Received: November 18, 2020, Date of acceptance: December 17, 2020, Date of publication: January 14, 2021, DOI: http://doi.org/10.26826/law-in-context.v37i1.125

A Law School Course in Applied Legal Analytics and AI

By **Jaromir Savelka,**[1] Adjunct Professor of Law at the University of Pittsburgh and a Postdoctoral Fellow in the School of Computer Science, Carnegie Mellon University, **Orcid**: http://orcid.org/0000-0002-3674-5456

Matthias Grabmair,[2] Assistant Professor in the Department of Informatics at the Technical University of Munich, **Orcid**: http://orcid.org/0000-0001-6586-2486

Kevin D. Ashley,[3] Professor of Law and Intelligent Systems at the University of Pittsburgh and a Fellow of the American Association of Artificial Intelligence, **Orcid**: https://orcid.org/0000-0002-5535-0759

[1] Carnegie Mellon University, USA

[2] Technical University of Munich, Germany

[3] University of Pittsburgh, USA

ABSTRACT

Technological advances in artificial intelligence (AI) are affecting the legal profession. Machine learning (ML) and natural language processing (NLP) enable new legal apps that, to some extent, can analyze contracts, answer legal questions, or predict the outcome of a case or issue. While it is hard to predict the extent to which these techniques will change law practice, two things are certain: legal professionals will need to understand the new text analysis techniques and how to use and evaluate them, and law faculties face the question of how to teach law students the required skills and knowledge to do so.

At the University of Pittsburgh School of Law, the authors have co-designed a semester-long course entitled, Applied Legal Data Analytics and AI, and twice taught it to combined groups of law students and students from technical departments. The course provides a hands-on practical introduction to applying ML and NLP to extract information from legal text data. It demonstrates applications of text analytics that support the work of legal professionals, researchers, and administrators and techniques for evaluating how well the new tools work. The course culminated in joint projects engaging small teams of law and technical students in applying machine learning and data analytics to legal problems. This article introduces the new text analytic techniques and briefly surveys law schools' current efforts to incorporate instruction on computer programming and machine learning in legal education. It then provides an overview of the course and explains how we taught law students skills of programming and experimental design that prepared them for the final course projects.

Keywords – *artificial intelligence and law, legal text analytics, machine learning, natural language processing, legal technology education*

Acknowledgments. *The authors gratefully acknowledge the University of Pittsburgh School of Law for offering this course and supporting the preparation of this article. We thank the reviewers and editors of Law in Context for their very valuable comments and suggestions*

Disclosure statement – *No potential conflict of interest was reported by the author.*

Suggested citation: Savelka, J., Grabmair, M. and Ashley, K., "A Law School Course in Applied Legal Analytics and AI" , *Law in Context*, 37 (1): 1-41, DOI: http://doi.org/10.26826/law-in-context.v37i1.125

Summary

1. INTRODUCTION

Technological advances in artificial intelligence (AI) are affecting the legal profession. Machine learning (ML) and natural language processing (NLP) enable new legal apps that, to some extent, can analyze contracts, answer legal questions, or predict the outcome of a case or issue. While it is hard to predict the extent to which these techniques will change law practice, two things are certain: legal professionals will need to understand the new text analysis techniques and how to use and evaluate them, and law faculties face the question of how to teach law students the required skills and knowledge to do so.

At the University of Pittsburgh School of Law, the authors[1] have co-designed a semester-long course entitled, Applied Legal Data Analytics and AI, and twice taught it to combined groups of law students and students from technical departments. The course provides a hands-on practical introduction to applying ML and NLP to extract information from legal text data. It demonstrates applications of text analytics that support the work of legal professionals, researchers, and administrators, and techniques for evaluating how well the new tools work.

This article introduces the new text analytic techniques and briefly surveys law schools' current efforts to incorporate instruction on computer programming and machine learning in legal education. Then it describes the 2020 version of the course, including the students, instructors, and course sessions in overview. We explain how we taught law students skills of programming and experimental design and engaged them in assignments that involve using Python programming environments

[1] Matthias Grabmair and Kevin Ashley taught the course in spring 2019. The course was jointly offered at Carnegie Mellon University (CMU) and the University of Pittsburgh (Pitt). Jaromir Savelka and Ashley taught the course in spring 2020, when the course was offered solely at Pitt.

to analyze legal data. The course culminated in joint projects engaging small teams of law and technical students in applying machine learning and data analytics to legal problems. The article explains how the instructors prepared the students for the final course projects, beginning early in the term with project ideas and databases of text, forming teams, working on the projects as a team and obtaining interim feedback, and finally completing the projects and reporting results. We draw some salient comparisons between the 2019 and 2020 versions of the course and report what worked well and what did not, the students' reactions, and lessons learned for future offerings of the course.

1.1 CHANGING THE FOCUS OF A COURSE ON AI AND LAW

Artificial intelligence is a subarea of computer science in which researchers build computational models of intelligent behavior. In the field of artificial intelligence and law (AI and Law), researchers have been building computational models of legal-reasoning behaviors for decades. While they have made great progress, for example, in modeling case-based arguments that account for analogical reasoning and underlying legal values, the work has remained largely academic.

With the advent of legal text analytics and citation network analysis, however, AI and Law techniques increasingly affect the legal profession. Legal text analytics are computational techniques that apply natural language processing, machine learning, and other methods to automatically extract information (e.g., summary statistics, keywords, names of entities, citations, or topics) from text archives of legal case decisions, contracts, or statutes (Simon et al. 2018, p. 253; Ashley 2019). *Machine learning* refers to computer programs that induce or learn models from data (given a set of assumptions about data representation and modeling), typically using statistical means, with which they can, for example, classify a document or predict an outcome for a new case. (Kohavi and Provost 1998; Bishop et al. 2006). General legal analytics apply ML methods to other kinds of data in the legal domain, such as bail recidivism data, to support decision-making. Another source of information comes from citation networks, graphs of the relations among legal cases or statutory provisions, that can be created automatically based on citation information in legal texts such as cases or patents. (Zhang and Koppaka 2007).

As the focus of research and development in AI and Law has changed, so should the focus of a course for law students like this one. Our course differs from a more traditional one on AI and Law in that it introduces students to the new techniques for processing legal texts that are now being applied in legal practice tools. Although it covers some prior work on computational models of legal reasoning with rules and cases, it focuses on how text analysis tools can extract legal information from the texts of cases and statutes, which may enable automatically acquiring the legal knowledge with which those models can reason.

As the learning outcomes, we intend that students understand the new AI and Law techniques for representing legal knowledge and argumentation, practice basic techniques for applying machine learning to legal data including case texts, learn to develop and assess research hypotheses in legal data and text analytics, and participate in designing, planning, and evaluating a legal data analytics project. In addition, the course introduced students to some topical legal issues arising in the context of AI, Big Data, and machine learning technology, including effects of electronic datamining on privacy, intellectual property rights in data, and bias in ML-based decision-making.

1.2 NEW LEGAL APPS CHANGE LEGAL PRACTICE

The new techniques in AI and Law have enabled new tools that promise (or threaten?) to upend traditional legal practice. Based on these techniques, new legal apps can, to some extent, analyze contracts, answer legal questions, or predict the outcome of a case or issue.

Programs like Ravn,[2] Kira,[3] and LawGeex[4] apply text analytics to contracts, identifying types of provisions, semi-automating the review of contracts for routine approval, highlighting issues, and referring apparently

[2] RAVN, imanage.com/product/ravn/ Accessed 6/7/2020

[3] KIRA, www.kirasystems.com/ Accessed 6/7/2020

[4] LAWGEEX, https://www.lawgeex.com/resources/aivslawyer/ Accessed 6/7/2020

unusual provisions for human review. Given a virtual data room full of contracts for review, they can generate a spreadsheet cataloguing provisions by types, such as non-disclosure agreement or disclaimer of warranties, greatly improving the efficiency of due diligence searches.

When answering a legal question, the Ross system,[5] based on IBM Watson (Ferrucci et al. 2010), searches a large collection of texts to locate sentences or short excerpts from cases and other documents that appear to answer a user's question. For example, upon inputting a question in plain English such as, "In New York, what is secondary liability with respect to copyright infringement and how is it established?" it responds with its top-ranked sentence from a federal district decision from the Southern District of New York to the effect that ". . . A party is liable for contributory infringement if, 'with knowledge of the infringing activity,' it 'induces, causes, or materially contributes to the infringing conduct of another.'. . . .", including citation, suggested readings, and updates.

Lex Machina,[6] now part of LexisNexis, predicts outcomes of new cases based on information about litigation participants and their behavior gleaned from a large repository of past cases (Surdeanu et al. 2011). Another program uses machine learning to predict the outcomes of decisions of the European Court of Human Rights based on the cases' textual descriptions of case facts (Medvedeva 2020). Ravel,[7] also acquired by LexisNexis, employs citation networks to present visual maps of citations of U.S. cases regarding a legal concept input by a user. If the user inputs "campaign finance", for instance, Ravel outputs a citation map of cases cited by or citing the U.S. Supreme Court's 2010 *Citizens United* decision, which permitted corporations to make independent political expenditures. In addition, it offers a "Judge Analytics dashboard" that highlights cases a judge has cited in the past and with which the judge may be more familiar, thus improving the chances of a favorable ruling. (Crichton 2015)

Some new legal apps are using text analytics and citation networks to obtain more information from the citation links between citing and cited cases with which to infer why a case is being cited. They may identify the topic of the paragraph where the citation appears or that of the cited case. For example, the "How Cited" tool in Google Scholar Cases,[8] employs such topic information to group into equivalence classes cases that cite a particular case for the same reason. CaseText's CARA A.I. system[9] is another such app. When users input a written legal memo, CARA A.I. suggests additional cases to cite in support of arguments in the memo based on text analytics and citation networks. The CARA A.I. Compose tool generates first drafts of memoranda supporting particular types of motions, for example, motions to "quash a subpoena, exclude expert testimony, file a motion for protective order or compel discovery or disclosure." (Hudgins 2020)

Sophisticated legal apps like these encourage the belief that legal practice is changing dramatically, threatening to shrink the profession and the prospects of law students and young attorneys. While this may be true to some extent, the commercial press has fanned expectations for the abilities of the techniques far beyond the reality.[10]

The legal academy should realize – and should teach law students – that, despite these advances, the text analytic applications are subject to significant limitations. If prediction programs do not represent substantive features of a legal dispute, they cannot explain their predictions in terms with which legal professionals would be familiar. Legal question answerers may map answers on to questions, but at present they understand neither the questions nor the answers. Similarly, contract review systems cannot read contracts as human attorneys would. These tools rely on matching, not reading. They cannot reason

[5] M ROSS INTELLIGENCE, https://rossintelligence.com Accessed 6/7/2020. On December 11, 2020, Ross Intelligence announced it was shutting down its business due to a copyright infringement claim brought against it by Thomson Reuters. The claim alleged that Ross copied Westlaw's copyrighted materials for use in machine learning, an interesting example of intellectual property limitations on datamining. Dipshan, R. 2020 "Ross Shuts Down Operations, Citing Financial Burden from Thomson Reuters Lawsuit." https://www.law.com/legaltechnews/author/profile/Rhys-Dipshan/ Accessed 14/12/2020

[6] LEX MACHINA, https://lexmachina.com Accessed 6/7/2020

[7] RAVEL, https://ravellaw.com Accessed 29/7/2019

[8] GOOGLE SCHOLAR, https://scholar.google.com Accessed 6/7/2020

[9] CASETEXT, https://casetext.com Accessed 6/7/2020

[10] For some examples, *see* (Simon et al. 2018, p. 253).

about how small changes in the question would affect the applicability of the answer. As a result, these tools require supervision by knowledgeable attorneys who understand what they can and cannot do and know how to evaluate their results.

These limitations in the current state of the art also raise interesting challenges for research and development. With more research, automated contract review might be able to draw inferences about due diligence across the content of multiple provisions in a single contract or across multiple contracts. It might also integrate other kinds of data available in a virtual data room such as tax returns, leases, and employee information. Also, of interest are the (still) primarily academic research efforts around "argument mining", which apply legal text analytics to identify argument-related information in case corpora.[11]

Rather than focus exclusively on computational models of legal reasoning that have been developed in AI and Law, this course focuses on the legal text and data analytics employed in the new legal apps that are changing legal practice. Increasingly, the legal profession employs – and relies upon – text analytic tools. Since, as soon as they begin practice, many law students will confront the need to employ these tools and to rely on their outputs, they need to understand how these tools work and what their limitations are.[12] In addition, as attorneys who have some knowledge of the technology, they can better participate in working with technical personnel and researchers in addressing the limitations and improving the tools.

1.3 ARTICLE OVERVIEW

Given the intensity of the law school academic community's interest in how to prepare students for legal practice with the new data analytic technologies, this article describes the course, its goals and implementation, and lessons learned. A continuing focus of the article is how we taught law students skills of programming and experimental design and engaged them in assignments that involved using Python programming environments to analyze legal data.

Section 2 of the article briefly surveys law schools' current efforts to incorporate instruction on computer programming, text analysis, and machine learning in legal education. Section 3 then describes the 2020 version of the course at Pitt, including the students, instructors, and the course sessions in overview. Along the way, we explain why we selected the readings and discuss various pedagogical techniques employed, including requiring students to prepare one-page abstracts of selected readings.

Section 4 discusses each of the course's four parts in some detail, beginning with the ways in which the instructors used programming environments to introduce Python, ML, and NLP. The section describes the introduction to AI and Law concepts and research with a focus on computationally modeling case-base legal argument, the more detailed treatment of how to apply text analytics to legal texts including statutes and cases, and discussions of issues of fairness, privacy, and liability affecting AI.

Section 5 explains how the instructors prepare the students for the final course projects, beginning early in the term with project ideas and databases of text, forming teams, working on the projects as a team and obtaining interim feedback, and finally completing the projects and reporting results.

Finally, in Section 6 we draw some salient comparisons between the 2019 and 2020 versions of the course. In Section 7, we report what worked well and what did not, the students' reactions, and lessons learned for future offerings of the course.

2. AI AND LEGAL DATA ANALYTICS IN LAW SCHOOL CURRICULA

A substantial number of law school curricula include courses that address AI. "As of the 2018 Spring Semester,

[11] *See*, for example, the seminal work of Mochales and Moens (2011). Argument mining has since become the focus of a research subcommunity and its ARGMINING workshop series (e.g., https://argmining2020.i3s.unice.fr/).

[12] There may also be professional ethical responsibility. American Bar Association Rules of Professional Conduct Model Rule 1.1 states that "A lawyer shall provide competent representation to a client. Competent representation requires the legal knowledge, skill, thoroughness and preparation reasonably necessary for the representation." Comment 8 to Rule 1.1 states (emphasis added), "To maintain the requisite knowledge and skill, a lawyer shall keep abreast of changes in the law and its practice, *including the benefits and risks associated with relevant technology,….*" https://www.americanbar.org/groups/professional_responsibility/publications/model_rules_of_professional_conduct/rule_1_1_competence/comment_on_rule_1_1/ Accessed 10/12/2020

... roughly 10% of ABA-accredited law schools offer at least one course explicitly concerning artificial intelligence." (Dalton 2019) This reflects a recognition in the legal academic community that, "Artificial intelligence is changing the practice of law." (Savkar 2019)

2.1 WHY TEACH AI IN LAW SCHOOL AND HOW?

There probably is no consensus about why AI should be included in the law school curriculum and how best to include it. "A broad range of ideas are taking root, somewhat experimentally, across the law school community." (Savkar 2019)

One reason cited for offering a course on AI is to help law students understand how the new technologies are changing models for delivering legal services and to enable them to participate in and even design those changes.

> If we can help students understand that technology, and specifically AI, can create a much more streamlined, efficacious means of connecting lawyers to consumers of legal services, and reorient or recalibrate what it means to provide legal services by lawyers, then that's an enormous benefit for us as legal educators in educating our students to the value and capacity of law to provide access to justice. (Dalton 2019, quoting Daniel B. Rodriguez, Dean, Northwestern Pritzker School of Law)

This reason includes preparing law students to be "practice-ready" for the new types of legal work the technology enables. "With the explosion of artificial intelligence and related cutting-edge technologies, law schools face huge opportunities to create graduates who efficiently and confidently rely on technology to better serve their clients and run more efficient practices." (Dalton 2019) An AI course affords an opportunity "to expose students to the same tools practicing attorneys use" (Dalton 2019) or to prepare them to fill the expected "entirely new categories of legal jobs in the future—perhaps legal data analyst or machine learning legal specialist." (Miller 2019, quoting University of Colorado Law Professor Harry Surden) More generally, the goal is said to be to produce lawyers for the 21st century who are "flexible, team-based, technologically-sophisticated, commercially astute, hybrid professionals," (Dalton 2019, quoting Richard Susskind) "able to identify how technology and other innovative methods can be used to deliver legal services better, faster, and cheaper." (Perlman 2017, p. 6)

One way some law schools expose students to these new tools and methods is by embedding examples and lessons in substantive or skills-oriented law school courses. In substantive courses, AI tools may be "woven into traditional classes, such as product liability classes and regulatory classes, administrative classes and the like." (Council 2019, quoting Robert Kantner, Jones Day) Indeed, "some law schools have begun the integration of technology within the law school curriculum by adding a few days of eDiscovery to Civil Procedure, offering courses in the law of technology, such as Cyber Law, or touching upon data and communication security during Legal Profession." (Eicks 2012, p. 5-9)

AI tools and techniques have also been embedded in skills-focused law school courses. Oklahoma University (OU) College of Law "had a transaction law practicum where [they]'re showing them the current smart contract software and contract analytic software out there." (Dalton 2019, quoting OU College of Law Professor Kenton Brice) The fact that these skills include performing legal research provides another natural opening for introducing AI since "every single one of these platforms [such as Westlaw and other research tools] to some degree or another is using a form of AI implementing legal research." (Dalton 2019, quoting OU College of Law Professor Kenton Brice)

AI may also be addressed in courses focusing on business, tech start-ups, and innovation "that train students in the business of law and operations, get them to think like entrepreneurs, and have them improve processes, gather data, and use technology." (Dalton 2019, quoting Daniel Linna, now Director of Law and Technology Initiatives and a Senior Lecturer at the Northwestern Pritzker School of Law) For example, a legal clinic at Cardozo Law aims to take "students through the life cycle of an actual area technology startup, from formation through IPO, and beyond." (Dalton 2019, quoting Professor Aaron Wright)

Law courses incorporating AI may aim to expose students to the engineering side of legal processes and modes of delivering legal services. A goal at Suffolk Law School is to expose students to "concepts like legal project management and process improvement, legal design ..., automated legal document assembly, expert system tools, electronic discovery, and other areas as well." (Perlman 2017, p. 7)

> Once process improvement has begun and continuous improvement is part of an organization's culture, we find many opportunities to exponentially increase efficiency and quality and improve outcomes through document automation, workflow automation, rules-based expert systems, and other readily available, basic technologies. Done properly, getting control of processes also means creating legal-services delivery metrics and capturing data from these processes .... [t]his ... also helps position organizations to identify opportunities for data analytics and data-driven artificial intelligence. (Dalton 2019, quoting Daniel Linna)

At the University of Minnesota, "the coding for lawyers course presents an overview of the changing role of lawyers, the role of lawyers as project managers, the key responsibilities of a project manager, and the organizational and operational structure of a legal tech project." (Contreras and McGrath 2020, p. 323)

Several law school courses are addressing AI in order to help law students to better understand "the emerging legal issues around AI" (Council 2019) and how to regulate AI in society.

> The law schools at Columbia University, Harvard University and Stanford University have within the past two years introduced courses that touch on AI topics including algorithmic bias and civil rights; liability for autonomous vehicle accidents; and regulating AI-powered applications and platforms. The goal is to give future lawyers, judges and regulators greater exposure to the technology, which is expected to become increasingly embedded in society and raise new questions around matters of legal responsibility, due process and regulatory frameworks. (Council 2019)

These societal issues include protection of privacy and data bias. "If students want to work on the cutting edge of things like AI, privacy, blockchain technology et cetera, you ... have to begin to really understand how these things work." (Dalton 2019, quoting Professor Aaron Wright)

Given the increasing reliance on data analytics and machine learning by agencies administering social benefits or criminal justice, the issue of data bias is a particularly important focus. "These predictive tools need to be examined to determine any algorithmic bias and what data was originally provided to make such an assessment." (Reid 2018, p. 481) At Stanford Law School, Professors Daniel Ho and David Engstrom teach a class on federal agencies' use of AI.

> It's hard to engineer a solution for algorithmic bias without understanding how decades of anti-discrimination law have dealt with these challenges, nor is it possible to understand what the future of anti-discrimination law should look like without a fine-grained understanding of the sources of algorithmic bias. (Council 2019, p.2, quoting Professor Daniel Ho, Stanford Law School)

An important focus in some of the courses is thus on teaching students how to assess the extent to which one can appropriately rely on the technology in the practice of law. "Inappropriate reliance on algorithms is certainly a theme we weave into more than one course, and there's a constant reminder that technology is there to serve people; and that when it can result in injustices, we have to be vigilant about how it's used." (O'Grady 2018, quoting Professor Gabriel Tanenbaum)

2.2 TEACHING LAW STUDENTS ABOUT CHALLENGES OF RELYING ON THE NEW TECHNOLOGY

Biased data and algorithms are not the only causes of problems confronting appropriate reliance on the new technologies. In their article on how artificial intelligence will disrupt legal ethics and professional responsibility, Murphy and Pearce (unpublished) point out that today, attorneys are in the unenviable position of relying on new technologies even though they are unable to verify the quality of the services. "[H]ow do lawyers know that they can reasonably rely on the legal search results of these businesses? Lawyers lack the software expertise to understand the algorithms...." (Murphy and Pearce unpublished)

A course on legal data analytics and AI can help law students to understand the assumptions underlying the new machine learning and text-analytic technologies and to learn how to assess the capabilities and limitations of these methods as they are applied generally in society and more specifically in legal practice. This is especially appropriate for applying machine learning to texts where the techniques may seem very counter intuitive. Harry Surden characterizes machine learning as "producing intelligent results without intelligence."

> For example, when a machine learning-based computer system produces a translation, it usually does so using statistical associations. However, such a pattern-based machine learning approach—while often producing decent translations—does not actually involve the computer "understanding" what it is translating or what the words mean in the same way a human translator might. (Miller 2019, quoting Professor Harry Surden)

This is true as well of applications of machine learning to legal documents and texts. According to Miller, Surden cautions that "a limited number of legal tasks may benefit from current machine learning approaches," including tasks in "e-discovery document review, litigation predictive analysis, and legal research." "Core tasks still require a great amount of problem solving and abstract reasoning that pattern recognition or machine learning is unable to replicate." (Miller 2019)[13]

That places lawyers relying on ML technology in an odd position. Increasingly in legal practice, they will need to rely on ML tools that draw inferences from textual data, but those tools have very limited information about what the texts mean. It would help if the ML technology could adequately explain the basis of its predictions and results in terms that lawyers could understand. Lawyers would then be able, at least, to assess the system's reasons for its predictions, to decide if they accept those reasons, and to factor the reasons into a determination of whether to rely on its advice. Most of the AI programs that employ machine learning, however, cannot explain their results in terms lawyers would understand. Most employ neural network architectures for which generating explanations is problematic. Much depends on the features that the machine learning considers. Neural networks may discover features that are statistically weighty but that do not correspond to legal concepts lawyers would expect to see in an explanation. Even if the features do make sense, the information about the features' weights is distributed across the network's nodes and difficult to fashion into an explanation.

Compounding the problem of reliance are the extraordinary claims made on behalf of the technology by commercial providers and the press as documented, such as:

> The question is how—and when—machines will enter the legal economy in full force and render lawyers obsolete. (Simon et al. 2018, p. 237)

> There is a lot of legal work that is routine…. But that routine work, sifting through documents for relevant information, is wrapped in language, which had protected lawyers from the effects of automation, but no longer. Indeed, language is no longer a barrier because of advances in `natural language processing' (NLP) technologies, as emphasized in the most recent McKinsey & Company report on technology and employment. (Simon et al. 2018, p. 271)

> Behind the numbers, many private law firm leaders believe that we are on our way to a future where at least the younger lawyers will be replaced by robots. (Simon et al. 2018, p. 285)

The claims are even more remarkable given that the new legal apps cannot read legal text like human lawyers can (Ashley 2019).

2.3 TEACHING LAW STUDENTS HOW TO EVALUATE THE NEW TECHNOLOGY

In such circumstances, assessing whether and how much to rely on inferences based on machine learning is difficult. A course in applied legal data analytics and AI can help law students to understand the extent to which such claims exaggerate the current state of the art.

> Most everyone would agree that the impact of algorithms, computational technologies, and artificial intelligence on everyday life, institutions, and society will only grow, and rapidly. Yet most law students and lawyers lack the foundational knowledge to generally explain how these technologies work, much less assess them. (Linna 2018)

[13] *See* (Surden 2014, p. 101) "there are a subset of legal tasks often performed manually today by attorneys, which are potentially partially automatable given techniques such as machine learning, provided the limitations are understood and accounted for."

It would help students to learn in some detail how the legal inferences are generated and how machine learning can be evaluated. As it appears today,

> There is no doubt that LegalTech will automate "legal work," such as contract drafting, legal risk management, and dispute resolution. If legal work will be dependent on and performed by algorithms in the future, it is crucial for the future lawyers (and lawyers who are at the start of their careers) to get a better understanding of "data analytics" and "artificial intelligence." Since all these near-benefits technologies are code based, the benefits of understanding code will be of value in the medium to long-term. (Fenwick et al. 2018, p. 17).

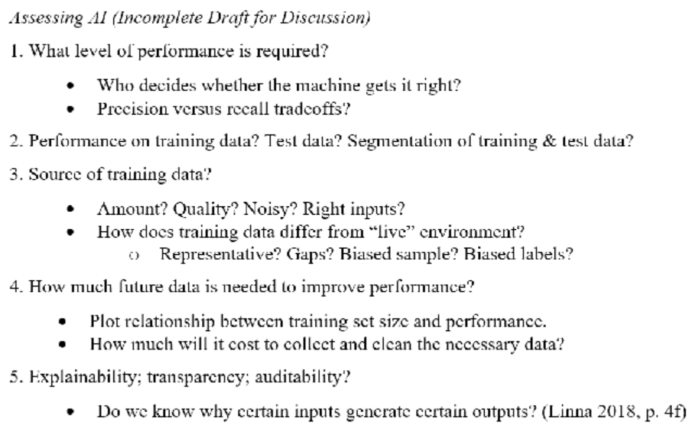
In some of the law school courses focused on AI, the instructors and students are pioneering approaches for assessing text analytic techniques as applied in the legal domain. In Daniel Linna's Artificial Intelligence and Legal Reasoning class, for example, an aim is to develop a framework for assessing artificial intelligence as used for legal-services delivery like that in Figure 1.

As the first question about the required level of performance suggests, students need to understand that the degree and criticality of the reliance depends on the "use case": how the technology is being used and the kinds of problems that it is being used to solve.

The Dutch ethicist Pim Haselager notes a distinction between two types of "use cases": I. AI as a component of a modular system delivering results that assist a human to make a legal decision, and II. AI as the overall legal decision maker or recommender. The former keeps humans on (or in) the loop, while the latter keeps them "under the loop" as Haselager put it, where they are much more likely to be forced to rely blindly and less critically on the AI system. (Haselager 2019) Understanding the implications of these different types of use cases is crucial for establishing how much reliance would be required, and for appropriately designing methods for keeping attorneys in the loop as they apply the new technologies.

2.4 PREPARING LAW STUDENTS TO INTERACT WITH TECHNOLOGISTS

Law school courses in AI help students to engage with students and faculty from other disciplines who focus on the new technologies and their applications, a kind of activity which, it is anticipated, will become increasingly important in legal practice. The courses provide a venue for law students to learn how to communicate with professionals who do not necessarily share their assumptions or methods and who do not understand their legal language.



Assessing AI (Incomplete Draft for Discussion)

1. What level of performance is required?
   - Who decides whether the machine gets it right?
   - Precision versus recall tradeoffs?
2. Performance on training data? Test data? Segmentation of training & test data?
3. Source of training data?
   - Amount? Quality? Noisy? Right inputs?
   - How does training data differ from "live" environment?
     - Representative? Gaps? Biased sample? Biased labels?
4. How much future data is needed to improve performance?
   - Plot relationship between training set size and performance.
   - How much will it cost to collect and clean the necessary data?
5. Explainability; transparency; auditability?
   - Do we know why certain inputs generate certain outputs? (Linna 2018, p. 4f)

FIGURE 1. Assessing AI as per D. Linna's class on AI and Legal Reasoning

AI courses focused on ethical and regulatory issues, for example, engage multidisciplinary students in evaluating the technology. See, e.g., Harvard Law School's course called "The Ethics and Governance of AI," which is part of a joint effort with Massachusetts Institute of Technology (Council 2019, p.2). The course of Stanford law professors Daniel Ho and David Engstrom on federal agencies' use of AI also combines law students and computer science students (Council 2019, p.2).

The multidisciplinary interactions may focus on designing and implementing the new technologies.

> "Legal tech" will profoundly disrupt the legal profession and since these technologies are code-based[,] lawyers need to be able to understand and talk about code to participate in the design of such legal technologies and to maximize their usefulness in supporting all legal work.... [A] combination of lawyers, mathematicians, and coders taught the course, and the students are encouraged in doing their assignment to partner up with students in other faculties to put together their final assignment. In that sense, the kind of multi-disciplinary teams of the future is re-created in the classroom. (Fenwick et al. 2018, p. 29)

> … "Co-creation" involving partnerships between developers and nondevelopers will be crucial to building a "better" digital future. And understanding coding enables lawyers to constructively engage with coders, programmers, and other software developers. (Fenwick et al. 2018, p. 11, 29, 30f)

In a similar vein, a Legal Analytics and Innovation Initiative at Georgia State University School of Law aims to enable "law students … to collaborate closely with computer science and business students … to design complex technologies that solve previously unsolvable legal problems (such as predicting to a high degree of accuracy how a particular judge will rule in cases defined by a large set of parameters)." (Savkar 2019)

While some students may participate in design, perhaps more will be called upon to oversee or supervise the application of the technology in practice, activities which also will involve them in multidisciplinary interactions with clients and staff members. "Students entering the current and future legal employment market must understand the implications and impact of AI and related technologies on the practice of law and be prepared to oversee their implementation and the resulting processes." (Dalton 2019)[14]

> [S]tudents should be taught to question the data produced by an Al program and avoid over-reliance on technology alone…. [O]verly broad tasks asked of Al programs can result in inaccuracies. Such operations need review and significant human oversight. … [J]unior lawyers will now find themselves in supervisory roles as they are asked to lead or assist employees assigned to the IT or Al department… Law schools need to find ways to emphasize such skills in the classroom and encourage group projects and team-oriented case study discussions. (Reid 2018, p. 484)

2.5 HOW MUCH PROGRAMMING TO TEACH LAW STUDENTS?

A still open pedagogical issue concerns how much programming a law student needs to know in order to understand how and how well machine learning and text analytics perform. Some argue that "Law students need to learn programming skills to understand how technologies can optimize their work and make the process of providing services easier." (Pivovarov 2019) This is often characterized as "coding for lawyers" but may be subject to some disclaimers illustrated below.

> [T]he "Coding for Lawyers" course is … not about teaching students "how to code," but about making them realize how important it is to think about our relationship with new technology and technology experts. In this way, students can see the new opportunities that technology creates, but also think about the new issues (practical and ethical) that such technologies create. For this reason, we plan to introduce the basics of machine learning and deep learning to the program. (Fenwick et al. 2018, p. 30)[15]

Beyond teaching coding to help law students understand the practical and ethical issues the new technologies raise, practice with coding may help law students to communicate with technologists.

> [W]e are not suggesting that law students should become professional coders. After all, coding is complicated and to become a serious coder does take time. However, we do believe that the ability to understand and communicate with coders is a necessary skill for the lawyer of the future. As such, we think that law students will benefit … [n]ot only by reading or hearing about it but by participating in and experimenting with coding projects in a classroom environment. (Fenwick et al. 2018, p. 30)

Our answer to how much programming law students need is enough to enable at least some of them to evaluate the new technologies as they are applied in legal practice. Murphy and Pearce (unpublished) warn of the growing need to teach students how to evaluate the new technologies.

> [L]egal services workers, including lawyers, will not need to code but they will need to understand how to apply and evaluate new methodologies powered by artificial intelligence. Pieces of these skills only exist today in marginal parts of the curriculum, whether a class session or course on electronic discovery, or a clinic on developing

[14] See also, (Contreras and McGrath 2020, p. 323)

[15] See also, (Contreras and McGrath 2020, p. 324, "Students are not expected to be fluent coders by the end of the course, but to have an appreciation and understanding of the capabilities of coding.")

ISSN: 1839-4183

> apps for consumers. Once artificial intelligence becomes central to the provision of legal services it will have to become central to legal education. (Murphy and Pearce unpublished)

We agree that law students need to engage closely with computer code in order to understand how the technologies work and how to evaluate them. In the Applied Legal Analytics and AI course, we stepped law students through programming with Python notebooks and organized teams combining law students and technical students to work on final projects. In each of the projects, the students implemented some Python code, designed empirical evaluations, ran the experiments, obtained the results, and then engaged in error analysis to see what the programs missed and why.

3. COURSE DESCRIPTION

As noted, the most recent version of the course in Applied Legal Analytics and AI took place at Pitt from January through April 2020.

3.1 THE STUDENTS

One of the unique features of the course was a mix of students from law backgrounds and technical backgrounds. The course comprised a total of ten students: five Pitt law students, four Pitt graduate students from the Department of Electrical and Computer Engineering, and one Pitt undergraduate student from the School of Computers and Information. A Duquesne University Law School professor with a longtime interest in AI and Law also attended on a regular basis.[16]

3.2 THE INSTRUCTORS

In order to teach this course successfully, the instructors need to combine an ability to: (1) teach students how to create computer programs in Python that apply machine learning and natural language processing to textual and other legal data, (2) help students design experiments evaluating applications of ML and NLP to such data, (3) explain how these experiments relate to research in AI and Law and to legal practice tools, and (4) instruct students about the legal issues that arise in contexts involving AI and machine learning.

It is likely that two instructors with complementary skills and knowledge will be required to cover all four abilities, and it is also helpful for the instructors to be able to share the responsibilities of teaching and guiding students. That was certainly our experience in teaching the course. As noted, Jaromir Savelka[17] and Kevin Ashley[18] co-taught the spring 2020 course; Matthias Grabmair[19] and Ashley co-taught in spring 2019. Their combined skills and knowledge covered the course's wide-ranging demands. As we learned by experience, it was also crucial

[16] The previous version of the course was offered at Pitt and CMU from January through April 2019. That version of the course comprised a total of fifteen students: eight Pitt law students, three Pitt graduate students from the School of Computing and Information, one Pitt graduate student from the Katz Graduate School of Business and three CMU undergraduate students.

[17] Jaromir Savelka is an Adjunct Professor of Law at the University of Pittsburgh and Postdoctoral Fellow in the School of Computer Science at Carnegie Mellon University. For four years as a Lecturer at the School of Law of Masaryk University in Brno, the Czech Republic, he taught Legal Informatics and related legal technology courses at the graduate and undergraduate levels. In 2013 he enrolled as a graduate student in the University of Pittsburgh Intelligent Systems Program where he developed expertise in computer programming and applying machine learning techniques to legal texts. In April 2020, he successfully defended his dissertation, entitled "Discovering Sentences for Argumentation about the Meaning of Statutory and Regulatory Terms." For the last three years, he has worked as a Data Scientist at Reed Smith, LLP, Pittsburgh, where he gained experience in evaluating and developing legal AI systems supporting eDiscovery and due diligence. He has a graduate degree in law and an undergraduate degree in computer science, both from Masaryk University.

[18] Kevin Ashley, a Professor of Law and Intelligent Systems and senior scientist at the Learning Research and Development Center at the University of Pittsburgh, has expertise in artificial intelligence and law and is a Fellow of the American Association of Artificial Intelligence. He is co-editor in chief of *Artificial Intelligence and Law*, the journal of record in the field of AI and Law and the author of *Artificial Intelligence and Legal Analytics: New Tools for Law Practice in the Digital Age* (2017). He has a JD degree from Harvard Law School and a PhD degree in Computer and Information Science from the University of Massachusetts. For five years before beginning his graduate studies in computer science, he practiced law as an associate at White & Case in New York City. His research has focused on developing computational models of case-based legal argument and ways to automatically analyze texts of legal decisions to populate those models.

[19] Matthias Grabmair is an Assistant Professor in the Technical University of Munich Department of Informatics. At the time of the spring 2019 version of the course, he was a Systems Scientist at Carnegie Mellon University's Language Technologies Institute teaching in CMU's Master's Program in Computational Data Science. He worked on solving problems in domain-specific question answering and knowledge engineering, including analysis of legal documents. His expertise reflects studies and experience in artificial intelligence and law, knowledge representation and reasoning, computer programming, natural language processing, applied machine learning, information retrieval and computational models of argument. He holds a PhD degree from the University of Pittsburgh's Intelligent Systems Program. He has an LLM degree from the University of Pittsburgh and a Dipl.-Jur. degree from the University of Augsburg, Germany.

for two instructors to be able to share the responsibilities of teaching students, providing feedback, and guiding student teams working on course projects. For additional help, guest speakers from law firms or legal service providers lectured the class on applications in legal practice such as automating contractual due diligence or patent review. It also makes sense to invite more traditional legal scholars to address such topics as the use of machine learning in empirical legal research or legal issues affecting machine learning such as intellectual property rights in data, effects of electronic datamining on privacy, bias in ML-based decision-making, and product liability of autonomous systems.

3.3 COURSE OVERVIEW

In overview, the course content comprised three major topics: (1) Text Analytics: introducing programming with Python, machine learning and natural language, and focusing ultimately on using text analytic techniques to extract argument-related information from case texts and to predict case outcomes. (2) AI and Law: introducing computational models of legal decision-making, arguing with rules and cases, and predicting case outcomes and how these models could interact with text analytics to improve legal information retrieval. (3) Legal Practice and AI: introducing students to legal text analytic tools that have been applied in legal practice and to substantive legal domain topics that affect AI such as data privacy, machine learning bias and fairness, and liability of autonomous vehicles.

Designing a syllabus for conveying this content presented at least two challenges: selecting appropriate readings and deciding on the ordering and duration of each topic.

Since no textbook appeared to cover all of the targeted content, we settled on the following plan. For the first area, Text Analytics, we designed workbooks of programming exercises supplemented with accessible research papers to illustrate applications of the techniques to various legal tasks and use cases. Selected chapters from *Artificial Intelligence and Legal Analytics: New Tools for Law Practice in the Digital Age* (Ashley 2017) would cover most of the subtopics in the second area, AI and Law. As to the third area, Legal Practice and AI, we selected law review articles to illustrate practical applications of text analytical tools and the legal ramifications of applying AI in selected areas of the law.

As far as ordering the topics and readings, there were two driving factors. We knew that it would be difficult and time-consuming for law students to learn programming skills. In addition, the students needed to learn such skills in time to apply them in working on their course projects. Hence, teaching programming skills and providing an intuition of how they would be relevant became the initial and extended focus of the course.

As illustrated in Table 1, we organized the lectures and discussion sessions around a progression of themes in four parts.   The readings and lectures associated with each part are discussed in the next section.

In Part I, the goal was to introduce the law students to some selected aspects of programming with Python, and to basic concepts in machine learning and natural language programming. We also sought to convey to the non-law students some intuitions about legal reasoning and provide all of the students with a glimpse of the impact of AI on legal practice.

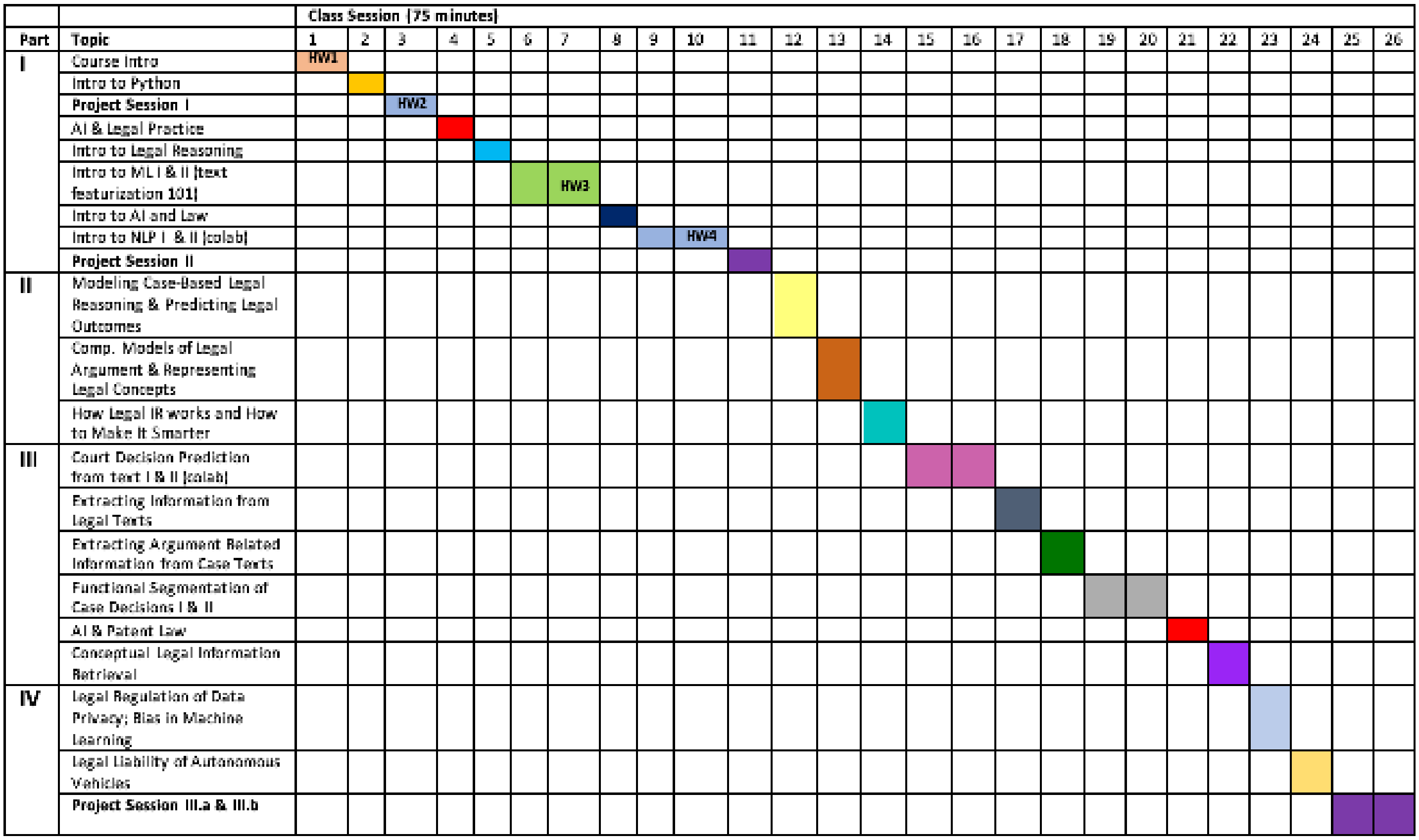
Part II provided a foundation in prior research in the field of Artificial Intelligence and Law on supporting legal decision-making, modeling case-based legal reasoning, and using the models to predict and explain legal outcomes. It described how legal information retrieval works and suggested how to make legal IR more intelligent.

Part III focused on techniques for technology assisted review, extracting information from statutory texts, extracting argument-related and other information from case texts, predicting outcomes directly from case texts, and segmenting case decisions by function.[20]

In Part IV, the last lectures in the course focused on legal domain topics that affect AI including patent law,

[20] The University shutdown due to covid-19 occurred at the end of Part III and continued through Part IV. As a result, classes were cancelled during the week of March 16. Upon resuming the course during the week of March 23, all lectures were delivered via Zoom or Panopto

TABLE 1. Course Sessions in Spring 2020



		Class Session (75 minutes)																									
Part	**Topic**	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
I	Course Intro	HW1																									
	Intro to Python		■																								
	Project Session I			HW2																							
	AI & Legal Practice				■																						
	Intro to Legal Reasoning					■																					
	Intro to ML I & II (text featurization 101)						■	HW3																			
	Intro to AI and Law								■																		
	Intro to NLP I & II (colab)									■	HW4																
	Project Session II											■															
II	Modeling Case-Based Legal Reasoning & Predicting Legal Outcomes												■														
	Comp. Models of Legal Argument & Representing Legal Concepts													■													
	How Legal IR works and How to Make It Smarter														■												
III	Court Decision Prediction from text I & II (colab)															■	■										
	Extracting Information from Legal Texts																	■									
	Extracting Argument Related Information from Case Texts																		■								
	Functional Segmentation of Case Decisions I & II																			■	■						
	AI & Patent Law																					■					
	Conceptual Legal Information Retrieval																						■				
IV	Legal Regulation of Data Privacy; Bias in Machine Learning																							■			
	Legal Liability of Autonomous Vehicles																								■		
	Project Session III.a & III.b																									■	■

legal regulation of data privacy, machine learning bias and fairness, and legal liability of autonomous vehicles.[21]

Table 1 also highlights two other aspects of the course: the four homework assignments and four project sessions. The homework assignments all occurred in Part I and served as the primary means for introducing students to a tool for legal question answering, introducing the non-law students to legal reasoning and a bit of legal practice, and engaging the students in working with Python programs. Table 1 shows when the homework was assigned; it was due a week or two later. These homework assignments are discussed in Section 4, Part I.

The second aspect was a semester-long focus on the course projects. As illustrated in Table 1, two project sessions at the beginning and end of Part I focused the students on identifying projects, assembling teams, and presenting preliminary proposals. Two sessions at the end of Part IV were devoted to team presentations of final projects. The course projects are discussed in Section 5.

4. COURSE CONTENT BY PARTS

The course content was presented primarily through weekly lectures and readings. In support of the lectures, in most weeks students were assigned to read one or two topical AI and Law research papers per week.

For selected readings students were asked to submit a brief, one-page abstract of the reading a day or two ahead of the class time. A template specified the form of the abstracts: a three-sentence summary of the reading, three positive aspects of the paper as well as three criticisms (about one sentence each), three questions the student would like to ask the authors because the

[21] The spring 2019 course covered similar topic matter. The primary differences were a more abbreviated treatment of AI and Law research, an extended treatment of the application of machine learning to, and quantitative models of, Supreme Court Justice voting (three sessions), fairness in machine learning, and the disparate impact of ML in bail recidivism prediction and credit scoring (one session), and a guest lecture about automated contract review for due diligence (one session).

student either did not understand something or would like more information, and a brief statement of how the reading relates to the student's research or other interests. The abstracts were due a day or two before the session so that the instructors could include answers and target discussion to address the students' points in the abstracts.

4.1 PART I: INTRODUCING PYTHON, MACHINE LEARNING, NATURAL LANGUAGE PROCESSING AND AI'S EFFECT ON LEGAL PRACTICE

An interdisciplinary course requires introducing students to many novel subjects. In Part I, we introduced them to examples of legal text analytic tools and their effects on legal practice, to some selected aspects of programming with Python, to basic concepts in machine learning (ML) and natural language processing (NLP), and to a brief history of research in Artificial Intelligence and Law.

4.1.1 INTRODUCING LEGAL TEXT ANALYTIC TOOLS AND THEIR EFFECTS ON LEGAL PRACTICE

An initial goal in Part I was to survey the new legal text analytic tools and methods. The readings (Ashley 2019) and lecture illustrated some techniques for applying machine learning to legal texts, using citation network diagrams, and question answering in the apps described in Section 1. Their limitations were also introduced: the fact that apps cannot read like lawyers, the difficulties the new machine learning techniques have in generating explanations, and the constraints imposed by the need for manually annotating training sets for machine learning. These limitations, we suggested, counteract the hype surrounding the impact of AI on the practice of law and its predicted effects on legal employment (Simon et al. 2018).

Two recent decisions concerning e-discovery and predictive coding helped introduce the non-law students to the court system and to such concepts as burden of proof, choice of law, types of motions to dismiss, the role of distinguishing, and the limits of precedential constraint.[22] In *Lola v. Skadden*, for example, the court held somewhat controversially that attorneys do not exercise "legal judgment" when they perform services that a computer could provide while reviewing documents (Simon et al. 2018, p. 253). We looked at a subsequent decision from the same federal district, in which a district court readily distinguished the decision in *Lola v. Skadden* despite the similarity of the fact situations. The *Dynamo* decision illustrated how courts are adapting to the role of machine learning in determining the relevance of documents in e-discovery.

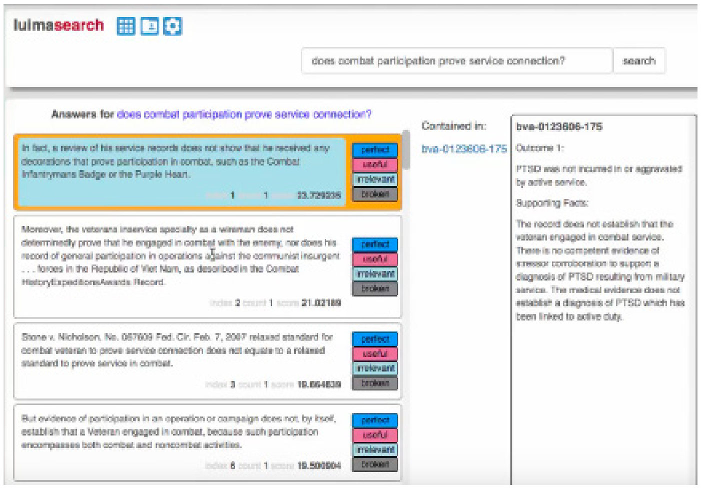
Homework assignment 1 introduced the students to using one of the new legal apps for an exercise in legal research and memo writing and the non-law students to some basics of the legal system. The LUIMA system, a prototype legal question answering system developed by Matthias Grabmair, accepts questions dealing with veterans' claims for compensation for posttraumatic stress disorders (PTSD) and retrieves answers (i.e., extracted sentences similar to the question) from a database of decisions of the Board of Veterans Appeals (BVA).[23]

The students began by reading a brief but informative introduction to the legal system, reading and briefing cases, using statutes, synthesizing cases, and drafting a legal memorandum.[24] The readings introduced basic legal reasoning concepts (e.g., standards of proof, presumptions, and preponderance of the evidence). Using these as a guide, students read a decision of the Board of Veterans Appeals and prepared a brief of the case. They then read a problem scenario and a list of questions to guide them in identifying some of the issues that a legal analysis would need to address.

[22] See *Lola v. Skadden, Arps, Slate, Meagher & Flom LLP,* 620 F. App'x 37, 620 Fed. Appx. 37 (2d Cir., 2015). *Dynamo Holdings et al. vs. Commissioner of Internal Revenue*, U.S. Tax Court, Docket No. 2685-11, 8393-12; July 13, 2016.

[23] The BVA is set in a reasonably limited yet intuitive legal universe. In performing administrative adjudications, it produces court-like opinions that follow typical legal argumentation, writing, and citation patterns. Running the assignment in the experimental LUIMA system rather than a commercial search engine had several advantages. First, it reduced the complexity of the interface and focused students on the research task by not providing headnotes or other forms of topic-based linking across decisions. Second, it made sure that only sentences from BVA decisions were retrieved as answer candidates, thereby preventing exposure to cases that use similar vocabulary but stem from different jurisdictions. Third, the student search queries and behavior in the system could be tracked.

[24] Excerpts from (Brostoff and Sinsheimer 2013).



does combat participation prove service connection?

Answers for does combat participation prove service connection?

In fact, a review of his service records does not show that he received any decorations that prove participation in combat, such as the Combat Infantrymans Badge or the Purple Heart.

index 1 count 1 score 23.729236

Moreover, the veterans inservice specialty as a wireman does not determinedly prove that he engaged in combat with the enemy, nor does his record of general participation in operations against the communist insurgent . . . forces in the Republic of Viet Nam, as described in the Combat HistoryExpeditionsAwards Record.

index 2 count 1 score 21.02189

Stone v. Nicholson, No. 067609 Fed. Cir. Feb. 7, 2007 relaxed standard for combat veteran to prove service connection does not equate to a relaxed standard to prove service in combat.

index 3 count 1 score 19.664639

But evidence of participation in an operation or campaign does not, by itself, establish that a Veteran engaged in combat, because such participation encompasses both combat and noncombat activities.

index 6 count 1 score 19.500904

Contained in:

bva-0123606-175

bva-0123606-175

Outcome 1:

PTSD was not incurred in or aggravated by active service.

Supporting Facts:

The record does not establish that the veteran engaged in combat service. There is no competent evidence of stressor corroboration to support a diagnosis of PTSD resulting from military service. The medical evidence does not establish a diagnosis of PTSD which has been linked to active duty.

FIGURE 2. Legal question answering with luimasearch

The scenario involved a mechanic in an Air Force base who witnessed a fatal crash of a helicopter on which he was scheduled to work. This led to alcohol abuse, a general discharge, morbid flashbacks, and drug abuse. The Veterans Administration (VA) dismissed the veteran's claim for disabilities as not substantiated. He has asked an attorney (the students) to prepare arguments for appeal to the BVA.

In order to prepare arguments for an appeal, the students needed to find out more about the relevant legal standards for proving issues related to PTSD claims and how to meet them. Eight questions at the end of the scenario guided their inquiries with the LUIMA system, such as:

1. What is the overall legal standard for determining if a veteran is entitled to compensation for disability?
2. What is the legal standard for proving that a veteran currently has PTSD?[25]

For each of the questions, the students were asked to find and write a very brief answer and to cite the case source[s] for the answer. Students could submit these questions or variations of them directly to the LUIMA search tool, using the interface shown in Figure 2. The program returned a list of sentences (left) rank-ordered by their responsiveness, a summary of the case in which the sentence appeared (right), and a link to the full case text (center). The students were asked to focus on finding similar prior decisions by navigating to the decisions from which the candidate answers had been extracted, and to identify patterns of arguments with which the BVA decides legal issues like those in the scenario. The students could also mark answers as more relevant to the query or less, a kind of feedback to LUIMA.

Finally, the students were asked to imagine that they worked in a Veterans Legal Clinic. The director has asked them to prepare a legal memorandum assessing legal arguments for an appeal to the BVA of the VA's rejection of the client's PTSD claim. The instructors advised the students to employ the results of their legal research with LUIMA in order to prepare their memos and provided a list of general criteria for assessing legal memoranda.

In this way, the first homework assignment gave students a chance to work with a basic legal question answering system and to experience using the system to perform legal research and generate a legal memorandum. Upon review of the students' memos, the instructor found them to be quite acceptable.

4.1.2 INTRODUCTION TO PYTHON PROGRAMMING

The first two lectures focused on introducing programming with Python. No prior experience with programming was assumed. Our goal was first to make students comfortable reading small pieces of code. Later we asked

[25] The other questions are:

1. What are the special regulations concerning proving service connection for PTSD?
2. What kind of evidence has been accepted to show a veteran has PTSD?
3. What kind of evidence has been accepted to show PTSD was service connected?
4. What if the evidence for and against service connection is evenly balanced?
5. What if the veteran is currently an alcoholic?
6. When is a veteran found to not be credible?

them to edit the code in order to adapt provided examples to their needs. Both lectures were organized as a guided walkthrough of prepared Jupyter notebooks[26] hosted on the Google Colaboratory (Colab) platform.[27] The idea was to provide students with small snippets of executable code with which they could explore coding outside of class, as well. The first lecture covered several foundational concepts including code layout, primitive types, and flow control. The second lecture addressed additional concepts, such as modules, functions, and data structures (lists, tuples, sets, and dictionaries[28]).

We did not expect that students with no prior programming experience would be able to learn enough from just two in-class lectures. While we did not aim to turn such students into full-fledged programmers, we recognized that it would require considerable effort for them to acquire the knowledge and skills necessary to successfully participate in the rest of the course. Hence, we assigned or recommended several other activities to further the aim. First, the students with no prior programming experience were required either to peruse a textbook covering the basics of Python programming[29] or take an online course covering the same material during the first few weeks. Furthermore, we provided the students with pointers to the PEP 8 – Style Guide for Python Code, the documentation to several modules (input/output,[30] pickle,[31] json,[32] and xml[33]), resources covering encoding (Zentgraf 2015) and regular expressions,[34] and an extensive list of Python resources for non-programmers.[35] As the students' backgrounds varied greatly, it was ultimately up to them to decide how much effort they would invest as we tried our best to inform them as to the necessary knowledge and skills for successful course participation. By including programming assignments in the required homework assignments, however, we did try to motivate them to invest the necessary time and effort.

The final activity focused on teaching Python programming was the second homework. It was based on a well-known Spelling Corrector described by Peter Norvig (2007). This simple program learns the frequencies of words from the document corpus and utilizes the model to correct the spelling of a word provided on input. The homework was designed to introduce students to a detailed analysis of a small-sized computer program. The goal of the exercise was to teach them how to follow the flow of the program and to understand what is happening in each step of the execution. This, in turn, would enable students to reason about potential weaknesses or flaws in the program. Students were instructed to first read the "How It Works: Some Probability Theory" section of the program webpage (Norvig 2007) and then to think about the program in general. They answered questions such as: How should a spellchecker behave if it sees a perfectly correct word? What should a spellchecker ideally do in case it sees an incorrect word? What does it mean for a word to be correct or incorrect?

The second part of the homework was designed to teach students how to read the source code. The assignment pointed them to specific lines while asking questions related to the role those pieces of code play in the program. For example, one of the first questions asked them to investigate the role of the imported modules and

[26] Jupyter notebooks (https://jupyter.org) are web-based, interactive environments for programming code in Python (and other languages), executing code, loading and inspecting data, conducting analyses, and plotting the results. Using Jupyter notebooks, an instructor can set up a classroom exercise or homework assignment in a file (an .ipynb file) providing students with code, a series of tasks, and instructions how to perform them with a provided file of data. Students complete the assignment in the notebook and turn it in to the instructor for grading.

[27] WELCOME TO COLABORATORY, https://colab.research.google.com/ Accessed 6/7/2020 For a similar pedagogical purpose Contreras and McGrath (2020, p. 330) employed the Repl.it integrated development environment.

[28] A Python dictionary is a collection of items which is unordered, changeable and indexed.

[29] We recommended the freely available (Halterman 2018).

[30] THE PYTHON TUTORIAL: INPUT AND OUTPUT. https://docs.python.org/3/tutorial/inputoutput.html Accessed 7/7/2020

[31] PICKLE - PYTHON OBJECT SERIALIZATION. https://docs.python.org/3/library/pickle.html Accessed 7/7/2020

[32] JSON – JSON ENCODER AND DECODER. https://docs.python.org/3/library/json.html Accessed 7/7/2020

[33] THE ELEMENTTREE XML API. https://docs.python.org/3/library/xml.etree.elementtree.html Accessed 7/7/2020

[34] REGULAR-EXPRESSIONS.INFO. https://www.regular-expressions.info/ Accessed 7/7/2020

[35] PYTHON FOR NON-PROGRAMMERS. https://wiki.python.org/moin/BeginnersGuide/NonProgrammers Accessed 7/7/2020

where they are being used in the program. This tied back to the lectures covering the concept of modules in Python. Other questions successively led the students from the lowest level functions toward the more complex ones. Eventually, the students had to understand the whole program in order to complete the homework successfully. They were also asked to run the program with carefully selected input and to explain the results. Interestingly, the program learned its model from the textual corpus. Hence, it provided a nice prequel for NLP and ML, topics covered later in the course.

4.1.3 INTRODUCING MACHINE LEARNING

Two lectures focused on introducing ML. The aim was to explain selected ML concepts for practical use in the context of the course, not to provide a comprehensive introduction to ML. Over the remainder of the semester, we planned to revisit and expand upon many aspects of ML, some in the context of specific AI and Law tasks or papers. The two lectures were a mix of a presentation and a guided walkthrough of prepared Jupyter notebooks. The idea was to leave the students with example code that they could use as the basis for the third homework (more details below).

The first lecture introduced the concept of an ML model and discussed training and optimization. After briefly introducing other paradigms (e.g., unsupervised, reinforcement, or transfer learning), we focused mostly on supervised learning, that is, inferring a classification model (or function) from labeled training data. We paid special attention to the experimental setup and evaluation of ML, explaining the use of training, validation, and test sets, as well as the use of k-fold cross-validation.[36] We also covered several evaluation metrics (e.g., accuracy, precision, recall, F1, AUC, and Jaccard similarity).[37] We explained the importance of more detailed inspection of the experimental results using a confusion matrix[38] and via means of detailed error analysis.

With the Jupyter notebook, we demonstrated how one turns a text into features (e.g., a bag-of-words[39]). This is important as text classification plays a major role in much AI and Law research and was frequently discussed throughout the subsequent lectures. The lecture concluded with a brief introduction of several ML models selected based on their conceptual appeal (i.e., logistic regression, decision tree, k-nearest neighbors[40]).

The second lecture was a guided walkthrough of a prepared Jupyter notebook. The goal was to provide students with fully functional code that performs an ML experiment starting from data preparation and finishing with evaluation of the results. As the data set, we used statutory interpretation data that are publicly available. (Savelka 2019) We chose to work with scikit-learn[41] for the ML experiments as it enabled writing the code in the way that largely corresponded to the concepts covered in the previous lecture. The first section of code walked students through the stages of (1) downloading the data in JSON format, (2) exploring the data (3) and preparing the

[36] *k*-fold cross validation is a standard procedure for evaluating an ML program in which the data are divided into *k* subsets or "folds." In each of *k* rounds, a different one of the *k* subsets is reserved as the test set. The ML model is trained using the *k* - 1 subsets as the training set (Ashley 2017, p. 395).

[37] Accuracy is the ratio of correct predictions over the number of all predictions. Precision is the ratio of the number of positive predictions that are correct over the total number of positive predictions. Recall is the ratio of positive predictions that are correct over the number of instances that were positive. F1 is the harmonic mean of precision and recall where both measures are treated as equally important. (Ashley 2017, pp. 393, 396, 400) AUC is an ML metric for evaluating a binary classifier; AUC relates to the probability that a classifier will rank a randomly chosen positive data point (e.g., relevant provision) higher than a randomly chosen negative one (non-relevant provision). (Ashley 2017, p. 393) Jaccard similarity measures the similarity between two sets as the ratio of the size of the intersection and the size of the union of the sets. DeepAI, Jaccard Index, https://deepai.org/machine-learning-glossary-and-terms/jaccard-index Accessed 16/12/2020

[38] A confusion matrix is a table that contains information about a classifier's predicted and actual classifications. (Kohavi and Provost 1998).

[39] A bag-of-words is a representation of a document as a collection of its terms that ignores the sequential order of the terms in the document (Ashley 2017, p. 394).

[40] Logistic regression is a statistical learning algorithm that predicts the odds of being an instance of a category based on the values of independent variables (predictors). It employs an iterative statistical procedure to estimate weights for the predictors. A decision tree is an ML technique that learns a tree-like set of questions or tests for determining if a new instance is a positive instance of a classifier. Each question is a test: for example, if the weight of a feature is less than a threshold value, branch one way, otherwise branch the other way. A k-nearest neighbor or k-NN algorithm compares a problem with cases to base a prediction on those that are most similar. One measures the similarity or dissimilarity between the features of the cases in terms of some metric. Then one predicts that a new case will have the same outcome as its closest neighbors. (Ashley 2017, pp. 108, 396, 398)

[41] The scikit-learn library provides an open-source set of software tools for predictive data analysis. scikit-learn, https://scikit-learn.org/stable/index.html Accessed 7/7/2020

data to be used in the ML models (including the division into training, validation, and test sets). As the task was simple text classification, the second section focused on (4) training decision tree, k-nearest neighbor, and logistic regression models. The final part was dedicated to (5) evaluating the three models using accuracy, precision, recall, F1, as well as confusion matrices.

Three related required readings supported the two in-class lectures on ML. Two of them introduced facets of ML concerning choice of appropriate representations and algorithms for learning and the potential for unsupervised learning, that is, automatically clustering data to reveal patterns that may correspond to concepts (Halevy et al. 2009; Domingos 2012). An early article about Lex Machina illustrated applying machine learning to predict outcomes of cases from texts quite independently of their substantive facts (Surdeanu et al. 2011). The goal of the required reading was to create a situation where the students would need to apply the concepts discussed in class in order to understand the articles.

Finally, the third homework assignment related to the two lectures introducing ML. The set of exercises was intended to reinforce students' understanding of the basic steps in carrying out a supervised ML experiment. The goal was to help them practice each step of such an experiment, beginning with acquisition and preprocessing of data and finishing with the evaluation of the experimental results. Practically, the students were supposed to start from the Jupyter notebook used in the second intro to ML lecture and adapt it so that they could solve the problems from the homework assignment. We also included two optional exercises that went beyond what was provided in the notebook from the class. One focused on using a different classifier from those presented during the lecture. The other involved improving a model's performance using grid search to optimize hyper-parameters.

4.1.4 INTRODUCING NATURAL LANGUAGE PROCESSING

Two lectures were dedicated to introducing NLP. Again, the goal was to explain selected concepts for practical use in the course context, not to provide a comprehensive treatment of NLP. We also planned to revisit and expand upon certain aspects of NLP over the remainder of the semester in the context of specific AI and Law tasks or papers. As in the case of the introduction to ML, the two lectures were organized as a mix of presentation and a guided walk through prepared Jupyter notebooks. Similarly, the idea was to leave the students with an example code that they could use as the basis of the fourth (last) homework (more details below). The NLP lectures supplemented the ML lessons by focusing on the initial stage where texts are transformed into features.

The first lecture interleaved examples of applying NLP methods using Jupyter notebooks. We introduced NLP as aiming to develop methods to solve practical problems involving language (e.g., search, information extraction, summarization) and contrasted it with computational linguistics, whose goal is to understand properties of human language. We then discussed several core analytical sub-tasks, such as tokenization, segmentation, grammatical and syntactic parsing, or named entity recognition. After a brief introduction of each such sub-task, we provided a practical demonstration with the spaCy NLP library.[42] We concluded the lecture with a case study on sentence boundary detection and its challenges in legal texts, specifically case law.[43]

The second lecture presented two more case studies via required readings. The first was an example of named entity recognition in legal texts using a mapping of a specialized legal ontology onto a general one.[44] The second case study concerned automatic recognition of facts and legal principles from case law.[45] Additionally, the students were also assigned to read a paper on an automatic analysis of jury verdicts which was not covered during the in-class lectures (Conrad and Al-Kofahi 2017). The second part of

[42] SPaCY, Industrial Strength Natural Language Processing, https://spacy.io/ Accessed 6/7/2020

[43] The case study was based on (Savelka et al. 2017).

[44] An ontology is a general, formal specification of the objects in a domain, their relations and properties. The case study was based on (Cardellino et al. 2017).

[45] The case study was based on (Shulayeva et al. 2017).

the lecture was a guided walk through a prepared Jupyter notebook. This walkthrough was like the one in the second introduction to ML. The goal was again to provide the students with fully functional code that performs an ML experiment starting from data preparation and finishing with evaluation of the results. The focus, however, was much more on text feature representation. Three different approaches were covered (i.e., bag-of-words, static word embeddings,[46] and transfer learning with contextualized word embeddings[47]). As the data set, we again used part of the statutory interpretation data that is publicly available (Savelka 2019).

The fourth and final homework assignment related to the two lectures introducing NLP. The set of exercises enabled students to practice performing the basic steps of integrating NLP components into a supervised ML experiment. The goal was to reinforce students' awareness of all the steps, but especially of the choices concerning feature representation. Practically, the students were supposed to adapt the Jupyter notebook used in the second introductory NLP lecture to solve the problems from the homework assignment. In order to complete the assignment, students could use the scikit-learn library and the same model as in the previous homework. Optionally, they could also train the model using static word embeddings, such as GloVe[48] or word2vec (Mikolov, et al. 2013), or fine-tune a transformer-based deep neural network language model (Devlin et al. 2018).[49] The example code to perform even the optional tasks was provided. This was intended to serve as a solid basis for the work the students carried out in their course projects.[50]

4.1.5 INTRODUCING ARTIFICIAL INTELLIGENCE AND LAW

A final goal of Part I was to convey a basic understanding of the field of AI and Law and its role in the future of legal practice based on the readings, two chapters of *Artificial Intelligence and Legal Analytics* (Ashley 2017).[51] The lecture introduced students to computational models of legal reasoning, including legal expert systems, their uses and limitations. We discussed applying logic programming to represent statutory rules as in a classic program where the researchers expressed the British Nationality Act (BNA) in the form of logical rules and effectively "ran" the statute as a computer program (Sergot et al. 1986). We focused on some problems for formalizing legislation with logic. These include semantic ambiguity and vagueness, syntactic ambiguity caused by the absence in natural language statutes of parentheses demarking the scopes of logical connectors, and legal indeterminacy: even if adversaries agree on the facts and applicable law, they can still generate reasonable pro and con arguments, and courts come to different conclusions (Berman and Hafner 1986). This is important because classical logic cannot readily support proving a proposition and its opposite, something that contending legal arguers frequently do.

Finally, we introduced a theme of the book and of the course: the role of text analytics in cognitive computing. Cognitive computing involves designing collaborations between humans and computers in which each performs the kinds of intelligent activities that they can do best. Legal text analytics, we hypothesize, can extract meaningful information from legal texts that will enable computational models of legal argument to connect more directly with

[46] Specifically, we used the GloVe embeddings described in (Pennington et al. 2014).

[47] Specifically, we used fine-tuning of the BERT model described in (Devlin et al. 2018).

[48] A static word embedding such as GloVe is a learned text representation that employs statistics about word co-occurrences across the whole text corpus to capture word contexts and meaning. "Static" indicates that the word embeddings are employed as a component of the machine learning model and not updated (Pennington et al. 2014).

[49] A language model is a neural network that has been trained to predict the next word in a sequence of words. A transformer, a special kind of neural network architecture, has an attention mechanism; each layer of its network assigns more weight to some features of an input sentence than to others. It learns associations between words that might be relatively far away from each other in complex sentences. By making multiple, parallel connections between certain words while ignoring others, it develops a "treelike representation of sentences [which gives] transformers a powerful way to model contextual meaning…." (Pavlus 2019)

[50] A course on teaching coding for lawyers at the University of Minnesota also combined lectures and "practical exercises based on Python and covered the fundamentals and practical exercises of machine learning and natural language processing techniques." "[S]tudents complete a code-based guided tutorial of an end-to-end machine learning project to understand the steps involved in the creation of this technology." (Contreras and McGrath 2020, pp. 325, 328-330)

[51] Ch. 1, Introducing AI & Law and its Role in Future Legal Practice and Ch. 2, Modeling Statutory Reasoning.

the texts. This, in turn, will expand the capabilities of cognitive computing by addressing some of the limitations of machine learning for answering legal questions and explaining the answers (Ashley 2017, pp. 11-12).

4.2 PART II: COMPUTATIONALLY MODELING CASE-BASED LEGAL ARGUMENT AND IMPROVING LEGAL IR

Building on this foundation, Part II provided a broader framework by exploring AI and Law research on modeling case-based legal reasoning and argument and on using the models to predict and explain legal outcomes. This is one way the field can assist machine learning to explain its answers and predictions in terms attorneys can understand.

A lecture and readings[52] introduced competing computational models of case-based reasoning in law – legal argument as theory construction, as *a fortiori* reasoning, or as explanation mapping. Each one employs a different knowledge-based technique to represent legal cases: in terms of prototypes and deformations, dimensions and factors, or exemplar-based explanations. Each one computes relevant similarity of cases in a different way, for example, as overlapping sets of factors or as ratios of criterial facts, and supports different kinds of counterexamples.[53] The competing case representations have different implications for using text analytic techniques to connect the texts of legal decisions directly to computational models. So far, only the factor-based representation seems likely to be able to be extracted automatically from case texts. In addition, only the factor-based models have been adapted to account for teleological reasoning with the values underlying the legal rules.[54]

Researchers in AI & Law have developed increasingly sophisticated knowledge-based methods for predicting outcomes of legal cases. The lecture surveyed these methods: from applying a nearest neighbor algorithm to tax cases on capital gains represented with features, to applying decision trees to model judicial decisions regarding bail, to predicting whether a U.S. Supreme Court Justice and the whole Court affirm or reverse a lower court's judgment. Katz, et al. (2017) applied extremely randomized forests of decision trees to past decisions represented as features from the Supreme Court Database and information about trends. In that reading students encountered another example of evaluating a machine learning program with stratified k-fold cross-validation as introduced in Part I.

The lecture and Chapter 4 also introduced work on predicting case outcomes based on case-based arguments and taking value effects into account as a kind of theory construction.[55] These knowledge-based prediction approaches can explain their predictions in terms of legal arguments.

The reading, Chapter 5, Computational Models of Legal Argument,[56] introduced students to a revolutionary development as AI and Law researchers produced general models of legal argument into which models of rule-based and case-based reasoning fit as complementary modules. The new models, such as Carneades, use attacking (and/or supporting) arguments instead of strict logical inferences.[57] Unlike relying solely on deductive logic, the argument models accommodate two realities: legal reasoning is nonmonotonic and defeasible. That is, inferences change as information is added or becomes invalid (nonmonotonicity) and arguments contradict and defeat each other (defeasibility).

The lecture featured two examples of computational models of legal argument. Given a case as a list of factors, Grabmair's Value Judgment-based Argumentative Prediction (VJAP) program outputs a predicted outcome, a level of confidence in the prediction, and an argument justifying prediction. It uses rule-based, case-based, and value-based reasoning to argue if a decision is coherent with tradeoffs among value effects in prior cases and bases its prediction on the arguments.[58]

[52] (Ashley 2017) Ch. 3, Modeling Case-Based Legal Reasoning, Ch. 4, Models for Predicting Legal Outcomes. (Katz et al. 2017).

[53] (Ashley 2017, Sec. 3.3)

[54] (Ashley 2017, Sec. 3.4, 3.5)

[55] (Ashley 2017, Sec. 3.5, 4.6)

[56] (Ashley 2017) Ch. 5, Computational Models of Legal Argument

[57] (Ashley 2017, Sec. 5.1 - 5.3)

[58] (Ashley 2017, Sec. 4.6)

The second example is a model of evidence-based legal arguments about compliance with legal rules, the Default Logic Framework (DLF). As described below, it inspired the LUIMA type system featured in the next readings.[59] A type system is a kind of ontology for annotating training sets of texts in terms of a hierarchy of concepts and relations so that machine learning in an annotation pipeline can automatically assign meanings to regions of text. The LUIMA type system captures the argumentation roles that sentences play in judicial decisions. These include a LegalRuleSentence, which states an abstract legal rule without applying it to a particular case, or an EvidenceBasedFindingSentence, which reports a factfinder's finding on whether or not evidence in a particular case proves that a rule condition has been satisfied. As the lecture explains, we hypothesize that state-of-the-art legal information retrieval systems could be more effective to the extent that they consider argument-related information such as sentence roles, a point elaborated in Part III.

4.3 PART III. APPLYING TEXT ANALYTICS TO LEGAL TEXTS

Part III's readings and lectures focused on applying machine learning to three types of legal documents: electronically stored documents produced in e-discovery, statutes, and the texts of legal decisions.[60]

4.3.1 APPLYING ML TO DOCUMENTS IN EDISCOVERY AND STATUTES

After recapitulating the machine learning methods, metrics and evaluation techniques introduced in Part I, a lecture addressed their application in the e-discovery domain that was introduced in the *Dynamo Holdings* case. Among other things, it featured a project applying network analysis of emails that could guide selecting witnesses for deposition and the possibility of constructing such a network automatically from the sender and receiver header information in emails produced in the litigation.

Another segment focused on applying machine learning and rule-based techniques to extract information from statutory or regulatory texts. Machine learning can identify the topic and type of a statutory provision, for example, whether it is a definition or imposes an obligation; knowledge engineered rules (or regex expressions) can then extract relevant information from these provisions, such as the obligor and obligee. The ML approach requires human experts to annotate training sets for learning and gold standard sets for evaluation, a labor-intensive and costly activity. Some research projects illustrated the utility of applying knowledge-engineered rules and templates to partially annotate texts in support of human annotation. In principle, this could make human annotators' work more efficient and enable non-expert humans to annotate as well as experts. Indeed, students could successfully annotate statutes and cases in a crowdsourced activity.

A lecture pointed out that despite these ML techniques, automatically extracting logical formulations of rules, instead of manually constructing them as in the BNA program, still seems technically out of reach. Combining text extraction and network diagrams, however, can help to partially automate the comparison of similarly purposed regulations such as different states' public health emergency statutes. This is of practical import, since commercial organizations, for example, insurance, health care, computer security, and privacy regulation, must comply with multiple states' similar, but somewhat different, regulatory schemes.

4.3.2 APPLYING ML TO LEGAL DECISIONS

Three articles reported on successive experiments predicting case outcomes based on their texts in a manner that accounted for some information about their substantive topics and the parts of the decision reporting the facts: (Aletras et al. 2016; Chalkidis et al. 2019; Medvedeva et al. 2020). These papers support the feasibility of automatically analyzing legal texts and building successful predictive models of judicial outcomes. The data comprised decisions of the European Court of Human Rights. Given textual evidence extracted from a case, the task is to predict the main conclusion of the court, that is whether there was a violation or not of any article of the European Convention on Human Rights or, alternatively,

[59] (Walker 2006) Ch. 6, Representing Legal Concepts in Ontologies, Ch. 7, Making Legal Information Retrieval Smarter and Type Systems. (Ashley and Walker 2013).

[60] (Ashley 2017) Ch. 9, Extracting Information from Statutory and Regulatory Texts.

a violation of a particular article. The results support a text-based approach for making *ex ante* predictions of case outcomes. Anonymizing information in the texts such as the name of the state involved in the case did not much affect the prediction. On the other hand, predicting outcomes simply from the names of the judges achieved lower rates of success, but not much lower. The third paper also introduced students to the application of new language models for text-based prediction such as BERT (Bidirectional Encoder Representations from Transformers) and attention-based neural networks that can highlight the most predictive portions of the text.

We supported the lectures on the court outcome prediction with a Google Colab notebook. As the basis we used a paper by Medvedeva et al. (2020) and their publicly available dataset.[61] The notebook first led the students through the data exploration and preparation phases. Then, they could go through the same experiments as those reported in the paper. Finally, students were encouraged to experiment with the code and try their own experiments. This exercise was important because it introduced the students to the idea of reproducing the results from a paper and what such an effort requires.

We then revisited the topic of extracting information from the texts of cases, this time using more knowledge-intensive techniques. Predicting case outcomes by themselves is not sufficient, especially if a program cannot explain its reasons. For explaining and for improving legal information retrieval, we maintain, one needs to extract argument-related information from case texts. In Part II, we spoke of factors and sentence roles in cases as a kind of argument-related information useful for modeling legal reasoning. Here we considered the extent to which those features can be extracted automatically through machine learning and related techniques. The class reviewed the classic argument mining work of Mochales and Moens (2011) in extracting argumentative propositions and classifying them as premises or conclusions. In one of the readings, Shulayeva et al. (2017) annotated “cited facts and principles in legal judgements”. That is the kind of information that could help attorney/users quickly decide if a case retrieved by an IR system is worth reading. Working with fifty common law case reports from the UK and Ireland dealing with civil matters, the researchers demonstrated the feasibility of humans reliably annotating applications in cited cases of legal principles to facts and of an ML system learning to perform such annotation.

As a final example of extracting argument-related information, we returned to the LUIMA project of Part II and its type system for annotating case texts in terms of the roles sentences play. An experiment described in the readings,[62] pitted LUIMA against a commercial legal IR system. LUIMA’s search module employed cases annotated in terms of certain sentence argument roles and retrieved them based on the annotations. LUIMA’s re-rank module learns weights from a human expert’s “true” rankings of cases in the training set that are responsive to queries; it uses these weights to re-rank responsive cases in the test set. The LUIMA evaluation showed that employing argument-related information in indexing and querying along with the re-ranking weights outperformed retrieval with a commercial IR system that did not use such information, at least for a restricted set of documents in the domain of vaccine injury claims.

By annotating sentence role types such as Evidence-BasedFindingSentence, an argument retrieval system could help users find “application cases”, that is, not just cases supporting a legal proposition in the abstract, but examples of applying a proposition to concrete facts which may be analogous to the user’s problem. We considered how LUIMA’s semantic annotation and argument-based reranking could be applied externally to a commercial system’s output for a user query, potentially improving the re-ranking, which would be a nice example of cognitive computing.

The class examined various approaches to automatically segmenting case decisions into argument-related functional parts. Two readings focused on the identification of rhetorical roles of sentences such as establishing facts, case history, arguments and analysis, the decision ratio, and the final decision (Saravanan and Ravindran 2010; Bhattacharya et al. 2019). The outcome is a partitioning

[61] https://ndownloader.figshare.com/files/8901952?private_link=6f7d9e7c375ff0822564

[62] (Ashley 2017) Ch. 11, Conceptual Legal Information Retrieval for Cognitive Computing. (Grabmair et al. 2015).

of a text at the sentence level into several "buckets," information that could be used to inform a summary of the case. Another reading explored automatically segmenting court opinions into high-level functional parts such as Introduction, Background, and Analysis as well as issue-specific parts with Conclusions (Savelka and Ashley 2018). The results for functional parts were not very far from human performance; the results for Conclusions were less successful. The lectures on case law segmentation were supported with a Google Colab notebook based on (Bhattacharya et al. 2019) employing the researchers' published code for their experiments.

Commercial IR providers, of course, are constantly enhancing their own systems for cognitive computing. Students read a paper describing Lexis Answers, a recent addition to the Lexis Advance legal research platform (Bennett et al. 2017). The paper presents the challenges that the system developers are addressing, for example, the need to identify irrelevant text, such as text quoted in the documents, the need to disambiguate terms by context, for example, distinguishing meanings of "fraud" in criminal and civil law, and the problem of creating an ontology and type system to support text understanding of Lexis users' queries. For example, is a user seeking the "standard of review in an appellate court" for a family of claims or for a specific claim? Finally, there is the question of how to evaluate the system's utility and effectiveness.

4.4 PART IV: AI IN LEGAL DOMAINS

In Part IV, the remaining readings and last lectures in the course focused on more specific legal domain topics including AI and patent law,[63] legal regulation of data privacy (O'Connor 2018) and machine learning bias (Angwin et al. 2016), as well as legal liability of autonomous vehicles (AVs) (Reed et al. 2016; Federal Automated Vehicles Policy 2016; Nooteboom 2017).

The class considered the effects of electronic data mining on privacy. Today, data collected can be aggregated to produce telling profiles of who users are, as revealed by what they do and say on the Internet. The information users generate as a by-product of this activity is quite valuable. Indeed, data mining of consumer information is big business. Data mining discovers patterns in distributed information regardless of how the data are formatted or of legal and ethical constraints on fairness. We then surveyed the generally limited extent to which American law under the U.S. Constitutional, federal, and state regulatory systems protects against these threats. We contrasted these with the protections under Europe's General Data Protection Regulation ("GDPR") and the new California Consumer Privacy Act (CCPA).

Machine learning and big data can be subject to bias, and correcting for a lack of fairness has been a perennial concern. The *ProPublica* article addressed fairness in machine learning and the problems of disparate impacts of ML in bail recidivism prediction, including the investigation of racial bias in the COMPAS re-offense prediction program (Angwin et al. 2016).

The lecture raised the question of the extent to which fairness can be achieved given practical and theoretical constraints. For example, if a model is unfair, one possibility is to blind it to that attribute. Indeed, a common safeguard in privacy regulation is to avoid using "prohibited" variables. A simple test for unequal treatment is to change the group attribute for a data point, and check if the prediction changes. Trying to achieve fairness by blinding the model to certain problematic features, however, does not always work. Apparently "harmless" variables can serve as proxies for group membership since they may not be statistically independent of the group membership. Imbalances in the dataset may still lead to different treatment of groups. In addition, one *needs* to use some prohibited variables such as Group membership information in order to test for equal treatment. Moreover, removing predictive information from the model may lead to lower accuracy.

Ensuring fairness, then, can be complicated. One can employ an optimized equation, one that tries to ensure that fairness constraints are met. Optimizing equations for fairness, however, is still subject to a problem: there are multiple conceptions of fairness, such as statistical/ demographic parity, accuracy parity, equalized odds, equal opportunity, and predictive parity / conditional

[63] A guest lecturer presented a talk entitled "NLP Customized to Fields of Law: Promises and Challenges" in which he focused on using AI to analyze patents' level of "indefiniteness" flagging a potential statutory violation.

use accuracy. Each fairness criterion has something to recommend it, but they can be mutually exclusive; one may not be able to achieve any two of them at the same time (see section 6.)

The readings and lecture on autonomous vehicle (AV) liability addressed the questions of how the American law of product liability and negligence will apply to the manufacturers of vehicles involved in fatal accidents. The lecturer noted that, even though AVs will increase traffic safety overall, inevitably, they will still cause accidents injuring people and property, and victims will sue manufacturers of AVs and AV software systems. Presumably, the legal standards affect the ways in which automated vehicles should be designed given certain ethical responsibilities. For example, any drivers, human or automated, must be prepared to deal with children on the street and other so-called "outlier" scenarios. These standards also give rise to legal and practical issues that affect machine learning, for instance, in determining whether negligence or product liability applies and in proving causation.

5. FINAL PROJECTS

The final projects were one of the most important elements of the course, accounting for 50% of a student's final grade. The goal of the projects was to provide students with an opportunity to apply what they learned during the course in the context of a larger work that they were partially responsible for designing. The projects were a team effort. The only requirement was that a team consist of at least one law student and at least one student pursuing a technical degree with a significant programming component (e.g., computer science or computer engineering). This increased the likelihood that each team had the necessary knowledge and skills to complete a successful project. Furthermore, it ensured that students with different backgrounds had to find ways how to interact with each other in a meaningful and productive way. As this is a very important professional skill, we considered it to be one of the projects' most beneficial aspects. Although it was largely up to the students to design the projects, there were certain fundamental requirements. Most importantly, each project had to have an empirical component such as surveying a dataset or evaluating a ML model. An exercise of reading literature and writing an essay would not suffice.

5.1 SUGGESTED PROJECT IDEAS AND DATASETS (FIRST PROJECT MEETING)

Based on the experience from the 2019 run of the course, we made sure that the students had an opportunity to start working on the projects very early in the semester. For this reason, we held the first project session in the second week of the course. During this session we explained the basic project requirements and milestones. First, we introduced several selected projects from the last year.[64] Second, we presented the three datasets that we had identified as a suitable basis for a project. These included the Supreme Court Database (SCDB)[65] with full texts of the Court's opinions,[66] Veteran Claims Decisions[67] comprising disability-claim decisions of the Board of Veterans' Appeals ("BVA") of the U.S. Department of Veterans Affairs, and the Enron Email Dataset with

[64] In the spring 2019 version of the course, the students completed five final projects:

1. Web Privacy Policy Analysis (Goal: To inform the public about privacy policies, apply supervised ML and NLP to a corpus of privacy policies to determine the kinds of data collected, by whom and if certain practices may be performed.)
2. Whose Line is it Anyway? (Goal: To inform readers about bias, apply supervised ML to dissenting opinions of Justices to classify sentences in legal opinions as taking a conservative or liberal stance.)
3. Risk Mitigation Strategy for International Patent Litigation (Goal: To inform potential litigants in international patent suits about their prospects in domestic U.S. courts, use ML to predict the results of patent litigation involving international parties based on database of U.S. patent litigation cases.)
4. Discover Legal Education Predictors of Bar Exam Passage and Failure (Goal: To identify law students who may need extra preparation, use data analytics and ML to identify early predictors for failing the bar exam and how those predictors change as students' legal education progresses.)
5. Tax Court Analysis (Goal: To assist tax professionals to advise clients on when to seek settlement based on assessed deficiencies, apply ML to a dataset of tax cases to determine the deficiencies assessed against individual taxpayers who lose their cases against the IRS and the most commonly cited sections of the Internal Revenue Code.)

[65] The Supreme Court Database, http://scdb.wustl.edu/ Accessed 6/7/2020

[66] Caselaw Access Project, https://case.law/ Accessed 6/7/2020

[67] LLTLab VetClaims - JSON https://github.com/LLTLab/VetClaims-JSON Accessed 6/7/2020

more than 600,000 employee emails from the period just before the energy company's collapse in December 2001 due to a mammoth accounting fraud. The Federal Energy Regulatory Commission generated the corpus as part of its fraud investigation.[68]

This was a considerable departure from the previous year when we assigned students the task of identifying a proper dataset as part of pitching a project and forming a team. We had reasoned that it would be educational for law students to identify domains and potential data sets as part of the exercise. While it did lead to successful projects, we found that, at about the halfway point in the course, some students had not yet obtained enough well-informed intuitions to assess the suitability of data for projects. In the end, organizing project pitches and supporting teams in their formation and technical setup required considerably more time and effort than we had expected, and it drew too much attention away from other tasks.

After briefly introducing the three prescribed datasets (via short code snippets in Google Colab notebooks), we suggested several project ideas for each of them. These were very high-level ideas that the students could adopt directly and develop into full proposals or use as an inspiration for their own ideas. For example, for the Enron Email Dataset we suggested identifying which emails are purely personal and express worry/anxiety, sadness/despair, or anger/agitation. For the SCDB, our suggestion was to predict if a justice will vote for or against affirming a new decision involving issues identified from the text of the decision itself. Finally, an example of a suggestion for the Veteran Claims Decisions was to automatically identify the role a sentence plays in a legal decision and determine if it is helpful to understand the sentences as tightly dependent ordered sequences or rather as independent units.

After the session, the students were expected to form teams and start working toward project proposals. We provided a template for proposing a project. It asked students to describe the legal domain problem the students proposed to tackle, including the process, task, or use case. Students should identify the current problem or limitation in their chosen legal domain or dataset that would benefit from a data-driven solution and their proposed solution. The focus was on demonstrating proof of concept; the template asked the students to explain the incremental progress that the project would make towards the envisioned solution, either by creating and statistically exploring a new dataset to reveal interesting patterns or by applying analytical/ML models to an existing dataset toward realizing some goal. To inspire project ideas, the instructors suggested reading a recent paper (Branting 2017) that surveyed some practical legal application areas.

5.2 PROJECT PROPOSALS (SECOND PROJECT MEETING)

The goal of the second project session was for the teams to present their project proposals and have them discussed. Eventually, the students formed three teams: two teams with three members and one team with four. The one smaller team was also joined by the Duquesne University Law School professor who was auditing the lectures. Each group was allocated 20 minutes for the presentation and 10 minutes for a discussion. Interestingly, none of the teams opted for one of the proposed project ideas and, hence, all the teams proposed their own projects. One of the teams even opted for using a different dataset from the three suggested ones. This shows that students were motivated to put extra effort into the projects.

The first team proposed a text classification project focused on an area of criminal procedure: the constitutionality of a search for and seizure of evidence under the Fourth Amendment. They proposed to automatically identify language in case decisions suggesting that a court had applied a “totality of circumstances” rule versus a “bright line” rule in determining the validity of the search. Here, the team had a clear idea of the work they wanted to carry out including the technology they would use (a transformer-based language model such as BERT). The team also committed to assembling their own data set including annotating the data.

Our initial feedback on the proposal was to take extra care in the data collection step. We suggested that the team work on explicitly defining the type system. We provided

[68] UC Berkeley Enron Email Analysis Project, http://bailando.sims.berkeley.edu/enron_email.html Accessed 6/7/2020

several examples of annotation guidelines and prompted the students to think about how to deal with the situation where a single decision contains language associated with both (or different) types of rules. We advised the students to use explicit, reasonable criteria to guide the selection of legal decisions to include in their data set. We emphasized the importance of demonstrating that human annotators could agree on the task of labeling instances of the types. Demonstrating such interrater reliability (IRR) is a crucial component in assessing the system. Hence, we recommended that two different people look at a certain number of the same decisions independently in order to measure IRR by computing a kappa/alpha statistic. Finally, we asked the students to think about a benchmark against which they could evaluate the system, such as a random or majority class baseline.

The second team wanted to investigate the concept of a swing vote in Supreme Court decisions. This phenomenon relates to the justice who typically breaks the tie in a decision that otherwise would involve an even number of votes for and against, (i.e., conservative vs. liberal). Such a justice plays a vital role in decisions that would otherwise be deadlocked, often involving a clash of political ideologies. The goal was to use empirical methods to predict which justice would likely play such a role in each case.[69] The team would use the SCDB to carry out the project.

Although, the proposal left certain aspects of the project open we deemed that the students were quite close to defining the scope of a successful project. In order, to give the effort more structure we provided the students with a list of suggested steps. First, we asked them to come up with an explicit definition of what it means for a case to be decided by a swing vote. Second, given the definition we advised the students to come up with a strategy for distinguishing these cases from the others. Third, we proposed that they explicitly define what it means for a justice to be a swing vote in a case. The final step would be to come up with a strategy for identifying such a justice based on the definition.

The third team struggled to come up with a proposal for a single project. Instead, they came up with several general ideas. We identified the two most promising ones and for each we suggested the steps that would be involved.

The first one would have resulted in a project focused on an analysis of a disagreement of the Supreme Court with the court of lower instance. The analysis would be driven by either a faceted definition of a disagreement (disagreement in terms of different aspects) or the strength of a disagreement on an ordinal scale (e.g., very strong, strong, medium, mild). In carrying out the project the students would likely have to filter the SCDB dataset to cover specific agencies and issues, come up with a detailed definition of the disagreement, manually annotate a non-negligible number of cases from the data set in terms of the disagreement, train an ML system using the textual features of a case to figure out the strength/type of a disagreement, and evaluate the system.

The second proposal was a project focused on an analysis of the effects of a veteran being represented by an attorney before the Board of Veterans' Appeals. In carrying out the project the students would likely have to filter the BVA dataset we would provide them, explicitly define what it means for a veteran to be represented by an attorney, define the possible types of outcomes, manually annotate a non-negligible number of cases in terms of the attorney representation and outcome, train a ML system using the textual features of a case to determine the outcome and if an attorney represented the veteran, evaluate the model, apply the system to the full dataset, and analyze the effects of attorney representation of the veteran. The team chose to work on the second proposal, primarily because of its practical importance to veterans.

5.3 WORK ON THE PROJECTS (BI-WEEKLY STAND-UPS)

After the meeting where the students proposed their projects, they were supposed to work toward their completion. In order to make sure the students were making steady progress during the semester, we held bi-weekly, 30-minute meetings with each group. Students began the meetings by explaining what they had done and their plans for the next steps. We usually commented on the progress and made sure the team remained on track

[69] Studies suggest there is substantial variation in the identity of the median justice across areas of the law. (Lauderdale and Clark 2012)

toward successfully concluding the project. Interestingly, each group required a very different approach.

The group working on automatically identifying the type of rule, "totality of circumstances" versus "bright line," a court applied in deciding the constitutionality of a search turned out to be self-reliant. Even though the group appeared to be able to complete the project unassisted, we focused on specific points where we could help. Specifically, our approach with this group was to point out weaknesses and limitations of their work and help the students think through if and how to remedy them.

The second team that worked on investigating the concept of a swing vote in Supreme Court decisions struggled with how to include some non-trivial empirical component as required. Specifically, we were concerned that their project appeared to be limited to a simple data filtering/analysis. Hence, most of the interaction with this team focused on discussion of how to make the task more ambitious. Toward the end of the semester, the students responded by redefining the project as predicting how justices would vote in individual cases.

The third group, that worked on analyzing the effects of a veteran being represented by an attorney before the BVA, plunged into a detailed inspection of the decision texts looking for textual cues suggesting representation (or the lack of it). Our main effort with this group was to help them see the whole picture and move beyond the initial step. The students responded well and brought the project to a successful conclusion. By the end of the semester, however, they had substantially scaled down their original plan.

5.4 THE RESULTING PROJECTS (FINAL PRESENTATIONS AND PROJECT REPORTS)

The students presented the results of their projects during two final project sessions. Each group had 20 minutes for the presentation and 10 minutes for the discussion. It was up to the students to decide how to deliver the presentation. Although there was no requirement that all the members of the group present, all three teams decided to split the presentation among all the members. The presentations took place about a week before the project reports were due. Hence, we focused the discussion on the points that we believed were important to be addressed in the reports.

The first group succeeded in delivering an exceptional project exactly as proposed. Specifically, they assembled a limited dataset of short excerpts from US Supreme Court Fourth Amendment cases retrieved from WestLaw. They manually labeled the case texts as suggesting the use of a "totality of circumstances" rule or a "bright line" rule. The team performed two sets of experiments. In the first set they evaluated the performance of the ML models (decision tree, support vector machines, multi-layer perceptron) trained on simple representation of text features (bag-of-words). The best performance achieved by these models was an F1 of 0.72 (weighted; 0.68 for the bright line class and 0.74 for the totality of circumstances class), a reasonably good combination of precision and recall. In the second set they fine-tuned a transformer-based language model based on BERT (Liu et al. 2019). This approach turned out to be much more successful with F1 of 0.87 (0.88 and 0.86 respectively).

The project exceeded all expectations. From the viewpoint of fulling the course project requirement, there was little to criticize. We suggested that the students consider the possibility of continuing their good work with the goal of presenting it at an AI and Law venue (either a conference or a workshop). In order to do so, we suggested, they would need to significantly extend the data set and define the annotation task with explicit guidelines. Additionally, an inter-annotator agreement study should be conducted to confirm the objectivity of the task and its level of difficulty for humans. Ideally, they would also be able to extend the system to deal with all the documents' text, not just the portions selected by humans.

The students performed some of these improvements during the summer of 2020, submitted an expanded version of their report, entitled "Transformers for Classifying Fourth Amendment Elements and Factors Tests" to the Jurix 2020 conference, one of the two main conferences in AI and Law, which accepted it as a full paper (Gretok, et al. 2020)!

The second team completed a successful project, as well. The team decided to abandon the original plan to predict a swing vote in Supreme Court decisions. Instead, they conducted a very detailed analysis of the SCDB features as to their power for predicting justices' voting (i.e., liberal or conservative) with the intention of training a multi-layer perceptron model on the subset of most predictive features. The team reported that features regarding issue, issue area, and lower court decision appear to be the most predictive of a justice's vote. A multi-layer perceptron model using the most predictive features, however, did not outperform the baseline (majority class). As regards this project we appreciated the students' detailed feature analysis. It was important for them to learn that some features are substantially more predictive than others. Ideally, the students would have detected the limited predictiveness of some of the features earlier in the semester in time to steer their project toward a more productive outcome.

The last group struggled the most but completed the project successfully, as well. The original plan was to determine from a case decision if a veteran was represented by an attorney before the BVA and then correlate the variable with outcomes. In the end, however, the team was not able to carry out the second step of the plan. We appreciated how the students delved into the actual texts looking for patterns indicating if the veteran was represented or not. We believe this effort was a substantial educational exercise; it helped them to understand some of the rewards and challenges of working with legal texts at scale. The team spent so much time and effort in identifying the project aim and in exploring, annotating, and preparing the data, however, that they left insufficient time to correlate the representation variable with the outcomes. Nevertheless, even with its limited scope, the work was acceptable as the final course project.

6. COMPARISON WITH SPRING 2019

Over the course of the spring 2019 semester, students engaged in three activities that were not included in spring 2020. They: (1) participated in a series of exercises using the Supreme Court Database (SCDB) and working in class with prediction code based on that developed by Katz, et al. (2017), (2) performed a programming assignment focused on fair machine learning, and (3) engaged in a multi-stage annotation and text analysis homework assignment and in-class text analytics workshop.

The 2020 course version afforded less time for these exercises because of the way in which we reorganized the process of preparing for final projects. As noted, we began formulating final project teams and topics much earlier in the semester and focused the student teams on selecting among three legal text corpora,[70] which we made available early in the semester. We tried to focus all the technical assignments on programming tasks that would support final projects involving legal text analytics (as opposed to solely the SCDB or other nontextual data). Thus, we revised the course curriculum to introduce students earlier to programming with Python and involving natural language processing and machine learning. This made the process of organizing final projects more efficient and contributed to their quality, but at the expense of leaving out technical exercises with non-textual legal data such as the SCDB and bail recidivism prediction data. In addition, we devoted more lecture time to placing legal text analytics within the context of research in AI and Law, leaving insufficient time for the annotation and text analysis task.

Nevertheless, since these three spring 2019 activities seemed to be valuable pedagogical exercises, we include brief descriptions here.

6.1 SCOTUS PREDICTION

This in-class Python programming effort tied into a series of sessions devoted to predicting outcomes of U.S. Supreme Court cases. It was originally intended to include a homework assignment, but the instructors decided to keep it in-class given the students' mixed levels of technical expertise and the higher technical difficulty in working with the dataset and models.

As noted, in some highly publicized work, Katz, et al. (2017, p. 2) applied machine learning to predict outcomes of SCOTUS decisions and the votes of individual justices. They applied a decision-tree-based learning

[70] As noted in section 5, these included the Enron Email Dataset, Supreme Court Data Base with the addition of full texts of the opinions, and the Veteran Claims Decisions. The three spring 2020 projects dealt with the latter two corpora, although one of the SCDB projects did not make use of the case texts.

[71] The Supreme Court Database, http://scdb.wustl.edu/ Accessed 7/6/2020.

model (so-called *extremely random trees*) to cases in the SCDB.[71] The model correctly forecast 70% of case outcomes and 71% of Justice-level vote outcomes over a sixty-year period. Cases are represented in terms of specially designed features from the dataset according to a codebook as well as aggregate features representing "trends". Textual features are not included.[72]

One learning objective was to give students some initial practice in using Python code to query the Supreme Court Database for information about the justices and certain decisions. Some initial exercises for students to perform included: Retrieve all votes where J. Scalia cast a vote coded as liberal AND the whole Court's decision was coded as conservative AND J. Scalia dissented AND drafted a separate opinion. Interestingly, that really did happen! In fact, there were 64 such cases in our release of the SCDB. Then students were asked to pick one of those cases, use the citation to retrieve the full text of the opinion, read J. Scalia's dissent and consider what information was missing from the dataset that would be necessary to capture the dissent's reasoning.

Some exercises in sensemaking about justices' votes focused on six cases for which students were asked to use Python code to identify information such as the decision date, the basis for the Supreme Court's jurisdiction, the types of legal issues, the legal sources cited, the lower court's outcome, the Supreme Court's decision, and the division of votes of the Court.

For the programming, students needed to gain some facility with the numerically coded features of the SCDB,[73] focusing on the justice-centered dataset at the issue level.[74] To that end, the exercises allowed the students to practice translating back and forth between legally meaningful statements and structured queries in the database format. This would then enable them to move on to the more technically demanding part: selecting parts of the database and adapting the numerical encoding toward training vote prediction models.

As the last increment, Grabmair conducted an in-class workshop on Supreme Court prediction demonstrating code he had written to partially reproduce the results of Katz, et al. (2017) based in part on their code.[75] The first step was to select the parts of the database that should be used as features for the prediction. This is important as the vote direction is coded in multiple variables of the numerical vector (i.e., the row representing an individual vote) in the database. The second was to add to the dataset some additional features that Katz, et al. had included in their model, such as the political parties of the president who had nominated each justice, circuit court information, and aggregate trend information regarding the justice's and court's probability of voting one way or the other. An important constraint here is to compute these aggregates based only on the votes of prior years. This ensures that the model does not have access to temporally implausible information during training and testing. The third step is then to train random forest prediction models using the scikit-learn Python library, similar to the original work by Katz, et al., and evaluate them. This includes comparing the prediction accuracy of models over the years in the dataset using different sets of features. One can also use built-in functions to assess each feature's contribution to the overall prediction. The trained model came very close to the performance reported in Katz, et al.'s paper. One interesting observation was that, while the aggregate trend features contributed most to the prediction of the

[72] The case representation includes features such as case-origin circuit, lower-court disposition, issue and issue area, the background of the justices and Court at that time, and historical trends. Background information includes justice, justice gender, Segal-Cover score, and party of appointing president. Trends include overall-historic Supreme Court, lower-court trends, current Supreme Court trends, individual Supreme Court justice, and differences in trends. The Segal-Cover score measures a justice's "perceived qualifications and ideology" based on expert analysis of newspaper editorials prior to confirmation. The behavioral trends and trend differences are human-engineered features. They track "the ideological direction" of individual and overall justice voting behavior. Differences in these trends "include general and issue[-] specific differences between individual justices and the balance of the Court as well as ideological differences between the Supreme Court and lower courts." (Katz, et al. 2017, pp. 6, 7, 14)

[73] See the coding handbook here: http://scdb.wustl.edu/documentation.php?s=1 Accessed 27/11/2020.

[74] The particular data of interest included, e.g., the case name, petitioner, respondent, petitioner's state, case court of origin, lower court's disposition of the case, the direction of the lower court's disposition of the case, the issue and issue area, the direction of the Supreme Court's decision, the authority cited for the decision, the type of law, the vote, and the direction of each justice's vote.

[75] MJBOMMAR / SCOTUS-PREDICT-V2, https://github.com/mjbommar/scotus-predict-v2/blob/master/src/legacy_model.py Accessed 7/7/2020

full model, removing them reduced performance only by a very small margin, and the model was able to leverage an almost as strong prediction signal from other features.

As mentioned, the original plan was to assign some of the SCOTUS prediction model training and evaluation in a homework assignment. While the students were able to query the database and analyze the retrieved results, the instructors had the impression that the class would benefit more from a guided exercise. While the Python notebook was almost completely prepared and the actual amount of code to be written was not substantial, we saw that many students in mid-February 2019 had not yet gained enough ability to see the analytical method "behind the code." Additionally, in preparing the assignment, we observed that the benefit in prediction performance gained from adding the non-standard features was not large enough to produce a satisfying experience suitable for a homework exercise.

6.2 FAIRNESS IN MACHINE LEARNING

This programming assignment involved training logistic regression models on bail recidivism prediction data, thereby focusing on a contemporary question of fairness in machine learning, an issue and debate affecting many aspects of AI. The most plausible angle for this course was the line of writing, work, and research initiated by ProPublica's well-known 2016 survey (Angwin, et al. 2016) on the workings of the COMPAS system for predicting a defendant's risk of committing another crime if released on bail. The inquiry found that the system's prediction had a higher false positive rate for African American defendants than for Caucasian ones (i.e., it falsely scored them as higher re-offense risks than Caucasians). A debate ensued between ProPublica and COMPAS's developer Northpointe[76] at the core of which was the question of how fairness can be defined quantitatively. While COMPAS itself is proprietary software, the dataset used by ProPublica was made publicly available (Larson et al. 2016) and greatly facilitated technical research around algorithmic fairness and its application in machine learning problems.

For purposes of this assignment, the features available for prediction were limited to a subset of those originally employed,[77] and the files needed to complete the homework were distributed to the students. Various definitions of fairness were made available in the posted slides from the lecture on fairness in machine learning.

The deliverables for this homework included the Python notebook with three completed tasks and a brief report with textual observations on specified topics. The tasks included surveying the data, predicting recidivism with non-blind and race-blinded models, and training a fair model. The data survey involved plotting histograms (using a Pandas plotting tool) of the COMPAS risk score data for the whole population and separately for Caucasians and African Americans.

The task of predicting recidivism comprised two subtasks: creating a logistic regression model that is not race blind and one that is blinded and comparing the fairness of their predictions on the whole population and for Caucasians and African Americans separately. Students began by training a logistic regression model[78] on all the data and assessing its accuracy. This was a non-blinded race-specific model. Then they applied the model separately for Caucasians and African Americans and calculated the true negative and false positive rates for each group, compared the fairness of the classifier across the groups, and considered which fairness criteria were satisfied: demographic parity, equalized odds, or predictive parity. They then trained a "blind" recidivism predictor using another logistic regression model and repeated the race-specific performance assessment to determine if the fairness changed.

The third task involved using Microsoft's Fairlearn library[79] to train a classification model whose behavior

[76] A summary may be found in (Corbett-Davies et al. 2016).

[77] That is, the defendant's gender, the age category of the defendant, the defendant's race, the number of juvenile felonies committed by the defendant, the number of prior convictions for the defendant, the category of crime with which the defendant is charged (misdemeanor, felony, or other), the risk score COMPAS assigned to the defendant (low/medium/high), the defendant's risk score on a scale of one to ten, and whether the defendant reoffended within two years.

[78] As noted, logistic regression is a statistical learning algorithm that predicts the odds of being an instance of a category based on the values of independent variables (predictors). It employs an iterative statistical procedure to estimate weights for the predictors.

[79] The assignment used an early version of the "Fairlearn" library: https://github.com/fairlearn/fairlearn.

ISSN: 1839-4183

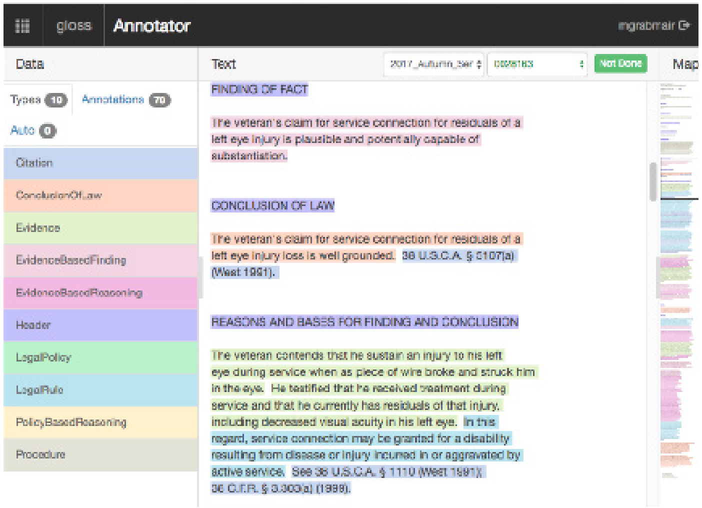
satisfies a desired fairness constraint at the cost of a possible reduction in accuracy. In the assigned exercise, the students forced the classifier to adhere to the equalized odds fairness policy, ensuring fairness to non-re-offending defendants as initially advocated by ProPublica. They computed the model's performance on the whole data set and on the Caucasian and Afro-American subsets and compared the results for fairness (i.e., demographic parity, equalized odds, and predictive parity) as well as accuracy across the groups and the tradeoffs between fairness and accuracy. This advanced exercise was optional for law students but mandatory for students from technical backgrounds. While bail recidivism prediction is an intuitive and legally very relevant task, we found that some of the students struggled with the mathematical nature of the programming exercise.

6.3 TEXT ANNOTATION ACTIVITIES

The third activity included a text annotation homework assignment and text analysis programming exercise extending over a span of three weeks. Students annotated sentences in the full opinions of Board of Veteran's Appeals (BVA) cases according to their legal rhetorical roles using an online annotation environment, Gloss, to classify the sentences.[80]

A sample use case motivated the annotation exercise. Students were asked to assume that the BVA seeks ways to search their decisions more effectively. It would like to build an Alexa-like question answering system that can talk about BVA cases. As a first pilot study, the dialog system should receive a question about a case, and answer based on one or more sentences from the decision, like the luimasearch function of the legal reasoning homework described above. Staff attorneys would like to ask the system about the facts of the case, how the evidence was assessed legally, and the case outcome. The system should also be able to answer abstract questions about Veteran's law from the text of available decisions.

As a first step, such a system would need to classify sentences in BVA decisions according to their rhetorical roles in the full opinion, and this requires annotating a training set of sentences, as positive and negative instances of those roles. The instructor (Grabmair) conducted the class sessions as an annotation workshop in which the students tried out an annotation scheme, wrote instructions to guide the annotation, and annotated the cases. The relevant annotation types (i.e., the type system) comprised ten roles that sentences play in legal decisions: citation, legal rule, evidence-based finding, and others.[81]



FINDING OF FACT

The veteran's claim for service connection for residuals of a left eye injury is plausible and potentially capable of substantiation.

CONCLUSION OF LAW

The veteran's claim for service connection for residuals of a left eye injury loss is well grounded. 38 U.S.C.A. § 5107(a) (West 1991).

REASONS AND BASES FOR FINDING AND CONCLUSION

The veteran contends that he sustain an injury to his left eye during service when as piece of wire broke and struck him in the eye. He testified that he received treatment during service and that he currently has residuals of that injury, including decreased visual acuity in his left eye. In this regard, service connection may be granted for a disability resulting from disease or injury incurred in or aggravated by active service. See 38 U.S.C.A. § 1110 (West 1991); 38 C.F.R. § 3.303(a) (1998).

FIGURE 3. Gloss annotation environment with legal rhetorical role types

[80] As noted above, annotation means marking up portions of text that are positive instances of a concept or "type" of interest, which markups can then be used in annotating higher-level, more composite types.

[81] The ten types and accompanying short explanations based on the LLT Lab BVA Type Explanations (https://github.com/LLTLab/VetClaims) include:
Citation: Reference legal authorities or other materials in standard notation.
Legal-Rule: Statement of legal rule(s) in the abstract, without application to current case.
Evidence: Statements of facts/evidence in the case, without legal assessment.
Evidence-Based Finding: Statement of authoritative findings, conclusion or determination of whether evidence satisfies a legal rule/standard.
Evidence-Based Reasoning: Statements of reasoning, based on the evidence, in making the findings of fact.
Legal Policy: Statement of abstract legal policies, principles or objectives without application to case facts.
Policy-Based Reasoning: Statement applying legal policies to decide legal issues in given case, e.g., to decide whether to adopt or reject a legal rule.
Conclusion of Law: Statement of ruling or holding about legal outcome of in case.
Procedure: Statements of procedural facts and formalities in case.
Header: Surface annotations for section headers.

The students had access to the Gloss annotation tool, a convenient web-based annotation interface developed by Jaromir Savelka. They employed it to annotate two BVA cases, in one of which relief was granted and in the other it was denied. As shown in Figure 3, a type system bar on the left side shows the 10 annotation types each with its own highlighting color. The text of the decision is shown in the center field. Students annotate a sentence by marking it with the cursor, selecting a type from a pop-up menu, whereupon the sentence is highlighted in the color corresponding to the type. In the subsequent class session, the instructor and students discussed the extent of the inter-annotator agreement with respect to their annotations of the two cases. They compared notes about any difficulties encountered and the need for clarifying the annotation instructions or revising the type system. By the end of the activity, a total of 16 cases had been annotated by the class.

These formed the dataset that would be used to develop and test the classifier models. This process involved students in examining the cases included in the collection, loading each case's sentences, putting it through preprocessing and "vectorizing" it, and finally training a classifier, examining its performance, and analyzing any errors. The features representing the sentence texts were its weighted major words minus the stop words, common small words that are filtered out. In the second step, students and the instructor transformed both training and test data into TFIDF vectors.[82] The last steps involved training decision tree classifiers for the sentence labels. Decision trees are particularly suitable for this type of exercise because they can be easily visualized, and students can assemble a series of decisions the model makes in assigning a sentence a type label based on the words it contains. Throughout the programming parts of the exercise, students used functionality provided by the scikit-learn Python library.

The instructor and students began by examining the highest TFIDF features of an example sentence, the highest TFIDF scoring words on the training dataset as a whole and per sentence type, and the performance of the classifier on the training and test set. Finally, they plotted a confusion matrix for the training data. The students examined mis-predicted sentences, focusing on the TFIDF features via the index of misclassified sentences. This kind of error analysis can lead to greater insights about the learned model and how to improve it.

Clearly, these activities in the spring 2019 course are worthwhile and would expand students' skills and knowledge in ways very relevant to the course. The challenge, quite simply, is finding the time to include them in a semester's activities.

7. STUDENTS' COMMENTS AND INSTRUCTORS LESSONS LEARNED

If the course was a learning experience for the students, it is also an opportunity for the instructors to learn. In this section, we report some of the lessons the instructors learned from this second experience of teaching the course, based in part on students' feedback in their final project reports and in the anonymous student evaluations.

7.1 REACTIONS TO COURSE AND INSTRUCTORS

According to the teaching surveys, students regarded the spring 2020 course in a generally favorable way. Six of eight responders in this course of ten students agreed or strongly agreed with the statement, "I would recommend this course to other students." Two were neutral and none disagreed. On a scale of strongly disagree (1) to strongly agree (5), with respect to the "instructor's overall teaching effectiveness" the median response of the eight responders regarding Prof. Savelka was 4.13 (.99 standard deviation) and of the eight responders regarding Prof. Ashley was 4.50 (.53 standard deviation).[83]

Some student comments described the course as a "Fantastic course, I learned a lot", "really well–done course

[82] TFIDF weights the terms in each document by frequency relative to a document and to the corpus. The weights are related positively to the term frequency in the document (TF) and inversely to its frequency in the whole corpus (IDF). To the extent a word appears often in a document or rarely in the corpus, the metric increases. To the extent it appears rarely in the document or frequently in the corpus, the metric decreases.

[83] For the spring 2019 version of the course, seven of ten responders in the course of fifteen agreed or strongly agreed with the statement, "I would recommend this course to other students." while two were neutral and one disagreed. On a scale of strongly disagree (1) to strongly agree (5), with respect to the "instructor's overall teaching effectiveness" the median response of the nine responders regarding Grabmair was 4.11 (.78 standard deviation) and of the ten responders regarding Ashley was 3.80 (.79 standard deviation).

with fascinating content," and a "Fun course!" Students commented that both instructors were "knowledgeable, passionate, and active in the field" and "presented relevant research on combining AI and Law". On the other hand, a comment complained, "We spend too much time on introducing entry level knowledges on two field[s]: computer science for law students, and law for computer students…."

The student reactions to the course reflected their different backgrounds. Understandably, comments apparently from law students expressed some frustration with the technical aspects of the course. "Honestly, this is not a class for most law students, perhaps pre–requ[isite] knowledge[] on computer should be included in course description before we choose course."[84] "I realize there may be no good solution to this problem, but I found the learning curve to be a bit steep. While I was able to understand the basics and I was able to complete the homeworks I still struggle to grasp many of the more technical concepts we discussed." Another comment, perhaps from an engineering student observing the law students, stated, "Much of the code and engineering aspect of the course I felt could be taught in a way that did not leave the law students as behind as they were. Frankly, I'm not entirely convinced that any understood what they were coding or what they were doing."[85]

Overall, comments apparently from engineering students were enthusiastic about the course, although they expressed some frustration with what they perceived as missing technologies: "I deeply appreciate your instruction and am glad that I took this course. Even as a non–law student, it has benefited my understanding of the field and the ML/DL textual tools that apply. I will take away questions and concepts beyond the scope of the class, your field, or my own." A commenter recommended that "it may be prudent to narrow the actual content down … to a few case studies and algorithms rather than the breadth of the subject. Depth is particularly important in a topic like ML, and much of the actual meaning and understanding is lost by going too quickly over the statistics and modeling aspect. Even very basic examples using pytorch or tensorflow in google colab with image recognition would be very helpful I think."

7.2 COVID-19 RESPONSE

Unfortunately, no description of a course in spring 2020 would be complete without mention of the response to the covid-19 crisis. As noted, beginning with the week of March 16, the University of Pittsburgh shut down to avoid the effects of the virus. All classes were cancelled that week to enable instructors to prepare for online instruction in the following weeks. Upon resuming classes, all lectures in the course were delivered asynchronously. Prof. Ashley recorded his lectures via Panopto; Prof. Savelka used Zoom. Each Wednesday, Prof. Savelka held a twenty-minute session with each team to review their progress on the final projects. Prof. Ashley divided each day's lecture into three or four pre-recorded sessions of between fifteen and twenty minutes each.

Students commented as follows[86]: "The remote lectures were a little slow, I will confess to listening to them at 2x speed, but they were clear and thorough." "I personally struggle learning remotely but the lessons were helpful and the best of a bad situation. My group work became very difficult once we switched to remote and the project no doubt suffered."

Following recommendations of the administration to instructors new to pre-recording lectures, Prof. Ashley embedded a small number of discussion questions in the recorded lectures and solicited students to submit brief written responses. This seemed like an effective way to gauge how well students were following the lectures. After the first week of pre-recorded lectures, however, an engineering student complained about the additional

[84] In spring 2019, some student commenters complained that the instructors had made "too many assumptions of what our prior knowledge is." "It appears like we needed to have recently taken classes in philosophy, statistics, and computing programming to understand much of what is discussed in this course." Indeed, that is one of the interesting aspects of AI and Law, that it brings all of these disciplines and law together!

[85] See Section *7.7 Planned Revisions re Teaching Programming Skills to Law Students.*

[86] The University-provided form for student comments included the following question: "What do you think the University should know about your experience as a student in the current remote learning situation?"

time required to answer the discussion questions.[87] Since the students were also writing the one-page abstracts of assigned readings, the instructors decided to forego including discussion questions.

Of course, the question remains how best to encourage student engagement with pre-recorded lectures. Recently, we learned that the continuing pandemic would require teaching the spring 2021 version of the course entirely remotely. In teaching another course remotely, Ashley has distributed discussion questions to students prior to publishing the pre-recorded lecture. Then, in a synchronous Zoom discussion session scheduled for the regular class time, the students and he discuss the answers to those questions. The total duration of the pre-recorded lecture and the Zoom discussion session is no longer than the scheduled duration of the class. Students actively participated in these discussion sessions.

7.3 REACTIONS TO READINGS AND ABSTRACTS

The instructors noted with interest that that there were fewer comments about the readings and one-page abstracts than in spring 2019. Only two comments mentioned the readings or abstracts in spring 2020. Prof. Ashley "assigned really interesting articles to read which always related to class lectures and got us thinking more about the implications of technology as it applies to the legal sector." "I also enjoyed doing the reading abstracts. It changed how I read the articles and I found it deepened my understanding of them." The abstracts were more controversial in spring 2019.[88]

This change may reflect that in the 2020 course, the instructors assigned fewer abstracts and more conspicuously integrated their responses to the students' abstracts in the lectures. We again found the students' abstracts to be quite perceptive and very useful. Beside preparing students to discuss the research papers in class, the abstracts provided the instructors with an indication of how well students understood the material. Given that the readings provided examples of how text analytic programs have been employed in experimentation, also a focus of the course, the readings and the abstracts continue to seem very important.

Regarding the materials on AI and Law, as noted, in spring 2020, the instructors employed chapters from Ashley's *Artificial Intelligence and Legal Analytics* (2017) as assigned readings for seven sessions.[89] According to the comments, Prof. Ashley helped students learn about "realistic possibilities of integrating machine learning in the legal field and how it can [be] useful in building a case," and "clear and thorough applications to the law in an accessible manner for those of other majors who we[re] a bit green on the subject." Two comments said he did "an excellent job introducing legal concepts to students who had no former experience with the law," and was "very knowledgeable about the history of the subject and … very forthcoming [about] the good and bad areas of research and how they were being addressed."

The focus on historical developments in AI and Law did not please everyone, however. A commenter complained, "some of the lectures and content felt a bit dated. Specifically, the modeling lectures were perhaps state of the art in the legal research, but frankly, engineering disciplines have moved past many of the modes of modeling described in the course. Specifically, logic models and decision models are rarely used anymore. While they are indeed helpful for subject understanding, I felt that they were taught in a way that might suggest that they would be valid for research today, which may or may not be the case."

Of course, Prof. Ashley did intend to suggest that these former approaches in AI and Law might still have relevance today. In fact, rule-based decision and process models are a very relevant topic in the discussion around the role

[87] An anonymous comment also noted, "There is a great tendency to overload students when they go remote, and the university should offer at least some high–level guidelines as to how to adjust the course load to compensate."

[88] In spring 2019 students expressed conflicting opinions about the one-page abstracts. One student found them "very helpful for us to learn the material effectively." Other students complained that they were too frequently assigned and too much work. "The abstracts are far too much busy work for law students." Another commenter recommended that the instructors give the students more feedback on the abstracts or make them optional for extra credit. In that way, it would free up time for code-based assignments.

[89] In spring 2019 chapters from this book were assigned readings in one session and suggested readings in 3 other sessions. One student in spring 2019 complained that "the book is terribly dense and is not helpful for beginners."

of modern technology in legal practice.[90] Indeed, that is a major theme of the chapters assigned from the book. To the extent text analytics provide the means to extract information from legal texts that can enable AI and Law computational models of legal reasoning to connect directly to cases and statutes, those models may provide a basis for the kinds of explanations that supplement machine learning's answers and predictions still lack.

7.4 REACTIONS TO PROGRAMMING INSTRUCTION AND ASSIGNMENTS

If there was a point of consensus in student comments from spring 2019, it was the recommendation to increase the amount of class time spent on learning programming. For example, commenters declared that, "It is okay to make a few classes a Python class; we need it," or that, "it is not a bad idea for law students to take a few more lessons about Python." Students perceived learning programming as the spring 2019 course's most difficult challenge. While one commenter declared that "the assignment is helpful for us to learn Python," another noted that, "The Python assignment is extremely hard for me." "Without enough foundations [in Python programming], it is really hard for us to apply our knowledge into real life practices."

Students in spring 2019 suggested ways to divide the programming activities into smaller, more learnable chunks, for example by spending one class session per week "as a working day manipulating pre–filled Python notebooks to understand what the code is doing, what the limitations are, and how it can be changed." Another declared, "To learn computer code, one must experience it."

> Small, digestible code–based assignments. Parsing sentences. Counting words. Writing small computer functions. Watching [the instructor] do the same. These things all contribute to the "experienced based" learning model.

One student recommended "screening law school class participants earlier to determine the level of python expertise in the room." Others suggested "having a tech student teach the law student how to use Jupyter, anaconda, etc. [referring to Python programming environments]" and noted that it would be beneficial to have more tech students in the course. As a quid pro quo, the law students could teach tech students "how to read a case within the first two weeks." Alternatively, a teaching assistant with technical expertise could "present in class and … help students facing technical difficulties by setting up Jupyter notebook etc."[91]

As described in Section 4, Part I, in spring 2020, we continued to use Python programming environments and homework assignments to teach skills. The classroom sessions were more focused on leading the students step-by-step through programming exercises with a few corpora relevant to the projects. In this way, we expected that the skills for much of the data pre-processing and feature engineering associated with the final projects, could be accomplished during the semester. Thus, students would better learn the skills and knowledge associated with data wrangling and feature preparation and devote the final weeks in the semester to model building, experimentation, error analysis, and improving results.

Regarding the instruction in computer programming, commenters credited Prof. Savelka with helping students to learn "basic coding skills that can be used in a larger program to achieve helpful machine learning tools for legal work," and doing "an excellent job teaching computer science to students who had little to no background in the subject." Others noted that he provided "really useful and helpful colab notebooks that made it easy to follow along and understand each step" and "thorough lectures on techniques with example code in accessible forms via excellent tools like Google Colab." One comment stated that he "made learning it fun, as fun as a wall of code can be."

Other comments suggested ways to improve the programming instruction. "Some more documentation to the provided code would be helpful in some cases. Some provided code, though mostly from external resources …, was not written for easy understanding by novices." "Less condensed rewrites with more comments could be helpful for those struggling with the coding portion."

[90] See, for example, DECISIONS. AUTOMATED. https://bryter.io/ Accessed 7/ 28/2020

[91] From the instructors' viewpoint in spring 2019, the main problem was that the law students did not spend sufficient time on homework assignments. It seemed that the law students simply underestimated that amount of time that learning to program requires, a reality with which tech students per force are familiar.

"Code examples displayed on the screen were sometimes difficult to see well." One comment suggested doing "more examples with colab notebooks with legal datasets."

7.5 REACTIONS TO COURSE PROJECTS

The Project Reports submitted in 2019 included several lessons learned, some of which recommended how to improve the timing and organization of the process for final projects given certain challenges.

Chief among these was a lack of sufficient time, especially to obtain and process appropriate data: "[T]ime was the biggest bottleneck with this project. With more time, … we could have modified how we selected data to better apply our results to [a] larger scope." "[E]nough time to encode a useful dataset can be crucial to make progresses in research projects." "A future version of this would benefit from students thinking about getting data for their projects earlier on."

One team suggested that:

> A future version of this course might allow students to brainstorm and select projects earlier in the semester. As [students] learned techniques in class, [they] could directly apply them to [the] project closer to when [they] learn them and with opportunities for better understanding of concepts throughout the class. While this may require the project ideas to come more from the professors, this could help with achieving a more complete project. However, it would still be feasible for students to come up with ideas based on their interests and have more guidance from the professors about the methods to create the project.

Somewhat similarly, another team recommended:

> If there was a way to introduce all of the topics with some of their use cases early in the semester, and have the project topics due…early [in the second month of the course] (with the understanding that they might evolve), this would give students a focus sooner that would help them explore data sources to use with their projects, giving more time for the actual programming and analysis. Another way to introduce what kinds of projects could be done earlier (in lieu of the topics) would be to present past student projects now that there are some. This may give a good idea of what sort of questions could be explored by the time the project really gets going.

A second challenge involved sufficient training in data preparation to make the process of preparing data from projects easier and less prone to errors.

> [A]ll projects will deal with data coming from some file and one of the biggest problems of the project was handling a Unicode Decode Error.[92] If there had been a day given to general project help, such as opening files, manipulating dictionaries, and displaying results, the entire course would have likely benefited.

As can be seen, the 2019 Project Report discussions of lessons learned focused on recommendations for improving the timing and process for identifying teams and preparing for projects.

By contrast, in the two of the three 2020 Project Reports that discussed them, the focus was on substantive lessons learned from the team's project experiments and activities. We take this as an indication that as instructors, we had learned from the lessons of 2019 and improved the process for choosing projects, teams, and data, removing some causes of frustration. Specifically, the 2019 comments led the instructors in spring 2020 to begin organizing the teams and projects in the beginning of the semester, to assemble a dedicated text corpus for three topic areas, and to suggest ideas for final projects involving these areas. From our perspective the changes that we introduced in 2020 had positive effects. Starting the projects early in the semester and providing datasets alleviated a lot of problems encountered in 2019.

This is a positive change, but the trade-off is the fact that students are defining their projects at a time when they have still not been exposed to most of the course content. Possibly this could be the reason why two of the three groups needed to change the scope of their projects toward the semester's end.

There are other lessons to be learned from 2020's project activities. One project report comment made a recommendation for future versions of the course. The Fourth Amendment team's project report stated:

[92] https://wiki.python.org/moin/UnicodeDecodeError

Assignments and in-class guidance prepared the authors well for this project. Additional opportunities for hands-on learning and comparison of different NLP techniques or various transformers would be a welcome addition. The simpletransformers library makes these complex systems much easier to train and integrate into a solution. Still, the approach of added bonus sections to assignments is appreciated to prevent students from being overloaded on tasks that may not be entirely relevant to their project work.

In addition, an anonymous student comments: "… the law students do not have a background in the machine learning…. My group primarily rested on technological skills of an undergrad and our limited skills which I learned for the first time this semester. This made the project very hard and compounded internal group frustrations."

7.6 PLANNED REVISIONS RE COURSE PROJECTS

We will try to mitigate these effects in spring 2021.

First, as mentioned earlier, two of the three teams had to change the project's scope towards the end of the semester. This is not necessarily a bad thing. We noticed, however, that students invested a lot of time and effort in activities that led to realizing that the path they had chosen was likely not leading to success. While we think it is important that students have ownership of their projects, in 2021 we would like to help them come to such a realization much earlier. Ideally, we hope to ensure that the bulk of time and effort would be spent on the substance of the actual final project. We plan to provide students with more assistance and feedback during the early stages of the project to overcome the fact that they need to define their projects before they have been exposed to much of the course content.

Second, as noted, despite our effort to create well-rounded teams we detected problems with one of the groups where a bit more technical proficiency would have been very beneficial. In 2021, we will consider deciding about membership in the teams ourselves instead of simply setting the ground rules. We will have to develop an effective way to assess students' backgrounds with respect to their potential contributions toward the final projects.

7.7 PLANNED REVISIONS RE TEACHING PROGRAMMING SKILLS TO LAW STUDENTS

Third, there is a lingering concern on how best to teach skills of programming and machine learning to prepare law students to participate actively and successfully in projects. While we have instituted more step-by-step instruction in the in-class and homework programming exercises, we still need to engage the law students more actively in the programming instruction and in programming aspects of the projects.

We have observed that it is more challenging for law students with little technical background to reach a productive level of programming for the course than it is for tech students to learn enough about law, legal reasoning and practice. Indeed, we may have been too accommodating to engineering students, while inadvertently leaving some law students behind. It may be a question of rebalancing.

One way to rebalance may be to design different homework for each group, with more incremental introductory exercises for the non-technical students. Another is to take advantage of the technical talent in the classroom. We will try to engage the tech students (e.g., the engineering students in spring 2020) to more actively provide feedback and assistance to law students about the programming exercises. If the tech students get into the habit of helping the law students with respect to programming exercises, they may more naturally engage the law students in programming aspects of the final projects. Conceivably, law students could do the same for tech students on legal exercises.

In any event, a goal of the course is to give law students an intense programming experience so that they can better understand how the programs work, how to evaluate them, and the methods of the technical personnel with whom they will need to collaborate in the future. That goal is met even if the law students do not learn to program effectively.

8. CONCLUSIONS

We hope that these modifications in the course on Applied Legal Analytics and AI will address the lessons learned from the course's first and second offerings. In these two versions of the course, we have observed law students engaging with computer code closely enough to gain an understanding of how the technologies work and how to evaluate them. We have stepped law students through programming with Python and participating in teams with technical students to work on final projects. In each of the projects, the teams implemented some Python code, designed empirical evaluations, ran experiments, obtained results, and engaged in some error analysis to see what the programs missed and why. They have learned how new text analytic tools work in legal practice and about their limitations and gained experience in communicating and working with technical personnel in applying and adapting the technology, a phenomenon which, we expect, will become increasingly central to legal practice.

The experience of teaching this course has convinced us that law students can benefit from learning to design and conduct experiments with legal texts and data, and that the skills and knowledge they learn will place them at the fore of a new generation of law students in tackling the impact of technology on the practice of law.

In meeting the challenges of teaching this course, our experience demonstrates the importance of transdisciplinary collaboration at multiple levels including among instructors and students from very different backgrounds. The instructors' expertise differs but together they cover research and teaching in law, AI and Law, computer science, experimental design, and computer programming with ML and NLP. The students have diverse skills and backgrounds, coming from multiple schools and departments including law, computing and information, business, electrical and computer engineering, and undergraduate computer science at Pitt and CMU. In guiding and participating in final project teams, instructors and students necessarily collaborated with those from other disciplines outside of their domains of expertise. We also encouraged collaboration beyond academia; inviting guest speakers from law firms or legal service providers to focus the class on applications in legal practice.

The subject matter of the course is also inherently transdisciplinary, ranging beyond law, AI and Law, and computer science to include political science in statistically analyzing judicial decisions, criminology and sociology in addressing the effects of bias in machine learning, engineering design and ethics tackling issues raised by the new technologies in the areas of professional responsibility, reliance, and liability of autonomous systems, and cognitive sciences concerning such issues as explanation in machine learning and cognitive computing. It would be interesting to include academics and professionals from these diverse domains in course-related collaborations or guest lectures.

While the course focused mainly on U.S. law and legal practice and on common law legal systems, there were strong international influences. Two of the instructors are European academics trained in civil law, and one is about to take up a faculty position in Germany. About one third of the students who took either version of the course comprised international graduate students; two international law students were enrolled in Pitt's program on International & Comparative Law. Nearly half of the readings in AI and Law research were international in origin. The substantive legal issues affecting data mining and machine learning such as bias, privacy, intellectual property limitations on data, and product liability, span international boundaries, where different jurisdictions take different approaches. Although still quite different, the civil law and common law systems exhibit some convergences and present complementary opportunities for learning. We look forward to exploring how the transdisciplinary collaborations so important to this course can be extended across international boundaries, for example, through remote guest lectures, now feasible in our increasingly virtual academic world.

9. REFERNECES

1. Aletras, N., Tsarapatsanis, D., Preoţiuc-Pietro, D. and Lampos, V. 2016. "Predicting judicial decisions of the European Court of Human Rights: A natural language processing perspective." *PEERJ Computer Science* 2: e93. https://peerj.com/articles/cs-93/?utm_source=mandiner&utm_medium=link&utm_campaign=mandiner_201912 Accessed 27/11/2020
2. Angwin, J., Larson, J., Mattu, S., and Kirchner, L. 2016. "Machine Bias," *ProPublica*, 23 May. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing Accessed 7/8/2019

3. Ashley, K. 2017. *Artificial Intelligence and Legal Analytics. New Tools for Law Practice in the Digital Age*. Cambridge, UK: Cambridge University Press.
4. Ashley, K. 2019. "Automatically Extracting Meaning from Legal Texts: Opportunities and Challenges." *Ga. St. U. L. Rev.* 35: 1117-1151.
5. Ashley, K. and Walker, V. 2013. "From Information Retrieval (IR) to Argument Retrieval (AR) for Legal Cases: Report on a Baseline Study." In K. Ashley (ed.), *26th Int'l Conf. on Legal Knowledge and Information Systems.* Jurix-2013. Amsterdam: IOS Press pp. 29-38.
6. Bennett, Z., Russell-Rose, T., and Farmer, K. 2017. "A scalable approach to legal question answering.", In *Proceedings ICAIL-17*. New York: ACM, pp. 269-270.
7. Berman, D. and Hafner, C. 1986. "Obstacles to the Development of Logic-Based Models of Legal Reasoning." In C. Walter (ed.) *Computer Power and Legal Language*. Santa Barbara: Praeger. pp. 183-214.
8. Bhattacharya, P., Paul, S., Ghosh, K., Ghosh, S., and Wyner, A. 2019. "Identification of Rhetorical Roles of Sentences in Indian Legal Judgments." In M. Araszkiewicz and V. Rodríguez-Doncel (ed.), *32d Int'l Conf. on Legal Knowledge and Information Systems*, Jurix-19. Amsterdam: IOS Press. pp. 3-12.
9. Bishop, C. 2006. *Pattern Recognition and Machine Learning*. New York: Springer.
10. Branting, K. 2017. "Data-centric and logic-based models for automated legal problem solving." *Artificial Intelligence and Law*, 25 (1): 5-27.
11. Brostoff, T. and Sinsheimer, A. 2013. *United States Legal Language and Culture: An Introduction to the US Common Law System.* Oxford, UK: Oxford University Press.
12. Cardellino, C., Teruel, M., Alemany, L., and Villata, S. 2017. "A low-cost, high-coverage legal named entity recognizer, classifier and linker." In *Proceedings ICAIL-17*. New York: ACM, pp. 9–18.
13. Chalkidis, I., Androutsopoulos, I., and Aletras, N. 2019. *Neural Legal Judgement Prediction in English,* Athens University of Economics and Business, https://arxiv.org/pdf/1906.02059 Accessed 26/11/2020
14. Conrad, J., and Al-Kofahi, K. *Scenario analytics: Analyzing jury verdicts to evaluate legal case outcomes.* 2017 In *Proceedings ICAIL-17*. New York: ACM, pp. 29-37.
15. Contreras, A. and McGrath, J. 2020. "Law, Technology, and Pedagogy: Teaching Coding to Build a "Future-Proof" Lawyer." *Minn. J.L. Sci. & Tech.* 21: 2 297-332.
16. Corbett-Davies, S., Pierson, E., Feller, A., and Goel, S. 2016. "A computer program used for bail and sentencing decisions was labeled biased against blacks. It's actually not that clear." *The Washington Post, Monkey Cage*. Oct. 17. https://www.washingtonpost.com/news/monkey-cage/wp/2016/10/17/can-an-algorithm-be-racist-our-analysis-is-more-cautious-than-propublicas/?noredirect=on Accessed 7/7/2020.
17. Council, J. 2019. "Top Law Schools Add AI Courses." *WSJ PRO Artificial Intelligence*. https://www.wsj.com/articles/top-law-schools-add-ai-courses-11555925401 Accessed 3/10/2019
18. Crichton, D. 2015. "With Judge Analytics, Ravel Law Starts to Judge the Judges." TechCrunch. April 16. https://techcrunch.com/2015/04/16/who-judges-the-judges/ Accessed 26/11/2020
19. Dalton, B. "Cognifying Legal Education." 2019. *Above the Law: Law 2020.* https://abovethelaw.com/law2020/cognifying-legal-education/ Accessed 2/10/2019
20. Devlin, J., Chang, M., Lee, K., and Toutanova, K. 2018. "Bert: Pre-training of deep bidirectional transformers for language understanding." *ARXIV Preprint* arXiv:1810.04805.
21. Domingos, P. 2012. "A few useful things to know about machine learning." *Communications of the ACM* 55: (10) 78-87.
22. Eicks, J. 2012. "Educating Superior Legal Professionals: Successful Modern Curricula Join Law and Technology." In O. Goodenough and M. Lauritsen (eds.), *Educating the Digital Lawyer* 12-1: 5-1 – 5-14, https://www.academia.edu/9202158/Educating_the_Digital_Lawyer Accessed 8/1/2020
23. Federal Automated Vehicles Policy. 2016. *Accelerating the Next Revolution in Roadway Safety,* NHTSA, US Dept. Transportation. (https://www.transportation.gov/AV/federal-automated-vehicles-policy-september-2016) Accessed 26/11/2020, pp. 5-14, 17-19.
24. Fenwick, M., Kaal, W., and Vermeulen, E. 2018 "Legal Education in a Digital Age: Why 'Coding for Lawyers' Matters." Lex Research Topics in Corporate Law & Economics Working Paper No, 2018-4, U of St. Thomas (Minnesota) Legal Studies Research Paper No. 18-21. Pp. 0-31 SSRN: https://ssrn.com/abstract=3227967 or http://dx.doi.org/10.2139/ssrn.3227967 Accessed 26/11/2020
25. Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A., Lally, A., Murdock, J., Nyberg, E., Prager, J.,

Schlaefer, N. and Welty, C. 2010. "Building Watson: An Overview of the DeepQA Project." *AI Magazine*, Fall, 31: 59-79.

26. Grabmair, M., Ashley, K., Chen, R., Sureshkumar, P., Wang, C., Nyberg, E., and Walker, V. 2015. "Introducing LUIMA: an experiment in legal conceptual retrieval of vaccine injury decisions using a UIMA type system and tools." In *Proceedings ICAIL-15*. New York: ACM. pp. 69-78.
27. Gretok, E., Langerman, D. and Oliver, W. 2020. "Transformers for Classifying Fourth Amendment Elements and Factors Tests." *33d Int'l Conf. on Legal Knowledge and Information Systems,* Jurix-2020. Amsterdam: IOS Press pp. 63-72.
28. Halevy, A., Norvig, P., and Pereira, F. 2009. "The unreasonable effectiveness of data." *IEEE Intelligent Systems,* 24 (2): 8-12.
29. Halterman. R. 2018 *Fundamentals of Python Programming*. Southern Adventist University (2018). https://archive.org/details/2018Fundamentals.ofPython. Accessed 27/11/2020.
30. Haselager, P. 2019. "Mediated action and the risk of entrapment", invited speech at ICAIL-2019, Montreal, June 18.
31. Hudgins, V. 2020. "Casetext Launches New Brief-Writing Automation Platform Compose." *LegalTech News.* Feb. 25. https://www.law.com/legaltechnews/2020/02/25/casetext-launches-new-brief-writing-automation-platform-compose/ Accessed 26/11/2020
32. Katz, D., Bommarito, I., and Blackman, J. 2017. "A general approach predicting the behavior of the Supreme Court of the United States." *PLOS One*. April 12. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0174698 Accessed 27/11/2020
33. Kohavi, R., and Provost, F. 1998. "Glossary of Terms." *Machine Learning,* 30 (2-3): 271–274.
34. Larson, J., Mattu, S., Kirchner, L., and Angwin, J. 2016. "How We Analyzed the COMPAS Recidivism Algorithm." *ProPublica* May 23. https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm Accessed 7/7/2020.
35. Lauderdale, B., and Clark, T. 2012. "The Supreme Court's many median justices." *American Political Science Review,* 106 (4): 847-866.
36. Linna, Jr., D. 2018. "Training Lawyers to Assess Artificial Intelligence and Computational Technologies." *LegalTech Lever* 1 https://www.legaltechlever.com/2018/09/training-lawyers-assess-artificial-intelligence-computational-technologies/ Accessed 3/10/2019.
37. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. 2019. "Roberta: A robustly optimized BERT pretraining approach." *ARXIV Preprint* arXiv:1907.11692.
38. Medvedeva, M., Vols, M., and Wieling, M. 2020. "Using machine learning to predict decisions of the European Court of Human Rights." *Artificial Intelligence and Law,* 28(2): 237-266.
39. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. 2013. *Distributed representations of words and phrases and their compositionality.* In *Advances in Neural Information Processing Systems*, 3111-3119.
40. Miller, S. 2019. "Artificial Intelligence and Law." *Colorado Law*, https://www.colorado.edu/law/2019/05/03/artificial-intelligence-and-law Accessed 3/10/2019
41. Mochales, R. and Moens, M. 2011. "Argumentation mining." *Artificial Intelligence and Law*, 19(1), 1–22.
42. Murphy, M. and Pearce, R. "Algorithms, Ethics, and Legal Services: How Artificial Intelligence Will Disrupt Legal Ethics and Professional Responsibility." Unpublished manuscript on file with author.
43. Nooteboom, L. 2017. "Child-Friendly Autonomous Vehicles; Designing Autonomy with all road users in mind." In *Developing Human Interactions with Autonomous Systems*. Nov. 13. https://medium.com/@HumanisingAutonomy/child-friendly-autonomous-vehicles-2880ca74165f Accessed 29/3/2020.
44. Norvig, P. 2007. "How to Write a Spelling Corrector." https://norvig.com/spell-correct.html Accessed 7/7/2020.
45. O'Connor, N. 2018. "Reforming the U.S. Approach to Data Protection and Privacy." Council of Foreign Relations, Digital and Cyberspace Policy Program. https://www.cfr.org/report/reforming-us-approach-data-protection Accessed 26/11/2020
46. O'Grady, J. 2018. "Suffolk Law School: Leading Transformation of Legal Education." *Practice Innovations 14.* http://static.legalsolutions.thomsonreuters.com/static/images/newsletters/pracinno/Mar18_PracticeInnovations.pdf Accessed 3/10/ 2019
47. Pavlus, J. 2019. "Machines Beat Humans on a Reading Test. But Do They Understand?" *Quanta Magazine*. Oct. 17. https://www.quantamagazine.org/machines-beat-humans-on-a-reading-test-but-do-they-understand-20191017/ Accessed 7/7/2020.
48. Pennington, J., Socher, R., and Manning, C. 2014. "Glove: Global vectors for word representation." In *Proceedings of the 2014 Conf. on Empirical Methods in Natural Language Processing*. EMNLP. pp. 1532-1543.
49. Perlman, A. 2017. "Reflections on the Future of Legal Services." Suffolk University Law School Research Paper

ISSN: 1839-4183

No. 17-10. https://ssrn.com/abstract=2965592 Accessed 3/10/2019. pp. 1-11.

50. Pivovarov, V. 2019. "Future Law School. What Does It Look Like?" *Forbes* 5 https://www.forbes.com/sites/valentin-pivovarov/2019/02/12/futurelawschool/#67a0dc2f6a84 Accessed 3/10/2019

51. Reed, C., Kennedy, E., and Silva, S. 2016. "Responsibility, Autonomy and Accountability: Legal Liability for Machine Learning." Queen Mary School of Law Legal Studies Research Paper No. 243/2016: 1-17, 26-31. October 17. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2853462 Accessed 26/11/2020

52. Reid, M. 2018. "A Call to Arms: Why and How Lawyers and Law Schools Should Embrace Artificial Intelligence." *U. Tol. L. Rev.* 50: 477-489.

53. Saravanan, M. and Ravindran, B. 2010. "Identification of rhetorical roles for segmentation and summarization of a legal judgment." *Artificial Intelligence and Law* 18, 1: 45–76.

54. Savelka, J. 2019. Statutory_Interpretation, https://github.com/jsavelka/statutory_interpretation Accessed 7/7/2020

55. Savelka, J. and Ashley, K. 2018. "Segmenting US Court Decisions into Functional and Issue Specific Parts." In *Proceedings of the 31st Int'l Conf. on Legal Knowledge and Information Systems.* Jurix-2018. Amsterdam: IOS Press pp. 111-120.

56. Savelka, J., Walker, V., Grabmair, M., and Ashley, K. 2017. "Sentence Boundary Detection in Adjudicatory Decisions in the United States." *Traitement Automatique des Langues*. 58: 21-45.

57. Savkar, V. 2019. "How Will Artificial Intelligence Change Law Schools? How law schools can evolve using artificial intelligence and machine learning." *Above the Law*. https://abovethelaw.com/legal-innovation-center/2019/06/20/how-will-artificial-intelligence-change-law-schools/ Accessed 3/10/2019

58. Sergot, M., Sadri, F., Kowalski, R., Kriwaczek, F., Hammond, P. and Cory, H., 1986. "The British Nationality Act as a logic program." *Communications of the ACM, 29*(5), pp. 370-386.

59. Shulayeva, O., Siddharthan, A., and Wyner, A. 2017. "Recognizing cited facts and principles in legal judgements." *Artificial Intelligence and Law* 25 1: 107–126.

60. Simon, M. Lindsay, A., Sosa, L., and Comparato, P. 2018. "*Lola v. Skadden* and the Automation of the Legal Profession." *Yale J.L. & Tech*. 20: 234-310.

61. Surdeanu, M., Nallapati, R., Gregory, G. Walker, J. and Manning, C. 2011. "Risk Analysis for Intellectual Property Litigation." In *Proceedings ICAIL-11.* p. 116-120.

62. Surden, H. 2014. "Machine Learning and Law", *Wash. L. Rev.* 89: 87-116.

63. Walker, V. 2007. "A default-logic paradigm for legal fact-finding." Jurimetrics 47: 193-244.

64. Zentgraf, D. 2015. "What Every Programmer Absolutely, Positively Needs to Know About Encodings and Character Sets to Work with Text." Kunstube, https://kunststube.net/encoding/ Accessed 7/7/2020

65. Zhang, P. and Koppaka, L. 2007. "Semantics-based legal citation network." In *Proceedings ICAIL-07*). New York: ACM. pp. 123–130.

CASE LAW

United States:

Dynamo Holdings et al. vs. Commissioner of Internal Revenue, U.S. Tax Court, Docket No. 2685-11, 8393-12; July 13, 2016.

Lola v. Skadden, Arps, Slate, Meagher & Flom LLP, 620 F. App'x 37, 620 Fed. Appx. 37 (2d Cir., 2015).