VIRGINIA JOURNAL OF LAW & TECHNOLOGY

SPRING 2016 UNIVERSITY OF VIRGINIA VOL. 20, NO. 01

Big Data Legal Scholarship: Toward a Research Program and Practitioner's Guide

FRANK FAGAN †

† Attorney Advisor, U.S. Department of Labor; Lecturer, University of South Australia School of Law.

ABSTRACT

This Article seeks to take first steps toward developing a research program for big data legal scholarship by sketching its positive and normative components. The application of big data methods to descriptive questions of law can generate broader consensus. This is because big data methods can provide greater comprehensiveness and less subjectivity than traditional approaches, and can diminish general disagreement over the categorization and theoretical development of law as a result. Positive application can increase the clarity of rules; uncover the relationship between judicial text and outcome; and comprehensively describe judicially-determined facts, the contents of legislation and regulation, or the contents of private agreements. Equipped with a normative framework, big data legal scholarship can lower the costs of judging, litigating, and administrating law; increase comparative justice and predictability; and support the advocacy of better rules and policies.

In addition to sketching theoretical foundations, this Article seeks to take first steps toward developing the core components of successful praxis. Handling and analyzing big data can be cumbersome, though the newcomer can avoid common pitfalls with care. Accordingly, there exist best practices for preprocessing, converting, and analyzing the data. Current analytical techniques germane to law include algorithmic classification, topic modeling, and large-dimension regression analysis.

First steps, by definition, are incomplete. The contours of big data legal scholarship and practice will undoubtedly shift over time to reflect new techniques and prevailing normative

questions. This Article merely aspires to generate a conversation about how big data can enhance our understanding of law — what it is, and what it should be.

TABLE OF CONTENTS

I. INTRODUCTION

Big data legal scholarship is relatively new. To date, only a handful of studies have appeared in law reviews and peer-reviewed legal journals.[1] It remains to be seen if legal scholars will embrace big data enthusiastically, but if the behavior of social scientists and humanities scholars from other fields is suggestive, a willingness to carry out big data research will increase as the methods simplify.[2] One observation in favor of an eventual and extensive embrace is that the type of data used by lawyers, judges, and most legal scholars is fundamentally textual.[3] Opinion texts constitute the raw materials that jurists mine for doctrinal advancement or retreat.[4] Analogously, big data analysts in other fields mine vast quantities of textual data for various medical, business, and political applications, among others.[5]

---

[1] *See infra* text accompanying notes 17–20.

[2] *See* CHRISTINE L. BORGMAN, BIG DATA, LITTLE DATA, NO DATA: SCHOLARSHIP IN THE NETWORKED WORLD *passim* (2015).

[3] *Cf.* Tom S. Clark & Benjamin E. Lauderdale, *The Genealogy of Law*, 20 POL. ANALYSIS 329, 330 (2012) ("While data on the justices' dispositional voting and which opinions the justices 'join' (which opinions they endorse in a case) can provide a great deal of information about the justices' preferences, opinion texts are the most important source of information about doctrine and its evolution.").

[4] *Id.* ("Ultimately, observable data about legal doctrine derives almost exclusively from the texts of opinions."). *See, e.g.*, Lea-Rachel Kosnik, *Determinants of Contract Completeness: An Environmental Regulatory Application*, 37 INT'L REV. L. & ECON. 198, 201 (2014) (mining several thousand licensing contracts). *See generally* EDWARD LEVI, AN INTRODUCTION TO LEGAL REASONING 2 (1949) (noting the announcement of rules in judicial opinions and their application in future opinions). Similarly, statutory and regulatory texts can be mined, and so can large volumes of textual private agreements.

[5] Ariel Porat & Lior Jacob Strahilevitz, *Personalizing Default Rules and Disclosure with Big Data*, 112 MICH. L. REV. 1417, 1435–36 (2014) (providing a summary of recent uses of big data for increasing the accuracy

This Article narrows its focus on the application of big data methods to legal scholarship. The goal is to develop a framework for, or at least take first steps toward, understanding how big data techniques can be used to address the positive and normative questions of conventional legal scholarship, i.e. what the law is and what the law should be. Thus, it is less concerned with evaluating whether big data techniques are particularly well-suited to legal scholarship as a question by itself, which is certainly relevant and central, and is more concerned with identifying potential avenues for application to conventional doctrinal scholarship.[6] Along the way, the analysis considers the value of traversing those avenues from a normative law and economics point of view. In other words, can big data legal studies increase social welfare, generally, through a reduction in the social costs of lawyering and

---

of search results, medical diagnoses, and credit ratings; for predicting an individual's race and ideology; for energizing a political base; for finding profitable customers, persuadable voters, and less risky insureds; and for personalizing services). *See also* Andrew McAfee & Erik Brynjolfsson, *Big Data: The Management Revolution*, HARV. BUS. REV., Oct. 2012, at 1, 9 (Oct. 2012) (asserting that "[i]n sector after sector, companies that figure out how to combine domain expertise with data science will pull away from their rivals."). For example, Daniel Katz has documented the application of big data methods for delivering legal services such as e-discovery. Daniel Martin Katz, *Quantitative Legal Prediction—or—How I Learned to Stop Worrying and Start Preparing for the Data-Driven Future of the Legal Services Industry*, 62 EMORY L. J. 909 (2013).

[6] For example, should legal scholars divert their attention from other forms of legal scholarship and allocate intellectual resources toward carrying out big data legal studies? One reason why this Article avoids that question is because a thorough answer would involve an assessment of intellectual ability across scholarly methods. Instead, comparative value of big data legal scholarship is simply assumed given that some scholars may find it beneficial to experiment with, or specialize in, its techniques. *See generally* Richard Posner, *Legal Scholarship Today*, 115 HARV. L. REV. 1314, 1324–26 (2001) (noting the benefits of scholarly specialization) [hereinafter Posner, *Scholarship Today*].

judging, or say, an increase in the awareness of socially undesirable rules (thereby placing pressure on their decline), or an increase in the use of socially desirable rules, and yet still others.[7]

	Lawmaking		*Rights Assertion*
Application	Justifying/Updating Doctrine	Justifying/Updating Non-judicial Rules	Asserting Preexisting Rights
Posture	Descriptive & Normative	Descriptive & Normative	Descriptive
Primary Object of Analysis	Opinion Texts	Extralegal Data	Extralegal Data

Fig. 1: Comparing *Big Data Lawmaking* with *Big Data Rights Assertion*

This Article does not evaluate the use of big data techniques for factual development in litigation and any attendant privacy concerns.[8] The conventional object of analysis when fashioning legal doctrine is the opinion text.[9] While the formation of a statutory or regulatory rule (or even a judicial rule) can draw upon extralegal texts and other data types such as large statistical repositories, their function when used in the context of lawmaking is generally focused on new

---

[7] Normative roots may certainly dig deeper, but as a useful point of reference, Richard Posner identifies the German historicist Fredrich Carl von Savigny as progenitor of the three crucial ingredients of American legal scholarship: (1) legal texts, (2) tacit extralegal purposes, and (3) an audience for reform comprised of other legal scholars, judges, legislators, and practicing lawyers who are all interested in improving the law and legal institutions. *Id*. at 1314–15. Early examples which follow this pattern are Holmes' *The Common Law*, Henry Maines' *Ancient Law*, and Thayer's well-known article on judicial review. *Id*. Later examples, unfortunately, too often depend on specialized political views that drastically limit the size of (3) and place the scholarship within an echo-chamber in proportion. Posner notes a quantitative increase in law professors and an attendant need for differentiation, which may explain the apparent lack of consensus. *Id*. at 1322. Yet on the other hand an increased quantity output of scholarship is not a sufficient (or necessary) condition for specialization and division.

[8] Similarly, this Article does not evaluate the use of big data in criminal sentencing and corrections policy.

[9] *See, e.g.*, LEVI, *supra* note 4, at 2.

rule justification.[10] In contrast, big data methods can also be applied to develop factual bases for asserting preexisting rights previously established by law. (Figure 1). No explicit rule-making or normative posture is preponderant in this type of application. For example, researchers at the University of Pisa have developed a technique for mining databases for evidence of discrimination.[11] Their algorithm considers group underrepresentation "as a quantitative measure of the qualitative requirement that people in a group are treated 'less favorably' than others, or such that 'a higher proportion of people without the attribute comply or are able to comply' to a qualifying criteri[on]."[12] As noted in a recent report of the Executive Office of the (U.S.) President:

> The same algorithmic and data mining technologies that enable discrimination could also help groups enforce their rights by identifying and empirically confirming instances of discrimination and characterizing the harms they caused. Civil rights groups can use the new and powerful tools of big data in service of equal treatment for the communities they represent. Whether big data will build greater equality for all Americans or exacerbate existing inequalities depends entirely on how its

---

[10] For an interesting and novel approach where news data is mined in real time in order to create an early warning system for prompting regulatory change, *see* Joshua Mitts, *Predictive Regulation*, SOCIAL SCIENCE RESEARCH NETWORK, June 27, 2014, *available at* http://ssrn.com/abstract=2411816.

[11] *See* Salvatore Ruggieri, et al., *DCUBE: Discrimination Discovery in Databases*, SIGMOD '10 (2010) (Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data), http://dl.acm.org/citation.cfm?id=1807298&dl=ACM&coll=DL&CFID=530829708&CFTOKEN=69183305.

[12] *Id*. at 1.

> technologies are applied in the years to come, what kinds of protections are present in the law, and how the law is enforced.[13]

So conceived, big data methods might be used to develop facts that enforce preexisting rights that have been previously established by law.[14] The objects of this type of analysis are the repositories of the facts themselves, e.g. databases of historical decision-making that range over various domains of protected rights.[15] Instead, big data legal

---

[13] Executive Office of the President, Big Data: Seizing Opportunities, Preserving Values (May 2014), *available at* http://www.whitehouse.gov/sites/default/files/docs/big_data_privacy_report_5.1.14_final_print.pdf. On the dangers of using big data to discriminate, *see* Solon Barocas & Andrew D. Selbst, *Big Data's Disparate Impact*, 104 CAL. L. REV. (forthcoming 2016).

[14] Similarly, this Article does not evaluate the use of big data for criminal sentencing and corrections policy. Like the use of big data tools for factual development in litigation, big data tools for criminal sentencing are simply used for applying preexisting law. There is a burgeoning literature with enthusiastic supporters who tout increased precision in predicting recidivism on the one hand and enthusiastic detractors who note that imprisonment should be primarily based upon past conduct on the other. For a thoughtful discussion of the potential benefits and costs, *see* Letter from Jonathan J. Wroblewski, Director, Office of Policy and Legislation, to Judge Patti B. Saris, Chair, United States Sentencing Commission (Jul. 29, 2014) (on file with author). For a critique, *see* BERNARD E. HARCOURT, AGAINST PREDICTION: PUNISHING AND POLICING IN AN ACTUARIAL AGE (2006). For a description of potential applications, *see* Lyria Bennett Moses & Janet Chan, *Using Big Data For Legal and Law Enforcement Decisions: Testing the New Tools*, 37 U. NEW S. WALES L. J. 643 (2014). Interestingly, Pennsylvania and Tennessee both mandate the use of risk assessments and explicitly permit use of empirical tools. *See* 42 PA. CONS. STAT. ANN. § 2154.5 (2009); TENN. CODE ANN. § 41-1-412(b) (2013).

[15] Ruggieri, et al., *supra* note 11, at 1 (identifying the domains of credit and insurance; sale, rental, and financing of housing; personnel selection and wages; and access to public accommodations, education, nursing homes, adoptions, and health care).

scholarship is both positive and normative. It shares with doctrinal legal scholarship the conventions of description and a forward-looking posture. Statistics and databases may be of interest, but the primary object of analysis is the legal text.

This Article is divided into two parts. Part one takes first steps toward establishing a way of thinking about big data legal scholarship. It begins by considering how big data methods can positively describe the law and identifies three types of study: of taxonomy, of realism, and of policy. First steps toward positive application are followed by first steps toward normative application. Three normative goals are identified: to clarify doctrine, to advocate comparative justice and predictability, and to advocate better rules. Part two aims to demystify the big data research process by providing a practitioner's guide, which details the steps a newcomer can take to develop a big data legal study. The guide has two goals: to convince an interested legal scholar to consider carrying out big data study and to facilitate broader dialog amongst those who study legal doctrine—regardless of whether they are interested in big data methods. By prying open the black box just a little, part two hopes to further the conversation about law—between big data legal scholars and their counterparts using other methods.

II. TOWARD A FRAMEWORK FOR BIG DATA LEGAL SCHOLARSHIP

A. Toward a Positive Application

As emphasized by Joshua Fischman, legal scholars should prioritize normative questions over positive assessments

and allow substantive questions of policy to drive their choice of methods.[16] Nonetheless, this Article leads with big data's positive component in order to place its normative component in sharper relief. Priority is assigned for exposition.

Broadly speaking, big data methods can be used to describe legal rules and legal theory. For example, Jonathan Macey and Joshua Mitts have applied big data methods to develop a taxonomy of corporate veil-piercing;[17] I have applied big data methods to develop a taxonomy of corporate successor liability.[18] Daniel Young has applied big data methods to study Bruce Akerman's constitutional moments;[19] Lea-Rachel Kosnik has applied big data methods to study Ronald Coase's transaction costs.[20] What has scientifically emerged thus far, at least in terms of big data's ability to describe legal rules, is essentially taxonomic. And in terms of its ability to describe legal theory, big data provides just one more empirical method for falsification.

---

[16] Joshua B. Fischman, *Reuniting 'Is' and 'Ought' in Empirical Legal Scholarship*, 162 U. PA. L. REV. 117, 154 (2013) [hereinafter Fishman, *Reuniting*].

[17] Jonathan Macey & Joshua Mitts, *Finding Order in the Morass: The Three Real Justifications for Piercing the Corporate Veil*, 100 CORNELL L. REV. 99 (2015).

[18] Frank Fagan, *Successor Liability from the Perspective of Big Data*, 9 VA. L. & BUS. REV. 391 (2015).

[19] Daniel Taylor Young, Note, *How Do You Measure a Constitutional Moment? Using Algorithmic Topic Modeling to Evaluate Bruce Akerman's Theory of Constitutional Change*, 122 YALE L. J. 1990 (2013).

[20] Kosnik, *supra* note 4, at 198 (noting that "[t]his paper investigates the completeness of hydroelectric license contracts over a nearly three decade time span and finds that as environmental concerns increase, so does contract flexibility, ultimately confirming the predictions of transaction cost theory").

1. Taxonomic Studies

Taxonomic studies are important nevertheless, especially where there exists scholarly dispute over which categorical set of facts leads to judicial application of a particular doctrine, or where there exists ongoing variation in judicial rationale for the same. Both of these discrepancies were present when Macey and Mitts approached veil-piercing with big data methods. Their effort was directed toward resolving (1) a scholarly dispute whether veil-piercing doctrinal standards exhibit coherence or whether those standards are "characterized by ambiguity, unpredictability, and even a seeming degree of randomness";[21] and (2) why judges appear to apply standards differently.[22] Either of these grounds present an intellectual opportunity for big data taxonomic studies. With respect to veil-piercing, the intellectual and doctrinal stakes were relatively high considering that Stephen Bainbridge called for the abolishment of the doctrine for its seeming lack of coherence and non-uniformity.[23] Through application of big data methods, Macey and Mitts were able to find coherence in the morass of over 9,000 veil-piercing decisions and establish an authoritative taxonomy.[24] The taxonomy draws its authority from the impressive fact that it parses 9,380 decisions, which were culled from a database of 2.5 million.[25] Size, for better or worse, has always been a hallmark of quality and thoroughness

---

[21] Stephen M. Bainbridge, *Abolishing Veil Piercing*, 26 J. CORP. L. 479, 507 (2001).
[22] Macey & Mitts, *supra* note 17, at 99.
[23] Bainbridge, *supra* note 21, at 479.
[24] Macey & Mitts, *supra* note 17, at 113.
[25] *Id*. at 141, n.166.

in doctrinal scholarship.[26] Big data methods can leverage that hallmark.

The taxonomy also draws authority from its limited subjectivity. Taxonomies based upon hand classification are inherently biased because an analyst must decide on a classification scheme.[27] While classifications based upon biological characteristics, for example, are relatively straightforward and mildly contestable at worst (think six legs means insect; thus, a centipede is not an insect)[28], classifications based upon legal characteristics are often messy and mildly contestable at best. As noted by Macey and Mitts, the messiness was particularly acute with veil-piercing doctrine:

> Coding schemes—indeed, quantitative analysis more generally—necessarily reflect an imperfect approximation of the qualitative complexity of each case. But using mechanical coding to identify determinants of veil piercing is particularly imprecise because it places substantial discretion in the hands of human coders, whose application of judgment can vary between individuals and even from case to case by the same individual.[29]

---

[26] *See, e.g.*, Brian Leiter, *Red Alert: Fordham Law Review is *Still* Interested in "Quality" of Articles—unlike Columbia, Cornell, Harvard, Michigan, Stanford, Texas, Virginia, et al.,* Law Professors Blogs Network (Aug. 16, 2005), http://leiterlawschool.typepad.com/leiter/2005/08/attention_fordh.html.
[27] Macey & Mitts, *supra* note 17, at 112–13.
[28] HOWARD ENSIGN EVANS, LIFE ON A LITTLE-KNOWN PLANET 26 (2d ed. 1984) (noting the difference).
[29] Macey & Mitts, *supra* note 17, at 112.

Heightened subjectivity can lead to differing classification results which limit the scope of academic consensus and places a downward pressure on producing a testable theory or taxonomy. This is because those that disagree with the initial classification results are more likely to reject a theory grounded in those results. And because the theory will have less appeal, the scholar will have less incentive to produce it and empirically test it (given that she values wide appeal of her scholarship of course).[30]

Big data methods, instead, offer classification techniques that severely limit the subjective bias of the analyst and thereby promote consensual advancement. Through a method known as topic modeling, the analyst can use an algorithm to evaluate a practically unlimited number of judicial decisions at once—without specifying qualitative classes or

---

[30] This pattern was plainly exhibited in veil-piercing scholarship:

> Indeed, the contradictory findings of Peter Oh and Robert Thompson with respect to tort versus contract cases serve as a useful case in point. It is likely that Oh's reclassification of many contract cases under a new category of "fraud" is responsible for explaining this difference, which Oh tacitly acknowledges. Regardless of whether fraud falls into contract or tort as a doctrinal matter, from an empirical standpoint this coding scheme is purely arbitrary: it reflects the coder's choice alone. It is impossible to view Oh's study as contradicting or lending support to Thompson's because the former simply uses a different coding scheme to classify cases than the latter. The coder has determined the result, not the data.

Moreover, none of these studies evaluate any theory of veil piercing.

Id. at 112–13 (footnotes omitted).

case characteristics ad hoc.[31] The algorithm, from its point of view, performs aimless work. The analyst simply specifies the number of topics to be modeled, and the algorithm simply outputs that number of word lists. The analyst must then create classification categories based upon the contents of the individual lists. For example, the topic model used in Fagan (2015) returned four lengthy lists with one consisting of terms such as: "petition reorganization," "legal interest," "transfer purchase," "liquidation distribution," and "public auction."[32] With the aid of that list, I was tasked to create a category (most cleanly related to bankruptcy).

Category creation requires a certain amount of subjectivity, but that amount is severely limited when compared to other methods—and in scientifically meaningful ways. Consider first, that categories are based upon algorithmically generated word lists that can be replicated by other researchers who use the same dataset.[33] Any dispute over category choice is therefore limited to subjective interpretation of the word list. Second, unlike traditional classification methods, topic modeling considers the complete contents of a legal text with equivalent importance. Terms such as "under capitalization" are evaluated with the same systematic rigor as "hot air." Any pattern or topical structure manifest in the word lists, therefore, is mechanically based upon the texts—irrespective of their contents. This type of frankness with

---

[31] *See generally* David M. Blei, *Probabilistic Topic Models*, 55 COMM. OF THE ACM 77 (2012); *infra* Section II.E.2.
[32] Fagan, *supra* note 18, at 416.
[33] Some researchers post their datasets online for replication and extended use. *See* Macey & Mitts, *supra* note 17, at 149 n.181. Conventionally, if one can speak of convention at this point, all big data legal studies describe how the dataset was created, typically by reporting the specific keyword search and database. *See, e.g.*, Fagan *supra* note 18, at 405 n.51 (reporting the specific keyword search performed in the CourtListener database).

textual data could, possibly, be sacrificed when the data are approached non-algorithmically and with a human touch. Entire topics, or at least subtle classification influences, might be missed or papered over as a result. Finally, topic modeling limits subjectivity through the use of substantially enlarged datasets. As the number of data increase, a more objective picture is more likely to emerge.[34] On balance then, big data topic modeling presents clear advancement in reducing subjectivity when compared with traditional hand-classification techniques. Combined with its even clearer advantage in comprehensiveness when an analyst mines a sufficient universe of legal texts, it holds immediate potential to generate taxonomies that yield broader acceptance.

2. Legal Realist Studies

In his book *How Judges Think*, Richard Posner notes wryly that if judges did nothing but apply clear rules of law, "[t]hen judges would be well on the road to being superseded by digitized artificial intelligence programs."[35] For their part, legalists claim that judges merely "apply rules made by legislatures or framers of the Constitution (or follow precedents, made by current or former judges, that are promptly changed if they prove maladapted to current conditions)."[36] Judges state those rules in their opinions and apply them without bias to sets of facts, which themselves are

---

[34] *Cf.* JEFFREY M. WOOLRIDGE, INTRODUCTORY ECONOMETRICS: A MODERN APPROACH 765 (5th ed. 2013) (explaining how an independent identically distributed variable converges toward its expected value as the number of times it is observed approaches infinity).
[35] RICHARD POSNER, HOW JUDGES THINK 5 (2008).
[36] *Id*. at 4.

also determined without bias.[37] Realists, instead, claim that the judicial temperament matters. Judicially-made doctrines and decisions depend in part upon the judges' incentives, "which may in turn depend on the judges' cognition and psychology, on how persons are selected (including self-selected) to be judges, and on terms and conditions of judicial employment."[38] Application and determination of rules and facts can, therefore, partly depend upon judges' "motivations, capacities, mode of selection, professional norms, and psychology."[39] Current judicial practice does not involve explicitly stating any of these up front, and it very likely will remain this way for the foreseeable future. Thus, if judicial temperament matters in decision making, it must be measured indirectly. Big data analytics provide a powerful research platform for empirically testing the tenets of legal realism. Empirical techniques now exist for untangling the (potentially) uneasy relationship between judicial texts and judicial outcomes. The analytical results are able to support the assertion that a given judicial rationale, even when made textually explicit, is not (or conversely, may be) driving a pattern of judicial decision making.

An imposing body of empirical literature supports the claim that judges pursue political objectives—just one of the judicial pursuits identified by legal realists.[40] Many of these

---

[37] *See* LEVI, *supra* note 4, at 8–9 (describing the movement of concepts into and out of law, but noting that there exists a period where "the concept is more or less fixed").
[38] *Id*. at 5.
[39] *Id*.
[40] *See, e.g.*, William M. Landes & Richard A. Posner, *Rational Judicial Behavior: A Statistical Study*, 1 J.L. ANALYSIS 775 (2009); Thomas J. Miles & Cass R. Sunstein, *Do Judges Make Regulatory Policy? An Empirical Investigation of Chevron*, 73 U. CHI. L. REV. 823 (2006); CASS R. SUNSTEIN, DAVID SCHKADE, LISA M. ELLMAN, & ANDRES SAWICKI, ARE

studies rely on specific attributes of the judge such as age, race, jurisdiction, and party of the appointing executive.[41] Big data analytics, by way of contrast, can rely on the judicially-determined facts of the case to the extent those facts are captured by actual judicial word choice in the opinion texts. For example, Macey and Mitts found that whether a corporation was undercapitalized, an alter ego, a mere instrumentality; whether its stock was held mostly by its owners; or whether it had failed to issue dividends or had rarely issued them at all, mattered little in judicial application of corporate veil-piercing doctrine—without looking at characteristics of individual judges or by hand-classifying cases categorically by fact pattern.[42] Their result might be interpreted a number of ways, but surely one interpretation involves that judges pay lip-service to those reasons and dispose of cases for other reasons.[43]

Not only can big data uncover the posturing of a particular rationale, it can additionally show that other rationales matter, particularly rationales grounded in judicial attributes. For example, judges with particular attributes may tend to hold for plaintiffs when there is commingling of assets by small business defendants or where plaintiffs are public entities. A number of possibilities exist across many areas of

---

JUDGES POLITICAL?: AN EMPIRICAL ANALYSIS OF THE FEDERAL JUDICIARY (2006) [hereinafter ARE JUDGES POLITICAL?].

[41] *See, e.g.*, SUNSTEIN ET AL., ARE JUDGES POLITICAL?, *supra* note 40 at 20.

[42] Macey & Mitts, *supra* note 17 at 102 ("Thus it is our view that all of the standard litany for justifications for disregarding the corporate form, which include failure to observe corporate formalities, undercapitalization, alter ego, mere instrumentality, ownership of all or most of the stock in the company, payment of dividends, failure to pay dividends, etc., are mere proxies for one of the three core reasons for piercing described above.").

[43] Macey and Mitts made this conclusion and asserted that other rationales were driving veil-piercing decisions. *Id.* at 152.

law. Areas where judicial bias is suspected already, e.g. criminal conviction and sentencing with respect to defendant attributes,[44] free speech with respect to speech contents,[45] and various areas of administrative law[46] are especially ripe for study.

Exactly how might this scholarship look? Big data legal scholars can regress opinion texts against judicial outcomes. Figure 6 depicts a spreadsheet of documents, ordered by rows, with the words contained within those documents ordered by columns.[47] Consider that each document represents a legal text associated with an outcome. For example, each document may represent a judicial opinion where a corporate successor is either found liable or not liable. Or each document may represent a judicial application of a criminal law doctrine where the defendant is found guilty or not guilty. All of the words in a given opinion are assumed to potentially relate to the judicial outcome. The analyst determines a relationship between the words and outcome by regressing the words as independent variables against the

---

[44] *See* David Abrams, Marianne Bertrand & Sendhil Mullainathan, *Do Judges Vary in Their Treatment of Race?* 41 J.L. STUD. 347, 347 (2012) (finding between-judge variation in incarceration rates for African-American and White defendants, but not sentencing).

[45] *See* Lee Epstein, Christopher M. Parker & Jeffrey A. Segal, *Do Judges Defend the Speech They Hate? In-Group Bias, Opportunism, and the First Amendment*, APSA 2013 Annual Meeting Paper 8 (2013) (finding that judicial bias along the liberal/conservative dimension impacts free speech decisions).

[46] *See* Thomas J. Miles & Cass R. Sunstein, *The New Legal Realism* 75 U. CHI. L. REV. 831, 846 (2008); Jason D. Vendel, Note, *General Bias and Administrative Law Judges: Is There a Remedy for Social Security Disability Claimants*, 90 CORNELL L. REV. 769, 771 (2005) (asserting judicial bias in disposition of social security claims).

[47] *Infra* Section II.D.2.

dependent variable of judicial outcome.[48] The results are then interpreted to support a given theory. The key difference between big data regression and traditional regression is the sheer number of covariates. A dataset consisting of a 20,000-word vocabulary will consist of 20,000 covariates. Obviously, judges' use of some of those words will contain no underlying theoretical connection to judicial outcome.[49] But word choice, just as obviously, is not meaningless. Otherwise judges could author opinions with randomly chosen words.

Regression of judicial texts can go a long way toward addressing omitted variables bias when studying the relationship between text and outcome. Because the analyst narrows her inquiry to how a judicial outcome is reached only through judicial word choice, and because every word contained within a judicial opinion is counted as a variable, omitted variables bias is severely reduced.[50] So long as the inquiry is properly narrowed and explicitly acknowledged while reaching conclusions, big data methods can provide an

---

[48] More precisely, groups of words, known as n-grams, are regressed against outcomes. Using groups of words as regressors acknowledges the lexical structure of texts. *See* Macey & Mitts, *supra* note 17, at 144 ("Because we examine two-word phrases rather than single words, the bag of words assumption is even more plausible, because it is even less likely that two sets of two-word phrases will systematically follow each other."). Judicial opinions, moreover, often fully discuss a doctrine before applying it to a set of facts. Thus, both application and non-application of a doctrine will use similar words or groups of words. *Id*. at 148 n.180. The unique facts of the case will likely provide the greatest source of variation used for prediction.

[49] *But see* Milton Friedman, *The Methodology of Positive Economics*, in MILTON FRIEDMAN, ESSAYS IN POSITIVE ECONOMICS 1 (1953) (asserting that no underlying theoretical connection between independent and dependent variables is needed for scientific contribution).

[50] However, it is not entirely eliminated when various words such as "stop words" and numbers are removed from the dataset or vocabulary. *See infra* notes 131–33 and accompanying text.

effective and powerful research platform for advancing our understanding of the limits and scope of legalism and realism, at least with respect to the relationship between judicial text and judicial outcome.

In addition to narrow inquiries that focus on text, big data methods can support wider inquiries into judicial behavior. When the words of the opinion texts are used as covariates, they can function as surrogates for the judicially determined facts of a case. One can think of the words as an enormous set of dummy variables, which can be used for holding the facts constant. This technique could represent an important advance in reducing subjectivity from traditional hand coding of cases as explained above in the section on taxonomy.[51] Once the facts are controlled for, the remaining judicial attribute covariates such as age, and appointed party can be evaluated more accurately.[52]

3. Policy Studies

To appreciate the role of regression analysis as a tool for the big data legal analyst, consider the recent history of one of the most influential forms of empirical legal scholarship, viz., econometrics applied to legal questions (aka empirical law and economics). In their survey of the field, Jonah Gelbach and Jonathan Klick note immediately that "the central problem in much empirical work is omitted variables bias."[53] Before

---

[51] *See* discussion *supra* Section I.A.1.

[52] What to do normatively with any finding of interjudge disparity is clearly discussed in Fischman, *Reuniting*, *supra* note 16, at 153–54 and in the discussion *infra* Section I.B.2.

[53] Jonah B. Gelbach & Jonathan Klick, *Empirical Law and Economics*, Research Paper No. 14-39, University of Pennsylvania School of Law Institute for Law and Economics (2015), *available at* http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2507324.

the mid-1990s empirical law and economics simply added more variables to address the problem.[54] However, if the variables are unknown, then they cannot be added. A follow-up approach was to admit bias but speculate about its nature.[55] However, the results of this approach were, naturally, always contestable because bias was always admitted.[56] By the mid-1990s, empirical law and economics began implementing the difference-in-differences approach to address research design problems.[57]

Typically, the analyst studies the effects of a new policy by comparing a jurisdiction that adopts the policy with a jurisdiction that does not. So long as unknown variables are realistically assumed to be the same across those jurisdictions, any difference between the two can be understood as caused by the change in policy.[58] However, jurisdictions do not adopt policies randomly. Something is happening within the jurisdiction that is driving the choice to adopt.[59] This "something" may or may not be happening in the other jurisdictions which are used for comparison.[60] For this reason, state-of-the-art studies use instrumental variables (IV) and similar techniques to purge the jurisdictionally-centered (or more generally, the endogenous or internal) nature of policy

---

[54] *Id*. at 2.
[55] *Id*.
[56] *See* Edward E. Leamer, *Let's Take the Con Out of Econometrics*, 73 AM. ECON. REV. 31 (1983).
[57] Gelbach & Klick, *supra* note 53, at 3.
[58] This is an oversimplification. Technically, when a handful of assumptions are true, then the results are "consistent . . . for . . . the average effect of the . . . change in law or policy—in the jurisdictions where the change occurred." *Id*. See the citation for a complete explanation.
[59] *Id*. at 11–12.
[60] *Id*.

choice.[61] Nonetheless, a valid instrumental variable requires an assumption that cannot be tested. Its validity can only be evaluated for reasonableness.[62] Thus, Gelbach and Klick conclude that "[u]ltimately, it is an unavoidable if uncomfortable fact of empirical life that untestable assumptions, and untestable good judgment about them, are indispensable to the measurement of causal effects of real world policies."[63] However, the good news is that new empirical techniques like instrumental variables have contributed to the restoration of some consensual level of credibility to empirical law and economics.[64] The new methods are not perfect, but they "have uncovered compelling evidence on [issues] of great policy import."[65]

Difference-in-differences research designs augmented with quasi-experimental approaches, such as IV, are usually concerned with isolating a crucial effect of policy change. This effect often constitutes the policy's justification. For example, does increased policing reduce crime?[66] Do liberal takeover policies increase market value of firms?[67] Do heightened

---

[61] *Id*. at 3.
[62] *See, e.g.*, Frank Fagan & Firat Bilgel, *Sunsets and Federal Lawmaking: Evidence from the 110th Congress*, 41 INT'L REV. L. & ECON. 1, 2–3 (2015) (assuming that number of legislative sponsor's offspring is not correlated with whether a bill becomes law).
[63] Gelbach & Klick, *supra* note 53, at 4.
[64] *See id*. at 9–17 for a discussion of other "quasi-experimental" methods that accomplish similar goals of the instrumental variables approach.
[65] *Id*. at 15 (highlighting the credibility of quasi-experimental methods in demonstrating that increased policing impacts crime).
[66] *See, e.g.*, Jonathan Klick & Alexander Tabarrok, *Using Terror Alert Levels to Estimate the Effect of Police on Crime*, 48 J.L. & ECON. 267 (2005).
[67] *See, e.g.*, Jonathan Klick & Robert H. Sitkoff, *Agency Costs, Charitable Trusts, and Corporate Control: Evidence from Hershey's Kiss-Off*, 108 COLUM. L. REV. 749 (2008).

pleading requirements increase settlement rates?[68] For some of these studies, especially those that involve the analysis of case law, controlling for judicially-determined facts may be useful. Big data methods can reduce subjectivity through classifying case types algorithmically with topic modeling or through using the words of opinion texts as covariates. In other instances, preliminary analysis with big data methods may reveal variation that can later be exploited with quasi-experimental methods.

This section has sought to take first steps toward constructing a research framework for the positive application of big data methods to legal scholarship. Over time, other contours will surely emerge in response to new techniques for evaluating data and sharpened normative questions posed by the research community. It is to this latter issue that this Article now turns.

B. Toward a Normative Application

1. Clarifying Doctrine

Doctrinal taxonomies constructed with big data methods are more advanced than their counterparts constructed with traditional methods.[69] By relying upon large datasets and algorithmic topic modeling, the analyst can severely limit subjectivity through machine-evaluation of the words contained within a body of judicial texts.[70] The results are

---

[68] *See, e.g.*, David Freeman Engstrom, *The Twiqbal Puzzle and Empirical Study of Civil Procedure*, 65 STAN. L. REV. 1203 (2014).
[69] *Supra* notes 27–34 and accompanying text.
[70] *Id*.

scientifically replicable.[71] Big data taxonomies, therefore, represent an advance in objectivity and can generate greater academic consensus as a result.

The normative potential of enhanced objectivity flows from big data legal scholars to judges, and the administration of law broadly, because more clearly defined taxonomies lower the costs of judging and litigating. Ideally, clarity comes from the ability of algorithmic processing to identify the general judicial rationales for application of a particular doctrine. General rationales that are not used for supporting doctrinal application, but that mistakenly became part of the hand-classified taxonomy, or a multi-factor checklist for application of a legal standard, will erode further.[72] As a result, the doctrine can become more streamlined and take more definitive shape over time. For example, the big data taxonomy of corporate veil-piercing has shown that courts pierce the veil for statutory mandate, misrepresentation, and bankruptcy values, but that courts do not pierce for undercapitalization and other reasons. Nonetheless courts historically have evaluated veil-piercing fact patterns for undercapitalization and other reasons, which marginally raises the costs of judging and litigating. As the veil-piercing study becomes more accepted and the scholarship continues to advance,[73] undercapitalization and

---

[71] *Id*.

[72] For a view on the undesirability of multi-factor lists, *see* Exacto Spring Corp. v. Comm'r, 196 F.3d 833, 833–34 (7th Cir. 1999) (Posner, J.).

[73] On the one hand, broader academic consensus should generate a greater confidence in a given theoretical taxonomy, which in turn, should generate greater incentive to build on that taxonomy in a given direction. *See supra* note 30 and accompanying text. For example, a big data theoretical taxonomy that eschews undercapitalization as a veil-piercing rationale may more likely generate further empirical study than a traditional theoretical taxonomy which relies on hand-classification. *See supra* note 30 and accompanying text. The fact that the big data study is more scientifically

those other reasons as rationales for veil-piercing should erode and lower judging and litigating costs.

Doctrinal clarity reduces the time taken to evaluate untenable rationales for doctrinal application and can increase uniformity between judges and across jurisdictions. For example, a judge may supplement her traditional case analysis by directly citing to an initial big data taxonomy study because she finds the study compelling. Over time, as the taxonomy percolates through the case law, the consensual strength of the taxonomy will develop throughout the judiciary, which in turn, will strengthen the ability of the taxonomy to provide a clear and persuasive rationale for a judicial holding. A stronger theoretical taxonomy that more sharply defines the general rationales for a holding will justify a required holding more often. In other words, higher quality will beget broader use.[74]

---

advanced increases the likelihood that a more tailored study, which shows for example that undercapitalization matters little and other rationales matter more, will produce results that generate broader acceptance. Scholars therefore have a greater incentive to produce it. *See generally*, THOMAS KUHN, THE STRUCTURE OF SCIENTIFIC REVOLUTIONS *passim* (1962) (describing the role of contemporary scientific study as promoting consensus). Over time, the general rationales for application of a particular doctrine which have been identified by algorithmic processing will clarify further as the scholarship continues to advance.

[74] Richard Posner correctly notes that "[t]he … theory that Supreme Court precedents depreciate more slowly because they are more authoritative, i.e., valuable is economically unsound [because the] rate at which a good depreciates is not a function of its value." RICHARD A. POSNER, ECONOMIC ANALYSIS OF LAW 588 (5th ed. 1998). He asserts instead that Supreme Court precedents depreciate more slowly because that court is more selective in its choices to review, and the cases that it chooses to hear apply far more generally. *Id*. Indeed, general precedents depreciate more slowly than specific precedents. William M. Landes & Richard A. Posner, *Legal Precedent: A Theoretical and Empirical Analysis*, 19 J.L. & ECON. 249 (1976). The broad appeal of doctrinal taxonomies is consistent with this reasoning. Doctrinal taxonomies are general and define broad categories of

To the extent that big data scholarship flows downstream to the judiciary then, judges will be able to make more broadly-accepted decisions, which will decrease the time to decision as a result. Any decrease in time could be further augmented by any increase in uniformity. Uniformity decreases the uncertainty of judgment and judging, and consequently increases time to decision, decreases reversible error, and increases settlement rates.[75] All of these benefits can lead to reduced costs of administering judgment. Thus, the normative goal of the scholar developing a big data taxonomy study should be to produce doctrinal clarity in order to reduce the decision making costs of individual judges and litigants. Doctrinal clarity is a clear normative objective that directly impacts the quality of the judicial process. By contrast, indirect proxies for judicial ability and performance such as citation counts, reversal rates, and interjudge disparities, require explicit connection to an "intended measure of merit."[76]

So conceived, big data theoretical taxonomies have potential to leverage objectivity to generate broader consensus. They streamline old doctrine by clarifying rationales that are used or not used for doctrinal application. Thus, the aim of streamlining doctrine follows the Llewellynian formula of determining what the law is doing before determining what the law ought to be doing, though with one important caveat:

---

judicial rationale that apply to widely varying sets of facts. For precisely this reason, once they are authoritatively established, they are widely used and depreciate slowly.

[75] Similarly, Dru Stevenson and Nicholas Wagoner have asserted that settlements will increase because lawyers themselves will perform big data analysis, predict the outcome of a case, and reach a bargain informed by their predictions in litigation and transactional disputes. *See* Dru Stevenson & Nicholas Wagoner, *Bargaining in the Shadow of Big Data*, 67 FLA. L. REV. 1337 (2015).

[76] Fischman, *Reuniting*, *supra* note 16, at 130.

streamlining involves eliminating rationales not simply on the basis of their desuetude, but on the additional basis that at least one other rationale is always present and controlling. This means that the normative force of an abrogation argument is not exclusively derived from the non-use of the rationale in question. Instead, normative force is derived from the non-use of a rationale only when at least one other judicial basis is always present and controlling. Just as too much icing on a cake can obscure its taste, too much icing on a case can obscure its rationale. And while the normative goal of lowering the costs of judging is no stranger to legal scholarship,[77] big data legal scholarship is uniquely positioned to generate a broader consensus about the conventional positive questions of law. Big data legal scholars should aim toward streamlining judicial doctrine with the new techniques.

2. Advocating Comparative Justice and Predictability

Just as big data legal scholarship can streamline judicial doctrine and reduce the costs of judging, it can raise awareness of judicial posturing and promote transparent application of a given rationale. Section I.A.2. highlighted the use of regression analysis for understanding the relationship between judicial texts and outcomes and for supporting wider inquiries into the behavior of judges. Both types of positive studies drive the same normative questions. Broadly, the normative question of interest is how does systemic change force transparent application of law, which increases comparative justice and predictability of outcomes in turn. With respect to

---

[77] *See, e.g.*, Richard A. Posner, *An Economic Approach to Legal Procedure and Judicial Administration*, 2 J.L. STUD. 399 (1973) (asserting that the normative role of judicial administration is to minimize the error costs and direct costs of judging).

the former, legal principles may require that justice is due in identical scenarios. For example, two offenders equally culpable should receive the same criminal sentence, "even if those principles do not uniquely determine what the sentence should be."[78] The normative thrust of big data legal realist studies should be to advocate systemic changes that raise transparent application of judicial rationale in order to promote comparative justice. With respect to the predictability of judicial outcomes, "notice will necessarily be inadequate to the extent that the application of law depends upon which judge [or judge type] is deciding each case."[79] Thus, another normative goal of big data legal realist studies should be to advocate systemic changes that raise transparency in order to promote predictability.

In the past, judicial posturing has been indirectly measured primarily through studies of interjudge disparity.[80] As noted by Fischman, "studies on interjudge disparity have enormous potential for improving the quality of systems of adjudication."[81] However, these studies raise complicated measurement questions.[82] Does disparity indicate judicial

---

[78] Fischman, *Reuniting*, *supra* note 16, at 149.
[79] *Id.* n.166.
[80] *See id*. at 146 (noting that "[a] large body of empirical research has sought to measure the degree to which systems deviate from the "central feature of the rule of law … [viz.,] application of legal force is governed by publicized rules rather than 'the predilections of the individual decisionmaker,'" and that research in this area has "typically documented statistical disparities among judges") (quoting RONALD A. CASS, THE RULE OF LAW IN AMERICA 17 (2001)).
[81] *Id*. at 153.
[82] Joshua B. Fischman, *Measuring Inconsistency, Indeterminacy, and Error in Adjudication*, 16 AM. L. & ECON. REV. 40 (2014) [hereinafter Fischman, *Measuring IIE*].

inconsistency[83] and a violation of comparative justice, or does inconsistency indicate indeterminacy of law or error about the "correct" outcome of case?[84] (And what is "correct")?[85] Fischman offers a useful metric for disparity. One can estimate the upper and lower bounds of inconsistency, defined as judicial disagreement on a randomly selected case.[86] However, in practice estimation of the bounds of judicial inconsistency rates "will be of limited use in policy making [though an] alternative approach is to exploit data on case characteristics."[87]

Big data methods allow the analyst to measure the relationship between judicial texts and outcomes.[88] By assuming that the words of the judicial text represent the judicially-determined facts of the case, the analyst can use those words as covariates and hold the effect of facts constant. This would permit the analyst to measure the causal effect of judicial attributes on the upper and lower bounds of judicial inconsistency rates—all while controlling for the facts of the case.[89] Using the words of opinion texts as a proxy for judicially-determined facts may be particularly suited to analysis of lower court decisions because lower courts

---

[83] Inconsistency is defined as judicial disagreement on a randomly selected case. *Id*. at 42.
[84] Fischman, *Reuniting, supra* note 16, at 153.
[85] *Id*. at 153 (noting that the premise that every case has a unique correct outcome is controversial and asserting that "interpretations of statistical disparity will necessarily require much stronger assumptions […] about the correct answers to various kinds of cases.").
[86] *Id*. at 153.
[87] Fischman, *Measuring IIE*, *supra* note 82, at 58.
[88] *See* discussion *supra* Section I.A.2.
[89] A rudimentary model could take the form of: Fischman bounds of judicial outcome = judicial attributes + claimant attributes + administrative or legal system attributes + dummies of all words contained within all judicial texts. Standard quasi-experimental techniques could supplement this form.

determine facts, and "the random assignment of judges to [lower court] cases is a sort of natural experiment that permits plausible causal inferences about the effect of judicial characteristics on outcomes."[90] In addition, using judicial texts to control for facts may represent an advance over hand-classification of case types based upon subjective interpretation of the coder dependent upon research design.[91]

3. Advocating Better Rules

Big data methods can be used to advocate better rules. Policy studies examine the impact of policy change on normative goals.[92] They advocate changes in policy by demonstrating favorable impacts.[93] Policy studies often examine changes that occur in the world such as reduced crime, increased market values, and increased settlement rates in response to policy choice.[94] In other words, policy studies examine changes external to the law itself and normatively advocate changes in policy accordingly. Big data methods can increase the accuracy of these studies when judicially-determined facts can be fruitfully used as additional independent variables and complement sound research design.[95]

---

[90] Miles & Sunstein, *supra* note 46, at 835; *see also* Fischman, *Measuring IIE, supra* note 82, at 7.
[91] For examples of research designs based upon case characteristics, see Fischman, *Measuring IIE*, *supra* note 82, at 19.
[92] *See* Gelbach & Klick, *supra* note 53, at 3.
[93] *See, e.g.*, *id*. at 17–19 (documenting research in the effect of increased police on crime).
[94] *Id*. at 17–24.
[95] *See supra* notes 91–93 and accompanying text. Similar to advocating changes in policy, Ariel Porat and Lior Jacob Strahilevitz assert that big data methods can be used to advocate changes in default rules and

A first sketch is necessarily incomplete. Surely there exist additional types of positive studies and normative goals that big data legal scholars will develop and embrace. This Part has merely attempted to take a first step toward generating a conversation about how big data methods can augment and enhance the traditional forms of legal scholarship. Even at an early stage, it seems that big data legal scholarship holds promise and has something exciting to tell us about what the law is, and about what the law should be.

III. PRACTITIONER'S GUIDE

Performing big data research can appear daunting for a legal scholar who may have little experience with statistics, linguistics, empirical research, or even computing. This section intends to demystify the analytical method and provide the newcomer with a brief foundation sufficient to perform a modest study using machine-learning classification, topic modeling, and regression analysis. Generally, the guide represents a modest step toward reducing the effect of big data's "Power Paradox" as identified by Neil Richards and Jonathan King by placing its tools in the hands of more people—in this case, legal scholars.[96] This guide also discusses some potential analytical pitfalls when applying regression analysis to machine-classified data and suggests a cautious response. The general approach of the guide is instructive; it may be helpful to have a paper copy on hand

---

disclosure. Porat & Strahilevitz, *supra* note 5, at 1417. Specifically, they assert that individuals should be assigned personal default rules and disclosure agreements on the basis of their own "personalities, characteristics, and past behaviors." *Id*.

[96] Neil M. Richards & Jonathan H. King, *Three Paradoxes of Big Data*, 66 STAN. L. REV. ONLINE 41, 42 (2013).

while installing and exploring the software tools for the first time. As an initial point of entry, it is useful to keep in mind that computational linguistics always involves at least a three-step procedure: collecting, preprocessing, and analyzing the data.

A. Collecting the Data

It is essential that the researcher thoroughly plans this phase of a study if she is to avoid laborious and tedious debugging work later. At its most granular level, computational linguistic data are individual words.[97] Words are component parts of documents, however, and researchers primarily collect data in the form of digital document files as a result. The technical objective of data collection, in contrast to the research objectives of a study, is to produce a group of documents that can be read by the computational software for analysis. This means that good practice involves verifying that the software can read the document type before collecting all of the document files. Most platforms, including the R computing language, can read files created with Microsoft Word (.doc), Adobe Acrobat (.pdf), or a standard text editor (.txt).[98] But not

[97] More precisely, linguists think of component parts of words, or morphs, which concern the formation of new words through a process called morphology. IGOR A. BOLSHAKOV & ALEXANDER GELBUKH, COMPUTATIONAL LINGUISTICS: MODELS, RESOURCES, APPLICATIONS 18, 100 (1st ed. 2004). For the purposes of legal research, which generally does not involve an interest in word formation or morphology, individual words suffice as objects of study.

[98] *See, e.g.*, Ingo Feinerer, *Package 'tm'* 20-25 (Jul. 3, 2015), http://cran.r-project.org/web/packages/tm/tm.pdf (detailing how to import .doc, .pdf, and .txt file types into the R computing language) [hereinafter Feinerer, *Package 'tm'*].

all file types use the same form of encoding.[99] For example, text files (.txt) can be encoded as ASCII, UTF-8, or Modified UTF-8. Similarly, Microsoft Word files (.doc) can take different forms when created with different editions of Microsoft Word. If faced with an encoding problem, the researcher will need to spend time (in inverse relation to her programming ability) in order to program the software to read various encoding types. At worst, the software will be unable to read the files and the researcher will have wasted a substantial amount of time collecting unreadable data. To avoid this problem, good practice employs a three-step process. First, before collecting an entire body of data, collect one representative document type. As a test, direct the software to read the document. Only if the software reads the document correctly, collect the remaining documents. Of course the remaining documents must be of the same type that the software has successfully read.

What does this imply in the context of a legal research project? Most computational linguistic research in law will analyze thousands of judicial decisions or large volumes of statutory code and regulations. Lexis or Westlaw will not suffice because most commercial licenses generally prohibit bulk downloading.[100] For example, Macey and Mitts analyzed 9,380 texts;[101] Fagan analyzed approximately 2,100 texts.[102] In order to download those texts from Lexis or Westlaw with a common license, each case would have to be saved to an individual file. Not only is this tedious, but the actual work

---

[99] Encoding is the form of representation of information within the file type. JOHN DAINTITH & EDMUND WRIGHT, A DICTIONARY OF COMPUTING 188 (6th ed. 2008).
[100] *See* Macey & Mitts, *supra* note 17, at 144 n.166.
[101] *Id*. at 141.
[102] Fagan, *supra* note 18, at 405.

could reach hundreds of hours. On the other hand, a relatively smaller study of say, 500 to 1,000 cases, could easily be completed within a reasonable amount of time, especially with research assistance. Larger studies, however, will benefit greatly from bulk downloading. Not only does bulk downloading save time, it also ensures uniformity of document type and encoding. One can download a single document, test it with the software's reader function, and confidently bulk download the remaining documents. CourtListener, a database used in previous studies, maintains a database of approximately 2.3 million precedential opinions and permits bulk downloading.[103] Coverage extends to most federal and state jurisdictions.[104] Unfortunately, bulk downloading is hardly straightforward with CourtListener's current functionality.[105] Users must perform a search and then download the contents of all of the search results by issuing cryptic commands from the command-line. On the other hand, a user can bulk download her search results with a simple script. As of 2014, CourtListener provides a script creation service and charges $100 per hour for academic and nonprofit-affiliated projects and $200 per hour for other projects.[106] A typical script for bulk downloading the entire contents of a search result requires, on average, three to four hours of work.[107]

Finally, successful data collection entails planning how to organize the data before and during collection. Useful

---

[103] *Non-Profit Free Legal Search Engine and Alert System*, COURTLISTENER, https://www.courtlistener.com (last visited Sept. 10, 2015).
[104] *Id*., https://www.courtlistener.com/coverage (last visited Sept. 15, 2015).
[105] E-mail from Mike Lissner, Executive Director and Chief Technical Officer, Free Law Project, to Author (April 17, 2014, 8:53 EST) (on file with author).
[106] *Id*.
[107] *Id*.

organization for later stages of the analysis requires consistent use of a standard file naming convention and simplicity in the organization and location of folders. Typically, document names are ordered numbers: for example, 0001.txt, 0002.txt, …, 1348.txt, etc. Descriptive file names provide no meaningful information in computational linguistics because analysis is carried out at the lexical level. All of the files for a single research project should be contained in one basic folder such as DataSet. This folder should be located inside of another folder, immediately one level above, appropriately named for the study. For example, data for Fagan was placed in a folder named "DataSet," which was placed inside of a folder named "SLdata." Strict maintenance of this directory structure is beneficial, because as the analyst develops the study, cases will be divided into new folders in order to carry out different types of analyses. Maintaining good order of the folders saves time when one needs to program the software to read different groups of cases.[108]

[108] *See* discussion *infra* Section II.E.1.

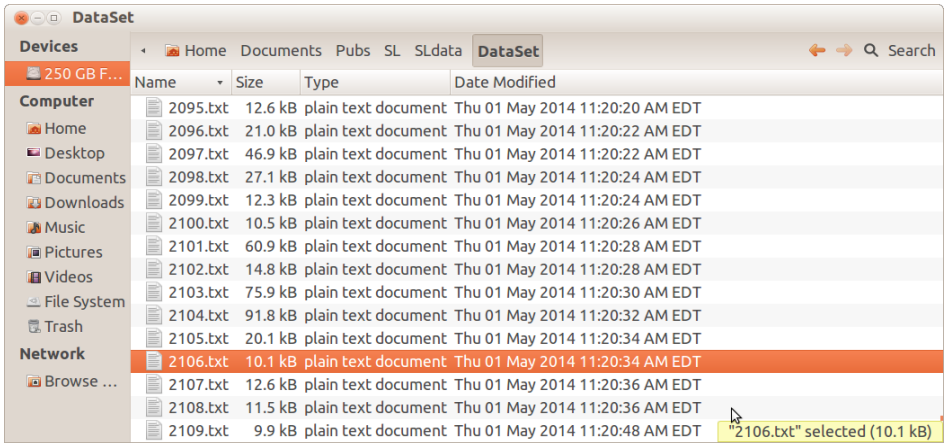

Fig. 2: Good Folder Structure

B. Setting Up a Computer for Research

1. Choosing the Hardware

The standard processing power, memory, and storage space of today's average personal computer are more than sufficient for most computational linguistic studies.[109] For example, Fagan analyzed 2,111 documents, which yielded vocabularies of 7,915 and 17,584 bigrams[110] for constructing a classifier and modeling topics, respectively. The analysis was completed with a consumer-grade personal computer, specifically an Intel Core i5 (1.6 gigahertz) processor, four gigabytes of memory, and adequate hard disk storage. Analytical construction of the classifier and topic model took between one to three minutes each. Larger projects approaching one million documents may benefit from better hardware, which is readily available in the personal computing market. Projects over one million documents may require

[109] BRETT LANTZ, MACHINE LEARNING WITH R 355–62 (2013) (discussing various advances in hardware and software to increase processing power and speed).
[110] Bigrams are singular, two-word units used for lexical analysis. *See* Fagan, *supra* note 18, at 410, 414.

industrial-grade hardware.[111] The point here is that one does not need a mainframe or a supercomputer to carry out basic computational linguistic analysis. A personal computer is sufficient.

2. Choosing the Software

A number of commercial and open source software applications are available for reading, preprocessing, and analyzing data. Open source software, particular R, has clear advantages over commercial applications. This guide assumes that most readers are legal scholars interested in big data and may be thinking about developing their own study. Thus, examining the research method with open source software before making a financial commitment is likely an attractive option. Trying before buying can reveal a capability and aptitude for computational linguistics. Perhaps more importantly, taking a test drive can reveal an appetite (or not) to persist and experiment when the coding becomes difficult or buggy. Open source software has two other notable advantages. First, help is readily available because the software is well-documented and supported by an enthusiastic community. Documentation ranges from introductory guides written for students to thorough and exhaustive manuals written for professional researchers. Sample code and examples, which are extremely useful for beginners, are widely distributed in print and on the World Wide Web; and what cannot be understood working alone can be asked in responsive

---

[111] LANTZ, *supra* note 109, observes that "a modern PC [can process] datasets of many thousands of records, but datasets with a million records or more can push the limits of what is currently possible with consumer-grade hardware". *Id.* at 360–61. However, the use of parallel cloud computing, or multiple networked workstations on the ground, can increase processing power. *Id*.

online forums.[112] Second, frequent updates and new features can be freely downloaded and installed. For example, the popular open source software R, permits the installation of "packages," which are developed by researchers for analyzing data in new ways.[113]

For these reasons, this guide recommends an open source application, at least for newcomers who are testing the waters of big data. Amongst open source options, the clear choice is R. R is powerful enough for performing most computational linguistic tasks. It is extremely well-documented and supported by an enthusiastic community, and relatively simple enough for beginners.

R is an implementation of S, a programming language developed by Bell Labs during the late 1970s. A team of researchers at the University of Auckland extended the architecture of S during the early 1990s. Today, R is primarily used for statistical programming and data mining. The language has grown in popularity and has polled as a top-ranked data mining solution.[114] The original implementation of R runs from a command line interface, but several graphical interfaces are available for Windows, Mac, and Linux. Among those, RStudio is a good choice. It is easy to install and

---

[112] *See, e.g.*, Stack Exchange, Questions Tagged 'R', http://stackoverflow.com/questions/tagged/r (last visited Sept. 10, 2015).
[113] Readers familiar with other statistical software such as Stata will recognize the term "package." An R package is conceptually identical.
[114] Revolution Analytics Blog, *R Tops Data Mining Software Poll* (May 31, 2012), http://blog.revolutionanalytics.com/2012/05/r-tops-data-mining-poll.html.

configure, presents the main components of R cleanly and intuitively, and hosts an integrated debugging environment.[115]

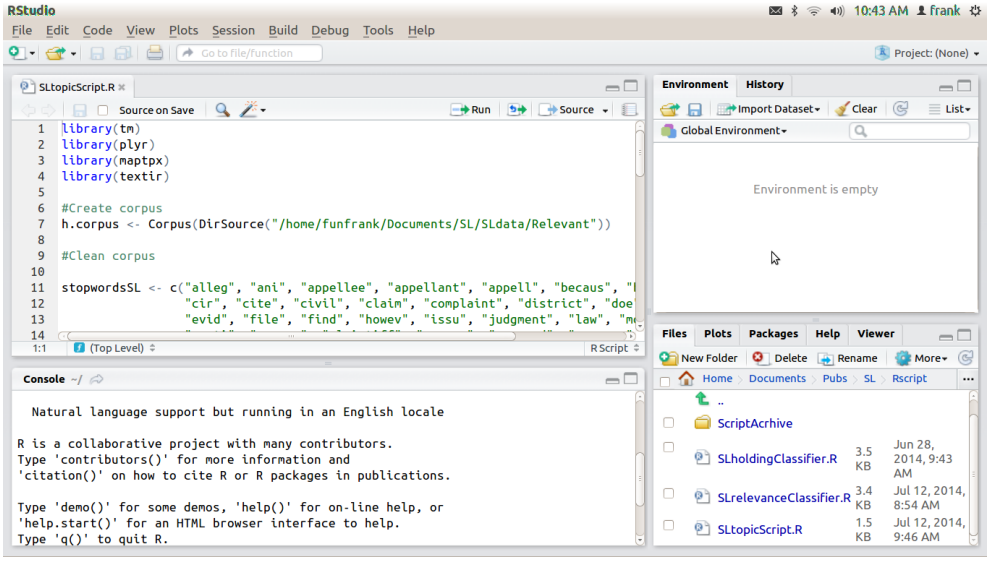

Fig. 3: A Typical RStudio Environment

Installation is straightforward. A user first installs R, the programming language, and then installs RStudio, the graphical interface. The RStudio website publishes installation instructions for Windows, Mac, and Linux.[116] Users agree to an AGPL version 3 license. Users who cannot agree to an AGPL license can purchase a license for $995. Ubuntu Linux users can install RStudio directly from the synaptic

---

[115] *See* Ari B. Friedman, Stack Exchange, *Good Gui for R Suitable for A Beginner Wanting to Learn Programming in R?* (March 6, 2014), http://stats.stackexchange.com/questions/5292/good-gui-for-r-suitable-for-a-beginner-wanting-to-learn-programming-in-r.

[116] RStudio, Products http://www.rstudio.com/products/RStudio/#Desk (last visited Sept. 10, 2015).

repositories, which will automatically install R as a dependency.[117]

The default environment of RStudio is divided into three panes. The left-most pane is the Console. The Console is essentially the command line interface of R in graphical form. All R commands can be passed through the Console just as if one were to pass commands through an operating system's command line. Users program in R by creating scripts, which are series of commands that run sequentially when directed. RStudio provides a fourth pane for creating and viewing scripts, which will open upon creating a new script. The Script pane opens immediately above the Console Pane on the left-hand side of the RStudio environment.

The Script Pane numbers each script line sequentially. These numbers are simply markers that indicate the physical location of the programming code. They facilitate quick reference for finding portions of code and debugging. R describes errors by first telling the user where they physically exist in the script. For example, there may be an error contained on lines 10–12. In addition to numbering lines of code, the Script Pane color codes various terms. The default text color is black, but other colors indicate specific types of objects or commands recognized by R. For example, the term "library" is blue. Library represents a command to call a group of functions into local memory for use. To save space and time, the default build of programming languages like R does not include every possible command or analytical method available to the user. Users load libraries of commands (or

[117] Detailed instructions for adding the relevant repository can be found at The Ubuntu R Blog, *Installing R in Ubuntu*, http://sites.psu.edu/theubunturblog/installing-r-in-ubuntu/ (last visited Sept. 10, 2015).

functions) only if they need them for performing specific tasks. The loading is accomplished by issuing the command "library," which is highlighted in blue. In Figure 2, notice in the upper left-hand pane that the script "SLtopicScript.R" loads four libraries: tm, plyr, maptx, and textir. Each of these libraries loads commands for (1) text mining; (2) combining and splitting data (think of a pair of pliers, hence the name "plyr"); (3) mapping topics; and (4) performing multinomial inverse regression. Without loading those libraries, R would not recognize the commands for performing those tasks.

The uppermost right-hand pane is primarily descriptive. The Environment displays all of the objects loaded into memory. For example, the script in Figure 2 contains a command to create a corpus object on line 7. A corpus is a linguistic term for a collection of data.[118] Once the user runs the command at line 7, the Environment pane displays the object, which is a VCorpus with various attributes that can be viewed by clicking the collapsible arrow button (Figure 3). Note that the commands Corpus and DirSource are part of the library tm; those commands would not have been available had the user failed to load that library. Finally, the uppermost right-hand pane contains a History tab, which simply displays a list of all commands previously issued by the user.

---

[118] Ingo Feinerer, *Introduction to the tm Package Text Mining in R*, 1 (2015), *available at* http://cran.rapporter.net/web/packages/tm/vignettes/tm.pdf.

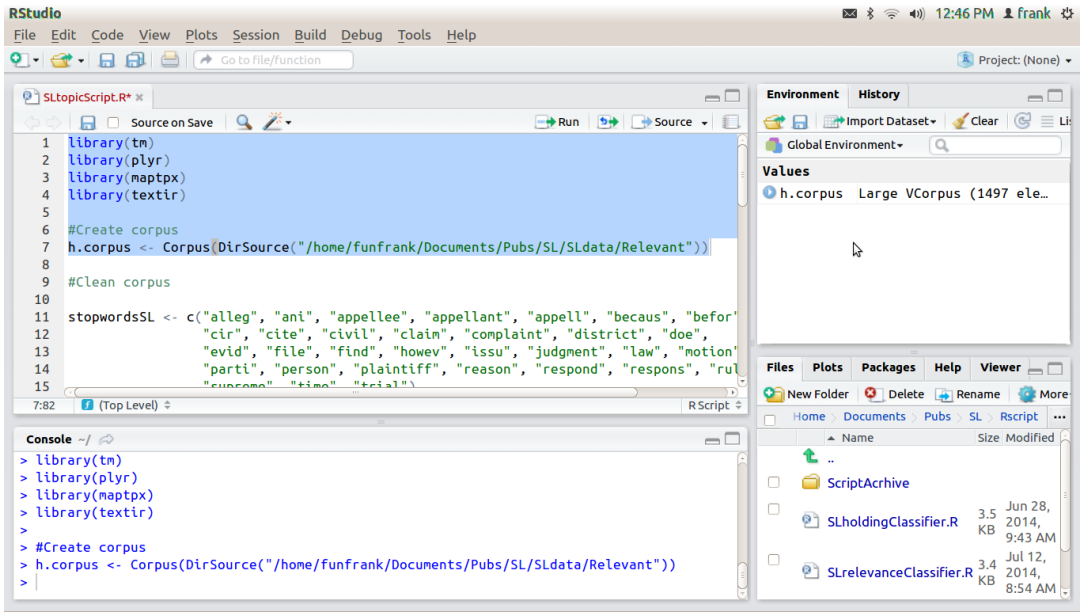

Fig. 4: A Newly-created VCorpus Object Displayed in the Environment Pane

The lowermost right-hand pane is very useful for new users, primarily for installing new packages such as tm, plyr, maptx, etc., and for viewing instructive vignettes in the Help tab. The Packages and Help tabs include a search bar for quick navigation. The remaining tabs are not particularly useful for beginner big data legal scholars or computational linguistics in general. The Files tab, for example, facilitates easy access to various scripts, etc. A beginner working on a modest study will likely work with a small number of scripts that can be opened and closed from the File menu. Similarly, RStudio generally provides the Plots tab for performing regression analysis; it is not particularly useful for performing computational linguistics.

3. Running the Software for the First Time

Even though a beginner will likely work with a limited number of files, it is good practice to maintain a simple and intuitive directory structure. Thus, a first-time user should proactively consider where to assign the working directory of an RStudio research project. RStudio uses a special architecture to facilitate organization of the working directory, workspace, history, and source documents.[119] All of this information is assigned to a Project, which is associated with a working directory chosen by the user. Select a name for a Project. For example, the Project in Figure 3 is named SL for successor liability. On the menu bar at the top of the screen, select the drop-down menu "File" and select "New Project". Select "New Directory" and then "Empty Project". Enter the Directory Name and click "Create Project". To confirm the working directory, input the command getwd() in the Console and compare the output.

Next, a first-time user should install relevant packages for analyzing large volumes of legal texts. Click the Packages tab. If the search bar is out of view, drag the left-hand border of the rightmost panes to the left in order to expand the pane size. RStudio displays all of the installed packages. Even though first-time users have yet to install any packages, a clean installation of R includes base packages. Base packages are automatically loaded into memory and do not need to be called with the library() command. Notice that the packages are hyperlinks. By clicking on one, a relevant help vignette will display within the Help tab. Within the Packages tab, select "Install Packages". The window in Figure 4 will appear.

---

[119] RStudio, Support, https://support.rstudio.com/hc/en-us/articles/200526207-Using-Projects (last visited Sept. 10, 2015).

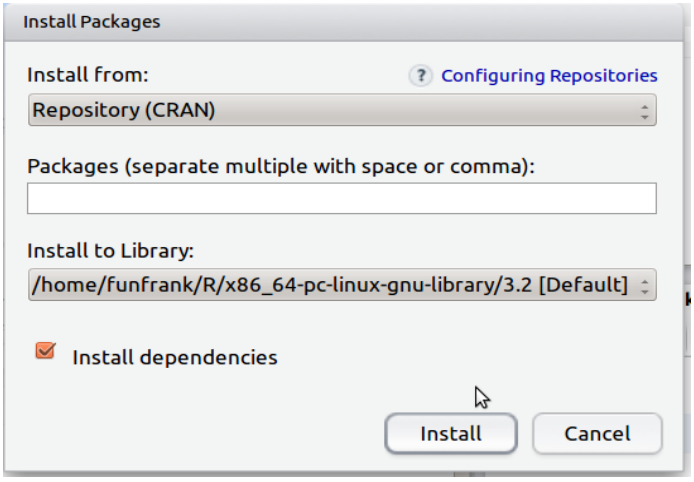

Figure 5: The "Install Packages" Dialog Box

R packages are stored in a central repository, named CRAN. The option to install from this repository should be the default option for first-time users. Packages will be installed to the user's "Library," which is the default installation directory of R. The default installation directory should already be assigned for first-time users. The option "Install dependencies" should be ticked. Try installing one package first. Type "tm" into the Packages bar and click install. RStudio will automatically pass a number of commands to the Console. At the end of the process, the tm package should be installed. Confirm installation by examining the "Packages" tab in the lower right-hand pane. Now install the following packages: caret, e1071, plyr, maptx, and textir. They can be installed separately or all at once by following the instruction for multiple package installation in the Install Packages dialog box.

To practice coding the examples that will be discussed in this section, create a script. Recall that a script is essentially a list of commands that could otherwise be passed through the console one by one. Scripts provide the ability to issue a set of commands at once. They also facilitate reuse and debugging of code. To create a script, select the drop-down menu "File",

point to "New File", and then select "R Script". RStudio will create an untitled script that will appear in a pane located above the Console. Save the script file in the Project Directory by clicking File → Save As. If the reader plans to work this guide as an example, save the script with a name such as PracticeScript. Notice that the script will appear in the lower right-hand pane under the "Files" tab. If you exit RStudio, it will ask the user to save the workspace image. Click yes. Reopen RStudio, and it will restore the identical workspace at exit.

C. Importing Data

The work environment is now ready for importing data. At this point, the guide assumes that the newcomer has collected a small handful of text files (.txt) for a test run, or several thousand for a study.[120] As explained in Section II.A., this can be accomplished through bulk download of a search result from Court Listener. Ideally, the text files are sequentially numbered, and are placed in a single folder named "DataSet" or something similar.[121] Move this folder into the RStudio Project directory.

In order to analyze the data, the text files must first be read into R. Ingo Feinerer of the Vienna Technical University has created an excellent and well-documented R package called text miner ("tm") for this task.[122] Given that the user has already installed this package, it can be called into RStudio's local memory. In the first line of the script enter the code:

---

[120] For reading other file types, *see* Feinerer, *supra* note 118, at 1.
[121] *See supra* Figure 2.
[122] For an introduction to the tm package, see Feinerer, *supra* note 118, *passim*.

```
library ("tm")
```

This code will call the tm library into memory for use. Enter a carriage return and type a comment on line 3. Comments describe the function of the code in plain language.[123] In R, comments are entered by first typing the number sign "#". The comment should read # Create corpus. On the line below, enter a command for reading the data into a VCorpus object,[124] for example:

```
tort.corpus <- Corpus(DirSource("path to your dataset folder"))
```

The code will create a VCorpus object, named tort.corpus that contains all of the text files contained within the dataset folder.[125] Select all of the code in the Script Pane and click "Run". The object tort.corpus will now appear in the Environment tab of the upper right-hand pane. By inputting the command print (tort.corpus), RStudio will output the object type (i.e., VCorpus) and the number of documents contained within that object.[126] By inputing the command inspect (tort.corpus [1672:1675]), RStudio will output the contents of documents 1,672 through 1,675. When analyzing data, it may

---

[123] Programmers often create a program simply with comments in plain English, and then insert the necessary code thereafter. As a beginner, it's useful to insert comments along with code. The comments remind one what the code means.

[124] VCorpus stands for volatile corpus, meaning that it is stored in local memory and can be changed. Feinerer, *supra* note 118, at 1.

[125] See Figures 2 and 3 for more examples. The backward arrow sign, <-, is the assignment operator in R. LANTZ, *supra* note 109, at 30. Other programming languages use "=". *Id.* Thus, in the code example here, the VCorpus object named tort.corpus is assigned the texts found in the path DirSource by means of the function Corpus().

[126] LANTZ, *supra* note 109, at 105.

be necessary to divide the corpus into sub-corpora. Division and concatenation of a corpus will be discussed below in Section II.E.1.

D. Preparing the Data for Analysis

1. Preprocessing

Most computational linguistic analysis is concerned with prediction. Words and artifacts that are not useful for prediction can be stripped in order to increase the accuracy and speed of the analysis.[127] The entire battery of stripping techniques falls under the general task of preprocessing. Again, the tm package is useful. The package performs preprocessing functions by transforming, or mapping, the noisy corpus to a clean corpus free of noise.[128] Accordingly, the tm package performs various preprocessing routines with the general function tm_map(). Standard corpus linguistic analysis typically strips whitespace, numbers, punctuation, and stop words (i.e., words that provide little predictive power because they appear so frequently, e.g. articles, helping verbs, infinitives, etc).[129] In addition to removing stop words, an analyst should remove words that frequently appear in a legal context.[130] This can be accomplished by creating a new object

---

[127] *See* C.J. VAN RIJSBERGEN, A NON-CLASSICAL LOGIC FOR INFORMATION RETRIEVAL (1980), *reprinted in* READINGS IN INFORMATION RETRIEVAL 268 (Karen Sparck Jones & Peter Willett eds., 1997); *see also* M.F. Porter, *An Algorithm for Suffix Stripping*, 14 PROGRAM 130 (1980).

[128] LANTZ, *supra* note 109, at 106.

[129] On whitespace, numbers, and punctuation, see LANTZ, *supra* note 109, at 106. On stop words, see David D. Lewis et al., *RCV1: A New Benchmark Collection for Text Categorization Research*, 5 J. MACH. LEARN. RES. 361, app. 11 (2004).

[130] This technique is controversial because it increases the amount of subjectivity in the analysis. *See* Fagan, *supra* note 18, at 409 n.67

called lwords and removing all lwords from the corpus. To carry out this task, enter a carriage return underneath the creation of the corpus, and enter the following code:

```
lwords <- c("alleg", "appellee", "appellant", "appell", "case", "cir", "cite", "civil", "claim", "complaint", "district", "doe", "evid", "file", "find", "issu", "judgment", "law", "motion", "order", "parti", "person", "plaintiff", "respond", "respons", "rule", "sct", "supreme", "trial")
```

The function c() combines each of the words separated by commas into a single character object called lwords.[131] The object can later be called when removing words from the corpus.

In addition to stripping, two additional transformations will enhance the predictive capability and speed of the analysis. First, all words should be set to lower case to eliminate variability based upon capitalization. Second, all words should be stemmed to eliminate variability based upon lexeme usage.[132] Westlaw and Lexis users are familiar with stemming. In those search engines, words can be searched without regard to their endings using the wildcard operator "!". Thus, liab! will return liable, liability, liabilities, etc. Stemming reduces all of those words in the corpus to "liab". In order to preprocess a VCorpus object in RStudio, the analyst calls the function tm_map(). Underneath the lwords object, enter a

---

("Selection of stop words provides perhaps the single largest potential for bias in application of computational linguistics to legal research.").

[131] Technically, the function combines the words into a character vector and assigns that vector the name lwords. LANTZ, *supra* note 109 at 30. *See* discussion *supra* Section II.D.2.

[132] *See* C.J. VAN RIJSBERGEN, ET AL., *supra* note 127, *passim*.

carriage return. On the next line, enter a comment # Clean corpus. On the line below, enter the following code:[133]

```
tort.corpus <- tm_map(tort.corpus, stripWhitespace)
tort.corpus <- tm_map(tort.corpus, tolower)
tort.corpus <- tm_map(tort.corpus, removePunctuation)
tort.corpus <- tm_map(tort.corpus, removeNumbers)
tort.corpus <- tm_map(tort.corpus, stemDocument)
tort.corpus <- tm_map(tort.corpus, removeWords, c(stopwords("SMART"), lwords))
```

This piece of code returns a clean VCorpus object where, all whitespace is stripped; all words are set to lower case; all punctuation is removed; all numbers are removed; all words are stemmed; and all stop words, including those contained in the SMART list[134] and the lword list, are removed. The clean corpus object retains the name tort.corpus. A final consideration for preprocessing a corpus is to remove sparse terms. Sparse terms provide little predictive power because they occur infrequently across the documents and can slow the analysis considerably. Generally, removal of sparse terms after preparing the data further enhances efficiency and increases analytical speed. Their removal will be discussed in the following section.

---

[133] Later versions of the tm package may require setting the parameter "lazy" to the TRUE value. *See* Feinerer, *Package 'tm' supra* note 98, at 15.
[134] The SMART stop word list includes frequently used words such as articles, helping verbs, infinitives, etc. *See* David D. Lewis et al., *RCV1: A New Benchmark Collection for Text Categorization Research*, 5 J. MACHINE LEARNING RES. 361, 386–87 (2004).

2. Conversion of Data Structures and Tokenization

Programming languages such as R use various types of data structures according to the type of task the data must perform. Think of a data structure as a uniform way of organizing a group of individual pieces of data. For example, in the stop word example above, the function c() created a character vector named lword. In R, the fundamental data structure is the vector, which contains an ordered set of elements.[135] Thus, lword is an object, specifically a character vector, that contains the ordered set of elements {"alleg", "appellee", ..., "trial"}; because the object is ordered, the vector understands that the first element is alleg, the second is appellee, etc.[136]

The other two important types of data structures in R look like spreadsheets. These types of data structures are important for computational linguistics because an analyst is often interested in all of the words contained in a single document across the entire vocabulary of the corpus. Individual documents can be represented by rows, and all of the words contained within the corpus can be represented by columns. If an individual document contains a word, the spreadsheet will contain that information in its columns. A spreadsheet organized by documents in rows and by words in columns is called a Document-Term Matrix.

---

[135] LANTZ, *supra* note 109, at 30.

[136] R uses two other special types of vectors called Factors and Lists. *Id*. at 31. Factors are used for storing categorical data such as gender (male or female), while lists are used for storing elements of different types such as characters, numbers, and factors. *Id*. at 31–34.

	coming	nuisance	boomer	cement	factory	closure	employees
Document 1	1	1	1	1	1	1	1
Document 2	1	1	0	0	1	1	1
Document 3	1	1	0	0	0	1	1
Document 4	1	1	0	0	0	0	1
Document 5	0	0	0	0	0	0	1

Fig. 6: A Bernoulli Document-Term Matrix

The topical analysis in Fagan produced a vocabulary of 17,584 bigrams across 2,111 documents.[137] Thus, the Document-Term Matrix used to construct the topic model consisted of 2,111 rows and 17,584 columns. Document-Term Matrices can be Bernoulli or continuous.[138] The Bernoulli type simply indicates if the word is contained in the document (1) or not (0). The continuous type indicates how frequently the word appears in the document. Continuous types are generally more predictive because they contain a greater amount of analytically productive information about the documents.

R handles spreadsheet-like data structures with two classes of objects: the data frame and the matrix.[139] Data frames must contain the same number of columns for each row. Thus, the Document-Term Matrix in Figure 5 could be converted to a data frame within R. Instead, if Document 5 contained no value in the column "employees", the programmer would have to "fill" that cell with a 1 or a 0 in order to convert the spreadsheet into a data frame.[140] Generally, the data frame is used for performing machine learning (e.g. data classification), while the matrix is used for performing mathematical operations.[141] The tm package

[137] Fagan, *supra* note 18, at 414.
[138] *See* CHRISTOPHER MANNING, ET AL., INTRODUCTION TO INFORMATION RETRIEVAL 243 (2009).
[139] LANTZ, *supra* note 109, at 36–38.
[140] This symmetrical requirement of the data frame is important to keep in mind when programming within R as it can create a bug for the beginner.
[141] LANTZ, *supra* note 109, at 35, 37.

includes a function for creating a Document-Term Matrix. Enter a carriage return underneath the final call of tm_map() for removing stop words, and input the comment # Create Document-Term Matrix. Input the following code:

```
tort.dtm <- DocumentTermMatrix(tort.corpus)
```

The function DocumentTermMatrix() constructs a continuous Document Term Matrix from the VCorpus object tort.corpus and assigns it the name tort.dtm.

The code above creates a Document-Term Matrix with single words in the columns. However, in order to increase predictive power, computational linguists tokenize texts, that is, they combine neighbor words into phrases, or tokens.[142] For example, the sentence "great big dogs fight animals" tokenizes into four bigrams: "great big," "big dogs," "dogs fight," "fight animals." Similarly, the same sentence tokenizes into three trigrams: "great big dogs," "big dogs fight," and "dogs fight animals." An n-gram is a token of n words. In order to tokenize a Document Term Matrix created with the package tm, the user needs to call a tokenizer function. Several options are available. Both the NLP and RWeka packages provide functions to construct bigrams.[143]

The RWeka package will be discussed here in order to illustrate that sometimes programming solutions, especially for beginners, can appear inelegant, but they nonetheless solve the problem. A user typically loads a package into memory by calling a library. Normally, to load the RWeka package, the

---

[142] Fagan, *supra* note 18, at 409.
[143] For use of the NLP package, *see* Ingo Fernerier, Frequently Asked Questions About tm, http://tm.r-forge.r-project.org/faq.html#Encoding (last visited Sept. 10, 2015).

script would contain the following line of code: library("RWeka"). However, RWeka conflicts with a base R package "parallel" in some configurations. In order to avoid the conflict, the user must call the RWeka pacakage at the time she calls a function instead of initially loading the library.[144] This is accomplished by using the namespace, i.e., the name of the library, within an encapsulated function:

```
BigramTokenizer <- function(x) {
   RWeka::NGramTokenizer(x,
RWeka::Weka_control(min = 2, max = 2))}
```

This piece of code creates a function called BigramTokenizer. Notice that in order to construct this function, the code calls NGramTokenizer and Weka_control from the RWeka library by using the "::" syntax. This type of creative solution is typical when working with code, especially when various packages are created by different people. Now, this time, create the Document Term Matrix with BigramTokenizer:

```
tort.dtm <- DocumentTermMatrix(tort.corpus, control =
list(tokenize = BigramTokenizer))
```

The Document Term Matrix now consists of rows of each document, and columns of each two-word couplet contained within the corpus. After completing bigram tokenization, Figure 5 would appear as:

---

[144] *See* Stack Exchange, Bigrams Instead of Single Words in Termdocument Matrix Using R and RWeka, http://stackoverflow.com/questions/17703553/bigrams-instead-of-single-words-in-termdocument-matrix-using-r-and-rweka (last visited Sept. 10, 2015).

	coming nuisance	nuisance boomer	boomer cement	cement factory	factory closure	closure employees
Document 1	1	1	1	1	1	1
Document 2	1	0	0	0	1	1
Document 3	1	0	0	0	0	1
Document 4	1	0	0	0	0	0
Document 5	0	0	0	0	0	0

Figure 7: A Document Term Matrix Tokenized with Bigrams

Before converting the data structures for analysis, one final transformation is needed, i.e. removal of sparse terms. Sparse terms provide little predictive power because they rarely occur across multiple texts.[145] A corpus typically contains a large number of sparse terms, which considerably slow the analysis. Enter a comment to indicate their removal, e.g. # Remove terms that occur in less than 2% of the documents, and enter the following code on the line below:

```
tort.dtm <- removeSparseTerms(tort.dtm, 0.98)
```

The Document-Term Matrix now consists of bigrams that occur across 98% of the texts. Analysts can experiment with removal at the 1% level by setting the parameter to 0.99. A change can sometimes increase or decrease the accuracy of the computational analysis.

In order to carry out computations with the Document-Term Matrix, R requires conversion into a familiar data structure. Recall that the two type of spreadsheet-like data structures in R are matrices and data frames. In order to carry out the analysis in the following section, the Document-Term Matrix needs to be converted into a data frame, in particular, a multinomial data frame where each column (bigram) is a categorical variable. Enter a comment such as # Create multinomial data frame where each column is a categorical variable. Input the following code:

---

[145] *See* Fagan, *supra* note 18, at 410 n.69.

```
tort.mat <- as.matrix(tort.dtm)
tort.df <- as.data.frame(tort.mat)

for(i in 1:ncol(tort.df)) {
        tort.df[,i] <- as.factor(tort.df[,i])
        }
```

The code performs a series of conversions. First, it converts the Document-Term Matrix, tort.dtm, into a matrix. R requires conversion of the Document-Term Matrix into a matrix data structure in order to create the data frame. On the second line, the code converts the tort matrix into a data frame. Finally, the function on fourth and fifth lines set each of the columns of the data frame to categorical variables. The code therefore creates a multinomial data frame where each column is a categorical variable. Preprocessing and conversion of the data is now complete.

E. Analysis

Topic modeling and regression analysis are the two most important tools for big data legal scholarship today. Topic modeling can produce taxonomic studies that are useful for resolving doctrinal disputes and updating confusing applications of law.[146] Regression analysis can uncover what judges do, despite what they say, and can thereby provide strong empirical bases for the legal realist view of judging (or support an opposing conclusion).[147] Regression analysis can also support taxonomic studies based upon topic modeling, and can help resolve doctrinal disputes and update confusing applications of law.[148] Each type of study plays an important

[146] *Supra* Section I.A.1.
[147] *Supra* Section I.A.2.
[148] *Supra* Section I.A.1.

normative role for lowering transaction costs, decreasing predictability, and increasing comparative justice.[149]

1. Using a Classifier

Topic modeling and regression analysis directly bear on the positive and normative conclusions of big data legal scholarship. However, in order to increase the accuracy and speed of the research, a big data scholar can perform the intermediate task of document classification. Document classification is a technique for classifying documents according to shared attributes. Successful document classification enables a researcher to collect thousands of documents and to partition them according to shared attributes broadly construed. For example, all email software use classification algorithms to divide incoming email messages between spam and not spam. Similarly, Fagan uses a classification algorithm to divide judicial opinions produced by a CourtListener search on successor liability as relevant or not relevant, where relevant is defined as the court applying a successor liability legal standard to the facts, and not relevant is defined as the court not applying a successor liability legal standard to the facts.[150] Because text-based searches can produce irrelevant documents that contain the search keywords, an analyst needs to evaluate the remaining words in the document in order to determine its relevance to her study. Consider that a simple Google keyword search produces millions of search results on average, but only some are relevant. An algorithm therefore sequentially orders each search result according to its relevance so as to increase the productivity of a search. Likewise, big data legal scholars may search a database of several million judicial opinions or legal

---

[149] *Supra* Section I.B.
[150] Fagan, *supra* note 18, at 406.

texts. A typical keyword search returns both relevant and irrelevant results. Classification algorithms, by evaluating the remaining words contained within a legal text, filter the search results according to their relevance to the study and increase the overall productivity of a keyword search.

The mechanics behind classification are straightforward. First, the analyst randomly samples a portion of her dataset and classifies the sample by hand. Second, the hand-classified sample trains the algorithm. Finally, the trained algorithm classifies the remaining documents of the dataset.

Random Sampling

The first question to consider is the size of the random sample. Because the randomly sampled data will be used to construct the classification algorithm, it is important to understand how much data is needed for construction. In general, classification algorithms require hand-classified data for both training and testing.[151] An analyst uses a portion of the hand-classified data to train the algorithm and uses the remaining hand-classified data to test the accuracy of that algorithm.[152] Thus, developing a classification algorithm requires constructing the classifier with a portion of the random sample and then testing that construction by evaluating its ability to machine-classify the remaining randomly sampled data. Because the analyst has previously hand classified that data, she can compare the results of the machine classification with the results of her hand classification and thereby evaluate

---

[151] *See* LANTZ, *supra* note 109, at 81, 108, 131, 152, 155 (observing the need for some portion of hand-classified data for the following models: kNN, naïve Bayes, C5.0, 1R, and RIPPER).
[152] *See, e.g.*, *id*. at 108 (partitioning data into training and testing sets).

the accuracy of the classifier. In practice, this means that the random sample must be sufficient in size to not only accurately represent the dataset, but also to provide a sufficient number of observations for construction and testing of the classifier.[153] A larger random sample allows for a greater number of observations to be used for constructing the algorithm, which increases its accuracy, i.e., the parameter estimates will have less variance.[154] A larger random sample additionally allows for a greater number of observations to be used for testing the algorithm, which increases its consistency, i.e., the performance statistics will have less variance.[155] Thus, randomly sampled data is often split into training and testing sets on an 80:20 or 90:10 basis.[156] Big data analysts often try different splits to maximize the algorithm's performance while maintaining its consistency; indeed, various packages exist within R for choosing splits that optimize these two measures.[157]

Given that an adequate number of observations is required for developing the classifier, what percentage of the dataset should the analyst randomly sample? There is no clear answer here. While a quantitative analyst generally evaluates a number of a priori considerations—including the desired level of confidence in the study,[158] qualitative analysts carrying out

---

[153] Stack Exchange, Is there a Rule of Thumb for How to Divide a Dataset into Training and Validation Sets?, http://stackoverflow.com/questions/13610074/is-there-a-rule-of-thumb-for-how-to-divide-a-dataset-into-training-and-validatio (last visited Sept. 10, 2015) (noting that a sufficient number of observations are required to divide the training and testing data).
[154] *Id*.
[155] *Id*.
[156] *Id*.
[157] LANTZ, *supra* note 109, at 293.
[158] STEVEN K. THOMPSON, SAMPLING 53–55 (2012).

big data research generally employ the saturation process, i.e., "[w]here no new themes emerge from qualitative analyses of the data set."[159] The difficulty of using saturation is that by the time the analyst understands that more data is needed, the time for data collection can be past.[160] Thus, big data analysts tend to rely on previously published studies as guidelines.[161] Macey and Mitts randomly sampled 1,000 of 9,380 observations (about 11 percent);[162] Fagan randomly sampled 500 of 2,111 observations (about 24 percent).[163] Depending upon the size of the dataset, therefore, saturation of themes may be expected with a sample size of 10 to 25 percent of the dataset.

Once the sample size has been determined, the analyst must carry out the random sampling. If the analyst has collected the data from CourtListener, it is likely ordered chronologically by decision date. Using the first several hundred observations as a sample will likely bias the results if the object of analysis correlates with time, e.g., updating judicial doctrine following an appellate court decision. While several programming techniques exist within R for randomly sampling the dataset and then using that random sample to construct the classifier,[164] recall that the random sample must be classified by hand. To avoid errors, it can be useful to partition the random sample into separate folders in order to

---

[159] Lynne M. Webb & Yuanxin Wang, *Techniques for Sampling Online Tex-Based Data Sets*, in BIG DATA MANAGEMENT, TECHNOLOGIES, AND APPLICATIONS 98 (Wen-Chen Hu & Naima Kaabouch eds., 2013).
[160] *Id*.
[161] *Id*.
[162] Macey & Mitts, *supra* note 17, at 141.
[163] Fagan, *supra* note 18, at 405.
[164] For example, a skilled programmer could create a vector of random numbers that correspond to a row in the data frame (which represents a document). The code could then reference those rows in order to construct the classifier.

avoid hand classifying documents that were not randomly sampled. Because the classifier will be used to classify relevant and irrelevant judicial decisions with respect to a keyword search in this example, the random sample should be divided into two folders: relevant and irrelevant. As a first step, the results of the random sample must be removed from the dataset and placed in its own folder.[165] Create a folder called RandomSample. Next, create a list of random integers in length of the random sample. This can be accomplished, for example, in Microsoft Excel.[166] Given that the dataset consists of documents named with sequential integers,[167] the document numbers that correspond with the randomly sampled integers may be moved from the DataSet folder to the RandomSample folder.[168] Now create two folders within RandomSample: Relevant and Irrelevant. Develop a clearly defined rule for classification. For example, Fagan develops a rule where relevant opinions are opinions where a court applies a successor liability legal standard to the facts. Application can involve finding that successor liability exists, finding that it does not exist, remanding and ordering a lower court to evaluate whether successor liability exists, etc.[169] Opinions where successor liability is merely mentioned or otherwise not evaluated against a set of facts are classified as irrelevant.[170] Development of a good rule is critical for later phases of a study and some rules are analytically superior to others. Rules

---

[165] In classification, the sample is always held out from the dataset because it is used to classify the remaining data. The entire dataset will be used in topic modeling and multi-inverse regression.
[166] *See* McGimpsey & Associates, Random Integers Without Repetition, http://mcgimpsey.com/excel/udfs/randint.html (last visited Sept. 10, 2015).
[167] *See supra* text accompanying note 111.
[168] This may be accomplished with the use of an automated script, though its construction is outside the scope of this Article.
[169] Fagan, *supra* note 18, at 406.
[170] *Id*.

that are complicated will lead to poor performance of the classifier. Simple binary rules like spam or not spam will lead to better performance. Similarly, rules that introduce a level of subjectivity to the research may lead to less convincing results to a wider research community. Upon determining the rule for classification, read through each opinion in the RandomSample folder and move them to the Relevant and Irrelevant folders accordingly.

Constructing the Classifier

Now that the random sample has been hand classified, the classification algorithm can be constructed within RStudio. Within the current project, create a new script and save it with a descriptive name such as RelevanceScript. Load the necessary libraries, each on a separate code line, beginning with line 1:

```
library(tm)
library(plyr)
library(e1071)
library(caret)
```

Next, enter a comment that indicates creation of the corpus that will be used to train the classifier. On the next two lines, create two VCorpus objects, one each for the documents contained within the Relevance and Irrelevance folders:

```
#Create training corpus
t.relevant <- Corpus(DirSource("path to Relevance folder"))
t.irrelevant <-Corpus(DirSource("path to Irrelevant folder"))
```

Directly underneath the line above, create another VCorpus object for the documents contained in the DataSet folder, and then combine all of the corpora into a single, full corpus:

```
d.corpus <- Corpus(DirSource("path to DataSet folder"))
full.corpus <- c(t.relevant, t.irrelevant, d.corpus)
```

Using the techniques described in the previous section, create a stop word vector, lwords, and clean the full.corpus corpus with the tm_map function.[171] Next, create a tokenized Document-Term Matrix and remove sparse terms that occur in less than two percent of the documents.[172] Finally, convert the Document-Term Matrix to a data frame:

---

[171] Specifically:

```
lwords <- c("alleg", "appellee", "appellant", "appell", "case", "cir", "cite", "civil", "claim", "complaint", "district", "doe", "evid", "file", "find", "issu", "judgment", "law", "motion", "order", "parti", "person", "plaintiff", "respond", "respons", "rule", "sct", "supreme", "trial")

full.corpus <- tm_map(full.corpus, stripWhitespace)
full.corpus <- tm_map(full.corpus, tolower)
full.corpus <- tm_map(full.corpus, removePunctuation)
full.corpus <- tm_map(full.corpus, removeNumbers)
full.corpus <- tm_map(full.corpus, stemDocument)
full.corpus <- tm_map(full.corpus, removeWords, c(stopwords("SMART"), lwords))
```

[172] Specifically:

```
# Create tokenized DTM
BigramTokenizer <- function(x) {
RWeka::NGramTokenizer(x, RWeka::Weka_control(min = 2, max = 2))
}
```

```
#Create multinomial data frame where each column is a categorical variable

full.mat <- as.matrix(full.dtm)
full.df <- as.data.frame(full.mat)

for(i in 1:ncol(full.df)) {
full.df[,i] <- as.factor(full.df[,i])
}
```

Before constructing a classifier, it is important to choose amongst various models. Different classification algorithms have varying strengths and weakness, and choosing a good classifier can enhance predictive accuracy and utility.[173] This latter distinction is worth discussing in detail. While a classifier may accurately predict test data, it may not perform well for its intended purpose because the test data is skewed, i.e., the data used for classification is highly imbalanced across classes.[174] Lantz explains the class imbalance problem as follows:

> [S]uppose a classifier correctly identified whether or not 99,990 out of 100,000 newborn babies are carriers of a treatable but potentially-fatal genetic defect. This would imply an

---

```
full.dtm <- DocumentTermMatrix(full.corpus, control = list(tokenize = BigramTokenizer))

# Remove terms that occur in less than 2% of documents
full.dtm <- removeSparseTerms(full.dtm, 0.98)
```

[173] LANTZ, *supra* note 109, at 293.
[174] *Id*. at 294.

> accuracy of 99.99 percent and an error rate of only 0.01 percent. ...this would appear to indicate an extremely useful classifier... What if the genetic defect is found [in the real world] in only 10 out of every 100,000 babies? A test that predicts "no defect" regardless of circumstances would still be correct for 99.99 percent of all cases. [However,] even though the predictions are correct for the large majority of data, the classifier is not very useful for its intended purpose, which is to identify children with birth defects.[175]

In legal research, the class imbalance problem can arise when a dataset is imbalanced across classes of interest. For example, Fagan used an algorithm to classify judicial decisions according to relevance, in particular whether or not a court applied a successor liability legal standard to the facts. The study initially hand classified the random sample according to whether (1) the court applied the legal standard and found the successor liable, (2) applied the legal standard and did not find the successor liable, and (3) the court applied no successor liability legal standard, i.e., the decision was irrelevant.[176] However, this preliminary classification performed poorly because a large number of the hand-classified decisions were irrelevant.[177] As a result, the classifier correctly classified a large number of irrelevant decisions and appeared to indicate a useful classifier. But it incorrectly classified a substantial number of incoming liable successor decisions as not liable and not liable successor decisions as liable—just as classifier that assigns "no defect" can still appear extremely accurate.

---

[175] *Id*.
[176] Fagan, *supra* note 18, at 406.
[177] *Id*.

The classification and regression training ("caret") package for R provides a number of functions for training and evaluating a classifier.[178] One type of classification model that can be useful for legal research is the decision tree. Decision trees are widely used for computing credit scores, conducting marketing studies, and diagnosing medical conditions based upon laboratory measurements, symptoms, and rate of disease progression, though they can be applied to almost any classification problem.[179] Perhaps their most significant weakness for legal studies is that small changes in training data can drastically impact analytical results.[180] A keyword search that only produces several thousand cases will, by necessity, produce a limited amount of data for training that may be susceptible to small variations between training and non-training data.[181] On the other hand, studies that consist of large and noisy datasets may benefit from their use.[182] In addition to decision trees, artificial neural networks may also be useful for legal research. Artificial neural networks are often used for automation tasks such as self-driving cars and self-piloting drones as well as for developing sophisticated models of weather, tensile strength, fluid dynamics, and various socio-economic phenomena.[183] While highly accurate, these classification models are computationally intensive and slow to

---

[178] LANTZ, *supra* note 109, at 302; *see also* Max Kuhn, *Building Predictive Models in R Using the Caret Package*, 28 J. STAT. SOFTWARE 1 (2008). In particular, the caret package can evaluate and improve the performance of about 75 classification models. *Id*. at 328.
[179] *Id*. at 120.
[180] *Id*. at 125.
[181] For example, Fagan performed a keyword search of 2.2 million CourtListener opinions that only produced 2,111 results. Fagan, *supra* note 18, at 405.
[182] *See* LANTZ, *supra* note 109 at 147–48 (discussing the RIPPER algorithm and its relatively good performance with large and noisy datasets).
[183] *Id*. at 206.

train.[184] One classification model that has been successfully used for legal research is naïve Bayes. Naïve Bayes uses training data to calculate an observed probability of each class based upon the number of documents in a class and the data attributes of the documents. In text analysis, the attributes are words. Thus, naïve Bayes calculates the probability that a document belongs to a class based upon the number of documents per class, and a comparison of class words to document words. The model has been successfully used in text classification applications such as junk mail filtering and plagiarism detection. Bayesian classifiers are particularly useful where numerous attributes (such as words contained within a document) are considered simultaneously in outcome probability (such as a 51 percent probability that document x belongs to class y).

With respect to legal research, naïve Bayes classification is particularly useful for filtering irrelevant results of a keyword search. Many useful studies of judicial doctrine will surely involve a keyword search of a large database that does not accurately sort results by relevance. Those studies will benefit from relevance filtering with naïve Bayes. The package e1071 contains several functions for creating a naïve Bayes classifier and interpreting its results. In order to compare the attributes of hand-classified texts with those of unclassified texts, naïve Bayes must know the attributes of all texts and the results of hand classification. In the coding example above, the analyst has created a data frame consisting of documents by rows and bigrams by columns for all legal texts contained within the dataset. Thus, all of the attributes (specifically, bigrams) are known. Recall too, that the Relevant, Irrelevant, and DataSet folders were combined

---

[184] *Id*. at 216.

with the c() function into a single corpus. The combine function maintains sequential order so that all documents contained within the Relevant folder are followed by Irrelevant documents and then DataSet documents. Ordering is critical for training the classifier in the code example that follows.

Let us assume that the Relevant folder contains 600 documents, the Irrelevant folder contains 400 documents, and the DataSet folder contains 3,000 documents. In that case, the total size of the dataset is 4,000 documents where 1,000 documents were randomly sampled for constructing a classifier. If three folders named Relevant, Irrelevant, and DataSet are created and filled according to a relevance rule, then each of those folders contains 600, 400, and 3,000 documents respectively. Following the code example above, the analyst creates a corpus with the documents contained within each folder and then combines each corpus into a full corpus, full.corpus and converts this object to a data frame. This data frame object will consist of 4,000 *ordered* observations, where the first 600 are observations that have been hand classified as Relevant, the next 400 have been hand classified as Irrelevant, and the next 3,000 will be machine classified with the algorithm. Because the first 1,000 observations of the data frame consist exclusively of the randomly sampled data used for training the classifier, the analyst can create a new data frame object consisting exclusively of the hand-classified data with the following code:

```
#Create training set dataframe
training.df <- full.df[1:1000, ]
```

This code simply assigns the first 1,000 rows of full.df (delineated by "1:1000") with all columns (delineated by ", "to the object training.df. Now that R knows which data will be used for training the classifier, the analyst must specify the results of the hand-classification. Recall that observations 1

through 600 have been classified as Relevant and that observations 601 through 1000 have been classified as Irrelevant. The analyst can therefore simply create a vector called relevance class with the following code:

```
#Create relevance-class vector

rclass <- c(rep("relevant", 600), rep("irrelevant", 400))
rclass <- factor(rclass)
```

This code creates a vector consisting of 600 repetitions of "relevant" and 400 repetitions of "irrelevant", and then converts the character vector to a factor vector.

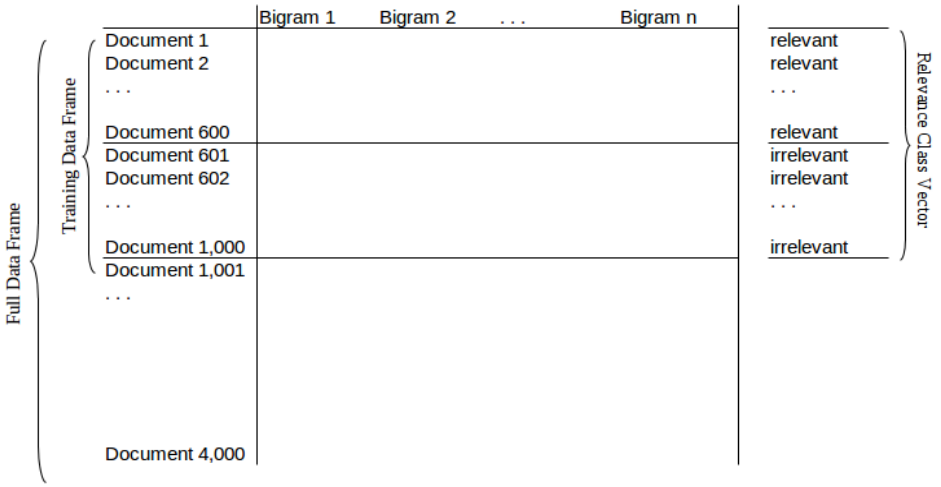

Fig. 8: A Visual Depiction of Data Structures Ready for Naïve Bayes

Before training the algorithm, the analyst must apportion a percentage of the training data frame observations for testing. Typically, the data will be divided on a 90:10

(training to testing) basis.[185] To estimate model performance, the data is further divided into folds, where each fold is a random partition.[186] For example, if the hand-classified random sample consists of 1,000 observations, 10-fold partitioning creates 10 folds of 100 observations each. The folds are used to cross-validate the algorithm against different portions of the training data. That is, the analyst trains the algorithm 10 times with 10 different (randomly selected) training sets and she predicts 10 different (randomly selected) testing sets in order to evaluate overall model performance. In essence, 10-fold cross-validation creates 10 versions of the model with the same universe of data and measures the performance of each model.[187] To create the folds, enter the following code in the classification script:

```
#Create folds for cross-validation

set.seed(123)
folds <- createFolds(rclass, k = 10)
```

The set.seed function ensures that selection of the observations for each fold is random, and the createFolds function creates ten folds of 100 rclass values.

With the following code, fit the naïve Bayes model and predict values of the test data across all folds:

```
# Fit model and predict values of each fold

cv.results <- lapply(folds, function(x) {
```

---

[185] *See* Stack Exchange, *supra* note 153.
[186] LANTZ, *supra* note 109, at 319.
[187] While any number of folds may be specified within R, empirical evidence suggests little benefit from using more than 10. *Id*.

```
   r.nb <- naiveBayes(training.df[-x, ], rclass[-x], laplace =
1)

r.nb.predict <- predict(r.nb, training.df[x, ], type = "class")

conf.mat <- table(r.nb.predict, rclass[x])
accuracy <- sum(diag(conf.mat)) / length(x) * 100
return(accuracy)
})
```

The lapply() function applies function(x) to each of the folds and assigns the results to cv.results. The function(x) consists of training the naïve Bayes model, predicting test data values with the trained model, and assessing its accuracy with a confusion matrix.[188] The function returns a percentage value called accuracy, which gives the percentage of correctly predicted test data for each fold. To output a list of the results for each fold, enter str(cv.results). The results of cross-validation should be reported in a study.

How does one interpret the results? Macey and Mitts obtained a 76.62% average accuracy;[189] Fagan obtained a 73.38% average accuracy. It remains unsettled what is considered acceptable for big data legal research, though considerations of acceptability should depend upon how exactly the machine-classified data will be analyzed. For example, a lower average accuracy may be acceptable for relevance filtering where the machine-classified data will only

---

[188] "A confusion matrix is a table that categorizes predictions according to whether they match the actual value in the data." *Id*. at 298. Predicted classes are columns and actual classes are rows. Thus, the diagonal consists of correct results and the off-diagonal consists of incorrect results. *See id*.
[189] Macey & Mitts, *supra* note 17, at 147.

be used for topic modeling. Instead, if the machine-classified data will be used for regression analysis, a lower average accuracy may produce doubtful results for questions of predictability.

Once satisfied with the classifier, use it to machine-classify the unclassified documents in the dataset with the following code:

```
# Classify full dataset with relevance classifier

d.nb.predict <- predict(r.nb, full.df[601:4000], type = "class")
```

Finally, remove the irrelevant documents from the DataSet folder. This can be accomplished by creating a folder named, for example, Irrelevant, and moving the irrelevant from DataSet to Irrelevant. To output a list of irrelevant documents for removal, create a vector of irrelevant opinions:

```
# Output irrelevant opinions

irrelevant.idx <- d.nb.predict[d.nb.predict=="irrelevant"]
```

Call the object from the Console with the command `str(irrelevant.idx)` and remove the documents from the folder DataSet.[190] The DataSet folder now consists only of relevant documents and the data is ready for further analysis.

[190] This may be accomplished with the use of an automated script, though its construction is outside the scope of this Article.

2. Topic Modeling

While learning algorithms such as decision trees, artificial neural networks, and naïve Bayes can be used to model topics (e.g. relevant vs. irrelevant), their primary function is to automate hand-classification tasks through machine learning. Thus, classification algorithms require hand classifying a portion of the data in order to train a classifier, which is then used to classify the remaining data.[191] Probabilistic topic modeling algorithms, instead, evaluate the entire dataset at once and discover thematic information without requiring any prior annotation.[192]

Consider for example, a law review article entitled, "Predicting Search and Seizure Rules of Mobile Data Devices with Big Data Text Analysis." Assume that the article contains words about (1) *data analysis* such as "computer" and "prediction"; (2) the *Fourth Amendment* such as "search" and "warrant"; and (3) *mobile technology* such as "cellphone" and "tablet." Knowing that the article blends those topics helps organize it within a collection of law review articles. Topic models formally define a topic as a distribution of words over a fixed vocabulary.[193] For example, the word "warrant" occurs within the Fourth Amendment topic at a high probability and the word "predict" occurs within the data analysis topic with a high probability. Topic models assume that topics exist before any data has been generated.[194] Which documents belong to which topics? With topic modeling, all documents belong to all topics, but "each document exhibits those topics in different

---

[191] *Supra* text accompanying note 165.
[192] Blei, *supra* note 31, at 77–78.
[193] *Id*. at 78.
[194] *Id*.

proportion."[195] Thus, each document is distributed over all topics. Likewise, each word within each document is distributed over a given topic's fixed vocabulary. Through a statistical technique known as Latent Dirichlet Allocation, the observed per-document words are used to uncover the hidden topic structure, where the topic structure is conditioned upon the observed per-document words.[196]

In practice, the analyst applies a topic model to a corpus of documents by choosing a number of topics. For example, if the analyst chooses to create a three-topic model, the algorithm will return three lists of words labeled topic one, topic two, and topic three. The analyst must then infer the nominal topics from the list of words that have been grouped by the algorithm. Thus in the veil-piercing topic model constructed in Macey and Mitts the topic model returned a list where the first five bigrams were "breach contract," "breach fiduciary," "implied warranty," "employment contract," and "term contract."[197] This list constituted a topic within the context of the study, which was nominalized as *Misrepresentation*.[198]

How many topics should the analyst choose to specify? Current studies model the results of three-, four-, five-, or six-topic models and then compare the results. For example, Macey and Mitts model varying number of topics and note that the topics "similarly show . . . organiz[ation] more or less along the lines of the three-theory justification we posit, with additional topics generally reflecting more nuanced distinctions

---

[195] *Id*. at 79.
[196] *Id*. at 80.
[197] Macey & Mitts, *supra* note 17, at 151.
[198] *Id*.

between specific types of statutory regimes."[199] Ideally, an optimal number of topics would be generated by the data itself. However, current methods for determining the optimal number of topics exhibit varying weaknesses and remain limited.[200]

Matt Taddy of the University of Chicago Booth School of Business has created an R package for modeling topics named map topics ("maptpx"). Create a new script within the current RStudio project and save it with a name such as TopicScript. Enter the following code to load the relevant libraries:

```
library(tm)
library(plyr)
library(maptpx)
```

Create a corpus with the text mining package function Corpus. Path to the results of the naïve Bayes classification analysis, which should be in a folder named Relevant. If classification was not performed, simply path to the collected data:

```
# Create corpus

tort.corpus <- Corpus(DirSource("path to Relevant data or complete dataset"))
```

Clean the corpus with the techniques detailed in Section II.D.1., create a tokenized Document-Term Matrix, and remove sparse terms that occur in less than two percent of the documents. In order to create a topic model with maptpx, the

---

[199] *Id*. Similarly, Fagan analyzes three- four- and five-topic models and notes no significant differences. Fagan, *supra* note 18, at 411 n.77.
[200] Matthew A. Taddy, *On Estimation and Selection for Topic Models*, Proc. 15th Int'l Conf. Art. Int. & Stat. 1184, 1186 (2012).

data must be in the form of a matrix data structure. Enter the following code into the script for conversion:

```
# Create matrix

full.mat <- as.matrix(full.dtm)
```

Now that the data has been converted, create a four-topic model with the following code:

```
# Model topics of relevant opinions

tortTopics <- topics(full.mat, 4)
```

The number of topics may be set with the numerical parameter that follows the comma. To output the lists of bigrams, enter:

```
# Output topics

tortTopics.mat <- as.matrix(tortTopics$theta)
write.csv(tortTopics.mat, "path to Project folder"/test.csv)
```

This code will output a file named test.csv, which can be opened inside Microsoft Excel, to the Project folder. Examine the lists of bigrams. Often, the lists will contain nonsensical bigrams that can be eliminated. For example, legal writing terms such as "omit emphasis" or geographical terms such as "New York" may provide no interpretive meaning.[201]

3. Regression Analysis

In addition to classifying documents and modeling topics, big data legal scholars perform regression analysis in

---

[201] *See* Macey & Mitts, *supra* note 17, at 148 (ignoring nonsensical words in the context of multinomial inverse regression).

order to understand relationships between words and legal outcomes or other points of interest.[202] Regressing text on dependent variables such as legal outcomes is generally difficult to incorporate into traditional statistics because of the large dimension of regressors, i.e., the words contained within many documents.[203] To the extent that numerous regressors preclude the analyst from using traditional regression models, inverse regression may provide a solution. Taddy has developed a novel estimation technique expressly for obtaining logistic estimations of very high dimension data that yields stable and effective results. Taddy explains its application to marketing, economics, and political science:

> [Multinomial inverse regression] investigates the relationship between text data—product reviews, political speech, financial news, or a personal blog post—and variables that are believed to influence its composition—product quality ratings, political affiliation, stock price, or mood polarity. Such language-motivating observable variables, generically termed *sentiment* . . . are often the main object of interest for text mining applications. When, as is typical, large amounts of text are available but only a small subset of documents are annotated with known sentiment, this relationship yields a powerful potential for text to act as a stand-in for related quantities of primary interest.[204]

---

[202] *See, e.g.*, *id*. at 147 (performing multinomial inverse regression in order to explore a predictive relationship between judicial decision to pierce the corporate veil and the text contained within a decision).
[203] Matt Taddy, *Multinomial Inverse Regression for Text Analysis*, 108 J. AM. STAT. ASS'N 755 (2013).
[204] *Id*. at 2.

In the context of legal scholarship, multinomial inverse regression, and regression analysis generally, investigates the relationship between legal texts—such as judicial opinions, statutes, regulations, etc.—and variables that are believed to influence their composition—such as judicial outcomes, various statutory or regulatory characteristics, etc. This type of analysis is typically of great interest. For example, Macey and Mitts demonstrate that various fact patterns represented by words help predict (or do not help predict) judicial outcomes.[205] Thus, they were able to support their hypothesis that undercapitalization does not impact a judicial decision to pierce the corporate veil, while fact patterns involving "misrepresentation," promotion of "bankruptcy values," and promotion of "statutory schemes" do so impact.

The base R package includes a number of regression models and Taddy has created an additional package that implements text inverse regression ("textir").[206] The analyst should apply the same code principles above: load the package, create and clean the corpus, and covert the corpus to the required data structure.[207] The analyst will additionally need to pass a matrix of outcomes to the multinomial inverse regression function. For example, if the court holds a successor liable in documents 1 through 500, and not liable in documents 501 through 2,000, then the analyst will need to create an outcome matrix which contains that information. This will be passed along with the corpus matrix.[208] Finally, output the data and evaluate the coefficients of the tokenized

---

[205] Macey & Mitts, *supra* note 17, at 148 Table 4.
[206] Taddy, *supra* note 203.
[207] Matt Taddy, *Package 'textir,'* 1, 4 https://cran.r-project.org/web/packages/textir/textir.pdf (last visited Sept. 10, 2015).
[208] *See id*. at 4–5.

words.[209] Obviously, the accuracy of the coefficients will be based upon the accuracy of the outcome matrix. If, for example, the analyst plans to create the outcome matrix with machine classification, the accuracy of the classifier must be considered when evaluating the validity of the regression, which will directly bear on the decision to perform or not perform multinomial inverse regression.[210]

The big data legal scholar has several other tools available to evaluate the relationship between texts and outcomes in the event that the accuracy of the classification algorithm prevents the use of regression analysis. Kosnik computed the word length of clauses contained within hydroelectric license contracts and regressed length against contract flexibility/rigidity.[211] She found that lengthier clauses resulted in higher levels of contract flexibility. Similarly, Kosnik regressed proportional word counts of high frequency words contained within her corpus against flexibility/rigidity. Because word length and frequency are not machine classified and are observable with certainty, their use as regressors are more likely to be accurate and robust. On the other hand, regressing a fact pattern represented with words against a legal outcome represents a powerful analytic tool for predicting legal outcomes such as judicial decision-making and statutory/regulatory contents to the extent that machine-classification is accurate.

---

[209] *Id*.
[210] *Cf.* Fagan, *supra* note 18, at 417 n.93.
[211] Kosnik, *supra* note 4, at 1.

IV. CONCLUSION

This Article has sought to take first steps toward developing a research program for big data legal scholarship by sketching its positive and normative components. The application of big data methods to the descriptive questions of law can generate broader consensus. This is because big data methods can provide greater comprehensiveness and less subjectivity than traditional approaches, and can diminish general disagreement over the categorization and theoretical development of law as a result. Positive application can increase the clarity of rules, uncover the relationship between judicial text and outcome, and comprehensively describe judicially-determined facts, the contents of legislation and regulation, or the contents of private agreements. Equipped with a normative framework, big data legal scholarship can lower the costs of judging, litigating, and administrating law; increase comparative justice and predictability; and support the advocacy of better rules and policies.

In addition to sketching theoretical foundations, this Article has sought to take first steps toward developing the core components of successful praxis. Handling and analyzing big data can be cumbersome, though the newcomer can avoid common pitfalls with care. Accordingly, there exist best practices for preprocessing, converting, and analyzing the data. Current analytical techniques germane to law include algorithmic classification, topic modeling, and multinomial inverse regression.

First steps, by definition, are incomplete. The contours of big data legal scholarship and practice will undoubtedly shift over time to reflect new techniques and prevailing normative questions. This Article merely aspires to generate a conversation about how big data can enhance our understanding of law—what it is and what it should be.