesque et al., *The Winograd Schema Challenge*, *in* PROCEEDINGS OF THE THIRTEENTH INTERNATIONAL CONFERENCE ON PRINCIPLES OF KNOWLEDGE REPRESENTATION AND REASONING 552, 558 (Gerhard Brewka et al. eds., 2012).

24. JURAFSKY & MARTIN, *supra* note 11, at 110.

25. Jeremy Howard & Sebastian Ruder, *Universal Language Model Fine-Tuning for Text Classification*, *in* PROCEEDINGS OF THE 56TH ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS 328, 330 (Iryna Gurevych & Yusuke Miyao eds., 2018).

26. Ciprian Chelba et al., *One Billion Word Benchmark for Measuring Progress in Statistical Language Modeling*, *in* 1 FIFTEENTH ANNUAL CONFERENCE OF THE INTERNATIONAL SPEECH COMMUNICATION ASSOCIATION 2635-39 (H. Li & P. Ching eds., 2014).

27. *About*, CASELAW ACCESS PROJECT, https://case.law/about/ (last visited June 30, 2019).

28. *Historical Trends,* CASELAW ACCESS PROJECT, https://case.law/trends/ (last visited June 30, 2019).

often each possible pair of words occurs together in the same sentence.[29] No comprehension at all is needed for this tedious but straightforward counting task. Even more advanced corpus linguistics statistical techniques allow the meanings of words to be suggested by the words with which they co-occur.[30]

Although the software does not understand any of the words it processes, calculating word co-occurrences permits NLP software to perform feats of apparent text comprehension. For example, imagine that NLP software has processed billions of words from Wikipedia pages in order to calculate co-occurrence for every possible pair of words in the English language. Upon completing these calculations, the NLP software is tasked with answering the following question: *"The lions ate the zebras because they are predators. What does 'they' refer to: the lions or the zebras?"*

Answering the question requires that the pronoun "they" be resolved, that is, connected to either the word "lions" or the word "zebras." The human reader answers this question by comprehending the text and employing simple reasoning about the nature of the two types of animals or at least using knowledge that predators typically do the eating. The software can take a much simpler approach that does not rely on reasoning or knowledge of predators. Armed with statistics on word co-occurrences, the NLP software can resolve the pronoun "they" to "lions" because the word "lion" occurs more frequently with the word "predator" than does the word "zebra."[31] Similarly, the same word co-occurrence data can allow the NLP software to answer a question such as *"Which animal is a predator: a lion or a zebra?"* Although this is more akin to an educated guess than a reasoned conclusion, it is a surprisingly effective NLP technique.

Beyond simple word co-occurrence data, more sophisticated statistical techniques can use additional information on the words that appear in sentences. For example, a general technique known as

---

29. As a starker example that no understanding is required in calculating word co-occurrences, every occurrence in a set of documents of a particular English word could be exchanged with a corresponding nonsense word. For example, every occurrence of "apple" in a document could be exchanged with "abcde," and every occurrence of "peel" could be exchanged with "vwxyz." The number of times that "apple" and "peel" co-occur is the same as the number of times that "abcde" and "vwxyz" co-occur. Therefore, the NLP software could still calculate the number of times that "abcde" and "vwxyz" co-occur.

30. *See generally* Stefan Th. Gries & Brian G. Slocum, *Ordinary Meaning and Corpus Linguistics*, 2017 BYU L. REV. 1417 (2017).

31. Altaf Rahman & Vincent Ng, *Resolving Complex Cases of Definite Pronouns: The Winograd Schema Challenge*, *in* PROCEEDINGS OF THE 2012 JOINT CONFERENCE ON EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING AND COMPUTATIONAL NATURAL LANGUAGE LEARNING 777, 781 (Jun'ichi Tsujii et al. eds., 2012).

language modeling considers the order of words in a sentence in predicting the next word that might likely follow. Neural language models can use all words in a sentence or set of sentences to predict the sequences of words that likely precede or follow a word.[32] Language modeling significantly increases the power of NLP systems to process text, albeit without endowing the software with any understanding of that text whatsoever.

The second type of NLP shortcut, "selectional restrictions," uses a constraint on the type of word that can be used in certain situations, often as a constraint on what type of word can be the subject or object of a particular verb.[33] For example, consider the following statement and question: *"Our graduate students published 20 papers this year and, apparently, a few of them authored some books. What does 'them' refer to: the graduate students or the papers?"*[34]

The pronoun "them" stands in for the subject of the verb "authored," and only people (and perhaps some AI software)[35] can author content. Therefore, the pronoun "them" should resolve to "graduate students," a type of person, rather than "papers," which are not a type of person. NLP software can properly understand that the pronoun "them" refers to "graduate students," if it knows that (1) only people can author, (2) graduate students are people, and (3) papers are not people. Of these three pieces of information, the first involves a common-sense relationship between the verb "author" and the subject of that verb, i.e., the subject of the verb "author" must be a type of person. The second and third involve "type-of" relationships between nouns: a student is a type of person, and a paper is not a type of person. Two online databases, WordNet and FrameNet, provide information on exactly these three relationships, allowing NLP software to utilize information on these relationships.

WordNet is a manually-constructed online database that links words and conceptual relations.[36] WordNet entries store "type-of"

32. YOAV GOLDBERG, NEURAL NETWORK METHODS FOR NATURAL LANGUAGE PROCESSING 109-12 (Graeme Hirst ed., 2017).

33. JURAFSKY & MARTIN, *supra* note 11, at 368.

34. WALID S. SABA, ON THE WINOGRAD SCHEMA: SITUATING LANGUAGE UNDERSTANDING IN THE DATA-INFORMATION-KNOWLEDGE CONTINUUM (2019), https://arxiv.org/abs/1810.00324.

35. *See, e.g.*, Russ Pearlman, *Recognizing Artificial Intelligence (AI) as Authors and Inventors Under U.S. Intellectual Property Law*, 24 RICH. J.L. & TECH. 2, 3 (2018) (arguing for recognition of AI authorship and inventorship); Shlomit Yanisky-Ravid, *Generating Rembrandt: Artificial Intelligence, Copyright, and Accountability in the 3A Era—The Human-Like Authors Are Already Here—A New Model,* 2017 MICH. ST. L. REV. 659, 660 (2017) (addressing the copyrightability of works generated by AI systems).

36. JURAFSKY & MARTIN, *supra* note 11, at 493; Christiane Fellbaum, *WordNet(s)*, *in* 13 ENCYCLOPEDIA OF LANGUAGE AND LINGUISTICS 665, 665 (Ron Asher ed., 2006).

relationships for thousands of words. For example, WordNet defines the word "hamburger" as a type of "sandwich," which in turn is defined as a type of food."[37] This hierarchical information can then be used in applying restrictions on which nouns may serve as the subject or object of particular verbs. Those restrictions are captured in the online FrameNet database.[38] For example, the FrameNet entry for "eat" explains that the subject of "eat" must be a person and the object must be a type of food.

Selectional restrictions capture aspects of knowledge in a way that purely statistical techniques do not. This knowledge has been meticulously sourced from human efforts over many years. When people manually created the entries in the WordNet and FrameNet databases, they reduced to digital form significant real-world concepts and how English words relate to those concepts. NLP programs can benefit from such codified knowledge, typically by determining whether nouns associated with a particular verb are of the correct type for that verb.

Both statistical techniques and selectional restrictions allow impressive performance on text processing challenges, masking the software's lack of text understanding. However, this shortcoming is revealed on more challenging tasks that require the software to possess some form of language understanding.

II. LIMITATIONS OF AI IN COMMON SENSE REASONING

In light of NLP's impressive successes, it is somewhat surprising that even state-of-the-art algorithms struggle with tasks that could reasonably be considered simple, or even trivial, for humans to perform. These deceptively-challenging tasks fall under the rubric of "common sense reasoning," a term that encompasses different types of cognitive skills and real-world knowledge. Software that cannot perform simple reasoning would be extremely unlikely to perform more complex legal reasoning tasks, which often rely upon simple reasoning and knowledge about the world.

Since NLP shortcuts can endow software with the appearance of human-level understanding, researchers have explored tests to distinguish software that truly understands the meaning of text from software that merely mimics such understanding. Below, I describe

---

37. JURAFSKY & MARTIN, *supra* note 11, at 369-70.

38. *See generally* JURAFSKY & MARTIN, *supra* note 11, at 362; Collin F. Baker et al., *The Berkeley FrameNet Project*, *in* PROCEEDINGS OF THE 17TH INTERNATIONAL CONFERENCE ON COMPUTATIONAL LINGUISTICS 86 (1998).

the Winograd Schema Challenge, problems which have been specifically designed so that they cannot be solved by merely using NLP shortcuts like statistical techniques or selectional restrictions. Successfully solving these problems is believed to require at least rudimentary commonsense reasoning by the NLP system.

Software that exhibits common sense reasoning at the level of a human is a formidable problem. Software that could do so would likely pass or come close to passing the Turing Test,[39] a touchstone for determining whether a computer is able to behave as intelligently as a human.[40] In the Turing Test, software engages in an extended conversation via teleprinter with a human, who does not know whether she is conversing with a human or machine.[41] If the human is fooled into believing the machine is another human, then the machine has passed the Turing Test and can be considered to be thinking to some extent.

Unfortunately, neither statistics nor selectional restrictions alone can be extended to handle even simple tasks of commonsense reasoning. In short, neither shortcut involves anything we might consider to be "reasoning" about the real world, nor do they encapsulate the types of real-world knowledge possessed by humans, including small children. Because these limitations are inherent to the techniques themselves, they are unlikely to be overcome with additional training data or much faster computers, two staples of increased Machine Learning performance in recent years.

Some simple examples will illustrate common sense reasoning and how software fails on such tasks. Consider the following question. *"The trophy doesn't fit in the brown suitcase because it is too small. What does 'it' refer to: trophy or suitcase?"* A human has no difficulty recognizing that the pronoun must refer to "suitcase." This conclusion is based on simple spatial reasoning.[42] If the suitcase is smaller than the trophy then the trophy cannot fit in the suitcase, but if the trophy is smaller than the suitcase there would be no problem fitting the trophy in the suitcase.

Statistical analysis does not provide an easy way to answer this question. There is no reason to believe that either of the words "trophy" or "suitcase" would co-occur with "small" more frequently than

39. *See generally* A.M. Turing, *Computing Machinery and Intelligence*, 59 MIND 433 (1950).

40. *See id.* at 459-60.

41. *See id.*

42. *See* Ernest Davis, *Qualitative Spatial Reasoning in Interpreting Text and Narrative*, 13 SPATIAL COGNITION & COMPUTATION 264, 264 (2013) (demonstrating that very simple natural language texts can "raise problems in commonsense spatial knowledge and reasoning of surprising logical complexity and geometric richness").

the other word. Selectional restrictions are also unavailing because both a trophy and a suitcase are the kinds of things that can be referred to as "too small."

The above trophy-suitcase question is selected from the Winograd Schema Challenge.[43] The Winograd Schema Challenge is "designed so that the correct answer is obvious to the human reader, but cannot easily be found using selectional restrictions or statistical techniques over text corpora."[44] To discourage the use of NLP shortcuts that do not require text understanding, Winograd Schema problems are arranged in pairs, each differing by only a single word. This small change results in a question with the opposite answer. For example, these two questions differ in only the underlined words:

"The trophy doesn't fit in the brown suitcase because it is too <u>small</u>. What does 'it' refer to?"

"The trophy doesn't fit in the brown suitcase because it is too <u>big</u>. What does 'it' refer to?"

The answer to the first is "suitcase" while the answer to the second is "trophy." The pairing of problems in this manner, each almost identical but requiring opposite answers, negates NLP shortcuts that might use the word order or other clever features of the remaining words in the problem as a hint to the proper answer. In other words, the problems are designed so that the other (unchanging) words in the problem pair do not provide any statistical clues to the correct answer. If they did, they would provide the same clue for both problems, which have opposite answers; use of statistical clues would guarantee that one of the pair is answered incorrectly.

The trophy-suitcase problem requires commonsense spatial reasoning. There are many other types of commonsense reasoning. Indeed, humans perform commonsense reasoning so easily that it can be difficult to identify exactly what kinds of knowledge and inferences are involved in answering a question. The designers of the Winograd Schema Challenge believe that software that successfully answers the majority of the questions "will need to have commonsense knowledge about space, time, physical reasoning, emotions, social constructs, and a wide variety of other domains."[45] As an example of commonsense reasoning about cause and effect, also

43. *See* Levesque et al., *supra* note 23, at 554; *see also* Ernest Davis, *Collection of Winograd Schemas*, NYU COURANT COMPUTER SCI., https://cs.nyu.edu/faculty/davise/papers/WinogradSchemas/WSCollection.html (last visited June 28, 2019).

44. Levesque et al., *supra* note 23, at 552.

45. *Id.* at 558.

known as causal implication, consider the following Winograd Schema problem pair.[46]

"Anna did a lot better than her good friend Lucy on the test because she had studied so hard. Who studied hard?"

"Anna did a lot worse than her good friend Lucy on the test because she had studied so hard. Who studied hard?"

The answer to the first is Anna, and the answer to the second is Lucy. This problem pair requires an understanding of the most likely effect of studying hard, which, in turn, requires an understanding of the nature of tests, studying, and the relationship between studying and test performance.

The next problem pair requires knowledge of other basic real-world relationships.

"Sam broke both his ankles and he's walking with crutches. But a month or so from now they should be better. What should be better?"

"Sam broke both his ankles and he's walking with crutches. But a month or so from now they should be unnecessary. What should be unnecessary?"

Understanding that the first answer is "ankles" and the second is "crutches" requires understanding rudimentary details about injuries, healing, and why people use crutches. This knowledge is the kind acquired through ordinary human interactions. The next problem draws upon knowledge of human social interactions.

"Joan made sure to thank Susan for all the help she had given. Who had given help?"

"Joan made sure to thank Susan for all the help she had received. Who had received help?"

To conclude that the first answer is "Susan" and the second is "Joan" requires that the software understand basic social aspects of gratitude and the general reasons that people express thanks.

The Winograd Schema questions are extremely simple and many could be answered by children.[47] Moreover, the Winograd Schema Challenge evaluates commonsense reasoning by employing an extremely narrow problem: resolving referential ambiguity, that is, deciding which noun a pronoun or possessive adjective refers to.[48] This problem is a limited test of common sense reasoning; only a

46. *See generally* Melissa Roemmele et al., *Choice of Plausible Alternatives: An Evaluation of Commonsense Causal Reasoning*, *in* AAAI SPRING SYMPOSIUM ON LOGICAL FORMALIZATIONS OF COMMONSENSE REASONING 21 (Ernest Davis et al. eds., 2011).

47. *See* Davis, *supra* note 43. Of the 150 problem pairs listed, many require only extremely basic real-world knowledge gained from routine human interactions.

48. Levesque et al., *supra* note 23, at 557.

single pronoun is to be resolved, and the noun to which the pronoun refers is known to exist in a single sentence. In contrast, it is fairly common for text to have more complicated structures in which information from multiple sentences must be aggregated to make an inference. It is also not uncommon that the referent of a pronoun is not present in a previous sentence but is instead implicit. In other words, the Winograd Schema Challenge appears to be a fair test of commonsense reasoning and certainly is not an unfairly challenging one.

People outside the field of AI can be forgiven for wondering what all the fuss is about. Is it really the case that software with access to billions of words and millions of web pages nevertheless cannot perform well at answering commonsense reasoning questions? After all, most people could solve the Winograd Schema problems easily and without thinking, in many cases exhibiting only the level of reasoning mastered by children in elementary school. Surely software can do so as well.

Unfortunately, software is not up to this task. The Winograd Schema Challenge was held at the 25th International Joint Conference on Artificial Intelligence (IJCAI-16) on July 2016. None of the NLP systems entered by contestants were able to advance from the first round to the second round.[49] The highest scoring entry answered less than half of all questions correctly.[50] These lackluster results on simple questions may be surprising, especially given the performance of NLP systems on seemingly harder tasks. For example, in 2011, IBM's Watson system answered "Jeopardy!" questions

---

49. Ernest Davis et al., *The First Winograd Schema Challenge at IJCAI-16*, AI MAG., Fall 2017, at 97, 97.

50. *Id.* at 98 tbl.1. However, in a paper released on June 19, 2019, Carnegie Mellon University and Google Brain researchers achieved 90.4% accuracy on WNLI, a modified version of the Winograd Schema Challenge. ZHILIN YANG ET AL., XLNET: GENERALIZED AUTOREGRESSIVE PRETRAINING FOR LANGUAGE UNDERSTANDING tbl.4 (2020), https://arxiv.org/abs/1906.08237. Their paper was released just days before the submission of this article and before publication of any subsequent analysis of the researchers' results. However, researchers have hypothesized that Machine Learning models that perform inordinately well on analogous reasoning challenges rely on the questions containing subtle "spurious statistical cues" to the proper answer. TIMOTHY NIVEN & HUNG-YU KAO, PROBING NEURAL NETWORK COMPREHENSION OF NATURAL LANGUAGE ARGUMENTS (2019), https://arxiv.org/abs/1907.07355; *see also* Abhijit Mahabal, *Do NLP Entailment Benchmarks Measure Faithfully?*, MEDIUM (July 19, 2019), https://towardsdatascience.com/do-nlp-entailment-benchmarks-measure-faithfully-e600212692b3. It is possible that some or all of XLNet's performance on the WNLI task comes from exploiting statistical clues in the data and not from the application of deep understanding of the real-world or common sense reasoning.

better than human champions of the game.[51] However, as explained above, NLP shortcuts can solve some types of natural language problems without requiring that the software understand anything about the text or the real world.[52] Researchers have noted that commonsense knowledge and reasoning played a limited role in Watson's success.[53] Among other strategies, Watson made use of words and noun phrases in the questions that specify the type of the answer "without any attempt to understand its semantics."[54]

A common thread among different flavors of commonsense reasoning is the need to draw upon different forms of "background knowledge": information that is not explicit in the text being processed. For example, solving the trophy-suitcase problem requires knowledge of spatial reasoning and how the size of objects has implications for fitting one inside of the other. Nothing in the problem's single sentence contains this information. Likewise, even if that single sentence was present in a more extensive document it is unlikely that any part of the document would contain an explanation of the requisite spatial reasoning information. Not only does unstated background knowledge help with understanding text, the answer to questions asked about text may actually be a word or phrase not present in the sentence or document. For example, the following question is similar to a Winograd Schema problem because it asks which noun the pronoun "it" refers to. However, it has a crucial difference that, although small, makes it inappropriate for the Winograd Schema Challenge.

"Dave told everyone in school that he wants to be a guitarist because he thinks it is a great sounding instrument. What does 'it' refer to?"

---

51. *See* Betsy Cooper, *Judges in Jeopardy!: Could IBM's Watson Beat Courts at Their Own Game?*, 121 YALE L.J. ONLINE 87, 87 (2011) (describing how IBM's Watson system beat the world's top "Jeopardy!" Champions); David Ferrucci et al., *Building Watson: An Overview of the DeepQA Project*, AI MAG., Fall 2010, at 59, 59.

52. *See* Davis, *supra* note 42, at 291 (explaining that the "many successes of applications of corpus-based [Machine Learning] to natural language text are very explicitly based on the avoidance of the kind of [common sense] inferences discussed here"); *see also* Ernest Davis & Gary Marcus, *Commonsense Reasoning and Commonsense Knowledge in Artificial Intelligence,* COMM. ACM, Sept. 2015, at 92, 94 ("Almost without exception, current computer programs to carry out language tasks succeed to the extent the tasks can be carried out purely in terms of manipulating individual words or short phrases, without attempting any deeper understanding; commonsense is evaded . . . .").

53. A. Kalyanpur et al., *Structured Data and Inference in DeepQA*, 56 IBM J. RES. & DEV. 1 (2012)*; see also* Davis & Marcus, *supra* note 52, at 94 ("The key techniques in Watson are mostly of the same flavor as those used in programs like Web search engines . . . . There is no evidence that Watson is anything like a general-purpose solution to the commonsense problem.").

54. Ferrucci et al., *supra* note 51, at 70.

Here "it" must refer to "guitar," a noun not mentioned in the text.[55] Clearly "it" cannot refer to any of the other nouns in the sentence because "it" must be a type of "instrument" and none of the other nouns are instruments. In the context of the sentence the implication is obvious, provided we possess the background knowledge that a guitarist plays a guitar. The ability to access unstated background knowledge is used to solve a wider array of problems besides understanding what a pronoun refers to.[56]

This simple example presents difficulties for common Machine Learning techniques, which generally rely exclusively on processing massive amounts of text data but do not attempt to codify and represent real-world background knowledge.[57] Such techniques will fail to recognize concepts that humans implicitly assume "due to our *shared* background knowledge of the world and the way we talk about it in ordinary spoken language."[58] In essence, purely data-driven Machine Learning techniques "cannot model what is not there."[59]

In summary, commonsense reasoning has largely remained impervious to the most powerful Machine Learning and NLP techniques. Solving most commonsense reasoning problems via software seems to require that the software have access to a large repository of the background knowledge most people possess but take for granted. There is no accepted method for creating or organizing such a repository of background knowledge.[60]

In Part III, below, this article explores legal analysis. In particular, this article explain how legal analysis typically involves background knowledge that is not disclosed in the text being processed, as well as other aspects of commonsense reasoning that continue to stymie NLP software. This characteristic of legal analysis suggests

---

55. WALID S. SABA, IS THERE A 'SIMPLE' MACHINE LEARNING METHOD FOR COMMONSENSE REASONING?: A SHORT COMMENTARY ON TRINH & LE (2018), https://arxiv.org/abs/1810.00521.

56. SABA, *supra* note 34.

57. SABA, *supra* note 55. Saba provides evidence that the NLP technique proposed by Trinh & Le, and more broadly any other data-driven technique, "will not scale into a workable and reasonable solution" because "data-driven approaches, true to their name, can only make generalizations based on the data they process," but not background knowledge the text does not describe.

58. *Id.* (emphasis in original).

59. *Id.*

60. Adam Richard-Bollans et al., *The Role of Pragmatics in Solving the Winograd Schema Challenge*, *in* PROCEEDINGS OF THE THIRTEENTH INTERNATIONAL SYMPOSIUM ON COMMONSENSE REASONING (Andrew S. Gordon et al. eds., 2017) (concluding that "it is clear that the necessary commonsense knowledge [for solving such problems] would involve the formalization of a notoriously extensive knowledge base. How to obtain and organize such a large knowledge base is unclear.").

that there will be limitations in automating legal reasoning with NLP software.

III. LEGAL ANALYSIS

A. *Legal Analysis: Common Sense and More*

Broadly speaking, the goal of legal analysis is to interpret or apply the law to particular facts. This involves identifying the legal authority that may apply and the issues in the fact pattern that may be relevant. It can sometimes be challenging merely to understand which law and legal principles apply to a fact pattern. Moreover, even if the relevant law and facts are certain, the resolution of the legal question might remain unclear because reasonable arguments can be made for contradictory conclusions.

One might think that the process of legal reasoning would be straightforward for computers. Once the facts and law are provided to the computer, the computer would diligently determine which legal conditions are satisfied by the facts and then provide the conclusion.[61] Unfortunately, NLP software that attempts to perform its own form of legal reasoning would have to contend with several conspicuous obstacles. The reasoning in judicial decisions may rely on unstated background knowledge or commonsense reasoning, which presents serious difficulties for NLP software attempting to understand the state of the law.[62] Compounding this difficulty is the fact that judges will often explicitly use intuition or a similar concept in their reasoning,[63] even though that intuition can be unexplainable and without explicit justification.[64] Courts will often link or even equate judicial intuition with common sense.[65]

An additional difficulty for NLP software conducting legal reasoning lies in determining the specific legal principles that are presented by a body of case law. The text of a decision does not always make clear the nature of the legal reasoning involved. The judge's reasoning can be incomplete because crucial assumptions are not

---

61. KEVIN D. ASHLEY, ARTIFICIAL INTELLIGENCE AND LEGAL ANALYTICS: NEW TOOLS FOR LAW PRACTICE IN THE DIGITAL AGE 50 (2017).

62. *See* Part II *supra*.

63. R. George Wright, *The Role of Intuition in Judicial Decisionmaking*, 42 HOUS. L. REV. 1381, 1385 (2006).

64. *Id.* at 1386.

65. *Id.*; *see also* Richard A. Posner, *The Jurisprudence of Skepticism*, 86 MICH. L. REV. 827, 838-41 (1988) (arguing that "practical" reasoning, including common sense and intuition, can answer most of the legal questions that logic cannot answer).

explicit.[66] Indeed, with enough of the legal reasoning lacking, the decision can appear to be arbitrary.[67]

Extracting the state of the law from judicial decisions can be especially thorny because such law might depend strongly on analogical arguments.[68] To understand judicial decisions and how to apply them, NLP software would need to be capable of drawing appropriate analogies between different fact patterns.[69] Crucially, legal reasoning involves identifying which similarities are relevant, not merely that similarities exist.[70] Just what makes distinct sets of facts similar to each other "depends on the principle for which the initial case is said, on reflection, to stand."[71] It would be especially challenging for software to identify principles because they depend on "evaluative judgments" that try to tease out whether the principle would be inconsistent with "anything . . . to which the legal system has committed itself."[72] These principles can depend on moral and political theories, intuitions of public policy, avowed or unconscious, and even the prejudices of judges.[73] In light of all of the above, it can be difficult even for legal experts to describe precisely what makes one legal argument stronger than another.[74]

Unsurprisingly, complete start-to-finish legal analysis by NLP software is an imposing and, for the moment, infeasible goal. Legal analysis requires commonsense reasoning, both to understand which issues arise from a fact pattern as well as to determine which legal principles should be applicable. Legal analysis also requires unstated background knowledge about the real world. Some of this background knowledge may, furthermore, be specific to a domain. For example, decisions in the field of patent law frequently rely on

---

66. Richard Warner, Note, *Three Theories of Legal Reasoning*, 62 S. CAL. L. REV. 1523, 1523 (1989) (discussing the appearance of arbitrariness in many legal decisions); *see also* Kevin D. Ashley & Stefanie Bruninghaus, *Computer Models for Legal Prediction*, 46 JURIMETRICS J. 309, 315-16 (2006) ("Judges may not have disclosed the features that influenced their decision or stated their rationales accurately or completely.").

67. Warner, *supra* note 66, at 1523.

68. *See generally* Scott Brewer, *Exemplary Reasoning: Semantics, Pragmatics, and the Rational Force of Legal Argument by Analogy*, 109 HARV. L. REV. 923, 927 (1996) (providing a detailed model of the process of reasoning by analogy); Emily Sherwin, *A Defense of Analogical Reasoning in Law*, 66 U. CHI. L. REV. 1179 (1999) (arguing that although analogical reasoning is an unscientific practice with imperfect results, it should be defended on the basis of its epistemic and institutional advantages).

69. Cass R. Sunstein, *Of Artificial Intelligence and Legal Reasoning*, 8 U. CHI. L. SCH. ROUNDTABLE 29, 31 (2001).

70. *Id.*

71. *Id.* at 31-32.

72. *Id.* at 32-33.

73. Oliver Wendell Holmes, Jr., THE COMMON LAW 1 (Dover Publ'ns 1991) (1881).

74. Brett G. Scharffs, *The Character of Legal Reasoning*, 61 WASH. & LEE L. REV. 733, 737 (2004).

the details of a particular field of technology or on the capabilities of people who practice in a particular field. Given that general commonsense reasoning is beyond the capabilities of state-of-the-art software, the additional burden of utilizing background knowledge and evaluating principles of case law renders general-purpose legal analysis by software an impossibility with current technology.

To address this daunting challenge, we begin by noting that the typical legal analysis task is composed of "subsidiary tasks," such as evaluating potentially relevant facts and other pieces of information and then drawing simple inferences from those facts. Making a legal analysis task tractable for partial automation by software involves, as a first step, decomposing the task into its subsidiary tasks. The subsidiary tasks will naturally depend on the nature of the legal analysis. It will be instructive to attempt to identify the subsidiary tasks present in the legal analysis task presented immediately below as well as in Section C of Part IV.

B. Legal Analysis Example: Patent Claim Obviousness

In this section I briefly summarize a core patent law doctrine, obviousness,[75] in order to provide a concrete example of the tasks performed in conducting a specific type of legal analysis. This example will illustrate aspects of legal analysis that are difficult for software to perform and will suggest a possible solution that allows properly-designed AI software to partially-automate legal analysis. It will also illustrate an aspect of the legal analysis that is rife with common sense reasoning.

Under § 103 of the Patent Act, a claim of a patent is invalid "if the differences between the claimed invention and the prior art are such that the claimed invention as a whole would have been obvious . . . to a person having ordinary skill in the art."[76] The goal of the non-obviousness requirement is to grant patents only for those inventions that represent a sufficiently large advance over previously-known technology, i.e., over the "prior art."[77]

---

75. *See, e.g.*, ROBERT PATRICK MERGES & JOHN FITZGERALD DUFFY, PATENT LAW AND POLICY: CASES AND MATERIALS 643 (3d ed. 2007) (introducing the nonobviousness requirement as "the most important of the basic patent requirements").

76. 35 U.S.C. § 103 (2012).

77. Lee Petherbridge, *On the Development of Patent Law*, 43 LOY. L.A. L. REV. 893, 907-08 (2010); *see also* Sensonics, Inc. v. Aerosonic Corp., 81 F.3d 1566, 1570 (Fed. Cir. 1996) (referring to a "significant and unobvious advance over" previous technology).

Obviousness is a question of law,[78] but it relies upon factual inquiries including the scope and content of the prior art, the differences between the prior art and the claims of the patent, and the level of ordinary skill in the art.[79] Additional facts such as commercial success of the invention, long felt but unsolved needs solved by the invention, and the failure of others to create the invention can also be relevant to determining whether a patent claim is obvious.[80]

The statute requires that obviousness be judged from the perspective of the "person of ordinary skill in the art," a theoretical construct that is not descriptive of any particular individual.[81] In this sense, the person having ordinary skill in the art is "not unlike the 'reasonable man' and other ghosts in the law."[82] This fictitious person is endowed with all existing information in the prior art.[83] The claims must be invalidated under § 103 of the Patent Act only if that hypothetical person would find the claimed invention to be obvious.

A full legal analysis of the obviousness of a patent claim requires understanding the patent's technology, the state of the art in the field of that technology, and the differences between the two. Although this is a necessary factual assessment, it is not sufficient. The analysis also mandates an inquiry into what exactly the person possessing ordinary skill would conclude about the obviousness of the claimed invention. This analysis must incorporate the scope of that skilled person's knowledge and technical abilities. These abilities can be assessed by considering factors such as the educational level of the inventor, the types of problems encountered in the technical field, prior solutions to those problems, how fast innovations are made, the sophistication of the technology, and the educational level of workers in the field.[84]

This difficult assessment is complicated by the fact that it involves considerations with very ill-defined boundaries. The obviousness determination must consider the creativity of a person of ordinary skill in the art.[85] Also relevant is that person's common

---

78. Graham v. John Deere Co., 383 U.S. 1, 17-18 (1966).

79. *Id.* at 17.

80. *Id.* at 17-18.

81. Endress + Hauser, Inc. v. Hawk Measurement Sys. Pty. Ltd., 122 F.3d 1040, 1042 (Fed. Cir. 1997) (quoting Custom Accessories, Inc. v. Jeffrey-Allan Indus., Inc., 807 F.2d 955, 963 (Fed. Cir. 1986)).

82. Panduit Corp. v. Dennison Mfg. Co., 810 F.2d 1561, 1566 (Fed. Cir. 1987).

83. Standard Oil Co. v. Am. Cyanamid Co., 774 F.2d 448, 454 (Fed. Cir. 1985).

84. Helifix, Ltd. v. Blok-Lok, Ltd., 208 F.3d 1339, 1347 (Fed. Cir. 2000) (citing *Custom Accessories, Inc.*, 807 F.2d at 962).

85. KSR Int'l Co. v. Teleflex Inc., 550 U.S. 398, 420-21 (2007).

sense reasoning abilities.[86] This can be especially difficult for data-driven methods that extract information from legal and other texts because the extent of creativity and common sense of the authors may not be reflected within these documents.[87]

Understanding the range of abilities possessed by the person of ordinary skill in the art is central to many patent law doctrines besides obviousness, including enablement[88] and indefiniteness[89] of patent claims. Briefly, the enablement requirement demands that the patent specification teach persons of ordinary skill in the art how to make and use the claimed invention without undue experimentation.[90] The purpose of the enablement requirement is "to extract meaningful disclosure of the invention and, by this disclosure, advance the technical arts."[91] Similarly, to satisfy the indefiniteness requirement the patent's claims must inform those skilled in the art about the scope of the invention with reasonable certainty.[92] This ensures that the public has clear notice of the exclusionary rights provided by the patent.[93]

Like the reasonable person standard present in so many other areas of the law, assessing the legal contours of the person of ordinary skill intimately involves considerations like common sense and creativity that lack clear boundaries. As described above in Part II, common sense reasoning is difficult for state-of-the-art NLP techniques to carry out. Moreover, legal text that is processed by NLP software does not explicitly describe much of the knowledge on which common sense reasoning is based. Therefore, even significantly more sophisticated software could not extract this knowledge from those texts. In summary, some portions of legal reasoning lie beyond the reach of AI technology and are likely to remain so for some time. Accepting this limitation can lead to an improved model for AI systems that assist in legal reasoning tasks, as explained in Part IV.

---

86. *Id.* at 420.

87. *Id.* at 419; *see also* Perfect Web Techs., Inc. v. InfoUSA, Inc., 587 F.3d 1324, 1329 (Fed. Cir. 2009) (explaining that the obviousness inquiry may include "common sense available to the person of ordinary skill [in the art] that do not necessarily require explication in any reference or expert opinion").

88. 35 U.S.C. § 112(a) (2012).

89. *Id.* § 112(b).

90. MagSil Corp. v. Hitachi Glob. Storage Techs., Inc., 687 F.3d 1377, 1380 (Fed. Cir. 2012).

91. Invitrogen Corp. v. Clontech Labs., Inc., 429 F.3d 1052, 1070 (Fed. Cir. 2005) (citing Koito Mfg. Co., Ltd. v. Turn-Key-Tech, LLC, 381 F.3d 1142, 1155 (Fed. Cir. 2004)).

92. Nautilus, Inc. v. Biosig Instruments, Inc., 572 U.S. 898, 901 (2014).

93. *Id.* at 909. Clear notice is necessary to avoid a "zone of uncertainty which enterprise and experimentation may enter only at the risk of infringement claims." *Id.* at 909-10 (quoting United Carbon Co. v. Binney & Smith Co., 317 U.S. 228, 236 (1942)).

IV. IMPROVING AI FOR LEGAL ANALYSIS

A. *Existing Methods of Legal Reasoning by AI*

There is a large and growing body of work on computational models of legal reasoning.[94] Ashley defines a computational model of legal reasoning as "a computer program that implements a process evidencing attributes of human legal reasoning," which "may involve analyzing a situation and answering a legal question, predicting an outcome, or making a legal argument."[95] Ashley distinguishes computational models of legal reasoning from legal text analytics, which is defined as the discovery of knowledge from legal text.[96] Informally, the difference is between the reasoning and obtaining the information on which reasoning is based. This distinction is important for the current discussion of AI and legal analysis.

Computational models of legal reasoning, like all forms of legal reasoning, depend on knowledge such as the state of the law. However, today's computers cannot extract all required knowledge directly from legal texts such as cases, statutes, regulations, and contracts. Instead, human experts must read legal texts, extract relevant knowledge in those texts, and manually translate this knowledge into a form that software can use to perform its legal reasoning.[97] This "knowledge representation bottleneck" prevents NLP software from performing legal analysis from start to finish without a significant investment of human labor.[98]

Legal text analytics is designed to extract relevant information from legal texts. For example, software can automatically annotate legal texts to indicate various concepts and their relations to each other.[99] Software also can process case texts for argument-related information about the roles of sentences in a case.[100] Although the software can intelligently process argument-related information, it does not understand those arguments to any profound extent.[101] Even if legal text analytics could identify all information it attempts to locate in legal text, this class of techniques nevertheless misses

---

94. *See generally* ASHLEY, *supra* note 61, at 50-201 (describing the literature on different computational models for statutory reasoning, case-based legal reasoning, predicting legal outcomes, and legal argument).
95. *Id.* at 12.
96. *Id.* at 13.
97. *Id.* at 12-13.
98. *Id.*
99. *Id.* at 205.
100. *Id.* at 334.
101. *Id.* at 441.

a significant type of information: background knowledge and other implicit information not stated in the text.

B. A Proposal to Advance AI-Driven Legal Analysis

In summary, two limitations of Machine Learning systems are particularly relevant to legal analysis: performing common sense reasoning and incorporating background knowledge that is not explicit in the text being processed. Both skills are required in legal analysis, and AI generally performs poorly at both. Therefore, we should minimize the role of software in both, while at the same time utilizing software as much as possible in the remaining steps of legal analysis.

The proposal below is founded on the assumption that it is futile to ask the software to perform the cognitive manipulations of information required in commonsense reasoning. However, we can, among other things, command the software to search for and present to the human legal expert the types of information that might possibly be relevant to the legal analysis being performed. This in turn requires that the software understand the steps involved in that legal analysis, what kinds of information are potentially relevant, and how those kinds of information might be expressed in text. The software will then provide the human with information that is helpful or indispensable to the legal analysis, though the human retains ultimate responsibility for drawing the conclusions that follow from that information.

This computational model of legal reasoning partitions responsibility for legal analysis between the human and the legal expert. The collaborative activity, known as "cognitive computing," allows humans and computers to each perform the kinds of intelligent activities that they can do best.[102] This is very much in line with Ashley's observation that although software cannot read legal texts in the sense that humans read, it can nevertheless intelligently process those texts to identify elements that are relevant to a problem and bring these elements to the user's attention.[103]

I begin by describing the proposed process for designing an AI system that is tailored to a specified legal domain. I then expound on this proposal by outlining a specific example of its usage for the patent law question of claim definiteness.

The first step in the design process is for the legal expert to select one or more particular forms of legal analysis in some area of law.

102. *Id.* at 3.
103. *Id.* at 22.

For example, one might decide that the AI system should analyze whether a valid contract has been formed under Virginia law. It may be acceptable to select several distinct but related forms of legal analysis if there is sufficient overlap between them.

Next, the legal expert develops an outline of the steps that must be performed in this legal analysis, including the details and permutations possible for each step. For example, in analyzing whether a valid contract has been formed, each of the requisite elements of contract formation must be established. The inquiry for a single element can be complex so the details of each must be outlined. For example, in determining whether conduct and words would convey to a reasonable person an intent to be bound, the expert would thoroughly review applicable case law to understand and outline exactly which kinds of conduct and words do or do not convey an intent to be bound.[104] This case law review would reveal many specific examples that the expert would use in the steps that follow.

The next step in the proposed process is to identify the information that is possibly, though not necessarily, useful to the legal analysis. Continuing with the contract example, information such as whether there is a prior arrangement to be bound by subsequently-passed rules,[105] or whether the agreement had been read by all parties,[106] can be relevant to whether an intent to be bound has been conveyed.

Extracting information from text can be deceptively difficult depending on the complexity of the information and how such information might be represented in text form. For example, a single type of information might be capable of being written using very different words and phrases. Where the information of interest exists in legal instruments such as contracts, wills, or patents that are drafted by an attorney, the experienced attorney can offer significant insights. Knowledge of the legal drafting process, and in particular how the drafter can use different patterns of text to represent a particular concept, can help in the design of NLP software that is better able to identify and extract the desired information from text. It is at this point that the legal expert must collaborate

104. *See, e.g.*, Lucy v. Zehmer, 84 S.E.2d 516, 522 (Va. 1954); Timothy S. Hall, *Magic and Contract: The Role of Intent*, 12 TEX. WESLEYAN L. REV. 464, 466 (2005) (describing *Lucy v. Zehmer* as demonstrating the "elementary principle of contracts that the relevant intent is the objective, expressed intent of the actor").

105. Falls Church v. Protestant Episcopal Church, 740 S.E.2d 530, 540-41 (Va. 2013).

106. Woodson v. Gilmer, 137 S.E.2d 891, 893 (Va. 1964).

with NLP experts to help understand and then define exactly what patterns of text can express the desired information.[107]

The final step of the proposed process is to define the preferred manner to aggregate and convey the information identified by the NLP software. For example, for the legal analysis of contract formation the software could aggregate, for each element of formation, the pieces of information for and against a conclusion that the element is satisfied. Such aggregation can also entail drawing simple inferences from the information. For legal analysis that is performed en masse over several fact patterns, such as in the analysis of hundreds of contracts, it can be advantageous to calculate scores for the legal analysis applied to a particular document. For example, by counting all evidence in favor of and against some conclusion (e.g., the contract lacks consideration) and noting for each contract whether the evidence strongly or weakly favors the conclusion or its negation, the software could provide a score for each contract or other document. The software could also quickly identify a subset of the contracts that are more likely, or less likely, to meet that conclusion based on their scores. This en masse scoring allows the decision maker's attention to be quickly focused on documents that are likely to meet their desired criteria, e.g., contracts that are very likely or very unlikely to lack consideration.

According to a common recommendation in innovation processes such as design thinking[108] and agile development,[109] iteratively developing the AI system can help to produce a higher quality design in a shorter amount of time. One should feel free to revisit the earlier stages of the process, such as identifying information, once insights are produced in later stages, such as defining how to aggregate information that is collected.

A noteworthy aspect of this process is that legal analysis is used to design the AI software, and therefore the entire process is imbued with significant domain knowledge. Domain knowledge can improve the performance of AI software.[110] Machine Learning systems often do not incorporate domain knowledge, instead being

107. *Cf.* David Lehr & Paul Ohm, *Playing with the Data: What Legal Scholars Should Learn About Machine Learning*, 51 U.C. DAVIS L. REV. 653, 717 (2017) (stating that collaboration between lawyers and technologists will be key for tackling some of the most intractable problems at the juncture of law and Machine Learning).

108. *See generally* Tim Brown, *Design Thinking*, HARV. BUS. REV., June 2008, at 84.

109. *See generally* Daniel E. Schoeni, *Long on Rhetoric, Short on Results: Agile Methods and Cyber Acquisitions in the Department of Defense*, 31 SANTA CLARA HIGH TECH. L.J. 385 (2015).

110. *See, e.g.*, Vincent Aleven, *Using Background Knowledge in Case-Based Legal Reasoning: A Computational Model and an Intelligent Learning Environment*, 150 ARTIFICIAL INTELLIGENCE 183 (2003) (arguing that it is necessary to represent and apply middle-level

driven purely by analysis of the raw data devoid of context.[111] This may be because Machine Learning practitioners can have difficulty understanding or applying domain knowledge.[112]

Note also that the proposed design process is somewhat contrary to the typical Machine Learning paradigm. The purely data-driven Machine Learning system learns all relevant patterns from being exposed to numerous examples, rather than by being told what kinds of patterns are useful or interesting.[113] In contrast, the proposed process delegates this cognitive task of identifying important data to the legal expert. Although this imposes a burden on the human, software would not be able to learn these patterns with sufficient accuracy.

The identification of information that is potentially relevant to the legal analysis has another benefit beyond bringing important information to the attention of the user of the NLP software. This information could be useful in future efforts to train a Machine Learning system to perform more advanced legal analysis with less human input than described in this article. Note that in the process proposed above, potentially-relevant information must first be identified. Moreover, when the NLP software proposed above is used by lawyers, the lawyers' evaluations and legal conclusions based on this information could eventually be captured. Such evaluations and conclusions could then form a set of training data of inputs (relevant information) and outputs (legal conclusions based on this information). This training data could be used to teach a future Machine Learning system to learn how to draw the same sorts of conclusions from relevant information.

C. *An Example of the Proposed Process: AI for Patent Indefiniteness*

In this section, I provide a brief overview of the legal standard for patent claim indefiniteness. The purpose of this exposition is to understand not only the analysis that the legal decision maker undertakes but also the types of information that are relevant to the

---

normative background knowledge in order to address case-based argumentation). The proposal offered in this article is similar to Aleven's in identifying the main issues raised by a problem. *Id.* at 193-94. However, Aleven's system, to my knowledge, does not incorporate the NLP systems for extracting information from text as proposed here.

111. Ting Yu et al., *Incorporating Prior Domain Knowledge into Inductive Machine Learning*, 73 NEUROCOMPUTING 2614, 2614 (2010).

112. *Id.*

113. Surden, *supra* note 2, at 90-93.

analysis. Both will be used in designing AI software to partially automate the indefiniteness analysis in an example that follows.

The definiteness requirement is specified in 35 U.S.C. § 112(b), which requires that the "specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter" of the invention.[114] The United States Supreme Court recently clarified this statutory standard by ruling that "a patent is invalid for indefiniteness if its claims, read in light of the specification delineating the patent, and the prosecution history, fail to inform, *with reasonable certainty*, those skilled in the art about the scope of the invention."[115] The "reasonable certainty" standard balances two interests. On the one hand, patent claims should provide the public with "clear notice" of the exclusionary rights provided by the patent.[116] Distinct claims "guard against unreasonable advantages to the patentee and disadvantages to others arising from uncertainty as to their [respective] rights."[117] On the other hand, "the definiteness requirement must take into account the inherent limitations of language."[118] Accepting some uncertainty is the "price of ensuring the appropriate incentives for innovation."[119]

Claim definiteness is a question of law that the courts review without deference.[120] This flows from a court's obligation to construe claims de novo.[121] Nevertheless, the definiteness inquiry depends on "the understanding of a skilled artisan at the time of the patent application, not that of a court viewing matters *post hoc*."[122] Thus, the level of skill of the person having ordinary skill in the art is relevant to definiteness.[123]

---

114. 35 U.S.C. § 112(b).

115. Nautilus, Inc. v. Biosig Instruments, Inc., 572 U.S. 898, 901 (2014) (emphasis added).

116. *Id.* at 909-10. Clear notice is necessary to avoid "[a] zone of uncertainty which enterprise and experimentation may enter only at the risk of infringement claims." *Id.* (quoting United Carbon Co. v. Binney & Smith Co., 317 U.S. 228, 236 (1942)).

117. General Elec. Co. v. Wabash Appliance Corp., 304 U.S. 364, 369 (1938) (citing Brooks v. Fiske, 56 U.S. 212, 215 (1853)).

118. *Nautilus, Inc.*, 572 U.S. at 909 (citing Festo Corp. v. Shoketsu Kinzoku Kogyo Kabushiki Co., 535 U.S. 722, 731 (2002)).

119. *Id.* (quoting *Festo Corp.*, 535 U.S. at 732).

120. Star Sci., Inc. v. R.J. Reynolds Tobacco Co., 655 F.3d 1364, 1372-73 (Fed. Cir. 2011).

121. Kinetic Concepts, Inc. v. Blue Sky Med. Grp., Inc., 554 F.3d 1010, 1022 (Fed. Cir. 2009).

122. *Nautilus, Inc.*, 572 U.S. at 911.

123. AllVoice Computing PLC v. Nuance Commc'ns, Inc., 504 F.3d 1236, 1240 (Fed. Cir. 2007) (citing Miles Labs., Inc. v. Shandon, Inc., 997 F.2d 870, 875 (Fed. Cir. 1993)).

One commentator has suggested that it can be helpful to consider two distinct types of definiteness, *linguistic* and *physical*.[124] Claims that can be construed in more than one way by the person of ordinary skill are linguistically indefinite, while claims whose single meaning does not sufficiently delineate a necessary relationship among claim elements are physically indefinite. For example, claims "with comparative terms or ambiguous spatial relationships between claim elements fail to meet the physical-definiteness requirement."[125]

The review of the legal standard for indefiniteness highlights several types of information, each of which can be relevant depending on the facts in the case at hand. Listed below are several types of information which can be automatically extracted from patent text. Accompanying each is a brief description of the potential relevance of the information to the indefiniteness inquiry. For simplicity, I present only three types of information, each of which involves only features related to claim terms. Many additional types of information are relevant to indefiniteness, some possessing a very different character than the types listed below.[126]

1. Claim term is not defined or not used in the specification.

The definiteness of a claim depends on whether the terms used in the claim have ascertainable meanings, so an inspection of claim terms is useful to the indefiniteness analysis.[127] The mere presence or absence of a definition for a claim term is information potentially useful to the indefiniteness analysis. If a claim term is not defined in the specification, then this suggests that the claim is (at least somewhat) less likely to be definite; the patent's specification might not provide the person of ordinary skill with enough information to

124. Gary M. Fox, Student Note, *Understanding Nautilus's Reasonable-Certainty Standard: Requirements for Linguistic and Physical Definiteness of Patent Claims*, 116 MICH. L. REV. 329, 347-48 (2017).

125. *Id.* at 342.

126. The Center for AI and Patent Analysis (CAPA) at Carnegie Mellon University conducts original research on new classes of AI tools for various users of the patent system. Several projects involve NLP systems for partially-automated patent analysis, including a more extensive version of the patent indefiniteness project described here.

127. Cox Commc'ns, Inc. v. Sprint Commc'n Co., 838 F.3d 1224, 1232 (Fed. Cir. 2016) ("[T]he common practice of training questions of indefiniteness on individual claim terms is a helpful tool. Indeed, if a person of ordinary skill in the art cannot discern the scope of a claim with reasonable certainty, it may be because one or several claim terms cannot be reliably construed.").

understand the meaning of the term.[128] Nevertheless, if no definition is provided, mere usage of the term in the patent can be sufficiently informative "if the meaning of the term is fairly inferable from the patent."[129] What is most relevant is whether the claim term is well understood by one of ordinary skill in the art and thus would not need any explanation or clarification in the patent.[130]

2. *Claim term is coined.*

The patent drafter is permitted to use claim terms of her own devising.[131] That is, the patent drafter may have invented a new term rather than used a term known in the literature of the relevant technical field. Such terms need not have ever appeared in any previous publication or patent. If the term has never appeared in any previous publication or patent, then it is possible that the person of ordinary skill would not ascribe a definite meaning to the term.[132] If so, it is incumbent on the patent drafter to define the custom term,[133] or risk the claim being considered indefinite.[134]

3. *Claim term is potentially vague.*

Terms denoting unspecified limits, including terms of degree and inherently vague adjectives, can be problematic. The inclusion of such words increases the likelihood that the claim does not have the requisite amount of certainty to satisfy the definiteness requirement. In patent claims, this can occur with the use of the modifier "substantially." For example, the claim may include a term of de-

128. Bancorp Servs., L.L.C. v. Hartford Life Ins. Co., 359 F.3d 1367, 1373 (Fed. Cir. 2004) (explaining that although claim terms need not necessarily be defined in the patent, a definition avoids a "time-consuming and difficult inquiry into indefiniteness").

129. *Id.*

130. Verve, LLC v. Crane Cams, Inc., 311 F.3d 1116, 1119-20 (Fed. Cir. 2002).

131. *See, e.g.*, Vitronics Corp. v. Conceptronic, Inc., 90 F.3d 1576, 1582 (Fed. Cir. 1996) (noting that a patentee may choose to be his own lexicographer).

132. *See* Advanced Ground Info. Sys., Inc. v. Life360, Inc., 830 F.3d 1341, 1348-49 (Fed. Cir. 2016) (holding that claim term "symbol generator" was not a term of art and was indefinite).

133. *Cf. Vitronics Corp.*, 90 F.3d at 1582 ("[A] patentee may choose to be his own lexicographer and use terms in a manner other than their ordinary meaning, as long as the special definition of the term is clearly stated in the patent specification or file history.").

134. Capital Sec. Sys., Inc. v. NCR Corp., 725 Fed. App'x 952, 959 (Fed. Cir. 2018) (affirming district court's holding of indefiniteness because the claim term "transactional operator" "has no commonly-accepted definition and its scope is unclear in view of the intrinsic evidence").

gree, such as a distance between components that must be "substantially equal to"[135] some amount, a balloon that must be "substantially filled,"[136] or a chemical that does not "interfere substantially"[137] with some capability. To avoid indefiniteness, there must be "some standard for measuring that degree,"[138] either in the patent itself or from the knowledge of a person of ordinary skill in the art.[139] If the claim provides "enough certainty to one of skill in the art when read in the context of the invention," then the claim is not indefinite.[140]

Other words besides modifiers introduce a potentially-indefinite term of degree into the claim. For example, adjectives such as "fragile" can be ambiguous as to the requisite degree of the fragility of the gel, thus rendering the term indefinite.[141] Similarly, the claim term "at least partially soluble in water" has been held to be improperly vague.[142]

However, definiteness does not require that the claim provide mathematical precision.[143] Terms of degree without numerical limits can nevertheless be considered definite, particularly if the relevant field of technology admits no more precise way of specifying the invention.[144] The key issue is whether the specification provides some standard for measuring that degree.[145]

In summary, it would be useful for AI software to assist in the indefiniteness analysis by identifying in the patent (and possibly in other patents as well): (1) whether the terms in the claims are defined or used in the patent, (2) whether the claim term appears to be coined rather than in common usage, and (3) whether any claim

---

135. Seattle Box Co. v. Indus. Crating & Packing, Inc., 731 F.2d 818, 821 (Fed. Cir. 1984).

136. Tinnus Enters., v. Telebrands Corp., 846 F.3d 1190, 1206 (Fed. Cir. 2017).

137. Enzo Biochem, Inc. v. Applera Corp., 599 F.3d 1325, 1329 (Fed. Cir. 2010). Though the term "not interfering substantially" does not provide a precise numerical measurement, the intrinsic evidence provided "a general guideline and examples sufficient to enable a person of ordinary skill in the art to determine [the scope of the claims]." *Id.* at 1335 (quoting *In re* Marosi, 710 F.2d 799, 803 (Fed. Cir. 1983)).

138. *Id.* at 1332 (quoting *Seattle Box Co.*, 731 F.2d at 826).

139. Verve, LLC v. Crane Cams, Inc., 311 F.3d 1116, 1119-20 (Fed. Cir. 2002).

140. Biosig Instruments, Inc. v. Nautilus, Inc., 783 F.3d 1374, 1378 (Fed. Cir. 2015) (quoting Interval Licensing LLC v. AOL, Inc., 766 F.3d 1364, 1370 (Fed. Cir. 2014)).

141. Halliburton Energy Servs., Inc. v. M-I LLC, 514 F.3d 1244, 1256 (Fed. Cir. 2008).

142. Standard Oil Co. v. Am. Cyanamid Co., 774 F.2d 448, 453 (Fed. Cir. 1985).

143. Invitrogen Corp. v. Biocrest Mfg., L.P., 424 F.3d 1374, 1384 (Fed. Cir. 2005); Sonix Tech. Co. v. Publ'ns Int'l, Ltd., 844 F.3d 1370, 1377 (Fed. Cir. 2017) ("Because language is limited," terms of degree are not inherently indefinite.).

144. Rosemount, Inc. v. Beckman Instruments, Inc., 727 F.2d 1540, 1547 (Fed. Cir. 1984) (affirming a district court's holding that the term "close proximity" is as precise as the subject matter permits).

145. Datamize, LLC v. Plumtree Software, Inc., 417 F.3d 1342, 1351 (Fed. Cir. 2005).

terms are inherently vague words. Further, aggregating this information, such as for each claim or each claim term, would help the lawyer quickly assess the totality of the evidence presented by the software. One way to aggregate this information would be to simply present all information in a list for the lawyer's review. Another way to aggregate this information would be to develop a simple score, such as counting the percentage of claim terms that lack a definition or counting the number of vague terms in the claims. This type of simple score would allow the lawyer to rapidly assess a large number of patents and would focus the lawyer's attention on the patents most likely to merit further review.

Conclusion

A review of contemporary NLP software reveals both impressive capabilities as well as serious shortcomings. The inability of NLP software to perform robust commonsense reasoning and utilize shared background knowledge prevents many types of legal analysis from being fully automated. Nevertheless, an appropriate division of labor between the lawyer and the computer enables a new class of partially-automated legal analysis. NLP tools designed with the aid of the legal expert can exploit concrete knowledge of case law, allowing software to identify in legal texts different types of information used in conducting specific forms of legal analysis. This type of NLP tool would be tailored to a narrow field of law but would thereby leverage profound expertise in that field to more accurately identify, aggregate, and display relevant information to the legal decision maker. Though such NLP software would not perform all of the steps in the desired legal analysis, the software would allow the user to search for and utilize necessary information faster than if the user worked unaided.

This paradigm is especially useful in the field of patent law because the patent document contains in text form much of the information necessary to perform different types of legal analysis. Moreover, knowledge of the drafting techniques of patent attorneys can be employed to better understand how different types of relevant information can be expressed in text form. This in turn leads to more effective NLP techniques to extract that information. It is hoped that this process for designing NLP software will facilitate greater exposure to NLP software by those in the legal field and foster more collaboration between the Machine Learning and legal communities.