Received July 24, 2019, Date of publication: September 25, 2019, DOI: https://doi.org/10.26826/law-in-context.v36i1.88

A Brief History of the Changing Roles of Case Prediction in AI and Law

By **Kevin D. Ashley** (Professor of Law and Intelligent Systems at the University of Pittsburgh and Fellow of the American Association of Artificial Intelligence)

University of Pittsburgh, USA, **Orcid**: https://orcid.org/0000-0002-5535-0759

ABSTRACT

Predicting case outcomes has long played a role in research on Artificial Intelligence and Law. Actually, it has played several roles, from identifying borderline cases worthy of legal academic commentary, to providing some evidence of the reasonableness of computational models of case-based legal reasoning, to providing the raison d'être of such models, to accounting for statistically telling features beyond such models, to circumventing features altogether in favor of predicting outcomes directly from analyzing case texts. The use cases to which case prediction has been put have also evolved. This article briefly surveys this historical evolution of roles and uses from a mere research possibility to a fundamental tool in AI and Law's kit bag of techniques.

Keywords – *artificial intelligence and law, prediction, machine-learning, case-based reasoning*

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

Suggested citation: *Ashley, K.D. 2019. "A Brief History of the Changing Roles of Case Prediction in AI and Law." Law in Context, 36 (1): 93-112. DOI https://doi.org/10.26826/law-in-context.v36i1.88*

Summary

ISSN: 1839-4183

1. INTRODUCTION

Predicting case outcomes has long played a role in Artificial Intelligence and Law (AI and Law), a branch of computer science which involves research and development of computer systems that can intelligently solve problems in the legal domain or assist humans in solving them. Machine learning (ML) has been frequently applied to predict case outcomes. ML refers to computer programs that use statistical means to induce or "learn" models from data with which to classify a document or predict an outcome for a new case. Other techniques have also been applied, however, notably computational models of case-based legal argument.

Through much of the history of AI and Law, the dominant approach to computationally modeling legal reasoning for prediction and other tasks has been top down. Researchers have employed legal expertise to decompose legal statutes and case decisions into rules or other components that could be applied to analyze problems and, where appropriate, to predict outcomes. Legal expert systems, for example, are a top down approach: "computer applications that contain representations of knowledge and expertise which they can apply—much as human beings do—in solving problems, offering advice, and undertaking a variety of other tasks," (Susskind 2010, p. 120 f.). The legal knowledge is assembled at the beginning, that is, top down, in manually creating the system's rules for solving problems. Expert systems apply the rules to analyze problems and make predictions, and they can explain the predictions in terms of the rules they applied, a kind of logical proof that attorneys will recognize and understand. It is, however, labor intensive and expensive to create, modify, or update the rules as the law changes, a phenomenon known as the knowledge acquisition bottleneck, which has limited the development of legal expert systems.

It has long been hoped that a bottom-up approach could sidestep the knowledge acquisition bottleneck. In theory, machine learning could extract legal knowledge by automatically inducing rules or generating statistical models from data such as decided cases. The learned rules or models could then be applied to predict outcomes of new problems. As discussed below, this is already happening to some extent. The learned models and features, however, may not correspond to legal knowledge human experts would recognize. As a result, the programs often cannot explain their predictions in terms that attorneys would credit.

In this article, we provide a short history of AI and Law research on predicting case outcomes. We then focus on recent developments in legal text analytics, which employs natural language processing (NLP), machine learning and other computational techniques automatically to extract meanings (or semantics) from archives of legal case decisions, contracts, or statutes. These developments present new opportunities for predicting case outcomes and for circumventing traditional approaches to modeling legal expertise although often at the expense of an inability to explain predictions.

2. MODELING LEGAL EXPERTISE

Throughout most of the history of AI and Law, building computational models that reason with legal rules, argue with legal cases and precedents, predict legal outcomes, and explain those predictions required addressing the following core questions: what legal domain to model and for what use case, how to represent the requisite legal knowledge, which inference methods to implement, how to acquire the legal knowledge, and whether the program will learn.

The first questions involve determining which areas of the law and problems of interest are to be modeled and what is the use case to which the model will be applied. Due to the difficulties of representing legal knowledge, models generally cover relatively narrow domains such as trade secret law or landlord tenant law. Since every model is a simplification, it is important to settle on a use case, that is, an application to which the model will be put, for which the model is sufficiently detailed and accurate. Use cases that involve classifying the relevance of documents for information retrieval and ranking, such as case decisions, contract or statutory provisions, keep human users in the loop and may impose less restrictive requirements than those that purport to replace human decision makers.

A second key question is how to represent the knowledge in the legal domain. Knowledge representation techniques may take many forms: formal representations of legal rules, representations of generalized case facts such as factors, stereotypical fact patterns that strengthen or weaken a side's claim, or representations of legal document texts as term vectors. A term vector represents a document in terms of its words, citations, indexing concepts, or other features; it is an arrow from the origin to the point representing the document in a large dimensional space with one dimension corresponding to each feature in the corpus. All such representations are simplifications; choices must be made as to which simplifying assumptions to make, for instance, discretizing concepts and values or adopting a closed-world assumption that what is not known to be true is false.

Since the computational model is usually meant to perform legal reasoning, there are questions about what inference methods to implement and whether to employ generic methods, such as logical inference with rules or statistical inference based on frequencies, or more specialized techniques, for example, drawing analogies to cases. These questions include not only how to perform the inferences but also how computationally expensive the methods. One must also consider how to explain the inferences. For instance, statistical inferences may make accurate predictions but not be able to explain those predictions in terms that legal professionals would recognize. Evaluation is a further consideration; it includes how to evaluate both the predictions and the explanations.

Finally, one must determine how to acquire information with which to populate the knowledge representations for the domain and use case. What technical and domain expertise is needed to create and fill domain representations? As noted, for expert systems this required enlisting humans with legal expertise to compile rules with which the system can analyze problem scenarios. For models of case-based legal reasoning, it has required humans to read the cases and index them by concepts such as factors.

The extent to which this knowledge acquisition can be automated has long been a tantalizing question. This is where machine learning comes into play. Models can learn from legal data. We will mostly consider supervised machine learning, which involves a training step and a prediction step. In the training step, the ML algorithm takes training instances as inputs. In legal text analytics, these will likely be chunks of text such as sentences from legal cases, represented as a vector of features and a target label (e.g., a binary decision whether a classification applies.) Feature vectors are like term vectors but use additional features beside terms and term frequencies; the value for each feature is the magnitude along some dimension of the feature in a text. The model statistically "learns" the correspondence between certain language features in the sentence feature vectors and the target label. In the prediction step, given the texts of new chunks from the test set, also represented as feature vectors, the model predicts the classification to assign to the sentence, if any. The model can be evaluated objectively by comparing the learned classifications to manually assigned ones for some gold standard test set.

Today, models can learn predictive rules from classified cases, boundaries between positive and negative instances of legal concepts, weights representing the predictive power of features, and likelihoods that a text chunk answers a legal question, expresses a factor, or is a particular type of contractual provision.

As discussed below, with neural networks, machine learning tackles even the knowledge representation step, automatically identifying the kinds of features that matter. Neural networks comprise input and output nodes connected to multiple layers of intermediary nodes via weighted edges. Propagating an input to an output involves a linear combination of the weights. The goal of the network is to learn weights that minimize the deviation of the computed output with the target output. Different architectures of networks, layers and depths are suitable for different tasks. Neural networks with multiple layers can perform feature learning via their hidden layers.

In short, machine learning is turning the top-down process of modeling legal knowledge on its head, enabling bottom-up approaches to acquire the knowledge to predict outcomes, if not to offer legally intelligible explanations of those predictions. Recently, computational models are learning to select sentences with which to effectively summarize legal cases. Some researchers today

ISSN: 1839-4183

are suggesting that, given enough data, machine learning based summarization techniques can learn to generate effective summaries without the need to represent and acquire legal knowledge at all. Others ask whether there can ever be sufficient data in the legal domain for this to be realistic.

In the next section, we briefly review the history of prediction in AI and Law and the events that have led to this current juncture in the field's history.

3. BRIEF HISTORY OF PREDICTION IN AI AND LAW

In the last 45 years, AI and Law approaches to predicting case outcomes have evolved in a number of directions. It all began with predicting case outcomes using a nearest neighbor algorithm or inducing rules via decision trees from substantive features of legal cases and outcomes. Gradually, more complex models of arguing from legal cases were applied to the prediction task, models that considered increasingly more legal knowledge of substantive factual strengths and weaknesses, rule-based issues, and underlying legal values, and which could explain their predictions in terms of these arguments.

Most recently, as discussed below, the predictive features have involved less information about substantive features and more about generic issues, historical trends and the identities of litigation participants, that is, courts, judges, parties, and their representatives. Machine learning text analytic programs using neural networks are predicting outcomes from case texts and automatically identifying predictive features without recourse to traditional legal knowledge representation. Researchers are attempting to tease out from these neural networks the constituents of legal explanations. For example, Hierarchical Attention Networks yield attention weights focusing on the most predictive parts of texts with which, it is hoped, meaningful explanations can be fashioned.

While one tends to think of use cases for predicting outcomes that guide settlement decisions or strategic decision making in litigation, sentence predictiveness can play other roles. For example, as discussed below, recent efforts have employed the predictiveness of sentences, that is, their correlation with case outcomes, to help generate extractive case summaries.

3.1 NEAREST NEIGHBOR

As early as the 1970's, researchers MacKaay and Robillard (1974) created a program to predict outcomes of Canadian tax cases in which courts determined whether real estate transactions generate ordinary income or capital gains and more favorable tax rates. They represented each of 60 Canadian tax cases in terms of 46 binary fact descriptors identified by previous courts as relevant. These included characteristics of the private party that sold the real estate, circumstances surrounding the purchase and sale of the property, use of the property during ownership by the private party, the private party's intention and whether the tax appeal board had upheld the taxpayer's claim. (MacKaay and Robillard 1974, p. 327-331).

The researchers applied a k-nearest neighbor (or k-NN) algorithm, which calculates "a measure of similarity or dissimilarity between the fact patterns of cases and [predicts] the decision in a new case to be the same as that of its [k] closest neighbor[s] in terms of the ... dissimilarity measure." (MacKaay and Robillard 1974, p. 307). The metric, Hamming distance, simply sums the number of variables, that is, descriptors, for which the two cases have different values.

The authors also generated two-dimensional displays such as Figure 1 using multidimensional scaling methods based on the same distance metric (MacKaay and Robillard

FIGURE 1. *Two dimensional representation of sixty capital gains tax cases*

1974, p. 317). One observes a fairly clear boundary between the cases decided pro and con the taxpayer. The authors examined commentaries in the *Canadian Tax Journal* for the period covered by the sample cases and discovered "that decisions which appeared new or extreme at the time of deciding in the view of an expert, turn out to lie on the frontier in the diagram." (MacKaay and Robillard 1974, p. 322).

Even in this early work, a central "question was raised as to what the 'prediction methods' try to achieve: minimization of prediction errors or elucidation of human understanding." (MacKaay and Robillard 1974, p. 322). This dichotomy has dogged legal prediction ever since, and it relates to another question that is especially relevant today: how descriptors are found. The authors discussed the alternatives: one can simply list all of the low-level factual circumstances that the cases have mentioned or one can seek to identify more general concepts that cover the instances and suggest how to recognize similar features in future cases. (MacKaay and Robillard 1974, p. 322). The authors call the latter alternative "the human process" of feature identification.

At a time when machine learning programs can predict outcomes from the raw text of cases and hierarchical neural networks can identify predictive features automatically, it is still a question if and how to "follow the human process", one that draws upon legal knowledge of a regulatory domain and the underlying values it protects to identify "general concepts" that can both guide recognition of relevant similarities but also explain the resulting predictions in terms that attorneys would recognize.

TABLE 1. *Should defendant be released on bail?*

Case	Injury	Drugs	Weapon	Prior record	Result
1	none	no	no	yes	yes
2	bad	yes	yes	serious	no
3	none	no	yes	no	yes
4	bad	yes	no	yes	no
5	slight	yes	yes	yes	no
6	none	yes	yes	serious	no
7	none	no	yes	yes	no

3.2. RULE INDUCTION AND DECISION TREES

Since Carole Hafner and Don Berman first presented them in the late 80s and early 90s, introductory tutorials on AI and Law have focused on an example of automatically inducing predictive rules, contrasting it with the human knowledge engineering process of creating rule-based expert systems. The latter involves collecting examples of legal decision-making, manually developing a rule to explain them in terms of legal concepts, and testing and refining the rule on more examples.

Compare that to the induction approach: collect a large set of examples and let the computer create rules using an induction algorithm like ID3 as illustrated in Table 1 and Figure 2. The first shows a small data set of seven cases involving the question of whether a defendant should be released on bail.

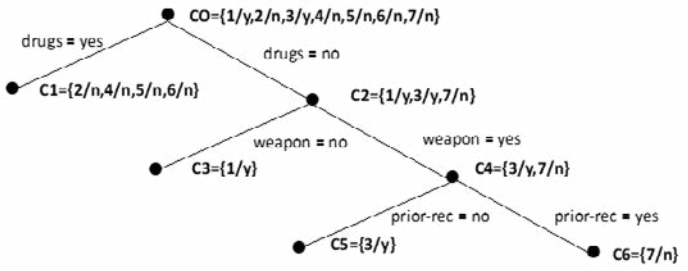

FIGURE 2. *Decision tree for bail decisions*

A decision tree algorithm learns a tree-like set of questions for determining if a previously unseen instance is a positive or negative example of a classification. Each question is a test: for example, if a particular feature has a value of "yes" branch one way or if "no" branch the other way. The test may also be if the weight of a particular feature is less than some threshold; if it is, branch one way, otherwise branch the other way.

An induction algorithm like ID3 generates decision trees like the one in Figure 2. It chooses one attribute to "split" the data. When each of the Ci nodes at the leaves all have instances with same result, the algorithm stops.

ISSN: 1839-4183

An expert system shell may then create rules based on the decision tree. Each rule records a path from the root node to a leaf, for instance, IF drugs = yes THEN bail = no, IF drugs = no AND weapon = no THEN bail = yes.

The advantage of the inductive approach is that it is automatic. Based on information theoretic criteria, ID3 minimizes the number of questions to ask. As an automatic process, it sidesteps knowledge engineering effort and avoids the need for human interpretation of results in fashioning rules. On the other hand, contradictory data, or the need to invent new conceptual terms in order to split the data cleanly, present challenges for an induction algorithm. Without more, decision trees also tend to overfit the data, that is, they learn rules from training data that do not generalize to previously unseen data. Moreover, beyond the induced rules, the approach does not generate legally cognizable reasons for a prediction.

Decision trees and related methods for learning treatment rules for judicial bail decisions (See, e.g., Figure 3, which employs Markov decision processes and a tree-pruning strategy (Lakkaraju and Rudin 2016)) are now addressing new problems. Algorithmic decision-making

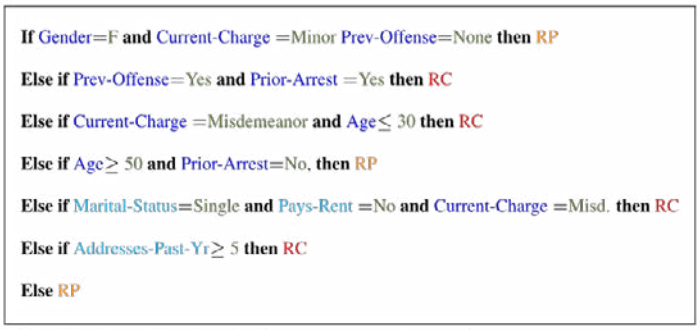

FIGURE 3. *Learning Cost-Effective, Interpretable Treatment Regimes for Judicial Bail Decisions using Markov Decision Processes (Lakkaraju and Rudin 2016)*

is correcting for the fact that the data only includes outcomes for released defendants, not for defendants that judges detained (Kleinberg et al. 2017). The new machine learning models consider predictions of counterfactuals, costs of gathering information, and costs of treatments. The models also need to address issues of fairness and bias (Hutchinson and Mitchell 2018) and be interpretable by a human decision maker.

3.3. PREDICTIONS VIA CASE-BASED ARGUMENT MODELS

Legal decision makers do more with case precedents than induce rules or compare features with neighboring cases; they cite them as authoritative examples, make arguments analogizing them to and distinguishing them from the case to be decided, and cite or distinguish counterexamples. A line of researchers in AI and Law has developed computational models of case-based legal argument and applied them to the task of predicting case outcomes in terms of the strengths of the competing arguments.

A prime example is the Value Judgment-based Argumentative Prediction (VJAP) program (Grabmair 2016). The author assumed that a judge makes a legal decision because the effect of the decision on applicable values is preferable over the effects of alternative decisions. That is, the judge makes a value judgement in determining that a decision's positive effects outweigh the negative effects.

Significantly, these value orderings are not preferences in the abstract; there is no single abstract hierarchy of values. Instead, judges assess the effects on values relative to the specific facts of the case to be decided. VJAP performs this kind of legal reasoning, applying value judgments across cases, mapping them from one factual scenario to another, constructing arguments that a target set of facts relates to the source factual context in a manner that justifies a particular conclusion in light of the applicable values.

In trade secret law, the domain of VJAP, parties can protect confidential product-related information from disclosure and use by competitors. The program employs a logical model of trade secret law, a set of rules derived from legal sources such as the Uniform Trade Secrets Act and the Restatement of Torts Section 757 which many courts have adopted.

This model provides a logical structure of trade secrets law as a set of rules (in the upper half of Figure 4), and for each leaf issue, such as "maintain secrecy" or "confidential relationship", a list of 26 factors that relate to that issue and whose presence strengthens or weakens the argument on that issue of a side (plaintiff "P" or defendant "D"). For

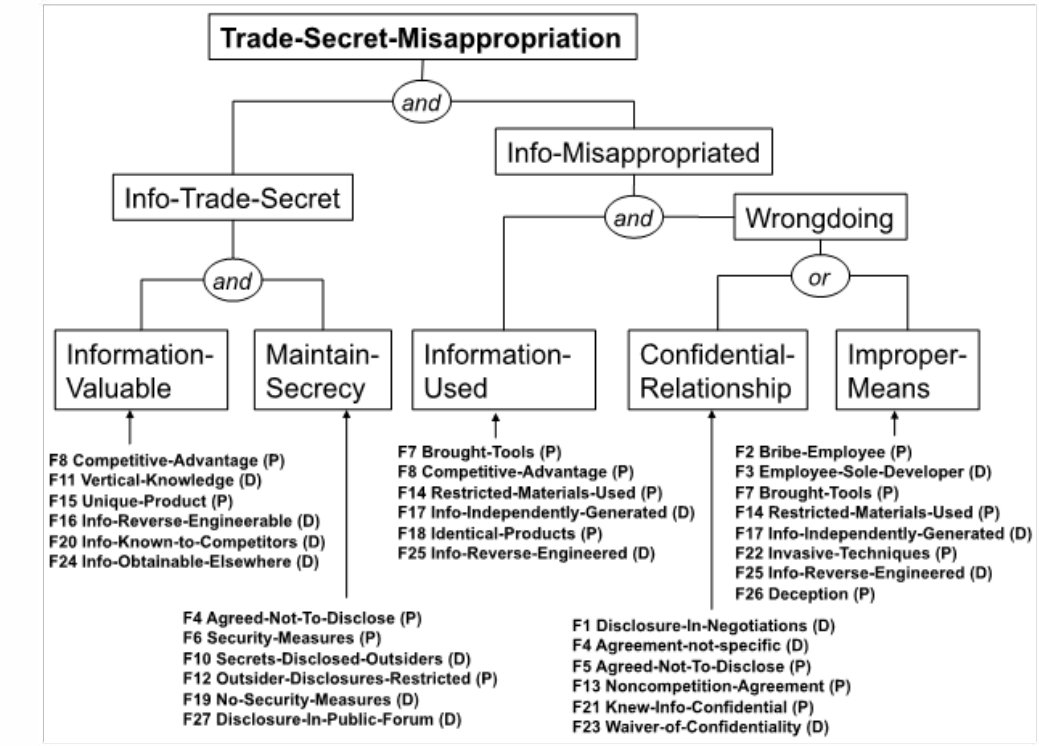

FIGURE 4. *VJAP Domain Model*

example, Table 2 describes three pro-plaintiff and one pro-defendant factor and the related issues. These issue-related factors, in turn, index cases in a database of 121 trade secret misappropriation cases, of which plaintiffs won 74 and defendants won 47.

Grabmair identified four value interests protected by trade secret law. It protects plaintiffs' property interests in competitively valuable information and in maintaining confidentiality. On the other hand, it also protects the general public's interests in the usability of publicly available information and in fair competition. This list of

TABLE 2. *Sample Trade Secret Misappropriation Factors*

Factor no.: Name	Side favoring	Meaning	Issue	Significance
F6: Security-Measures	pro-plaintiff	Pltf. adopted security measures.	Maintain secrecy	It helps to show that Pltf. took reasonable steps to protect his property.
F15: Unique-Product	pro-plaintiff	Pltf. was the only manufacturer making the product.	Information valuable	It helps to show that Pltf.'s trade secret is valuable property.
F16: Info-Reverse-Engineerable	pro-defendant	Pltf.'s product information could be learned by reverse-engineering.	Information valuable	It helps to show that Pltf.'s property interest is limited in time.
F21: Knew-Info-Confidential	pro-plaintiff	Def. knew that Pltf.'s information was confidential.	Confidential relationship	It helps to show that Def. knew Pltf. claimed a property interest.

ISSN: 1839-4183

protected interests is an interpretation of trade secret law, but one based in scholarly treatments.

In a key contribution, Grabmair also identified four ways in which various factors affect a value making it more protected, indicating that it has been waived, making it less legitimate, or interfering with it. For example, consider the value associated with plaintiff's interest in confidentiality. Plaintiff's confidentiality interest is more protected given the confidentiality of outside disclosures (Factor F12), the known confidentiality of the information (F21), the security measures plaintiff took (F6), and the noncompetition (F13) or nondisclosure agreements (F4) the plaintiff entered. The confidentiality interest is less legitimate because of the public availability of the information (F24), the nondisclosure agreement's lack of specificity (F5) or the information being known to competitors (F20). Plaintiff's value interest in confidentiality is waived because of a waiver of confidentiality (F23), the absence of security measures (F19), a public disclosure (F27), its disclosure during negotiations (F1) or its disclosure to outsiders (F10). Finally, the interest in confidentiality is interfered with or violated because of the defendant's use of restricted materials (F14) or its payment to an employee of plaintiff to switch employment (F2). Similar relations between factors and value effects are represented for the other three values underlying trade secret protection.

One inputs a new problem to VJAP represented as set of factors. As suggested in Figure 5, the program then generates all possible arguments about who should win the case by applying argument schemes, templates for making arguments by analogizing the current case to, and distinguishing it from, the cases in its database. These arguments by analogy and distinguishing consider the value tradeoffs in the previous cases as well as in the current case, that is, they consider the effects of factors on underlying values in terms of protection, legitimacy, waiver, and interference.

The argument graph of Figure 6 illustrates these arguments, in oval-shaped nodes, related to propositions in rectangular nodes, via diamond-shaped confidence propagation nodes. This argument structure is a variation of the Carneades argument framework (Gordon et al. 2009): the edges connecting the nodes represent consequence and premise relations (Grabmair 2016, p. 48-51). The upper part of the argument graph corresponds to arguments in the domain model while the lower part contains in depth "arguments about leaf issues, tradeoffs, precedents, and analogy/ distinction arguments between precedent and the case at bar.... " (Grabmair 2016, p. 50).

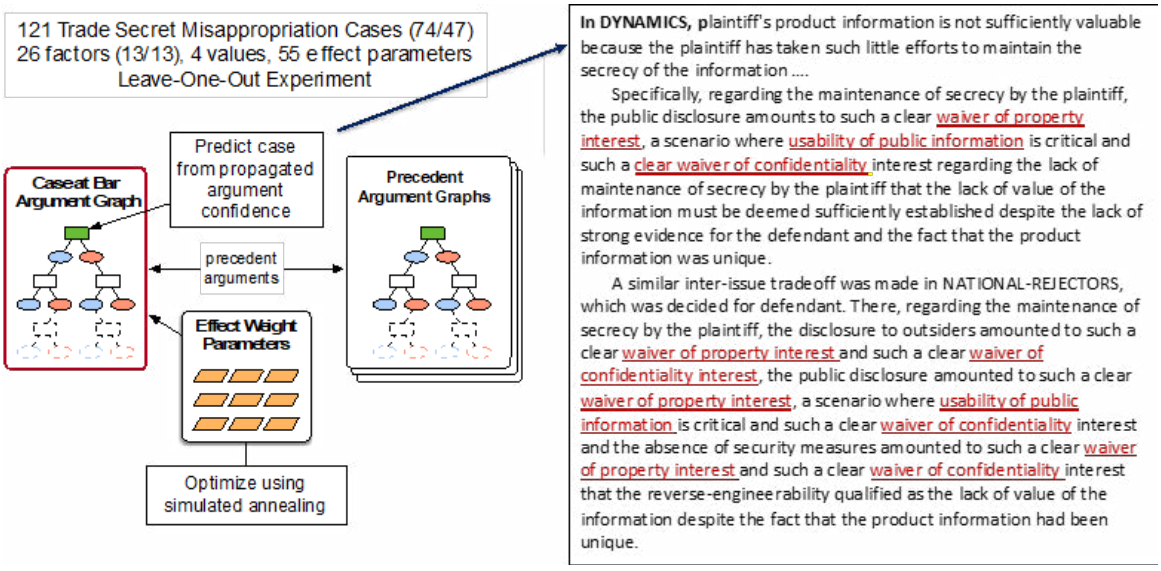

FIGURE 5. *VJAP's process makes predictions and explains them with arguments*

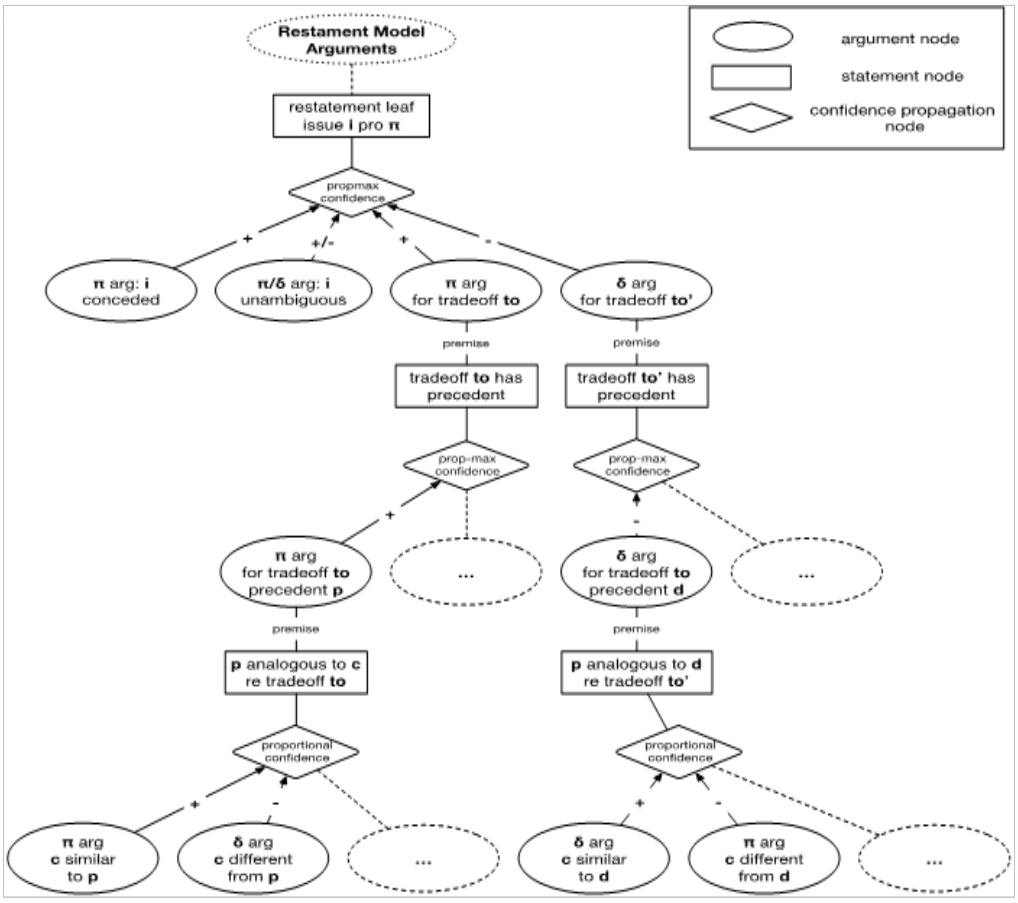

FIGURE 6. *VJAP's process makes predictions and explains them with arguments*

Basically, the program attempts to fit the new case into the existing database, using the arguments as a kind of mapping from one fact situation to another. The program propagates quantitative weights across a graphical model representing its confidence in a prediction based on the magnitude of promotion or demotion of the value in past case contexts. In short, using the quantitative graphical model in Figure 6, it scores the competing arguments and predicts an outcome based on the best fit.

VJAP optimizes the weights iteratively in a process of simulated annealing, adjusting the weights to reflect the degree of confidence that argument premises can be established, which depends on the strength of arguments pro and con the premises. VJAP considers local value tradeoffs involving only one issue as well as inter-issue tradeoffs; the confidence measure is increased in relation to the strength of the analogy between a precedent and the case and decreased to the extent they can be distinguished. (Grabmair 2016, p. 71). The simulated annealing takes place in an argument construction-propagation-prediction loop during which the system iteratively searches for the optimal weight map.

As shown in Figure 5, VJAP then outputs both its predicted outcome and a textual argument justifying the prediction. The underlined phrases indicate where VJAP refers to the relevant values and value effects in analogizing the current case to and distinguishing it from past cases such as the *National Rejectors* case.

The VJAP domain model in Figure 4 and the specification of possible value effects, of course, are examples of explicit, top-down representation of legal knowledge. The question is, however, whether a program can explain

ISSN: 1839-4183

its predictions without such legal knowledge. One can demonstrate empirically the contribution of the values to predictive accuracy. A virtue of creating a knowledge-based AI system is that one can turn on and off (that is, ablate) the various sources of knowledge in experiments to assess their effects on predictive accuracy. In cross validation experiments, Grabmair evaluated various versions of the VJAP program. Cross validation is a standard procedure for evaluating an ML program. The data is 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. Grabmair computed the versions' predictive accuracy, including a version that employs: only local value tradeoffs (.69), local plus inter-issue value tradeoffs (.79), no arguments based on precedents (.71), and arguments based on precedents that occur chronologically prior to the problem case (.84).

The last is particularly interesting. Basing legal arguments on chronologically preceding cases is a practical constraint of real-world legal argumentation. Mackaay and Robillard (1974, p.323) had raised the issue of "the development over time of the case …, a feature which is not found in work reported by earlier researchers in this area." That was 1974. Grabmair (2016) is the first to have evaluated a computational model of case-based legal reasoning imposing such a chronological constraint.

VJAP is only the latest computational model for predicting case outcomes based on arguments. As far as I know, the first work to employ case-based argument strengths as a basis for predicting outcomes involved the CATO (Aleven 2003) and IBP (Ashley and Brüninghaus 2006) programs.

Aleven argued persuasively that predictive accuracy was one measure of the reasonableness of a computational model of argument.

> *A useful supplementary approach is to look at how well a program predicts the outcome of cases, based on its arguments or judgments of case relevance. Good predictive performance would inspire confidence that the arguments made by the program … have some relation to the reality of legal reasoning. (Aleven 2003, p. 212)*

Brüninghaus developed a prediction technique in the Issue-based Prediction (IBP) program that involved hypothesis testing. Using a model like that shown in Figure 4, one counted the pro and con cases involving issue-related factors for each issue in the case to be decided, posed a hypothesis that the case should be decided with the majority on that issue, and then tested the hypothesis by attempting to explain away the counterexamples, that is, the cases decided for the other side. If it could distinguish all of the counterexamples, it would confirm the prediction for the majority side on that issue, otherwise it would abstain. It then used the logical model like that of Figure 4 to combine the issue-based predictions. In comparative evaluations, IBP's predictive accuracy was greater than that of CATO, k-nearest neighbor, decision trees, and a baseline that always predicted the majority class. (Ashley and Brüninghaus 2006).

Unlike CATO or IBP, the AGATHA program represented values associated with factors but differently from VJAP. In a kind of theory construction, AGATHA induced a set of preference rules from the outcomes of past cases. The rules captured preferences between sets of factors in those cases and between sets of the associated values. The theory could then be applied to determine and explain the outcome of new cases. AGATHA's search algorithm constructed a theory in the form of a tree-like set of argument moves, including citing an analogous case, distinguishing it, and countering it with a contrary case. Since it can construct more than one theory, AGATHA selects the best theory (that is, tree) according to theory evaluation criteria operationalized quantitatively, including simplicity (the number of preference rules), explanatory power (the number of cases predicted correctly), tree depth, and completeness (whether additional theory construction moves could be performed). For each theory, these measures are combined into an evaluation number, "a value with which to compare the theories based on how well they explain the background, [and] their structure …. They can be used to… guide a heuristic search." (Chorley and Bench-Capon 2005, p. 48).

As is apparent, each of these approaches, VJAP, CATO, IBP, and AGATHA, not only predicts case outcomes, but explains those predictions in terms of substantive legal knowledge concerning the merits of a case to be decided and precedents. The explanations are in the form of legal arguments or reports of hypothesis testing or theory construction, explanations that employ explicit legal

knowledge about legal rules, issues, factual strengths and weaknesses of particular cases, or underlying values.

3.4. PREDICTION VIA MACHINE LEARNING

It is interesting to compare with the above prediction approaches, two relatively new legal applications that do not consider information about the merits of a case. Katz et al. (2014) have developed and evaluated the first one, a supervised machine learning program to predict if a US Supreme Court Justice or the whole will affirm or reverse a lower court's judgment (referred to here as the KBB program). It employs an advanced form of decision trees, an extremely randomized forest of decision trees, to evaluate a case, input as a set of feature values, and to predict its outcome, based on all previous decisions for that Justice, the Court, and all previous cases. An extremely randomized forest of decision trees is a technique to transform a set of relatively weaker learners into a collectively strong one. It generates a large number of diverse trees and averages across the entire forest. As a result,

> *In total, over the period from 1816-2015, our model exhibits accuracy of 71.9% at the Justice vote level.... Starting in 1816 and carrying through the conclusion of the October 2014 term, our model correctly predicts 70.2% of the Court's decisions. (Katz et al. 2017, p. 8)*

The cases are represented in terms of 95 features from the Supreme Court Database [S], Segal-Cover Scores [SC], and feature engineering [FE] by the authors. As illustrated in Figure 7, the first two sources, prepared by academics,[1] provide case information and background information about the Justices and the Court.

As close as any of the features comes to representing the substantive merits of a case are the Issue Area [S] and Issue [S]. Issue Area values comprise 14 broad categories of legal issues before the Court such as Criminal Procedure, Civil Rights, or First Amendment. Criminal Procedure, in turn, comprises 60 issues, including involuntary confession, habeas corpus, plea bargaining, retroactivity, search and seizure, and others. Thus, features of a case that capture particular facts, strengths, or weaknesses are not represented.

With respect to background information about the Justices and the Court, the political party of the president who appointed the Justice is represented as a stand-in for the Justice's views, conservative or liberal. The Segal Cover Scores measure the "perceived qualifications and ideology" of a Supreme Court nominee. The trend features are engineered by the authors to record decision directions over time periods such as recent, prior, or cumulative terms with respect to legal issue areas.

The authors reported that, "collectively, individual case features account for approximately 23% of predictive power.... Justice and Court level background information account for just 4.4%." Most of the model's predictive power (72%) is driven by tracking the behavioral trends, including the ideological direction of overall voting and voting of various Justices, general and issue-specific differences

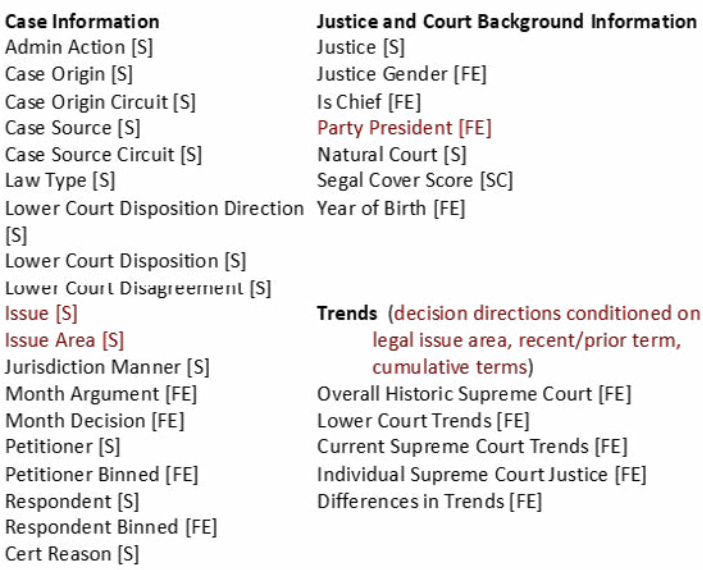

FIGURE 7. *Overview of Case Features in KBB Program*

between individual Justices and the balance of the Court, and ideological differences between the Supreme Court and lower courts (Katz et al. 2014, p. 17-18).

Like the KBB program, Lex Machina, another supervised machine learning approach to prediction, did not consider the substantive merits of cases, but focused on very different case features: the litigation participants and their behavior, namely the lawsuit parties, their attorneys

[1] http://scdb.wustl.edu/ ; https://gist.github.com/jeremyjbowers/f36efe6db30056b1a587

ISSN: 1839-4183

and law firms, the judges assigned to a case, the districts where the complaints were filed, judicial and district "bias" (computed as the ratio of cases won by the plaintiff from the set of past cases assigned to the corresponding judge or district) and the case outcomes. Developed at Stanford University by law professor Mark Lemley and colleagues in the Computer Science Department, Lex Machina employed logistic regression, a statistical machine learning model, to predict the outcomes of intellectual property claims based on all IP lawsuits in a 10+ year period with an accuracy of 64% (Surdeanu et al. 2011).[2] The most significant contributions to accuracy were the judge's identity,[3] followed by the plaintiff's law firm, the defendant's identify, the district where the claim was filed, the defendant's law firm, and the defendant's attorney.

Remarkably, Lex Machina's participant-and-behavior features can be extracted automatically from the texts of cases. For most features, it required identification of named entities (firms, courts, people) and checking names against directories or lists of names. Extracting the outcomes of cases was more difficult. Three IP experts annotated sentences stating the outcomes for cases in a training set, and a machine learning model was able to learn to extract the outcomes automatically.

The authors emphasized that the model is "agnostic to the merits of the case". Given enough data, participant-and-behavior features alone were a substitute for information about a case's merits. Lacking substantive information about the case, however, Lex Machina was unable to explain its predictions in terms that legal professionals would recognize as legal explanation or argumentation. One wonders if the program could predict better with such substantive information and, given the AI and Law prediction work described thus far, how that could be accomplished. It is a problem of representation: factors and other substantively relevant features work well where the cases are all of a type, but here the IP-related cases presumably ranged across many types of legal claims. Even if only one type of legal claim, an appropriate representation needs to be available; factors are appropriate for trade secret law and some issues in trademark, but have not been applied in other IP-related areas.

3.5. PREDICTION WITH SUBSTANTIVE INFORMATION EXTRACTED FROM CASE TEXTS

Assuming an appropriate representation for the substantive merits of a case is available, can such information be extracted automatically through legal text analytics and used to predict case outcomes? That approach was first tried in the SMILE+IBP project and continues to be a focus of current research in AI and Law. Alternatively, can one dispense with representing substantive legal features altogether and make predictions directly from the text of cases? That approach has been applied in recent work predicting outcomes of cases before the European Court of Human Rights. We will examine both approaches here in turn.

The SMILE+IBP program learned how to identify trade secret misappropriation factors in summaries of case texts. It employed supervised ML. Some examples of the kinds of sentences it learned to classify (from the *Mason* case, a trade secret dispute concerning the recipe for a cocktail, Lynchburg Lemonade) for the factors described in Table 2 include:

> F6: Security-Measures (pro-plaintiff): He testified that he told only a few of his employees--the bartenders--the recipe. He stated that each one was specifically instructed not to tell anyone the recipe. To prevent customers from learning the recipe, the beverage was mixed in the "back" of the restaurant and lounge.
>
> F15: Unique-Product (pro-plaintiff): It appears that one could not order a Lynchburg Lemonade in any establishment other than that of the plaintiff.
>
> F16: Info-Reverse-Engineerable (pro-defendant): At least one witness testified that he could duplicate the recipe after tasting a Lynchburg Lemonade.

[2] LexisNexis acquired Lex Machina in 2015.

[3] It appears that the use of Lex Machina with French judicial data would be illegal in France. The French Parliament has adopted a law prohibiting the use of judicial data for purposes of prediction: "No personally identifiable data concerning judges or court clerks may be subject to any reuse with the purpose or result of evaluating, analyzing or predicting their actual or supposed professional practices." Article 33 of the Justice Reform Act. Violation of this law could result in a prison term of five years. http://www.abajournal.com/news/article/france-bans-and-creates-criminal-penalty-for-judicial-analytics (accessed 23/7/2019).

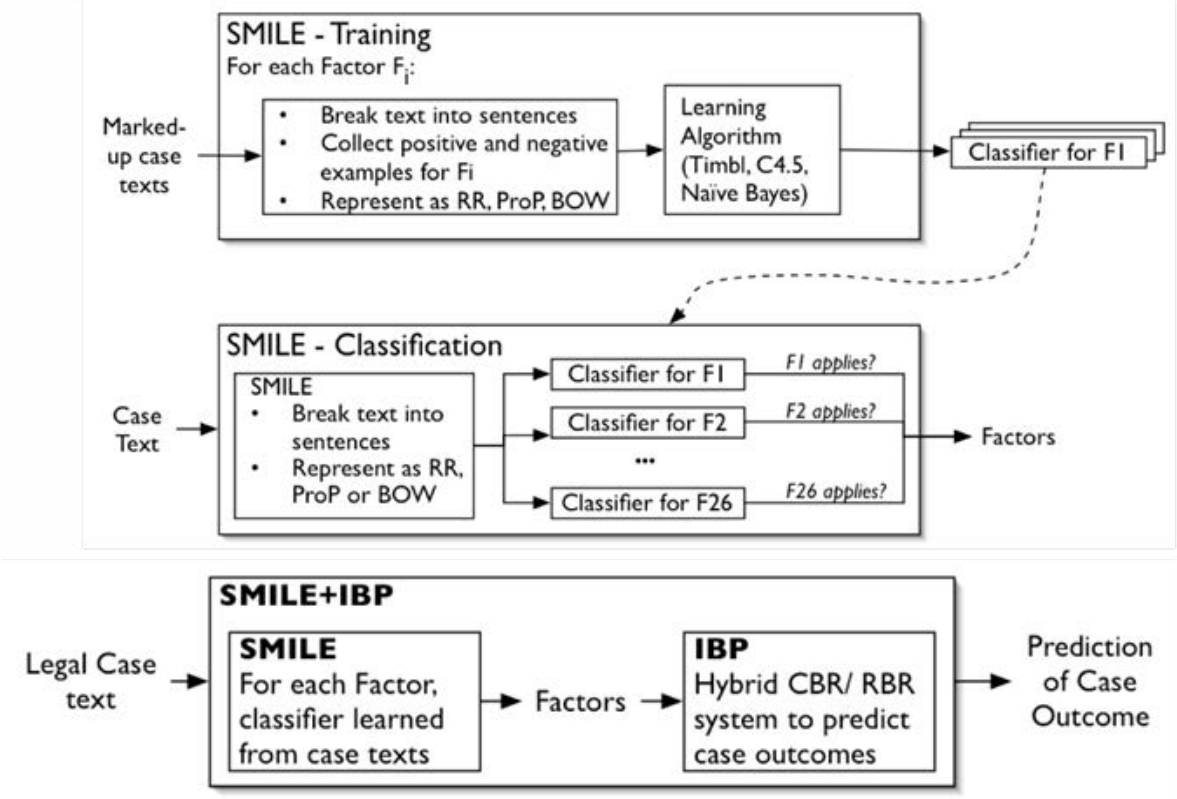

FIGURE 8. *SMILE+IBP learned to identify factors*

F21: Knew-Info-Confidential (pro-plaintiff): On cross-examination Randle agreed that he had been under the impression that Mason's recipe for Lynchburg Lemonade was a secret formula.

As indicated in Figure 8 at the top, in a training step, a classifier was learned for each factor from the texts of positive and negative instances of factor-related sentences from trade secret cases represented in three ways (as bags of words, that is, as term vectors, with roles represented (for example, substituting "plaintiff" or "defendant" for party names), or in terms of propositional patterns capturing subject-verb, verb-object, verb – prepositional phrase, and verb – adjective relationships). In a prediction or classification step, the full text of a new trade secret case was input, broken into sentences represented in the three ways, and all of the factor classifiers were applied. SMILE's output, the resulting list of factors representing the case, were then input to the IBP program, which used the hypothesis-testing approach described above to predict an outcome. In an evaluation, we compared the predictive accuracy and F1 metric of SMILE+IBP, IBP using human-determined case factors as inputs, and a biased-coin-toss baseline (F1 is the arithmetic mean of accuracy and coverage). While scoring lower than IBP's accuracy and F1 measures (.92, .96), SMILE+IBP scored higher (.63, .70) than the baseline (.49, .66) indicating that it was processing some factor-related signal in the case texts (Ashley and Bruninghaus 2009). One must recall that this was long before the rise of legal text analytics. As Samuel Johnson observed of a dog walking on its hind legs, "It is not done well; but you are surprised to find it done at all."

In more recent efforts, Falakmasir and Ashley (2017) assembled 1600 trade secret cases from CourtListener,[4] employed a word-embedding text representation technique, Doc2Vec, to capture contextual semantic information in the texts, focused on 179 cases in the IBP corpus, trained a machine learning model for each factor, and predicted the factors that apply in each case document. In an evaluation applying the model to 30% of the documents as a

ISSN: 1839-4183

hold-out test set, the result was an F1 measure (arithmetic mean of precision and recall) of .69/.65 (micro/macro) (Falakmasir and Ashley 2017).

Why is this important? If legal text analysis programs can learn how to automatically identify factors in case texts, then computational models of legal argument (e.g., VJAP, AGATHA, IBP) can accept case texts as inputs and output predictions and explanations or arguments. Thus, machine learning (ML) from manually annotated (or marked-up) texts is likely to be essential for scaling up AI and Law programs. (Ashley 2017).

Annotating case texts, for example, marking-up which sentences are positive instances of a given factor, is time-consuming and expensive in terms of expert labor. Any techniques for minimizing the amount of annotation required are worth exploring. In a recent paper, Branting et al. (2019) present the SCALE approach as an alternative, a semi-supervised machine learning method for achieving explainable legal prediction. SCALE employs a small set of annotated data and maps it onto a larger set of candidate documents. The approach may be useful wherever courts describe legal concepts in stereotypical terms across domains of legal cases. They need not use identical language; SCALE applies word-embedding representations and clustering algorithms that can identify semantically similar descriptive text segments across the case texts. The technique could be used to predict outcomes based on the relevant resulting clusters and explain the predictions in terms of cluster-related concepts. Alternatively, it could be applied as a pre-processing technique to make detailed annotation more efficient.

Branting presents a useful table of paradigms for explainable decision prediction. See Table 3. Items five through seven are approaches whose inputs are not case texts but features such as factors or rule predicates and include VJAP, IBP, and AGATHA. Items three and four do take case texts as inputs and include SMILE+IBP and SCALE, which then identify members of the feature set of factors or rule predicates in the texts and use them to predict and explain outcomes.

Items one and two move toward a radical alternative. They avoid the need for engineering sets of features altogether, using unsupervised machine learning to identify features such as topics (Aletras et al. 2016) or making predictions directly from the texts represented only as feature vectors (Medvedeva et al. 2019). Unsupervised ML employs techniques such as clustering algorithms that infer groupings of unlabeled instances based on their content.

The goal of Aletras et al. (2016) was to predict violations of particular articles of the European Convention on Human Rights (ECHR) from the texts of cases tried by the European Court of Human Rights (ECtHR). They hypothesized that the case texts and the parts of the text dealing with the facts, the law, and arguments are "important factors" influencing the case outcome. The work focused on three articles of the ECHR: Article 3: Prohibition of Torture, Article 6: Right to a Fair Trial, and Article 8: Right to Respect for Private and Family Life. The corpus comprised the published decisions of cases that had survived the first stage of admissibility, including 250 cases for Article 3, 80 for Article 6, and 254 for Article 8, all balanced in terms of the numbers of decisions granting or denying relief. Each case was represented in terms of its outcome and as bags of terms, that is, feature vectors, with n-grams from one to four contiguous terms for its sections on procedure, facts (circumstances, relevant law), law (which includes the parties' arguments and the court's reasons) and for the full case as a whole.

Case texts were also represented in terms of "abstract semantic topics" comprising word clusters associated

TABLE 3. *Paradigms of explainable decision prediction (Branting, et al. 2019, p. 23) with additions*

Paradigm	K-E artifacts	Execution-time representation	Output	Example
1. text → output			prediction, relevance weights	(Medvedeva, et al. 2019
2. text → predicates → output	predicate set, rule set		prediction, per-predicate relevance weights	(Aletras, et al. 2016
3. text → features → output	feature set		case-based argumentation	SMILE+IBP
4. text → features→ predicates → output	feature set, predicate set, rule set		hybrid case/rule-based argumentation	SCALE
5. features → output	feature set	featural case representation	case-based argumentation	IBP, AGATHA, VJAP
6. features → output	rule set	featural case representation	rule-base argumentation	
7. features→ predicates → output	feature set, predicate set, rule set	featural case representation	hybrid case/rule-based argumentation	NIHL (Angelic methodology)

[4] https://www.courtlistener.com/

with the ECHR articles. For each article, the word clusters were trained using an automated, unsupervised technique (spectral clustering on an n-gram similarity matrix). The presence of a cluster in a case text was treated as an additional feature representing that text. Thus, although their approach employed a kind of semantic legal features, human experts were not involved in compiling the features, which was done automatically.

The researchers applied a machine learning model (a linear support vector machine) and evaluated the trained model (using 10-fold stratified cross validation). Their model achieved an accuracy of 79% accuracy at the case outcome level. The circumstances and topics n-grams were the best predictors, the law n-grams predicted badly, a correlation was observed between facts and outcomes, and the topics revealed groups of non-violation and violation cases.

Subsequent researchers criticized the results in (Aletras et al. 2016), pointing out a confound in that the language of the circumstances sections of ECtHR cases was not neutral but prefigured the outcomes (Medvedeva et al. 2019). In that later work, the researchers assembled a database of more ECtHR cases for nine ECHR articles, balanced between Violation and Non-violation cases, applied a supervised machine learning algorithm (support vector machine), and achieved the predictive accuracies on a held-out test set shown in Table 4, with an average accuracy of 0.74.

Interestingly, in another experiment, they enforced a chronology constraint for Articles 3, 6, and 8 such that the cases used to predict a case's outcome had to have occurred prior in time to the case. When using training cases up to 2013 to test 2016-17 cases, the accuracies decreased to 0.70, 0.63, and 0.64, respectively. Compare this with the VJAP results, where enforcing the chronology constraint increased the accuracy of prediction. Could knowledge be the difference?

Finally, Medvedeva et al. also assessed prediction based simply on the surnames of the judges, achieving an average accuracy of 0.66! Prediction is always full of surprises!

4. PROSPECTS FOR EXPLAINING PREDICTIONS FROM TEXT

In this respect, one sees the influence that features like judges' names can have on prediction. Some programs like Lex Machina exploit such features, but they do not support explaining predictions in substantive terms. As discussed below, in some use cases considering such features is a virtue. In others, researchers must take steps to mask the predictive contributions of features like judges' names or certain citations.

In Aletras et al. (2016) the SVM algorithm weights the topic-related clusters in terms of how strongly they support an outcome of Violation or No Violation. Some of these clusters make some intuitive sense. For instance, the third highest ranked cluster for a finding of Violation of Article 3, "No one shall be subjected to torture or to inhuman or degrading treatment or punishment," labeled by the authors as "Treatment by state officials," contained terms such as "police, officer, treatment, police officer, July, ill, force, evidence, ill treatment, arrest, allegation, police station, subjected, arrested, brought, subsequently, allegedly, ten, treated, beaten." Although it is not obvious why some of these terms are included, others seem as though

TABLE 4. *Dataset and results per ECHR article in (Medvedeva, et al. 2019)*

Article	'Violation'	Drugs	Weapon	Prior record	Result
2	57	57	114	398	0.82
3	284	284	568	851	0.81
5	150	150	300	1118	0.75
6	458	458	916	4092	0.75
8	229	229	458	496	0.65
10	106	106	212	252	0.52
11	32	32	64	89	0.66
13	106	106	212	1060	0.82
14	144	144	288	44	0.84

ISSN: 1839-4183

they could be relevant to a conclusion of Violation. While perusal of the clusters for other topics shows that some are intuitively relevant, others are equivocal, and none conveys much confidence that these clusters could form the basis of an explanation (Aletras et al. 2016, Table 3).

Branting et al. (2017) have employed Hierarchical Attention Networks (HANs) to induce models from previous case decision texts that can predict case outcomes. HANs employ stacked recurrent neural networks, that is, neural networks that can process temporal sequences of inputs. One such network operates at the word level and has an

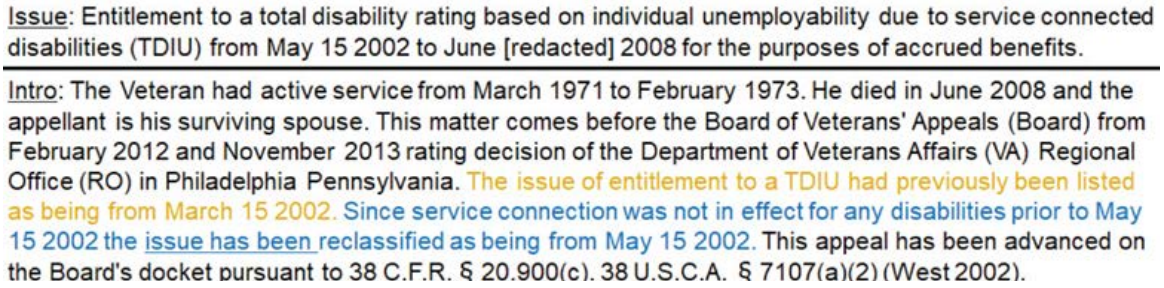

FIGURE 9. *A portion of a BVA case. The sentence with the highest proportion of attention weight, 74%, is shown in blue, and the sentence with the next highest weight, 9%, is shown in yellow. (Branting, et al. 2017 Figure 3)*

attention model to extract into a sentence vector those words important to the meaning of the sentence. A similar procedure is applied to the derived sentence vectors which then generates a document vector representing aspects of the meaning of a given document.[5]

For present purposes, the attention model is the important point. It assigns higher weights to the text portions that have greater influence on the model's outcome prediction. The hope is that the attention model can be used to highlight those salient portions of the text and that these highlighted portions would amount to an explanation of the prediction.

Branting applied the HAN approach to predict outcomes from the full texts of cases in a corpus of decisions of the Board of Veterans Appeals (BVA) and of WIPO domain name disputes. The results were good: F1 metric: BVA 0.74; WIPO 0.94. He found that decision outcomes could be predicted using various models from the texts of motion, contention, and factual background sections alone. Then, he used the network attention weights from the predictive models to identify salient phrases in decisions and presented the information in an interface specially designed to improve decision making. Figure 9 illustrates the output of the interface for a portion of a BVA decision in which two highly weighted sentences are highlighted.

In an experiment, Branting, et al. (2019) engaged 61 experts and non-experts on a task involving analyzing WIPO decisions in which attention-weight-based highlighting was employed in the user interface. "Each participant was asked to decide the issue of 'No Rights or Legitimate Interests' (NRLI) in two separate cases and to provide a justification for each prediction" as well as to comment on the experience. There were four conditions, two of which involved highlighting portions of case texts based on attention weights. The results showed that the "[h]ighlighting had no effect on correctness" of the predictions.

> *Perhaps the most illuminating comments by participants were that they had difficulty understanding the connection between the highlighted text and the issue that they were supposed to decide. These comments, and the overall results of study, indicate that useful decision support should help the user understand the connection between relevant portions of the case record and the issues and reasoning of the case (Branting et al. 2019, pp. 24f).*

[5] https://medium.com/analytics-vidhya/hierarchical-attention-networks-d220318cf87e

Although one experiment is not determinative, this finding is a blow to hopes that HAN attention weights can explain legal predictions.

5. USES OF AUTOMATED PREDICTION

We know that using legal predictions for some purposes requires explanations and that some ML models © feature weights do not yield accessible explanations. The use of hierarchical attention networks to predict case outcomes yielding network attention weights offered the hope of using the attention weights to highlight relevant portions of the document. Branting's careful evaluation, however, suggests that attention weights are *not* a basis for intelligible explanations.

This may not deter commercial applications of text-based case prediction. One provider's website touts a recent experiment pitting legal experts versus the company's ML prediction algorithm to see which could better predict whether complaints for payment protection insurance (PPI) mis-selling will be granted or rejected. CaseCruncher Alpha predicted outcomes of complaints submitted to it as texts based on target data from historical decisions using a multilayer (convolutional) neural network classifier. They compared the system's accuracy of prediction, 86.6%, with that of 100+ UK lawyers, (62.3%). Although details of the evaluation/dataset were not published, the promoters suggested that the

> *... main reason for the large winning margin seems to be that the network had a better grasp of the importance of non-legal factors than lawyers .... The experiment also suggests that there may be factors other than legal factors contributing to the outcome of cases. Further research is necessary ...*[6]

Presumably, given its use of a neural network, CaseCruncher Alpha does not explain its predictions. If its predictions are more accurate, however, because they

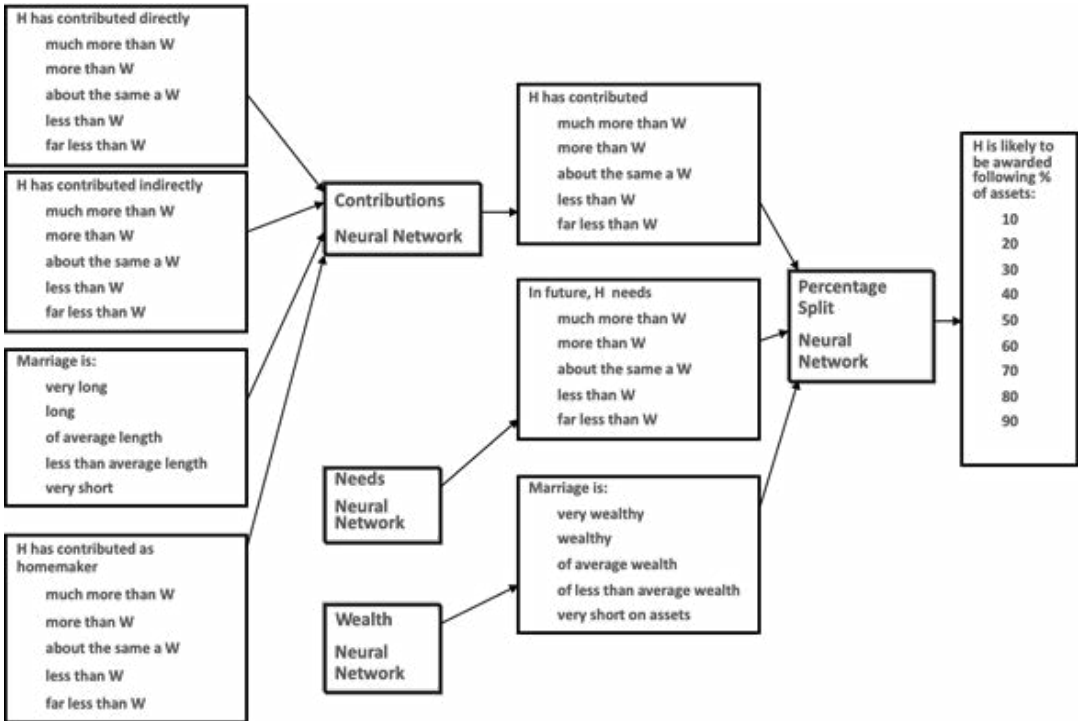

FIGURE 10. *Split-Up, a divide and conquer approach to explanation*

ISSN: 1839-4183

account for apparently "non-legal factors", that may better fit certain use cases such as valuing a dispute for purposes of betting or settlement, or strategic planning and resource allocation. Lex Machina also considered non-legal factors; one could imagine using it to inform big firm's lateral hiring decisions since it provides readily available statistics about litigators' performance.

For those use cases requiring explanations, a divide and conquer approach may be more appropriate.

In a program that predicted judicial allocations of marital assets in divorce proceedings, Zeleznikow and Stranieri (1995) implemented neural networks but addressed their inability to explain decisions by using a divide-and-conquer approach. As suggested, in Figure 10, they employed a structural framework of multiple neural networks, one for each issue such as needs, direct and indirect contributions, wealth, etc., and generated explanations based on the overall structure of the outputs, not just on the individual outputs.

In a sense, divide-and-conquer is also the approach in items three and four of Table 3. Programs like SMILE+IBP and SCALE divide predictions into a framework of legally relevant issues, the sets of factors and rule predicates. The programs employ multiple ML models to predict the presence of the sets' features in an individual case, and employ the framework, such as a domain model illustrated in Figure 4, to predict and explain an overall outcome.

Finally, there are other kinds of use cases in AI and Law in which prediction plays more of a supportive role. For example, in automatically summarizing legal cases Zhong et al. (2019) employed as a criterion for including a sentence in the summary, a sentence's outcome-predictiveness, that is, its correlation with the outcome of a case.

The work focused on 35,000 Board of Veterans' Appeals (BVA) cases involving a single issue, Post-Traumatic Stress Disorder (PTSD). The outcome distribution was 3:2:1 with respect to the three possible outcomes: remanded, denied and granted cases. The researchers employed a template for generating summaries comprising one sentence each stating: the source of the appeal (e.g., "This is an appeal from the Department of Veterans Affairs Regional Office in Seattle, Washington."), the issue on appeal (e.g., "The issue is entitlement to service connection for posttraumatic stress disorder (PTSD)."), the military service history (e.g., "The veteran had active military service from November 1967 to December 1970."), and the conclusion (e.g., "Service connection for PTSD is granted.") Each of these were identified with regular expressions (i.e., regex rules) (Zhong et al. 2019).

The summaries also contained, however, up to three sentences stating the court's reasoning and evidential support e.g., "The evidence of record establishes that the Veteran has been diagnosed with PTSD related to in-service stressors that have been corroborated by credible supporting evidence."). These were generated by an iterative process of ranking the sentences by predictiveness, selecting the most predictive sentence for possible inclusion in the summary, masking that sentence, and repeating the process until the sentences' predictiveness dipped below a threshold.

Interestingly, not all of the most predictive sentences contain useful information such as facts and evidence worth incorporating into a summary. Some sentences were statistically correlated with outcomes but comprised only citations, names of judges, or statements of high-level legal rules. For example, an excerpt like "STEVEN L. KELLER, BVA", may be highly correlated with outcomes and thus predictive because the model learned that Steven L. Keller remanded 78 percent of the cases. Thus, the researchers implemented measures to filter out citations, high-level legal rules, and judges' names, non-legal features like those on which Lex Machina based its predictions (Zhong et al. 2019).

In addition, only sentences of a particular type, Reasoning & Evidential Support, would be appropriate for the summary template. A random forest decision tree classifier, trained on an annotated training set, identified reasoning and evidential support sentences from the set of most predictive sentences. Finally, since it was important that the summary minimize duplication, an algorithm was applied (Maximal Marginal Relevance) to select the most diversified of the remaining most predictive sentences,

[6] https://www.case-crunch.com/index.html#progress-bars3-o

and the selected sentences were fit into the summary template (Zhong et al. 2019).

The results were somewhat equivocal. The researchers had enlisted two law students to write 20 summaries for comparison. The researchers determined that the frequently applied quantitative measures of summary quality (i.e., ROUGE-1 (unigram overlap) and ROUGE-2 (bigram overlap) (Lin 2004)) were inadequate in that they could not distinguish the human-prepared summaries of presumably higher quality from machine-generated ones. A human expert determined that while 10% of the human-generated summaries did not adequately identify issues and resolutions for the 20 cases, 50% of the machine generated summaries did not do so. Nevertheless, the predictiveness of sentences appeared to be a useful, if not sufficient, criterion for inclusion in meaningful summaries. (Zhong et al. 2019).

6. CONCLUSIONS

We have seen the different roles that predicting case outcomes has played in the history of Artificial Intelligence and Law research, from identifying borderline cases worthy of legal academic commentary, to providing some evidence of the reasonableness of computational models of case-based legal reasoning, to providing the raison d'être of such models, to accounting for statistically telling features beyond such models, to circumventing features altogether in favor of predicting outcomes directly from analyzing case texts.

We have also seen a variety of use cases to which predicting case outcomes has been put. Some of these require explanations to help humans assess whether to believe the prediction. For these uses, some combination of model-based and text analytic approaches could be best at predicting legal outcomes, providing explanations, and enabling arguments in terms attorneys can understand. Other uses involving valuing a dispute for purposes of betting or settlement, strategic planning and resource allocation, or lateral hiring may benefit from taking into account features instead of or beyond substantive legal merits. Still other uses, the most recent ones, employ prediction in support of some other intelligent task such as summarizing legal cases, tasks for which some legal knowledge is required but that do not necessarily require explanations of predictions.

Throughout this evolution, the question recurs that Mackaay and Robillard first posed. What do the prediction methods try to achieve: "minimization of prediction errors or elucidation of human understanding?" Both are important, but the use case determines the balance of their importance for particular tasks.

7. REFERENCES

1. Aletras, N., D. Tsarapatsanis, D. Preoţiuc-Pietro, and V. Lampos. 2016. "Predicting judicial decisions of the European Court of Human Rights: A natural language processing perspective." *PeerJ Computer Science*, 2: e93.
2. Aleven, V. 2003. "Using background knowledge in case-based legal reasoning: a computational model and an intelligent learning environment." *Artificial Intelligence,* 150 (1–2): 183–237.
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. and Brüninghaus, S. 2006. "Computer models for legal prediction." *Jurimetrics*, 46 (3): 309–52.
5. Ashley, K. and Brüninghaus, S. 2009. "Automatically classifying case texts and predicting outcomes." *Artificial Intelligence and Law,* 17 (2): 125–65.
6. Branting, K. et al. 2019. "Semi-Supervised Methods for Explainable Legal Prediction." *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law,* ICAIL '19, June 17–21, Montreal, QC, CA, pp. 22-31.
7. Branting, L. K., Yeh, A., Weiss, B., Merkhofer, E., and Brown, B. 2017, October. "Cognitive Assistance for Administrative Adjudication." In 2017 *AAAI Fall Symposium Series. Cognitive Assistance in Government and Public Sector Applications.* AAAI Technical Report FS-17-02, pp. 134-140.
8. Branting, L. K., Yeh, A., Weiss, B., Merkhofer, E., and Brown, B. 2017. "Inducing predictive models for decision support in administrative adjudication." In U. Pagallo et al. (eds.) *AI Approaches to the Complexity of Legal Systems. AICOL International Workshops 2015-2017*, LNAI 1071, Springer, Cham, pp. 465-477.
9. Chorley, A. and Bench-Capon, T. 2005. "AGATHA: using heuristic search to automate the construction of case law theories." *Artificial Intelligence and Law*, 13 (1): 9-51.

ISSN: 1839-4183

10. Falakmasir, M. and Ashley, K. 2017. "Utilizing Vector Space Models for Identifying Legal Factors from Text", *30th Int'l Conf. on Legal Knowledge and Information Systems*. Jurix 2017, Amsterdam: IOS Press, pp. 183-192.
11. Gordon, T., Prakken, H., and Walton, D. 2007. "The Carneades model of argument and burden of proof." *Artificial Intelligence,* 171 (10–5): 875–96.
12. Grabmair, M. 2016. *Modeling Purposive Legal Argumentation and Case Outcome Prediction using Argument Schemes in the Value Judgment Formalism.* Ph.D. thesis, University of Pittsburgh, Pittsburgh, PA.
13. Hutchinson, B., and Mitchell, M. 2019, January. "50 Years of Test (Un) fairness: Lessons for Machine Learning." In *Proceedings of the Conference on Fairness*, *Accountability, and Transparency,* ACM, pp. 49-58.
14. Katz, D., Bommarito, M. and Blackman, J. A. 2017. "General Approach for Predicting the Behavior of the Supreme Court of the United States." *PLoS ONE* 12(4): e0174698. https://doi.org/10.1371/journal.pone.0174698
15. Katz, D., Bommarito, M. II, and Blackman, J. 2014. "Predicting the Behavior of the Supreme Court of the United States: A General Approach." ARXIV.ORG, at 6 (2014), https://arxiv.org/pdf/1407.6333.pdf [https://perma.cc/JXX5-WQBY].
16. Kleinberg, J., Lakkaraju, H., Leskovec, J., Ludwig, J., and Mullainathan, S. 2017. "Human decisions and machine predictions." *The quarterly journal of economics,* 133 (1): 237-293.
17. Lakkaraju, H., and Rudin, C. 2016. "Learning cost-effective treatment regimes using Markov decision processes." *arXiv preprint arXiv:1610.06972*.
18. Lin, C.-Y. 2004. "Rouge: A package for automatic evaluation of summaries." *Proceedings of the ACL-04 Workshop Text Summarization Branches Out*", Barcelona, Conference held in conjunction with ACL, pp. 74–81.
19. Mackaay, E., Robillard, P. 1974. "Predicting judicial decisions: the nearest neighbor rule and visual representation of case patterns", *Datenverarbeitung im Recht*, 3: 302–31.
20. Medvedeva, M., Vols, M., and Wieling, M. 2019. "Using machine learning to predict decisions of the European Court of Human Rights." *Artificial Intelligence and Law,* first online 16 June, pp. 1-30. https://doi.org/10.1007/s10506-019-09255-y
21. Surdeanu, M., Nallapati, R., Gregory, G., Walker, J., and Manning, C. 2011. "Risk analysis for intellectual property litigation." *Proceedings of the 13th International Conference on Artificial Intelligence and Law.* New York, NY: ACM, pp. 116–20.
22. Susskind, R. 2010. *The End of Lawyers? Rethinking the Nature of Legal Services.* Oxford: Oxford University Press.
23. Zeleznikow, J. and A. Stranieri. 1995. "The Split-Up System: Integrating Neural Networks and Rule-Based Reasoning in the Legal Domain." In *Proceedings ICAIL-95*. New York: ACM, pp. 185-194.
24. Zhong, L., Z. Zhong, Z. Zhao, S. Wang, K. Ashley, and Grabmair, M. 2019. "Automatic Summarization of Legal Decisions Using Iterative Masking of Predictive Sentences." In *Proceedings ICAIL-19*, New York: ACM, pp. 163-172.