Suffolk Journal of Trial and Appellate Advocacy

Volume 21 | Issue 1 Article 4

1-1-2016

Peeking inside the Black Box: A Preliminary Survey of Technology Assisted Review (TAR) and Predictive Coding Algorithms for Ediscovery

Shannon Brown
Suffolk University Law School

Recommended Citation
21 Suffolk J. Trial & App. Advoc. 221 (2015-2016)

PEEKING INSIDE THE BLACK BOX: A PRELIMINARY SURVEY OF TECHNOLOGY ASSISTED REVIEW (TAR) AND PREDICTIVE CODING ALGORITHMS FOR EDISCOVERY

Shannon Brown, Esq., MA, JD[*]

---

[*] Attorney Shannon Brown practices technology law in Pennsylvania, teaches eDiscovery as an adjunct professor of law, holds a technical cybersecurity certification, and develops machine learning software for the legal community. He frequently writes on technology-related topics and consults on complex, technology law matters.

> *[W]hat we call a "wrong datum" is one which is inconsistent with all other known data. It is our only criterion of right and wrong. It is the Machine's as well. Order [the Machine] . . . to direct agricultural activity on the basis of an average July temperature in Iowa of 57 degrees Fahrenheit. [The Machine] won't accept that. It will not give an answer.—Not that it has any prejudice . . . or that an answer is impossible; but because, in light of all the other data fed it over a period of years, it knows that the probability of an average July temperature of 57 is virtually nil. It rejects that datum.*[1]

0 INTRODUCTION

Technology assisted review (TAR) and predictive coding software tools may allow attorneys to more efficiently and more effectively respond to eDiscovery requests—requests increasingly mired in a seemingly endless data deluge. But, unfortunately, busy attorneys may view these new tools as yet another type of new-fangled software that "I just need to somehow use." But TAR and predictive coding, and even their predecessor, keyword search, require more than just knowing how to mechanically "press buttons" to use the software. Knowing when, how, if, and what type of tool to deploy, or what type of tools to chain together, become part of the lawyer's duty and the successful lawyer's toolbox. And fulfilling that duty or creating that toolbox requires at least some technical understanding of how the technologies work. Thus, the new software may seem like magic but isn't once you understand some basics.[2]

---

[1] ISAAC ASIMOV, I, ROBOT 217 (Bantam Books 2008) (1950). Robots, or "the machines," paralleled what we think of today as computers and software. While science fiction in 1950, such software, which analyzes patterns in data and makes decisions and predictions, is in full use today in many industries including use in the legal community.

[2] Likewise, a basic technical understanding of these technologies is not so far beyond the

A number of eDiscovery articles and books address the procedural and case law aspects of eDiscovery.[3] Very few articles, however, address the technical aspects of technology assisted review (TAR) and predictive coding eDiscovery tools.[4] Yet, as will become evident, the "technical" aspects of eDiscovery raise important legal issues and reflect the transformation of the legal profession into one where attorneys will need both technical and legal skills to competently represent clients. Simply stated, attorneys can no longer uncritically rely on outside advisors or blindly accept "black box" results.

This article provides a much-needed overview of:

1. the ethical obligations arising from a lawyer's requirement to "know technologies,"

---

comprehension of diligent attorneys as to represent rocket science—as demonstrated in classes that I teach on this topic.

[3] *See, e.g.*, MANAGING E-DISCOVERY AND ESI: FROM PRE-LITIGATION THROUGH TRIAL 1 (Michael D. Berman et al. eds., 2012) (consisting of ABA overview of eDiscovery topics); SHIRA A. SCHEINDLIN ET AL., THE SEDONA CONFERENCE'S ELECTRONIC DISCOVERY AND DIGITAL EVIDENCE IN A NUTSHELL 1 (2009) (providing comprehensive overview of eDiscovery topics); Thomas Y. Allman, *The Sedona Principles and the 2006 Federal Rule Amendments Addressing E-Discovery*, 1 FED. CTS. L. REV. 15 (2006) (identifying future eDiscovery issues in early work); Steven C. Bennett, *E-Discovery: Reasonable Search, Proportionality, Cooperation, and Advancing Technology*, 30 J. MARSHALL J. INFO. TECH. & PRIVACY L. 433, 433-463 (2014) (providing overview of electronic evidence); Robert D. Brownstone, *Preserve or Perish; Destroy or Drown—eDiscovery Morphs into Electronic Information Management*, 8 N.C. J. L. & TECH. 1, 1-2 (2006) (providing early discussion of preservation duties); Millberg LLP & Hausfeld LLP, *E-Discovery Today: The Fault Lies Not in Our Rules . . .* , 4 Fed. Cts. L. Rev. 131, 131-183 (2011) (giving 2010 position on eDiscovery challenges); Burke T. Ward et al., *Electronic Discovery: Rules for a Digital Age*, 18 B.U. J. SCI. & TECH. L. 150, 150-198 (2012) (detailing eDiscovery rules and procedures).

[4] The rare exceptions are the works by Gordon Cormack and Maura Grossman. *See, e.g.*, Gordon V. Cormack & Maura R. Grossman, *Evaluation of Machine-Learning Protocols for Technology-Assisted Review in Electronic Discovery*, PROCEEDINGS OF THE 37TH INT'L ASS'N OF COMPETING MACHINERY SPECIAL INT. GROUP INFO. RETRIEVAL CONF. 153, 153-62 (2014), http://dl.acm.org/citation.cfm?doid=2600428.2609601 [hereinafter Cormack & Grossman, *Evaluation of Machine-Learning Protocols*] (comparing effectiveness of three machine-learning protocols for technology assisted reviews); Maura R. Grossman & Gordon V. Cormack, *The Grossman-Cormack Glossary of Technology-Assisted Review*, 7 FED. CTS. L. REV. 1, 1-34 (2013), http://www.fclr.org/fclr/articles/html/2010/grossman.pdf [hereinafter *The Grossman-Cormack Glossary*] (introducing common framework and set of definitions relating to TAR); Maura R. Grossman & Gordon V. Cormack, *Technology-Assisted Review in E-Discovery Can Be More Effective and More Efficient Than Exhaustive Manual Review*, 17 RICH. J.L. & TECH. 11, 11-61 (2011) (arguing technology-assisted processes are more efficient and precise compared to manual review process); *see also* Jacob Tingen, *Technologies-That-Must-Not-Be-Named: Understanding and Implementing Advanced Search Technologies in e-Discovery*, 19 RICH. J.L. & TECH. 1, 3 (2012) (grappling with technologies in a more direct manner).

2. the latent problems in the legal profession stemming from the legal community's troubling confusion about the various technologies applicable to eDiscovery tasks,[5]
3. some of the predicate, preprocessing techniques used by the tools to transform human-readable materials to machine-readable materials, and
4. some of the main algorithms and methods used in TAR and predictive coding tools to allow the legal community to better understand how to apply specific technologies to specific problems.

Thus, this article provides a solid, preliminary introduction, since a comprehensive analysis could fill volumes, as to why attorneys must delve further into understanding the actual technologies and why the application and implementation of these technologies represents modern law practice.

Section 1 addresses the important and growing recognition that lawyers simply must both know how technologies function and know how to apply the technologies to specific problems.[6] The section addresses the recent changes to the model ethics rules requiring technology knowledge as a condition of legal competence.[7] The section also addresses relevant case law and commentary by the judiciary on technology awareness.[8] As the section illustrates, however, even judicial commentary sometimes avoids the technical issues or misconstrues technologies.[9]

Section 2 provides a brief summary of the origins of TAR and predictive coding systems.[10] Lawyers should be aware that a number of

---

[5] This article pays particular attention to a latent, but fundamental, problem in the legal community related to misunderstanding the current technologies. The misunderstanding arises from incorrect conflation of eDiscovery technologies used for search tasks versus eDiscovery technologies used for classification or grouping tasks. *See infra* Section 2.1 (noting differences in terminology) and Section 4.1 (distinguishing TAR and search). "Search" technologies retrieving specific documents or items based on *a priori* knowledge of the document set usually employ "keywords" to retrieve "matching" documents. *See infra* Section 4.1. In sharp contrast, classification algorithms group documents into like groups (or classify documents into categories) based on meaning or similarity attributes between the documents and typically require some type of preliminary analysis by a lawyer of a subset of documents which trains the eDiscovery classification algorithm. *See id.*

[6] *See infra* Section 1 (discussing legal practitioner's duty to know technology).

[7] *See infra* Section 1.1 (outlining changes in Model Rules regarding lawyer's use of technology and confidentiality).

[8] *See infra* Section 1.2 (discussing federal cases that address technical challenges related to electronically stored information (ESI)).

[9] *See id.*

[10] *See infra* Section 2 (discussing origins of data analytics tools).

academic disciplines contribute to TAR and predictive coding.[11] Lawyers should also be aware that practitioners of each discipline may bring discipline-specific terminology and discipline-specific predispositions to the legal community.[12] Lawyers must be aware of these predispositions and terminologies to avoid silly disputes and wasted time.[13]

Section 3 addresses the "pre-processing" of digital documents.[14] Essentially, pre-processing transforms human-readable documents into formats suitable for analysis by TAR and predictive coding tools.[15] Pre-processing involves unfamiliar techniques such as stemming, feature selection, text-to-matrix conversion, and parsing.[16] As will become evident, pre-processing demonstrates the types of "technical" decisions that a seemingly "technical" task requires—and why attorneys must be well-versed in the techniques to assure proper application.[17]

Section 4 delves into some of the algorithms and methods used by TAR and predictive coding tools.[18] This section explores how the algorithms function, identifies some of the strengths and weaknesses of the algorithms, and explains the practical application of the algorithms.[19] Lawyers, albeit perhaps unfamiliar, should not fear the plots, graphs, tables, and diagrams necessary to illustrate the algorithms.

1 A LEGAL PRACTITIONER'S DUTY TO KNOW TECHNOLOGY

Lawyers have a legal duty to "know technology."[20] That duty applies specifically to eDiscovery.[21] Computer technologies pose practical and pragmatic benefits (and risks) for both the legal practitioner and the

---

[11] *See id.* (listing disciplines such as machine learning, applied mathematics, statistics, robotics, artificial intelligence, economics, and others).

[12] *See id.* (noting each community has its own terminology, disciplinary mindset, and emphases).

[13] *See id.*

[14] *See infra* Section 3 (outlining data pre-processing issues).

[15] *See id.* (explaining range of techniques to transform documents into forms acceptable for processing by TAR algorithms).

[16] *See infra* Sections 3.2-3.5.

[17] *See infra* Section 3.9 (noting importance of understanding how features are extracted in a machine learning context and implications).

[18] *See infra* Section 4 (distinguishing Keyword Search, Technology Assisted Review (TAR), and Predictive Coding Systems).

[19] *See id.*

[20] *See* MODEL RULES OF PROF'L CONDUCT R. 1.1 cmt. 8 (2015) ("To maintain the requisite knowledge and skill [to remain ethically competent], a lawyer should keep abreast of changes in law and its practice, including the benefits and risks associated with relevant technology . . . ."); Tingen, *supra* note 4, at 3.

[21] *See* Monica McCarroll, *Discovery and the Duty of Competence*, 26 REGENT U. L. REV. 81, 102-10 (2013-14) (discussing application of lawyer competence to e-discovery tasks).

practitioner's clients. Clients increasingly demand, and expect, technology competence from lawyers and legal staff.[22] Judges, especially in federal courts, increasingly comment on the need for technical competence.[23] Thus, put simply, lawyers can no longer hide from computer technologies[24]—especially when the technologies promise enhanced efficiency at significantly lower costs.[25]

1.1 Duty Stated in Ethics Rule Changes

The American Bar Association recently revised the definition of competent law practice—or at least emphasized an already existing, albeit latent, change.[26] In 2012, the American Bar Association's 20/20

---

[22] *See* Monica Bay, *Small Firms Steal Business from Big Law*, LAW TECH. NEWS, June 29, 2014, *available at* https://advance.lexis.com/api/permalink/207a090c-cfd8-4eeb-90ae-f85f3c205362/?context=1000516 (commenting on technology competence as law firm differentiator); *see also* Marlisse S. Sweeney, *Suffolk Law Launches Legal Tech Audit*, LAW TECH. NEWS, Sept. 3, 2014, *available at* https://advance.lexis.com/api/permalink/0337b75d-7919-4cab-aeac-a2666763c4b8/?context=1000516 (discussing results from legal tech audit).

[23] *See* William A. Gross Constr. Assocs., Inc. v. Am. Mfrs. Mut. Ins. Co., 256 F.R.D. 134, 135-36 (S.D.N.Y. 2009) (holding attorneys must cooperate and craft appropriate keywords for non-party to use in searching emails). In a somewhat amusing and blunt opinion, Federal Magistrate Judge Andrew Peck stated, "[t]his case is just the latest example of lawyers designing keyword searches in the dark, by the seat of the pants, without adequate (indeed, here, apparently without any) discussion with those who wrote the emails." *Id.* at 135. The opinion earlier opens with:

> This Opinion should serve as a wake-up call to the Bar in this District about the need for careful thought, quality control, testing, and cooperation with opposing counsel in designing search terms or "keywords" to be used to produce emails or other electronically stored information ("ESI"). While this message has appeared in several cases from outside this Circuit, it appears that the message has not reached many members of our Bar.

Id. at 134. While only one example, such comments illustrate the frustration of judges with apparently less-then-technically-sophisticated legal counsel. *See id.*

[24] The legal profession experienced a similar technological shift in the 1990s—although the eDiscovery shift foretells a shift of larger magnitude and of more technical detail. Earlier, the legal profession shifted from primarily book-based research to online legal research tools. *See, e.g.*, Jean McKnight, *WEXIS Versus the Net*, 85 ILL. B.J. 187, 187 (1997); Jesse J. Richardson, Jr., *How a Sole Practitioner Uses the "Electronic Office" to Maintain a Competitive Law Practice*, 3 DRAKE J. AGRIC. L. 141, 142-43 (1998) (discussing how new electronic technologies level playing field for small firms). While online, or computer-based, research was novel at the time, online research quickly became an indispensable tool for competent law practice. Lawyers needed to quickly adapt to the new, technology-driven tools.

[25] *See* Patrick Oot, Anne Kershaw & Herbert Roitblat, *Mandating Reasonableness in a Reasonable Inquiry*, 87 DENVER U. L. REV. 533, 533-35, 551 (2010) (providing hard-hitting, but much needed, analysis of need for technical competency for efficiency and reasonableness).

[26] *See* MODEL RULES OF PROF'L CONDUCT R. 1.1 cmt. 8 (2015) ("To maintain the requisite knowledge and skill, a lawyer should keep abreast of changes in the law and its practice,

Committee issued a report recommending changes to the *Model Rules of Professional Conduct*—used by most states as a template for state-based *Rules*.[27] The updates included: "to maintain the requisite knowledge and skill, a lawyer should keep abreast of changes in the law and its practice, *including the benefits and risks associated with relevant technology* . . . ."[28] The updates also emphasize that the technology-competence duty falls to the individual lawyer.[29] Furthermore, lawyers must properly supervise internal and external non-lawyer staff such as eDiscovery-outsourcing vendors.[30] To date, twenty-five states have already incorporated the *Model Rule* updates into state *Rules*.[31] Axiomatically, the rule changes apply in an eDiscovery context.[32]

The California State Bar recently issued a request-for-comments on a formal, advisory, eDiscovery ethics opinion that states, in part, that attorneys must become familiar with eDiscovery technologies, associate with attorneys (professionals) with those skills, or decline representation.[33]

---

including the benefits and risks associated with relevant technology . . . ."); ABA Comm. on Ethics & Prof'l Responsibility, Rep. to the House of Delegates 105A Revised, at 3 (Aug. 6, 2012), http://www.americanbar.org/content/dam/aba/administrative/ethics_2020/20120808_revised_resolution_105a_as_amended.authcheckdam.pdf (approving new language regarding lawyer's use of technology and confidentiality).

[27] *See* ABA Comm. on Ethics & Prof'l Responsibility, Rep. to the House of Delegates 105A Revised, at 3.

[28] *Id.* (emphasis added).

[29] *See* MODEL RULES OF PROF'L CONDUCT R. 1.1 cmt. 8 (2015); ABA Comm. on Ethics & Prof'l Responsibility, Rep. to the House of Delegates 105A Revised, at 3; *see also* Rachel K. Alexander, *E-Discovery Practice, Theory, and Precedent: Finding the Right Pond, Lure, and Lines Without Going on a Fishing Expedition*, 56 S.D. L. REV. 25, 44-45 (2011) ("It is important to bear in mind that many commentators and bar organizations recognize that an attorney's ethical duties, particularly those of competence and diligence, are applicable in the e-discovery context.").

[30] *See* MODEL RULES OF PROF'L CONDUCT R. 5.3 cmts. 1-3 (2015) (outlining responsibilities regarding nonlawyer assistance); ABA Comm. on Ethics & Prof'l Responsibility, Rep. to the House of Delegates 105C, at 2-3 (Aug. 6, 2012), http://www.americanbar.org/content/dam/aba/administrative/ethics_2020/2012_hod_annual_meeting_105c.authcheckdam.pdf (recommending change to *Model Rules* regarding outsourcing of client matters).

[31] *See* ABA Ctr. for Prof'l Resp. Pol'y Implementation Comm., *Chronological List of States Adopting Amendments to their Rules of Professional Conduct based upon the August 2012 policies of the ABA Commission on Ethics 20/20* (Dec. 21, 2015), http://www.americanbar.org/content/dam/aba/administrative/professional_responsibility/chron_adoption_e_20_20_amendments.authcheckdam.pdf.

[32] *See* Alexander, *supra* note 29, at 44-45 ("It is important to bear in mind that many commentators and bar organizations recognize that an attorney's ethical duties, particularly those of competence and diligence, are applicable in the e-discovery context.").

[33] *See* The State Bar of California Standing Comm. on Prof'l Resp. and Conduct, Formal Op. Interim No. 11-0004, at 1 (June 24, 2014), http://www.calbar.ca.gov/Portals/0/documents/publicComment/2014/2014_11-0004ESI03-21-

The opinion generated significant comment from the bar[34] but plainly illustrates the transformed nature of the profession much as the ABA's *Model Rule* updates reflect such changes.[35]

Lawyers relying on non-lawyers has become a significant problem because many "technical" issues involve significant legal judgment and legal analysis.[36] For example, the California State Bar advisory opinion implies that attorneys may rely on a "non-lawyer technical expert" to fulfill California expectations on ethical competence related to eDiscovery.[37] The ABA's *Model Rules* also state that lawyers "may use non-lawyers outside the firm to assist the lawyer in rendering legal services to the client."[38] But, the ABA *Model Rules* carefully note that a lawyer cannot blindly rely on non-lawyer staff or non-lawyer vendors and that the lawyer retains full responsibility for assuring that the non-lawyer assistance fully complies with the lawyer's ethical duties.[39] In other words, lawyers must know about the technologies themselves.

Interestingly, the District of Columbia Court of Appeals recently addressed the growing problem of non-lawyer ownership of discovery vendors and held that eDiscovery-services companies who are not authorized to practice law cannot offer services in Washington, D.C.[40]

14.pdf (outlining responsibilities regarding electronically stored information ("ESI") and discovery requests).

[34] *See* Debra C. Weiss, *Botched E-Discovery Can Be an Ethics Violation, Proposed Opinion Says*, A.B.A. J. (Apr. 14, 2014), http://www.abajournal.com/news/article/botched_e-discovery_can_be_an_ethics_violation_proposed_opinion_says (discussing requirements imposed by California state bar); Philip Favro, *What California's e-Discovery Ethics Opinion Means for In-house Counsel*, INSIDE COUNSEL (May 9, 2014), http://www.insidecounsel.com/2014/05/09/what-californias-e-discovery-ethics-opinion-means (describing steps in-house counsel can take to protect client's interest).

[35] *See* The State Bar of California Standing Comm. on Prof'l Resp. and Conduct, *supra* note 33, at 3. The opinion concisely summaries a core insight: "Not every litigated case ultimately involves e-discovery; however, in today's technological world, almost every litigation matter *potentially* does." *Id.* (emphasis in original).

[36] *See* Shannon Brown, *Potential Problems & Perils of eDiscovery Outsourcing*, SHANNON BROWN L. BLOG (June 3, 2013), http://www.shannonbrownlaw.com/archives/1719.

[37] *See* The California Standing Comm. on Prof'l Resp. and Conduct, *supra* note 33, at 3.

[38] MODEL RULES OF PROF'L CONDUCT R. 5.3 cmt. 3 (2015) (outlining responsibilities regarding nonlawyer assistance).

[39] *See* MODEL RULES OF PROF'L CONDUCT R. 1.1, 1.0, 5.3 cmt. 3 (2015). Model Comment 3 to Rule 5.3, along with Rule 1.0 and Rule 1.1, plainly state that the lawyer cannot simply delegate the lawyer's own duty of competence. *Id.* That means that the lawyer must personally be familiar with the technologies to provide the proper oversight and responsibility. *See id.* at R. 1.1 cmt. 6, 5.3 cmt. 3.

[40] *See* Applicability of Rule 49 to Discovery Services Companies, D.C. Cir. Comm. on Unauthorized Practice of Law, UPL Op. 21-12, at 8 (Jan. 12, 2012), http://www.dcappeals.gov/internet/documents/21-Opinion-21-12.pdf.

Other jurisdictions would likely make similar conclusions once lawyers better understand the technical aspects of eDiscovery.[41]

The interplay of technology competence as an integral part of legal competence may confound many attorneys accustomed to simply "delegating" such "computer" tasks to subordinates or non-lawyer staff. But, as this article illustrates, the technologies themselves require significant legal judgments and legal analysis, and the revised ethical duties reflect the reality of these changes.[42]

1.2 Courts Sending Wakeup Call

Several federal cases sent a "wake-up call" to the legal profession regarding eDiscovery.[43] The substantive law varies in the cases. But, all

---

[41] Lawyers seem to confuse the evidentiary use of experts with a lawyer's *personal* duty of legal competence. *See* Alice Nelson, *Expert Testimony*, *in* FEDERAL PRACTICE MANUAL FOR LEGAL AID ATTORNEYS § 6.6 (2014) (discussing evidentiary use of expert testimony). Magistrate Judge Andrew Peck presciently noted this conflict in an early article regarding the inapplicability of *Daubert* in standard eDiscovery use. *See* Andrew Peck, *Search, Forward: Will Manual Document Review and Keyword Searches Be Replaced by Computer-Assisted Coding?*, LAW TECH. NEWS, Oct. 2011, at 25, 29, *available at* https://law.duke.edu/sites/default/files/centers/judicialstudies/TAR_conference/Panel_1-Background_Paper.pdf.

[42] *See* Helen Geib, *Three E-Discovery Trends Spurred by Proposed FRCP Amendments*, LAW TECH. NEWS, Dec. 29, 2014, *available at* https://advance.lexis.com/api/permalink/69d266d9-5591-46bb-96da-518b6420dafe/?context=1000516 ("Calls for a higher standard of technology competence among litigators gathered momentum throughout 2014 across case law, ethics opinions, and best practices commentary."). Increasingly, even the legal profession has started to realize the lack of technology skills. *See id.*

[43] *See* William A. Gross Constr. Assocs., Inc. v. Am. Mfrs. Mut. Ins. Co., 256 F.R.D. 134, 135-36 (S.D.N.Y. 2009) (using "wake-up call" language). The mid-2000s saw a flurry of eDiscovery-related cases on key topics such as duty of preservation, disclosure, adequacy of "search", and cooperation in addition to exploration of the 2006 updates to the Federal Rules of Civil Procedure. What becomes material here was the alleged novelty of the evidentiary issues in these cases and the uncertainty in the legal community during those years. Within a span of less than ten years, electronic evidence went from curious novelty to a core legal competency. *See* Pension Comm. of the Univ. of Montreal Pension Plan v. Banc of Am. Sec., LLC, 685 F. Supp. 2d 456, 475 (S.D.N.Y. 2010) (finding attorney duty to issue written litigation hold preserving electronic evidence); Rimkus Consulting Grp., Inc. v. Cammarata, 688 F. Supp. 2d 598, 613 (S.D. Tex. 2010) (addressing non-custodial discovery preservation); Victor Stanley, Inc. v. Creative Pipe, Inc., 250 F.R.D. 251, 263 (D. Md. 2008) (considering privileged evidence inadvertently disclosed due to eDiscovery error); Mancia v. Mayflower Textile Servs. Co., 253 F.R.D. 354, 357-58 (D. Md. 2008) (stating compliance with electronic discovery rules requires party cooperation); *see also* Zubulake v. UBS Warburg LLC (*Zubulake I*), 217 F.R.D. 309, 318-19 (S.D.N.Y. 2003); Zubulake v. UBS Warburg LLC (*Zubulake III*), 216 F.R.D. 280, 287-89 (S.D.N.Y. 2003); Zubulake v. UBS Warburg LLC (*Zubulake IV*), 220 F.R.D. 212, 218 (S.D.N.Y. 2003); Zubulake v. UBS Warburg LLC (*Zubulake V*), 229 F.R.D. 422, 436 (S.D.N.Y. 2004) (implying increased awareness of eDiscovery issues such as preservation in watershed series of cases).

address some technical challenge related to identifying, preserving, analyzing, or producing relevant, electronically stored information (ESI).[44] All also recognize the fundamental nature of technology in law practice—and a lawyer's duty to know how the technology works, how to apply the technology to specific problems, and how to defensibly address increasingly technologically-sophisticated courts.[45]

In an early case, *United States v. O'Keefe*,[46] the defendant challenged the efficacy of the prosecution's search of government records including keyword searches of "electronic record files."[47] Regarding the keyword searches, Magistrate Judge John M. Facciola responded:

> Whether search terms or "keywords" will yield the information sought is a complicated question involving the interplay, at least, of the sciences of computer technology [sic], statistics and linguistics . . . . Given this complexity, for lawyers and judges to dare opine that a certain search term or terms would be more likely to produce information than the terms that were used is truly to go where angels fear tread.[48]

The opinion holds that lawyers challenging keyword-search results must support such challenges with expert analysis compliant with Federal Rules of Evidence Rule 702.[49]

Despite the Rule 702 problem and the seemingly exaggerated "angels fear tread" language in this context, the opinion nevertheless implies that attorneys must have a working knowledge of eDiscovery-related technologies.[50]

---

[44] *See The Grossman-Cormack Glossary*, *supra* note 4, at 15 (defining "ESI"); sources cited *supra* note 43 (citing eDiscovery cases related to duty of preservation, disclosure, adequacy of "search", and cooperation).

[45] *See* sources cited *supra* note 43 and accompanying text (observing how electronic evidence shifted from novelty to core legal competency); *William A. Gross*, 256 F.R.D. at 135 (issuing "wake-up" call).

[46] 537 F. Supp. 2d 14 (D.D.C. 2008) (fairly rare criminal case implicating eDiscovery).

[47] *See id.* at 16, 17-18, 22-24.

[48] *Id.* at 24. The criminal case involved accusations of improperly expediting visa requests for a co-defendant's company. *See id.* at 15-16. The specific search terms included "early or expedite* or appointment or early & interview or expedite* & interview." *See id.* at 18.

[49] *See id.* at 24. The applicability of Rule 702 to eDiscovery remains an open question. *See* Peck, *supra* note 41, at 29.

[50] *See O'Keefe*, 537 F. Supp. 2d at 16, 17-18, 22-24. The technology duty included not just keyword searches, but also the duty to preserve and produce documents. *See id.*

Also in 2008, Federal Magistrate Judge Paul Grimm addressed a lawyer's duty of technical competency in eDiscovery.[51] Magistrate Judge Grimm held that privilege may be waived by voluntary, albeit perhaps unintentional, disclosure of otherwise privileged materials during eDiscovery.[52] The opinion specifically addresses the technical problems leading to the waiver.[53] The methods used involved keyword searches by "forensics" experts hired to identify relevant materials and, at least in part, to identify privileged materials.[54] According to the Defendants, only 4.9 Gigabytes of the materials reviewed were electronically, text-searchable, and 33.7 Gigabytes required page-by-page review.[55] Some documents slipped through.[56]

Regarding the lawyer's duty to know technology, Magistrate Judge Grimm aptly and powerfully noted:

> While it is known that M. Pappas (a party) and Mohr and Schmid (attorneys) selected the keywords, nothing is known from the affidavits provided to the court regarding their qualifications for designing a search and information retrieval strategy that could be expected to produce an effective and reliable privilege review. As will be discussed, while it is universally acknowledged that keyword searches are useful tools for search and retrieval of ESI, all keyword searches are not created equal; and there is a growing body of literature that highlights the risks associated with conducting an unreliable or inadequate keyword search or relying exclusively on such searches for privilege review. Additionally, the Defendants do not assert that any sampling was done of the text searchable ESI files that were determined not to contain privileged information on the basis of the keyword

---

[51] *See* Victor Stanley, Inc. v. Creative Pipe, Inc., 250 F.R.D. 251, 253-54 (D. Md. 2008) (finding defendants failed to take reasonable precautions to prevent disclosure of privileged information).

[52] *See id.* Magistrate Judge Grimm noted that some of the materials in question did not qualify as privileged or protected. *See id.* at 254 n.1.

[53] *See id.* at 257 (discussing facts leading to waiver of privilege).

[54] *See id.* at 254-56.

[55] *See id.* at 256. Interestingly, the Plaintiffs "vigorously disput[ed]" the alleged inability to search the materials for relevant and privileged information and claimed that all or most of the applicable materials were text searchable using "a readily-available desktop search tool." *Id.* at 257. Again, the dispute centers on the technical methods used. *See id.*

[56] *See id.* at 263 (noting defendants produced 165 asserted privileged/protected documents to the plaintiff).

> search to see if the search results were reliable. Common sense suggests that even a properly designed and executed keyword search may prove to be over-inclusive or under-inclusive, resulting in the identification of documents as privileged which are not, and non-privileged which, in fact, are. The only prudent way to test the reliability of the keyword search is to perform some appropriate sampling of the documents determined to be privileged and those determined not to be in order to arrive at a comfort level that the categories are neither over-inclusive nor under-inclusive. There is no evidence on the record that the Defendants did so in this case.[57]

The opinion emphasizes that lawyers need to use proper technical methodologies to handle eDiscovery including selecting the proper technologies for the task.[58] Magistrate Judge Grimm also summarized (the 2008 understanding of) the known risks arising from reliance on keyword searches to identify relevant or privileged materials.[59] Impliedly, the burden falls to the lawyers and not to the "forensics" experts to properly monitor and employ good technical methodologies.

Victor Stanley, Inc. v. Creative Pipe, Inc.[60] and *O'Keefe* raised eyebrows in the legal profession and, at least in federal courts, technical competence became obvious.[61] Nevertheless, in 2009, Magistrate Judge Andrew Peck issued a wake-up call to the Bar in the Southern District of

---

[57] *Id.* at 256-57 (highlighting growing literature concerning unreliable keyword searches).

[58] *See id.*

[59] *See id.* at 259-60, 260 n.9. In a rather lengthy footnote, Magistrate Judge Grimm attempted to clarify the troubling Rule 702 problem allegedly raised in *O'Keefe* by narrowing the potential applicability of Rule 702 in eDiscovery to cases where a party challenges the effectiveness of disclosure. *See id.* at 261 n.10.

[60] 250 F.R.D. 251 (D. Md. 2008).

[61] A flurry of articles followed. *See, e.g.*, Joshua P. Rosenberg, *A Step Too Far?* Victor Stanley v. Creative Pipe *Decision Is Latest Judicial Alarm Bell In Risk of Spoliation as Relates to Handling of Litigation Holds*, 23 INTELL. PROP. & TECH. L.J. 10, 10 (2011) ("The *Victor Stanley* decision is the latest in a series of decisions in the federal courts to take aim at litigants and their counsel (both in-house attorneys and outside law firms) with respect to FRCP compliance responsibilities for electronic discovery."); Thomas Y. Allman, *Conducting E-Discovery after the Amendments: The Second Wave*, 10 SEDONA CONF. J. 215, 216 (2009) ("[C]ounsel, both inside and retained, must accept responsibility, along with and apart from their clients, for discovery compliance."); Charles Skamser, *The New Generation of eDiscovery Search*, EDISCOVERYTIMES (Feb. 12, 2009), http://ediscoverytimes.com/the-new-generation-of-ediscovery-search/ ("[T]here is a tremendous amount of confusion and trepidation among litigators in regards to potential malpractice claims, sanctions and adherence to Rule 702 and Daubert challenges associated with employing the New Generation of eDiscovery Search technology.").

New York regarding technology and eDiscovery.[62] Judge Peck succinctly summarized:

> This case is just the latest example of lawyers designing keyword searches in the dark, by the seat of the pants, without adequate (indeed, here, apparently without any) discussion with those who wrote the emails.[63]

While the court also emphasized the need for cooperation among counsel in eDiscovery, the opinion concludes with a plain statement about technical proficiency related to keyword search-based eDiscovery:

> [W]here counsel are using keyword searches for retrieval of ESI, they at a minimum must carefully craft the appropriate keywords, with input from the ESI's custodians as to the words and abbreviations they use, and the proposed methodology must be quality control tested to assure accuracy in retrieval and elimination of "false positives." It is time that the Bar—even those lawyers who did not come of age in the computer era—understand this.[64]

In 2012, Federal Judge Shira Scheindlin addressed the technical obligations of eDiscovery in a complex case involving the federal Freedom of Information Act (FOIA).[65] Notably, the suit involved a number of federal agencies and the "largest FOIA search in the history of ICE and an enormous search for [Department of Homeland Security] and the FBI . . . ."[66] Interestingly, the government Defendants conducted the extensive searches using only simple, keyword search methods—a fact that the court notes in detail when finding the searches, at least in part, inadequate.[67] The court goes on to suggest newer technologies such as predictive coding or

---

[62] *See* William A. Gross Constr. Assocs., Inc. v. Am. Mfrs. Mut. Ins. Co., 256 F.R.D. 134, 136 (S.D.N.Y. 2009) (holding attorneys must cooperate and craft appropriate keywords for non-party use in searching for emails).

[63] *See id.* at 135-36.

[64] *Id.* at 136.

[65] *See Nat'l Day Laborer Org. Network v. U.S. Immigration & Customs Enforcement Agency*, 877 F. Supp. 2d 87, 93 (S.D.N.Y. 2012) (finding federal agencies did not sufficiently search files upon request despite obligation to do so).

[66] *See id.* at 111.

[67] *See id.* at 106 (analyzing defendants' search). The court, however, noted the availability of more robust methods. *See id.* at 106-07, 107 n.103, 108-109 (noting keyword searches are usually not effective and verification tests should have been used).

computer assisted review that may produce better results[68]—"defendants must learn to use twenty-first century technologies to effectuate congressional intent."[69] Thus, a significant portion of the opinion addresses the technical aspects of eDiscovery analysis and subtly admonishes some defendants for failing to use "twenty-first century technologies" in 2012.[70]

Finally, Magistrate Judge Peck issued another influential opinion allegedly "approving" predictive coding technologies in eDiscovery.[71] The case involved analyzing approximately three million electronic documents for relevance.[72] The plaintiffs, seeking class action status for alleged sexual discrimination, originally agreed to use predictive coding since keywords searches were problematic.[73] Later the plaintiffs balked at using predictive coding and moved to recuse Magistrate Judge.[74] Nevertheless, the opinion does address technical issues in fair detail. First, the opinion repeatedly acknowledges that predictive coding might be appropriate in certain, but not all cases.[75] Second, the opinion addresses some of the technical requirements of predictive coding such as disclosing the seed set,[76] recognizing the implications of mis-coding a specific document,[77] and

---

[68] *See id.* at 109-11 (suggesting parties should frequently rely on more complex search technologies).

[69] *Id.* at 111.

[70] *See id.* at 109-11 (detailing best practices for searches and obligation of government defendants to take part in analysis).

[71] *See Da Silva Moore v. Publicis Groupe*, 868 F. Supp. 2d 137, 141 (S.D.N.Y. 2012). *Da Silva Moore* relies on an earlier article by Magistrate Judge Peck. *See id.* (identifying Peck's *Search, Forward* article); Peck, *supra* note 41, at 29 (reviewing problems with keyword searches and manual review). Some popular, legal-news outlets claimed the opinion "approved" predictive coding. *See, e.g.*, Philip H. Cohen, *Federal Judge Approves Predictive Coding Technology for e-Discovery*, NAT'L L. REV. (Mar. 13, 2012), http://www.natlawreview.com/article/federal-judge-approves-predictive-coding-technology-e-discovery ("Magistrate Judge Andrew Peck of the Southern District of New York issued the first judicial opinion formally approving the use of 'predictive coding' technology . . . ."); Martha Neil, *Is Judge Peck the First to Require a Predictive Coding Protocol for Automated Doc Review?*, A.B.A. J. (Feb. 14, 2012, 9:24 PM), http://www.abajournal.com/mobile/article/is_federal_magistrate_the_first_to_require_computerized_predictive_coding_p/ (citing Law Technology article stating Peck may be first judge to require predictive coding protocol).

[72] *See Da Silva Moore*, 868 F. Supp. 2d at 140.

[73] *See id.* at 140-42, 145-47. The plaintiffs' eDiscovery consultant even issued a press release regarding using predictive coding with the text included in the opinion. *See id.* at 145.

[74] *See id.* at 147, 159-60 (discussing plaintiffs' attempt to remove Magistrate with favorable view of computer assisted technology).

[75] *See id.* at 141.

[76] *See id.* at 141, 170.

[77] *See id.* 170. The insights here run deeper because the attorney was specifically asked whether her response was due to lack of authority to make a legal decision or lack of understanding regarding the technical and legal implications of the decision. *See id.*

stabilization of algorithms.[78] The opinion notes issues related to technical experts, technical writings, and technical details.[79]

Thus, federal opinions reflect a common theme—lawyers must be aware of the various eDiscovery technologies and be able to apply those technologies in specific cases to address specific problems. Failing to properly apply eDiscovery technologies, as the cases attest, may result in re-doing work, using different technologies to achieve the legal objectives, or embarrassing admonishment of the lawyer's lack of skills in a federal opinion.

2 THE MULTIPLE ORIGINS OF DATA ANALYTICS TOOLS

Many writings about eDiscovery technologies assume that the products all work essentially the same, yet in reality, these tools employ a diverse set of research, computational methods, and algorithmic techniques.[80] That is, while the legal community sometimes attempts to assign a simple, universal label such as "TAR" or "predictive coding" to such tools,[81] the tools actually vary widely in functionality, algorithmic basis, research origins, and scientific scope.[82] Put simply, eDiscovery tools use a dizzying array of unique, although sometimes related, techniques—each with its strengths and weaknesses.

Also, legal practitioners must understand that the basic techniques associated with advanced TAR arise from multiple academic disciplines. The multiple origins explain, as described below, some of the difficulties with developing consistent and appropriate terminology for these techniques from within the legal community. Origins and applications of the underlying technologies arise from machine learning, applied mathematics, statistics, robotics, artificial intelligence, economics, psychology, neuroscience, biology, finance, information retrieval, natural language processing, engineering, and medicine, among others.[83]

---

[78] *See id.* at 145 (noting courts acceptance of proposed protocol system if it could be stabilized).

[79] *See id.* at 140-47.

[80] *See* Sharon D. Nelson & John W. Simek, *Predictive Coding: A Rose by Any Other Name*, 38 ABA L. PRACTICE, Jul.-Aug. 2012, at 20. Sharon Nelson and John Simek noted the problems with assuming that all TAR systems are the same. *See id.*

[81] Currently, the legal community struggles to adopt a simple and common term to generally describe these types of augmented legal analysis methods—leading to my preference, the accurate computer augmented legal analysis. *See id.* (discussing terms used for predictive coding).

[82] *See The Grossman-Cormack Glossary*, *supra* note 4, at 6 (discussing TAR and confusion over terms).

[83] *See, e.g.*, STUART RUSSELL & PETER NORVIG, ARTIFICIAL INTELLIGENCE: A MODERN APPROACH 5-28 (3d ed., Prentice Hall 2010) (providing history of disciplines that contributed ideas, viewpoints, and techniques to artificial intelligence); Zoubin Ghahramani, *Unsupervised*

However, the most significant differences in terminologies for legal practitioners and for these otherwise conceptually similar technologies probably exist between the statistics and machine learning communities.[84] Carl Rasmussen succinctly describes the well-known differences between these two communities as:

> One could say that in statistics the prime focus is often in understanding the *data* and relationships in terms of *models* giving approximate summaries such as linear relationships or interdependencies. In contrast, the goals in machine learning are primarily to make *predictions* as accurately as possible and to understand the behavior of *learning algorithms*.[85]

Each community has its own terminology, predispositions, and emphases, and legal practitioners must be aware of these predispositions. Whereas machine learning may focus on the prediction efficiency and prediction effectiveness of an algorithm (citing, perhaps, overfitting or underfitting), a statistical focus may focus on confidence intervals, sample size, and the appropriate probability curves given the data (citing, perhaps, normal distributions or a confidence interval of 95%±2.7).[86]

Neither community is necessarily better or right. But legal practitioners must avoid being misled by these predispositions—avoid misunderstanding the terminology and avoid assuming that one school somehow provides more "precision" than warranted for the legal community's needs.[87] The legal community must also adopt legally

---

Learning, *in* 3176 ADVANCED LECTURES ON MACHINE LEARNING 72-73 (Olivier Bousquet et al. eds., 2003) (providing overview of field of unsupervised learning from statistical modeling perspective); James O. Berger, *Bayesian Analysis: A Look at Today and Thoughts of Tomorrow*, *in* STATISTICS IN THE 21ST CENTURY: MONOGRAPHS ON STATISTICS AND APPLIED PROBABILITY 276-77 (Adrian E. Rafferty et al. eds., 2002) (providing overview of ongoing activity in Bayesian analysis). Professor Joel Henry also alludes to the need for multiple fields of study to better optimize TAR systems—albeit implying that the new systems will summarily replace older systems. *See* Joel Henry, *Expect an Eclipse: Predictive Coding Is So Yesterday*, LAW TECH. NEWS, Feb. 18, 2014, *available at* https://advance.lexis.com/api/permalink/1738f908-0732-42c5-9ff5-8345b9862fbc/?context=1000516.

[84] *See The Grossman-Cormack Glossary*, *supra* note 4, at 32 (hinting at this issue). Christopher Manning also notes the multiple origins of information retrieval techniques, used in some TAR implementations. *See* CHRISTOPHER MANNING ET AL., INTRODUCTION TO INFORMATION RETRIEVAL xxxi-xxxiii (2009) (outlining history of information retrieval and beliefs about future developments).

[85] CARL E. RASMUSSEN & CHRISTOPHER K. I. WILLIAMS, GAUSSIAN PROCESSES FOR MACHINE LEARNING xiv (2006).

[86] *See generally id.* (summarizing distinction between statistical and machine learning communities).

[87] One of the few cases discussing the details of one type of predictive coding project is *Da*

sufficient terminology for these technologies that focuses on the legal requirements associated with applying these technologies.[88]

2.1 A Note on Definitions

This article largely uses the term "technology assisted review" (TAR)[89] to refer to the computer algorithms that *classify*, or sort, documents into discrete categories. TAR algorithms generally require a legal review of a small subset of documents or materials by an attorney familiar with the project.[90] The attorney, based on legal analysis, classifies the documents into categories and assigns a category to each of the items in the subset.[91] The TAR algorithms then "learn" from the attorney's legal analysis and develop some type of mathematical model that can then predict the classifications of other documents in that dataset.[92] Thus, *classification* distinguishes TAR and serves as the hallmark of TAR systems—as opposed to Boolean search systems that merely retrieve specific information based on search terms known *a priori*.[93]

This definition intentionally differs from an early definition of TAR provided by *The Grossman-Cormack Glossary of Technology-Assisted Review* ("*Glossary*").[94] The *Glossary* definition too narrowly defines TAR systems because the definition artificially limits TAR systems to binary classification—for example, restricting the categories to just relevant or

---

Silva Moore v. Publicis Groupe, 287 F.R.D. 182, 184-88 (2012). The opinion addresses selection of confidence intervals as related to random sampling of the dataset. *See id.* at 186-87. But, the opinion also comments on the "judgmental sampling" which would presumably run contrary to random sampling and thus the citation to confidence intervals. *See id.* Thus, this opinion provides an interesting illustration of some of the various terminologies associated with predictive coding but also the pitfalls of misconstruing the terms. *See id.*

[88] The legally sufficient hallmarks are proportionality, lawyer certifications of legal documents, and lawyer ethics. These issues may differ markedly from the issues that statisticians, mathematicians, or machine learning academics emphasize.

[89] I strongly prefer the more accurate term "computer augmented legal analysis" (CALA). CALA emphasizes the legal-analysis aspects of such systems. However, non-lawyer vendors may resist the fact that TAR, predictive coding, or CALA, no matter which term is used, is law practice. *See* D.C. Bar, *Ethics Op. 363: Non-lawyer Ownership of Discovery Service Vendors* (June 2012), http://www.dcbar.org/bar-resources/legal-ethics/opinions/opinion362.cfm.

[90] *See* Karl Schieneman & Thomas C. Gricks, III, *The Implications of Rule 26(g) on the Use of Technology-Assisted Review*, 7 FED. CTS. L. REV. 241, 259-63 (2013) (discussing issues associated with training technology-assisted review tools).

[91] *See id.* at 259-60.

[92] *See id.* at 260.

[93] *See infra* Section 4.1 for a detailed discussion of Search vs. TAR & Predictive Coding.

[94] *See The Grossman-Cormack Glossary*, *supra* note 4, at 31 (defining TAR as "[a] process for Prioritizing or Coding a Collection of Documents using a computerized system that harnesses human judgments of one or more Subject Matter Expert(s) on a smaller set of Documents and then extrapolates those judgments to the remaining Document Collection.").

not-relevant.[95] However, TAR algorithms can handle much richer multi-class classification tasks, and attorneys may need the richer capacity to classify document sets beyond simple relevant and non-relevant.[96] Uses for relevant-privileged, relevant-non-privileged, medical records, memos, email, or other classifications easily come to mind. Any task that can assist the attorney and staff to classify or sort documents into meaningful categories is a candidate for TAR algorithms.

This latter realization raises a second issue with current TAR and predictive coding definitions. The definitions implicitly assume that TAR or predictive coding applies to production.[97] Yet these algorithms have much broader applications related to sorting and classifying rather than limiting such systems to only production—for example the tools might be used for culling or preliminary case assessment.[98]

Finally, I mention so-called predictive coding[99] in this article simply because the few lawyers already familiar with eDiscovery systems probably know of predictive coding. Outside the legal community and eDiscovery vendors, "predictive coding" means little.[100] The *Glossary* essentially defines "predictive coding" as a subset of TAR and implies that

---

[95] *See id.* The limitation reflects the academic background of one of the authors who works in information retrieval. This is an example of how a specific academic background can influence the working definition of a term used for the legal community. *See* discussion *supra* Section 2 (noting multiple academic disciplines associated with TAR has created inconsistent terminology). The *Grossman-Cormack Glossary* remains a good resource. However, the *Glossary* cannot substitute for actually understanding the technologies.

[96] *See* Schieneman & Gricks, *supra* note 90, at 248-49, 254-57 (suggesting broader use for algorithms beyond data collection).

[97] *See The Grossman-Cormack Glossary*, *supra* note 4, at 26, 32 (defining predictive coding and TAR).

[98] Schieneman and Gricks imply a broader application for such algorithms beyond data collection in discovery by recognizing that these algorithms also serve as culling tools. *See* Schieneman & Gricks, *supra* note 90, at 248-49, 254-57 (distinguishing collection uses from disclosure uses). Such algorithms can certainly be used by the recipients of the disclosed documents (by the requesting party) to internally process the documents. This usage, however, goes beyond mere binary classifications and would entail more complex classifications such as sorting medical reports, from email, from financial reports, from memoranda. Technically, the same algorithms can typically achieve such results—greatly helping teams to use the disclosed documents such as forwarding medical reports to an expert while forwarding emails to the litigation team for analysis. So far, most vendors have not offered such options, which is why they have garnered little notice.

[99] *See The Grossman-Cormack Glossary*, *supra* note 4, at 26 (defining "predictive coding" as "[a]n industry-specific term generally used to describe a Technology-Assisted Review process involving the use of a Machine Learning Algorithm to distinguish Relevant from Non-Relevant Documents, based on Subject Matter Expert(s)' Coding of a Training Set of Documents.").

[100] *See* Nelson & Simek, *supra* note 80, at 20. After much research on the origins of the term "predictive coding," Sharon Nelson and John Simek conclude that the term "predictive coding" was apparently put forward by a vendor, Recommind. *See id.*

predictive coding is the legal community's term for TAR.[101] Thus, attorneys should simply be aware that predictive coding may be a legal-community-term-of-art for some types of TAR and not a distinctive technology in itself.

3 STARTING A PROJECT: DATA PREPROCESSING ISSUES

Before discussing some of the TAR algorithms, a fundamental question must be addressed: how do we get the documents from human-readable form into a form usable by TAR algorithms?[102] While often overlooked, how this occurs and which decisions are made during preprocessing may have significant legal consequences. Put simply, preprocessing decisions largely shape what the TAR algorithms "see," and thus may affect the outcome of a TAR project.

Preprocessing encompasses a wide range of techniques to transform documents from their native format into a form acceptable for processing by the various TAR algorithms.[103] Those techniques may include optical character recognition (OCR), parsing, tokenizing, indexing, stemming, and stop-word-removal.[104]

The discussion here focuses on some of the preprocessing topics that lawyers might frequently encounter. One of the most important topics is feature selection. Feature selection describes the process of identifying and extracting the important "concepts" from documents—one might view this as extracting the individual words from a document but features may be far more complex than simple "words."[105] Feature selection encompasses the process of extracting the features, such as using parsing, and the methods related to processing those extracted features ("words") such as stemming,

---

[101] *See The Grossman-Cormack Glossary*, *supra* note 4, at 26 (defining "predictive coding" as "[a]n industry-specific term generally used to describe a Technology-Assisted Review process involving the use of a Machine Learning Algorithm to distinguish Relevant from Non-Relevant Documents, based on Subject Matter Expert(s)' Coding of a Training Set of Documents.").

[102] However, lawyers should be aware that many of the algorithms used in TAR and predictive coding have wider application in many areas of science. The algorithms can handle financial analysis, genetic analysis, medical diagnosis, and many other subjects. The discussion here focuses on the likely application in legal contexts and assumes that documents are the target of for analysis.

[103] *See* MANNING ET AL., *supra* note 84, at 3-7.

[104] The topics identified here are the major topics that lawyers will likely incur in typical eDiscovery projects. However, lawyers should be aware that "preprocessing" topics remain an area of vigorous academic research and may encompass many other topics. *See id.* at 3-47.

[105] *See The Grossman-Cormack Glossary*, *supra* note 4, at 17 (describing "Features" as "the units of information used by a Machine Learning Algorithm to Classify or Prioritize Documents.").

n-grams, and stop words.[106] Extracted features must be stored for use by the eDiscovery algorithms and storage typically involves data vectors, data matrices, indexing, and data dictionaries.[107]

Finally, a word of caution. Lawyers tend to focus on minute details. However, preprocessing involves fundamental decisions about "what to let go" *because not every feature carries significance for eDiscovery algorithms*.[108] The latter statement may cause attorneys to pause: "how can we just leave stuff out?" But current eDiscovery algorithms may become clogged by "noisy data"—that is, by excessive feature sets with many features conveying little material information. Lawyers do this all the time: if you are sorting through a stack of papers, you quickly spot spreadsheets (with salient features such as numbers, columnar format, and mathematical formulas) as opposed to memos (with salient features such as a title, paragraphs, and headings). That is, the attorney does not need to read every word to quickly differentiate between spreadsheets and memos. Features work very similarly in some eDiscovery algorithms.

Thus, delving into preprocessing provides some deep insights into how the eDiscovery algorithms work—and indirectly, reveals the strengths and weaknesses of the algorithms due to the data as derived during preprocessing. The first topic addresses the need, albeit reduced as more documents originate in electronic format, to transfer paper documents to electronic format.[109]

3.1 Optical Character Recognition

Optical character recognition (OCR) should be familiar to many attorneys.[110] OCR may apply to an eDiscovery project if the documents truly originated in paper form and were subsequently scanned into a

---

[106] *See* DANIEL JURAFSKY & JAMES H. MARTIN, SPEECH AND LANGUAGE PROCESSING 46, 68, 295-96, 640, 768 (2d ed. 2009); CHRISTOPHER M. BISHOP, PATTERN RECOGNITION AND MACHINE LEARNING 2 (2006) (providing features general example); MANNING ET AL., *supra* note 84, at 6-8.

[107] *See, e.g.*, MANNING ET AL., *supra* note 84, at 6-8 (building inverted index); BISHOP, *supra* note 106, at 294 (feature vector kernel method core component).

[108] *See generally* ETHEM ALPAYDIN, INTRODUCTION TO MACHINE LEARNING 110 (2d ed. 2010) (discussing feature or dimensionality reduction techniques without losing significant fidelity).

[109] *See* David Isom, *Electronic Discovery Primer for Judges*, FED. CTS. L. REV. 1, 1 n.1 (2005) (citing Peter Lyman and Hal R. Varian, *How Much Information 2003?*, SCH. OF INFO. MGMT SYS., U.C. BERKELEY (Oct. 27, 2003), http://www.sims.berkeley.edu/research/projects/how-much-info-2003/) (explaining that over 99% of documents originate in electronic format).

[110] *See* JAY E. GRENIG & WILLIAM C. GLEISNER, III, 1 eDISCOVERY & DIGITAL EVIDENCE § 16:7 (2005) (discussing OCR scanned documents).

computer. The scanning typically creates a digital photograph of the original paper document—usually in TIFF format but other image file formats are possible.[111] The OCR software then analyzes the digital images of the documents and extracts the usable text from the documents using the OCR engine.[112] An electronically searchable, text-copy of each original paper document typically results.[113]

Not all matters require OCR. Because documents increasingly originate in electronic format, the cumbersome process of printing the original electronic document, scanning the paper copy, applying OCR, and verifying[114] the results adds needless complexity and labor to an eDiscovery project. Most ESI should be produced in native format when possible to avoid this cumbersome, and trouble-prone, process.[115] Nevertheless, OCR may be needed in some cases to preprocess data for use in eDiscovery systems.

3.2 Parsing and Tokenizing

Assuming the target document set exists in text-based, electronic format, preprocessing typically begins with the concept of parsing and

---

[111] *See id.*

[112] *See* Ahmad Abdulkader & Matthew R. Casey, *Low Cost Correction of OCR Errors Using Learning in a Multi-engine Environment*, 2009 10TH ANN. CONF. ON DOCUMENT ANALYSIS AND RECOGNITION, 576, 576 (2009) (discussing accuracy of OCR technology converting scanned images of documents into readable text).

[113] *See* DANIEL J. FETTERMAN & MARK P. GOODMAN, DEFENDING CORPORATIONS AND INDIVIDUALS IN GOVERNMENT INVESTIGATIONS § 18:15 (2014) (describing production issues related to ESI). In some cases, lawyers may be familiar with the somewhat outdated concept of TIFF load files. TIFF load files associated text extracted via OCR with the TIFF image of the original paper document. *See id.* Thus, in these older systems, the lawyer could view the TIFF image of the original paper document and search the OCR-extracted text of the document. *See id.*

[114] OCR engines commit errors. Good practice requires verifying the error rates by sampling the OCR documents to assess the specific error rate and to correct problems with the OCR recognition. *See* JAY E. GRENIG ET AL., ELECTRONIC DISCOVERY AND RECORDS AND INFORMATION MANAGEMENT GUIDE § 14:24 (2014) (". . . OCR is at best only 70% to 80% accurate . . . ."); JURAFSKY & MARTIN, *supra* note 106, at 72-73 (noting OCR systems have higher error rates than human typists); *see also* Abdulkader & Casey, *supra* note 112, at 576 ("[M]ean word level error rates for OCR ranges roughly between 1 to 10%.").

[115] *See* FED. R. CIV. P. 34(b)(2)(E)(ii). The Federal Rules of Civil Procedure allude to the native format disclosure. *See id.* If the request for production "does not specify a form for producing electronically stored information, a party must produce it in a form or forms in which it is ordinarily maintained or in a reasonably usable form or forms." *Id.* Thus, at least in a federal court, "printing" electronic documents, unless requested by the recipient, violates the F.R.C.P. *See id.* For an example, resulting in sanctions, *see* eBay Inc. v. Kelora Sys., LLC, Nos. C 10–4947 CW (LB), C 11–1398 CW (LB), C 11–1548 CW (LB), 2013 WL 1402736, at *1 (N.D. Cal. 2013) (detailing costs associated with fairly simple case arising from TIFF file conversion); Bray & Gillespie Mgmt. v. Lexington Ins. Co., 259 F.R.D. 568 (M.D. Fla. 2009) (imposing sanctions for failing to produce discovery in form requested by insurer).

tokenizing. Tokenization reads the text in from the document file and parses the documents into individual parts or tokens—for now, think of this as "words" but more complex structures are possible.[116]

Tokenization and parsing may remove punctuation and unneeded white space (such as spaces and tabs); might transform the text into all lower case characters; and might substitute a unique token for specific structures in a document.[117] The substitution, called normalization,[118] might replace dollar values with a common token—for example, replacing $1,234.56 with a token such as <TOK_MONEY>.[119]

Because parsing looks through every document, parsing can be computationally rigorous and thus can take quite some time to accomplish. Typically, in practice, once complete the parsing does not need to be replicated.

The parsed words are then typically placed into some type of index, and the index is then used to efficiently identify documents containing the target words.[120] Depending on the academic background, an index may be referred to as a postings list or data dictionary.

3.3 Data Dictionaries & Indexing

Many TAR systems use data dictionaries or indexes.[121] In essence, a data dictionary or an index operates much like the index in a book.[122] However, instead of associating terms with *pages* within a book, the data dictionary or index associates the words with the *documents*.[123] Thus, in essence, one can look-up a term (word) and immediately determine in

---

[116] *See* MANNING ET AL., *supra* note 84, at 22-26 (providing mechanical overview of parsing). While parsing seems simple, parsing actually represents a complex area of research especially in natural language processing. *See* JURAFSKY & MARTIN, *supra* note 106, at 427-86. Syntactic parsing attempts to parse language documents into constituent parts such as nouns, verbs, and phrases. *See id.*

[117] *See* MANNING ET AL., *supra* note 84, at 22-24 (discussing issues of tokenization).

[118] *See* JURAFSKY & MARTIN, *supra* note 106, at 252-54.

[119] *See id.* While beyond this introduction, normalization can enhance the ability of algorithms to process language by reducing the incidence of non-standard words and ambiguities. Thus, in the example, rather than seeing every dollar amount as unique, the token allows the algorithm to begin to slowly learn about money as a concept, which can sometimes be more effective. *See id.*

[120] JURE LESKOVEC ET AL., MINING OF MASSIVE DATASETS 10-11 (2014) (outlining methods and strategies for efficient data mining); MANNING ET AL., *supra* note 84, at 19, 21-44 (illustrating issues with creating inverted indexes).

[121] *See The Grossman-Cormack Glossary*, *supra* note 4, at 19 (noting automatic indices are used in Information Retrieval systems to identify documents containing particular search terms).

[122] *See id.*

[123] *See id.* (defining term "Index" in TAR context).

which documents include that term.[124] Indexing or data dictionaries provide overwhelming computational efficiency as opposed to naively stepping through each document each time to search for a term with little loss of fidelity.

As mentioned, the index maps word tokens to documents.[125] Look at Table 1 and Table 2 below. Think of an index in a book. The index maps a word to the pages on which that word appears. Conceptually similar, an index in a machine learning context maps words to the *documents* where that word appears.[126]

Note how the two tables relate. Table 1 maps each tokenized word to a database Word ID number.[127] Then, Table 2 maps each Word ID to the Document IDs in which that word appears.[128] Table 1 shows a list of the parsed and tokenized words in the right-hand column. In the left hand column of Table 1, the indexing system assigned each respective word a unique numerical identification number (ID) or *Word ID*. Now assume that there are 100,000 unique documents in the data set and that each document in the document set was assigned a unique *Document ID*—somewhat like a BATES number for each document as familiar to many attorneys. In Table 2,[129] the left-hand column contains the unique *Word ID* from Table 1. The right-hand column of Table 2 contains a list of all the unique *Document IDs* that contain that word.

Now, looking at Table 1, assume that I want to identify all documents that mention the word "photovoltaic"—or Word ID 2. I quickly scan Table 2 looking for the table entry for Word ID 2 and see that Document IDs 11, 788, 2345, and 51222 contain "photovoltaic." Thus, rather than slowly iterating through 100,000 documents to see if any contain "photovoltaic," the index permits an immediate lookup capability. This immediate lookup capacity is the both the magic and the computational benefit of indexing.

---

[124] *See* LESKOVEC ET AL., *supra* note 120, at 10-11.

[125] *See The Grossman-Cormack Glossary*, *supra* note 4, at 19 (defining term "Index").

[126] *See id.*

[127] *See infra* Table 1.

[128] *See infra* Table 2.

[129] Take a moment to look at Table 1 and Table 2, *infra*. What can you infer about the overall topic of the documents? This seemingly innate ability of humans to infer topics simply from related words based on past experience conceptually mirrors what predictive coding systems do. A predictive coding system might see these documents as conceptually related to the topic of energy or, specifically, renewable energy. Thus, other documents with similar words might also be related to that same topic.

Word ID	Word
1	turbine
2	photovoltaic
3	wind
4	energy

Table 1: Word ID to Word Mapping

Word ID	Document IDs
1	30, 566, 2345, 35677, 41000
2	11, 788, 2345, 51222
3	30, 566, 2345, 41000
4	11, 30, 566, 788, 2345, 35677, 41000

Table 2: Word ID to Document ID Mapping

Thus, the indexing provides efficiency. In contrast, a naïve approach would simply store every document in full-text and then a search would sequentially step through every word in every document.[130]

3.4 An Introduction to the Features Concept

So far, I have used the terms "word" or token to describe the parsed and tokenized output of the initial preprocessing.[131] However, the generic term "feature" more precisely describes the output.[132] The more generic, and accurate, term "features" recognizes that many of the algorithms discussed later in this article are also generic algorithms that can handle a wide-array of feature types—not just words.[133] But, what is a feature?

Assume an attorney needs to identify (and assume manual review) all of the medical-test records in a stack of documents. The attorney begins flipping through the stack and starts pulling out medical records. The medical records might be identifiable due to the format of the document, such as a table with numbers or perhaps due to the presence of a graph. Other indicia of medical records might include a laboratory name in the letterhead, an attached X-ray image, a certain color paper, or the presence

---

[130] *See* David C. Blair & M.E. Maron, *An Evaluation of Retrieval Effectiveness for a Full-Text Document-Retrieval System*, 28 COMM. ASS'N OF COMPUTING MACHINERY 289, 289-99 (1985) (discussing indexing concept in litigation context in 1985).

[131] *See supra* Section 3.2 (discussing parsing and tokenizing).

[132] *See The Grossman-Cormack Glossary*, *supra* note 4, at 17 (defining Features as "[t]he units of information used by a Machine Learning Algorithm to Classify or Prioritize Documents.").

[133] *See* discussion *supra* note 102 (noting algorithms can handle financial analysis, genetic analysis, medical diagnosis, and many other subjects).

of physician name. The point here is that certain aspects of a document help give pointers or indicators determining the type of document. Generally, humans can scan a document and recognize letters, words, word-phrases, overall grammatical structure, the presence of images, or a certain document format (such as an invoice).[134]

Similarly, a machine learning algorithm must identify key aspects of documents in a dataset to properly distinguish and classify the documents. Machine learning generally uses the terms "features" to describe the distinguishing elements of documents used by a machine learning algorithm to classify documents[135] and "feature extraction" to describe the process of developing the best set of features for a particular matter (or classification task in machine learning vocabulary).[136] Machine learning has whole fields dedicated to feature extraction.[137]

Understand that the features provide the predictive capability from the learning model, and thus feature selection has important legal consequences.

3.5 Unigrams, Bigrams, n-Grams

With a basic understanding of features, machine learning algorithms do not limit features to simply individual words. Thus, in some instances, algorithms may perform better with aggregated features.[138]

---

[134] Note that such analysis can be far more precise or nuanced such as identifying a writer by looking for certain words or word phrases the writer is known to use. That is, features might simply be more complex syntactic analysis rather than looking at physical features of a document.

[135] *See* ALPAYDIN, *supra* note 108, at 109-110.

[136] *See* Andrew Ng, *CS229 Lecture Notes*, *Introduction to Machine Learning: Lecture 5 Part VII, Regularization and Model Selection*, STANFORD UNIV. 1, 4 (2011), http://cs229.stanford.edu/notes/cs229-notes5.pdf (discussing model selection and feature selection).

[137] *See, e.g.*, Ghahramani, *supra* note 83, at 77-80 (providing factor analysis); ALPAYDIN, *supra* note 108, at 109-39 (giving overview of dimensionality reduction); Bernhard Schölkopf & Alexander J. Smola, LEARNING WITH KERNELS 427-452 (2002) (explaining and analyzing Kernel Feature Extraction).

[138] Note that performance of algorithms and feature sets remain an open research question. Generally speaking, aggregated features may or may not affect the performance of all algorithms or all applications of algorithms. In fact, at least some evidence indicates little or no gain from aggregated feature sets. *See, e.g.*, Ng, *supra* note 136, at 4 (discussing Regularization); Constantinos Boulis & Mari Ostendorf, *Text Classification by Augmenting the Bag-of-Words Representation with Redundancy-Compensating Bigrams*, *in* PROCEEDINGS OF THE WORKSHOP ON FEATURE SELECTION FOR DATA MINING: INTERFACING MACHINE LEARNING AND STATISTICS 9, 9 (April 23, 2005), http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.86.3116&rep=rep1&type=pdf (noting mixed results with complex features, such as bigrams or part-of-speech tags).

An *n-gram* is a generic name for a feature that typically associates two, three, four, or more words together as a single feature.[139] That is, the machine learning algorithm "sees" one feature when that feature may in fact represent one or more words or tokens.[140] In application, a single word is a unigram, word pairs are bigrams, word triples are trigrams, and so on.[141] Generically, these types of aggregated features are conceptually called n-grams (where the variable "n" serves as a placeholder for the number of aggregated words).[142]

Formal Type	Alternate Names	Number Words
unigram	1-gram or bag-of-words[143]	1
bigram	2-gram	2
trigram	3-gram	3
n-gram		*n*

Table 3: Types of Aggregated Features

N-gram generation essentially creates a sliding mask that moves along each individual sentence until encountering a sentence-ending punctuation mark.[144] Consider the following sentence as a simplified example of how n-gram generation works:

> [Courage is] when you know you're licked before you begin but you begin anyway and you see it through no matter what. You rarely win, but sometimes you do.[145]

For example, using a trigram "mask" the trigrams in the example text would include:

---

[139] *See* JURAFSKY & MARTIN, *supra* note 106, at 83-120 (providing general discussion of n-grams).

[140] *See id.*

[141] *See id.*

[142] *See id.* at 86-88; William B. Cavnar & John M. Trenkle, *N-Gram Based Text Categorization*, *in* PROCEEDINGS OF SDAIR-94, 3RD ANNUAL SYMPOSIUM ON DOCUMENT ANALYSIS AND INFORMATION RETRIEVAL (1994), http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.53.9367&rep=rep1&type=pdf (describing n-gram approach to test characterization that is tolerant of textual errors).

[143] Use of unigrams, or 1-grams, is sometimes called a "bag-of-words feature set." *See* Boulis & Ostendorf, *supra* note 138, at 1. The term bag-of-words simply represents that a unigram has no context—as if you dumped each word into a bag and pulled single words out randomly. *See* JURAFSKY & MARTIN, *supra* note 106, at 643. Nevertheless, bag-of-words or unigram feature sets remain a robust modelling technique. *See* Boulis & Ostendorf, *supra* note 138, at 1.

[144] *See* JURAFSKY & MARTIN, *supra* note 106, at 83-120 (providing overview of n-grams).

[145] HARPER LEE, TO KILL A MOCKINGBIRD 128 (Harper Collins 2002) (1960).

Trigram Number	Trigram Contents
1	courage is when
2	is when you
3	when you know
4	you know you're
5	know you're licked
6	your licked before
. . .	. . .
29	sometimes you do

An n-gram generally does not traverse a punctuation symbol. So, in the example above, the sliding mask would end at "no matter what" due to the period and the next trigram would be "You rarely win."

In contrast, a unigram "mask" of the same paragraph results in only 23 features—as the mask omits second instances of duplicate words.

The most commonly used n-grams in practice are unigrams, bigrams, or trigrams.[146] Note, however, that simple unigram implementations typically do well in most instances.[147]

3.6 Stemming

Attorneys already use stemming-like concepts when doing online research. For example, when doing legal search, it is sometimes helpful to use Westlaw's, LexisNexis', or FastCase's "root extender" (!) or "universal character" (*) connectors.[148] For example, to find instances of *contract*, *contracted*, *contracting*, *contractor*, or *contracts*, the root extender requires *contrac!*. The connector acts as shorthand to avoid needing to specify each possible variation of the target term.[149] The connector works, in most cases, because the target words are all somewhat related.[150]

---

[146] *See* JURAFSKY & MARTIN, *supra* note 106, at 772.

[147] *See id.* at 117. Although higher performance might result from applying additional methods or from more complex analysis. *See* Levent Ozgur & Tunga Gungor, *Analysis of Stemming Alternatives and Dependency Pattern Support in Text Classification*, *in* 41 ADVANCES IN COMPUTATIONAL LINGUISTICS, 195, 195-97 (2009) (discussing how different search pattern types enrich solutions).

[148] MATTHEW S. CORNICK, USING COMPUTERS IN THE LAW OFFICE 652 (7th ed. 2014) (explaining how to use root expander and universal character in Westlaw).

[149] *See* MANNING ET AL., *supra* note 84, at 15 (providing root expander example).

[150] *See* JURAFSKY & MARTIN, *supra* note 106, at 772; MANNING ET AL., *supra* note 84, at 15 (providing specific Boolean search example using Westlaw).

In some circumstances, reducing data-noise generated by related words can increase the predictive capacity of the algorithms used in TAR.[151] One common method to reduce the data noise is stemming.[152]

Stemming describes the concise process of reducing inflectionally-related or morphologically-related word forms to a base form of the word.[153] For example, consider the related words *compute*, *computes*, *computed*, *computing*, *computer*, *computation*, *computerize*, or *computational*. In simple terms, a stemming algorithm uses a heuristic pattern to essentially chop-off the end of the word and leave the stem.[154] The stem, *comput* in the example above, results after applying a stemmer to each of the examples. The Porter Stemmer[155] remains a popular stemming algorithm[156] but other variations exist.[157]

---

[151] *See* MANNING ET AL., *supra* note 84, at 32-35 (providing overview of stemming and lemmatization); Scott Deerwester et al., *Indexing by Latent Semantic Analysis*, 41 J. AMER. SOC. INFO. SCIENCE 391, 403 (1990) [hereinafter Deerwester et al., *Indexing by Latent Semantic Analysis*] (observing that stemming improved performance). The apparent reduction in noise occurs because the stemming creates a single root word token. *See The Grossman-Cormack Glossary*, *supra* note 4, at 30-31 (defining "stemming"). Thus, the frequency counts of this token increase, as opposed to separate counts for each variation of the word, and thus raise the root token above the "noise" of less common terms. Basically, the stemming might boost the frequency or perceived incidence of the term. If an important term, this boosting can enhance the TAR algorithms. *See generally* JURAFSKY & MARTIN, *supra* note 106, at 772.

[152] *See* Mohammed A. Wajeed & T. Adilakshmi, *Text Classification Using Machine Learning*, 4 J. THEORETICAL & APPLIED INFO. TECH., 119, 121-23 (2009), *available at* http://www.jatit.org/volumes/research-papers/Vol7No2/4Vol7No2.pdf (describing preprocessing phase).

[153] *See* MANNING ET AL., *supra* note 84, at 32-35 ("Stemming usually refers to a crude heuristic process that chops off the ends of the words in the hope of achieving this goal correctly . . . .").

[154] *See id.* The "chopping" distinguishes stemming from true root word derivation. *Compare* JURAFSKY & MARTIN, *supra* note 106, at 47-56 (describing morphological parsing using word stems in morphological classes), *with* JURAFSKY & MARTIN, *supra* note 106, at 772 (describing stemming as "the process of collapsing together the morphological variants of a word").

[155] *See* Martin F. Porter, *The Porter Stemming Algorithm*, TARTARUS.ORG http://tartarus.org/~martin/PorterStemmer/ (last visited Mar. 26, 2016) (citing C.J. van Rijsbergen, S.E. Robertson & M.F. Porter, *New Models in Probabilistic Information Retrieval*, BRITISH LIBRARY RESEARCH & DEVELOPMENT REPORT NO. 5587 (1980) (original algorithm stemming paper)). Martin Porter later developed a more advanced stemming language called Snowball. *See* Martin F. Porter, *Snowball*, TARTARUS.ORG, http://snowball.tartarus.org/ (last visited Feb. 1, 2016).

[156] MANNING ET AL., *supra* note 84, at 33 (noting Porter Stemmer is most common algorithm for stemming English).

[157] *See, e.g.*, Julie Beth Lovins, *Development of a Stemming Algorithm*, 11 MECHANICHAL TRANSLATION AND COMPUTATIONAL LINGUISTICS 22, 22–31 (1968) (discussing the Lovins Stemmer); MANNING ET AL., *supra* note 84, at 33 ("Other stemmers include the one-pass Lovins stemmer and newer stemmers like the Paice/Husk stemmer."); Ozgur & Gungor, *supra* note 147, at 202-03 (discussing newer research into alternatives for stemmers).

Using the following example text,

> [Courage is] when you know you're licked before you begin but you begin anyway and you see it through no matter what. You rarely win, but sometimes you do.[158]

and applying the Porter Stemmer, the stemmed result is:

> Courag is when you know you re lick befor you begin but you begin anywai and you see it through no matter what You rare win but sometim you do[159]

Stemming may help reduce noise or extraneous features and thus enhance the predictive capacity of the learned model.[160]

3.7 Stop Words

Some words occur with very high frequency in each language. For example, in English, the words *the*, *be*, *to*, *of*, *a*, *and*, *in*, *that*, or *have* occur commonly.[161] Some TAR algorithms look at the frequency of word occurrences in the document set to calculate probabilities or other predictive information. But, because these common words occur so frequently in any document, their appearance in any specific document probably adds little to distinguishing one type of document from another.[162] These words thus might be seen as "data noise"—conveying little real, discriminative information.[163]

Thus, removal of stop words might occur during preprocessing because to a data scientist, the stop words convey little information and thus can be

---

[158] LEE, *supra* note 145, at 128.

[159] Bo Luo, *Porter Stemmer Online*, KUTZTOWN UNIV. INFO. & TELECOMMUNICATIONS TECH. CENTER, http://www.ittc.ku.edu/~bluo/eecs767sp10/stemmer.php (last visited Feb. 1, 2016) (Stemming using the Porter Stemmer Online Tool).

[160] *See* MANNING ET AL., *supra* note 84, at 34-35.

[161] *See generally The OEC: Facts About the Language*, OXFORD DICTIONARIES, http://www.oxforddictionaries.com/words/the-oec-facts-about-the-language (last visited Feb. 1, 2016) (providing statistics regarding base words).

[162] *See* MANNING ET AL., *supra* note 84, at 27. Whether stop words should be removed in a legal context remains an open research question. The natural downside of stop word removal is the inability to search for specific phrases that might use the stop words. *See* JURAFSKY & MARTIN, *supra* note 106, at 772. But whether removing stop words in a classification context negatively affects results of classification remains open to evaluation. *See infra* Section 4.1 Search vs. TAR & Predictive Coding and 4.2 Boolean, "Search" Systems & Information Retrieval (discussing distinction between search and TAR purposes).

[163] *See generally* MANNING ET AL., *supra* note 84, at 117-19.

discarded.[164] But, as many have noted, stop word removal can be especially detrimental to TAR algorithms using indexed search strategies.[165] In search contexts, search for a precise phrase such as "to be or not to be" would presumably yield no results if stop words such as *to* and *be* were removed from the index.[166] However, in practice, other TAR algorithms might not be as drastically affected, and in fact might be improved by stop word removal.

3.8 Data Vectors and Data Matrices

While attorneys relate to words and documents, many machine learning algorithms instead use a numerical, vector representation of a single document or a matrix representation of an entire document corpus (or subset).[167] The representational concepts of vectors and matrices fundamentally underlie the feature extraction (or preprocessing) steps and the deployment of the machine learning algorithms.[168] That is, preprocessing often "translates" documents or words into numerical equivalents to permit the machine learning algorithms to process the documents. Some examples show how this translation takes place.

Data preprocessing typically results in some form of vector representation of a document.[169] A vector, in this context, is a one-dimensional object. For example, the following illustrates an unremarkable vector: [1,1,0,0,1,0,0,0,0,1]. The vector contains ten elements. The elements of this vector would typically be referenced from 0 to 9. Element 0 is a 1, element 1 is a 1, element 2 is a 0, and so on.

Typically, the preprocessing step may reduce each document to a vector representation. The reduction might be called "creating a postings list"[170] or a "data dictionary"—depending on disciplinary background of the analyst. Basically, a postings list or dataset dictionary takes every word identified in every document in the dataset during preprocessing, and maps

---

[164] *See* Wajeed & Adilakshmi, *supra* note 152, at 121-22.

[165] *See The Grossman-Cormack Glossary*, *supra* note 4, at 31 (defining "Stop Word" as "a common word that is eliminated from indexing.").

[166] *See id.* (noting phrase contains exclusively stop words that would not be indexed).

[167] *See* Ozgur & Gungor, *supra* note 147, at 196-97 (discussing stemmer analysis in text preprocessing).

[168] *See id.* at 195-97; *see also* HARALAMBOS MARMANIS & DMITRY BABENKO, ALGORITHMS OF THE INTELLIGENT WEB 34-37 (2009) (providing techniques to improve search results based on hyperlink analysis).

[169] *See, e.g.*, Jurafsky & Martin, *supra* note 106, at 765-70 (illustrating vector space model in information retrieval); PETER HARRINGTON, MACHINE LEARNING IN ACTION 67-69 (2012) (showing example of vector representation); MANNING ET AL., *supra* note 84, at 123-24, 291 (showing documents as vector spaces).

[170] MANNING ET AL., *supra* note 84, at 19 (discussing information retrieval).

that word to a numerical index value.[171] Thus, each document becomes a vector.

If we "stack" a group of vectors on "top" of each other, we get a two-dimensional matrix. This would represent the sequential documents in a document set or corpus. Such a matrix, including the first row as the vector example above, we start to get:

Example of a Matrix by Row and Column[172]	
Row	Columns
Document 1	[1,1,0,0,1,0,0,0,0,1]
Document 2	[0,0,0,0,1,0,0,0,0,0]
Document 3	[1,1,1,1,1,1,0,0,1,1]

At this point, simply understand that the machine learning algorithm probably does not view actual words but a vector or matrix representation of the documents.

3.9 Getting Too Good: Generalization, Over-fitting, and Under-fitting

While technically not part of preprocessing, the concept "generalization" must be understood to shed light on why feature extraction and preprocessing play a significant role in many TAR algorithms. Most TAR algorithms learn, in a computational model sense, from a specified subset of the entire data set.[173] That subset represents a training set,[174] or sometimes called a seed set.[175] The models then predictively apply the

---

[171] *See id.* at 10.

[172] More accurately, the software algorithm probably sees a series of arrays that look something like this:

[[1,1,0,0,1,0,0,0,0,1],
[0,0,0,0,1,0,0,0,0,0],
[1,1,1,1,1,1,0,0,1,1]].

This would be an array of arrays. The first row, and third column would be referenced as [0,2] in array format that usually begins with the 0 element in many computer languages. Many linear algebra libraries include more complex and functional vector and matrix objects. *See, e.g.*, *Topic: numpy.matrix*, SCIPY.ORG (Oct. 18, 2015), http://docs.scipy.org/doc/numpy/reference/generated/numpy.matrix.html (containing popular library for python programming language).

[173] *See* MEHRYAR MOHRI ET AL., FOUNDATIONS OF MACHINE LEARNING 4-5 (2012).

[174] *See* HARRINGTON, *supra* note 169, at 7-10 (providing approachable discussion of machine learning technologies).

[175] *See The Grossman-Cormack Glossary*, *supra* note 4, at 29 (defining "seed set" as "[t]he initial Training Set provided to the learning Algorithm in an Active Learning process). At least one case mentions the concept of a seed-set. *See* Da Silva Moore v. Publicis Groupe, 868 F.

learning derived from the training set to unseen (to the algorithm) documents in the remaining dataset[176]—resulting in the "predictive" in predictive coding. That is, learning about a subset of the documents should permit the best possible predictions in the remaining dataset.

But, a Goldilocks-like Dilemma occurs and is often related to the task of feature selection.[177] Perhaps startling and a little disconcerting for attorneys, a learned-model based on the training set might simply be too good. That is, the model over-fits the training data but does not generalize well to the rest of the dataset.[178] In contrast, a learned-model might exhibit poor predictive performance represented as not being able to adequately handle the complexity of the entire dataset.[179] The poor performance is called under-fitting.[180]

The generalization, under-fitting, and over-fitting dilemma relates, at least in part, to the number and quality of the features selected.[181] Intuitively, more features may seem to be better—especially to an attorney trained to focus on details and minutia.[182] However, practice demonstrates some rather unintuitive aspects of machine learning—aspects that attorneys

---

Supp. 2d 137, 141 (S.D.N.Y. 2012) (referring to seed sets three times in opinion); *see also* Peck, *supra* note 41, at 29 ("[C]omputer-assisted coding involves a senior partner (or team) who review and code a "seed set" of documents. The computer identifies properties of those documents that it uses to code other documents.").

[176] *See* HARRINGTON, *supra* note 169, at 9-10; *see also* Peck, *supra* note 41, at 29 ("[C]omputer-assisted coding involves a senior partner (or team) who review and code a "seed set" of documents. The computer identifies properties of those documents that it uses to code other documents.").

[177] In fact, feature selection represents a whole sub-field in machine learning; for an intensive discussion of the complexity of the issues as related to PC Learning, VC-Dimension, and *Rademacher* Complexity, *see* MOHRI ET AL., *supra* note 173, at 11-28, 33-54; ALPAYDIN, *supra* note 108, at 27-30 (providing more approachable analysis); Ghahramani, *supra* note 83, at 99-100.

[178] *See* MARMANIS & BABENKO, *supra* note 168, at 226-227; Ng, *supra* note 136, at 4 (discussing regularization at depth).

[179] *See* MARMANIS & BABENKO, *supra* note 168, at 227 (describing under-fitting).

[180] *See id.* at 226-27; Ng, *supra* note 136, at 4 (discussing regularization and problem of over-fitting).

[181] *See* MARMANIS & BABENKO, *supra* note 168, at 226-27; Ng, *supra* note 136, at 4.

[182] Also, many attorneys come from a search-oriented background where the attorney looks for documents containing a specific term—such as FastCase, LexisNexis, or Westlaw searches. *See* Douglas W. Oard et al., *Evaluation of Information Retrieval for E-discovery*, 18:4 ARTIFICIAL INTELLIGENCE & L. 347, 354 (2010) (discussing computerized databases of case law such as Lexis and Westlaw). But, that information retrieval orientation, in machine learning terms, does not necessarily apply to a machine learning classification task. The information retrieval algorithm focuses on returning all documents with the specific term. A machine learning algorithm focuses, instead, on developing adequate features to distinguish one type of document from another type of document. *Compare* MANNING ET AL., *supra* note 84, at 1-6 (discussing information retrieval limitations), *with* HARRINGTON, *supra* note 169, at 7-10 (noting benefits of machine learning techniques).

must be aware of due to the direct implications on the results achieved by augmented legal analysis systems.[183] Rather than more-is-better, in machine learning, the task suggests getting the number of features "just right" so that the learned model generalizes adequately to the unseen documents in the project.[184] Proper generalization, getting the features just-right, gives the machine learning algorithm its predictive "accuracy."[185] Thus, for legal practitioners, a careful understanding of what features are, how features are extracted in a machine learning context, why you sometimes need to let features go, and the implications of features on overall predictive performance is essential to understand the legal implications of the feature extraction task.

3.9 Preprocessing Summary

While only a brief introduction to a very complex and research-intensive topic, attorneys should be aware of the preprocessing step and especially aware of the implications of the decisions made during preprocessing. Parsing, tokenizing, stop word removal, and stemming may alter the data—albeit in the scientifically rigorous manner.[186] Feature selection and generalization are extremely important when applying algorithms.[187] Achieving the optimal number and optimal types of features takes analysis and skill.[188] And understanding, at least basically, the role of indexing and data vectors helps an attorney to better understand how eDiscovery technologies work.[189]

4 DISTINGUISHING KEYWORD SEARCH, TECHNOLOGY ASSISTED REVIEW (TAR) & PREDICTIVE CODING SYSTEMS

Preprocessing provides some insight into the complexity related to simply preparing the data for use in TAR. But preprocessing simply creates raw data sets. The TAR systems use a series of algorithms—each

---

[183] *See supra* note 182 and accompanying text (describing differences between information retrieval algorithms and machine learning algorithms).

[184] *See* ALPAYDIN, *supra* note 108, at 24, 76-80. Also see techniques such as principle component analysis (PCA) which provides a scientific method for reducing the number of features (called reducing dimensionality) to those with the most effect on the classification task. *See id.* at 113-25; HARRINGTON, *supra* note 174, at 269-79.

[185] *See* ALPAYDIN, *supra* note 135, at 24, 76-80.

[186] *See supra* Sections 3.2, 3.6, and 3.7 (discussing parsing, tokenizing, stemming and stop words).

[187] *See supra* Section 3.4 (providing overview of feature selection).

[188] *See id.*

[189] *See supra* Section 3.4 (discussing data vectors and matrices).

with strengths and weaknesses—to predictively analyze the preprocessed data to generate results.[190]

But before turning to the algorithms, a critical distinction must be made between Boolean (or keyword) search systems and TAR (and predictive coding) systems.[191]

---

[190] *See The Grossman-Cormack Glossary*, *supra* note 4, at 32 (defining "TAR").

[191] *See infra* Sections 4.1-4.2 (discussing Boolean, search systems, TAR, and predictive coding); *The Grossman-Cormack Glossary*, *supra* note 4, at 10, 32 (defining "Boolean" and "TAR"). The *Grossman-Cormack Glossary* is the best resource so far for defining basic terms associated with eDiscovery technologies. *See The Grossman-Cormack Glossary*, *supra* note 4, at 4. The *Glossary* makes a subtle distinction between TAR and predictive coding and asserts that predictive coding uses machine learning algorithms while the more generic TAR supposedly can include non-machine learning algorithms. *Compare id.* at 32 (defining "TAR" as "[a] process for Prioritizing or Coding a Collection of Documents using a computerized system that harnesses human judgments of one or more Subject Matter Expert(s) on a smaller set of Documents and then extrapolates those judgments to the remaining Document Collection."), *with id.* at 26 (defining "Predictive Coding" as "[a]n industry-specific term generally used to describe a Technology-Assisted Review process involving the use of a Machine Learning Algorithm to distinguish Relevant from Non-Relevant Documents, based on Subject Matter Expert(s)' Coding of a Training Set of Documents."). The distinction means little in practice and perpetuates the use of brand-name-like terms such as "predictive coding." The definitions also downplay the fact that classifications other than simply relevant or non-relevant may be needed in eDiscovery projects—for example, relevant-privileged or relevant-non-privileged are commonly needed. *See* Victor Stanley, Inc. v. Creative Pipe, Inc., 250 F.R.D. 251, 253-54 (D. Md. 2008) (holding privilege may be waived by voluntary, but unintentional, disclosure of privileged materials during eDiscovery). As mentioned in *Section 2: The Multiple Origins of Data Analytics Tools*, the disciplinary mindset of the experts—here one of the authors coming from an information retrieval background which emphasizes binary, relevant and non-relevant distinctions—are reflected in the *Grossman-Cormack Glossary*. *See* discussion *supra* Section 2.1 and note 95 (commenting on academic background of author). While this is not necessarily wrong, the language, especially a glossary, shapes the discourse so these distinctions become important when applying concepts to the legal community. *See id.* (noting how academic background can conflict with needs of working definition for legal community).

4.1 Search vs. TAR & Predictive Coding

Contrary to assumptions in the legal community, TAR and predictive coding are *not* general search, or document retrieval, tools. Instead, predictive coding describes *classification* algorithms used to place documents into related categories.[192] In contrast, search systems provide "search" capabilities which return specific documents related to a search query.[193] The distinction is fundamental and underlies much of the confusion related to the application of these distinctive tools. In a nutshell, TAR and predictive coding tools are best for sorting or classifying items into general categories such as relevant, privileged, medical records, and emails.[194] Search tools are best for retrieving *specific* documents based on *a priori* knowledge of the search terms necessary to retrieve those specific documents.[195]

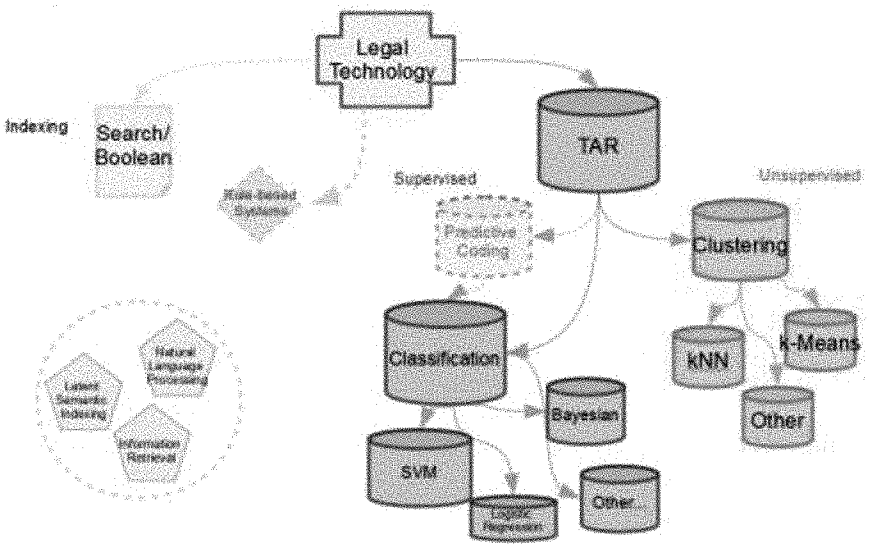

Figure 1: Schematic of eDiscovery-related Legal Technologies & Algorithms

---

[192] *See* SEAN OWEN ET AL., MAHOUT IN ACTION 230-31 (2012).

[193] *See* Blair & Maron, *supra* note 130, at 289-90 (describing 1985 test involving corporate litigation documents). While predictive coding remains fairly new in the legal community, document retrieval in legal contexts stretches back for decades, as this article attests. *See id.*

[194] *See* Schieneman & Gricks, *supra* note 90, at 248-49, 254-57 (recognizing algorithms serve as both culling and as data collection tools in discovery).

[195] *See* Blair & Maron, *supra* note 130, at 289, 295.

Thus, s*earching* differs markedly from *classifying* items—such as classifying items as relevant or irrelevant.[196] But, most lawyers fundamentally misunderstand the critical difference between search and classification.[197] Again, a classification task involves sorting or organizing as opposed to a search which zeroes in on specific documents containing known terms. At its core, a classification system predictively assigns a document to a categorical classification (or assigns a probability that a document belongs to a categorical classification).[198] Classification represents a fuzzier task—a task where overall document context and semantics play a core role and not just a limited set of specific search terms or keywords.[199] Thus, attorneys venturing into TAR, predictive coding, or search must consider the best tool for the task and avoid using the wrong tool for the wrong task just because the attorney, or a vendor, is familiar with a specific tool.

A basic understanding of how some of the primary algorithms work provides insight into the differences between the tools and how to use the tools. The following discusses the primary distinctions between keyword (or Boolean) search and TAR, including predictive coding.[200] The discussion also distinguishes algorithms and systems from more general natural language processing and latent semantic indexing techniques or methods that apply to a diverse array of systems and algorithms.[201]

---

[196] *See supra* Section 4.1 and *infra* Section 4.2 (discussing Boolean, search systems, TAR, and predictive coding).

[197] *See, e.g.*, Tingen, *supra* note 4, at 15 (conflating keyword search and TAR as "search" tools); Jason R. Baron, *Law in the Age of Exabytes: Some Further Thoughts on 'Information Inflation' and Current Issues in e-Discovery*, 17 Rich. J.L. & Tech 9, 14-15, 33-35 (2011) (conflating keyword search and TAR as "search" tools); Maura R. Grossman & Terry Sweeney, *What Lawyers Need to Know About Search Tools*, NAT'L L.J. (Aug. 24, 2014), *available at* http://www.ned.ucourts.gov/internetDocs/cle/2011-01/National%20Law%20Journal%20(Aug%202010).pdf (referring to Bayesian and TAR-type systems as "search" tools). This confusion is also the source of the so-called Go Fish! Problem with using keyword search to try to identify a *class* of relevant documents within a document set. *See* Steven Bennett, e-*Discovery's Balancing Act*, LAW TECH. NEWS, Jul. 18, 2014, *available at* https://advance.lexis.com/api/permalink/7bd4adf3-f03c-4a17-9193-bf4ed15f0ad8/?context=1000516 (noting "go fish" approach to discovery requests is ill-suited to creation of efficient discovery process); Peck, *supra* note 41, at 29; *infra* Section 4.2 (discussing Go Fish! Dilemma in more detail).

[198] *See* OWEN ET AL., *supra* note 192, at 231.

[199] Think of this as similar to the difference between doing legal research where one knows specifically the legal term at issue, say adverse possession, versus finding cases that might represent real property boundary issues. The former is a candidate for search while the latter might require more subtle methods related to classification or natural language processing to understand context.

[200] *See infra* Sections 4.2-4.3.

[201] *See infra* Section 4.4.

4.2 Boolean, "Search" Systems & Information Retrieval

Most attorneys have some familiarity with the concept of "searching." Online, legal research tools such as FastCase, LexisNexis, or WestLaw and common Internet search tools such as DuckDuckGo, Yahoo!, and Google all use search technologies.[202] "Search" tools typically return a list of results that match the searcher's specified "keywords" and Boolean connectors (such as AND, OR, and NOT).[203] Search, while powerful, is a rather simple tool.[204]

Search systems consist of:

1. a predicate creation of an index by analyzing (or indexing) the terms in the documents;
2. the construction of a search phrase typically using Boolean connectors (but "natural language" searches become increasingly common);
3. parsing of the search phrase by the search algorithm;
4. matching the parsed, search phrase against the index; and
5. returning any search matches—which may or may not be relevant.[205]

Note the language—searching, matching, indexing, parsing, and relevance. All of these terms are hallmarks of search technologies.

Data scientists sometimes use the more precise term "information retrieval" to describe the technical aspects of "searching" or finding information relevant to the search phrase.[206] Fundamentally, search systems retrieve specific, previously-indexed, documents based on the specified Boolean criteria and the specified search terms.[207] While seemingly obvious, the objective of search is to return "matching" documents and assumes that the words alone in the document, and not

---

[202] *See* Oard et al., *supra* note 182, at 354 (discussing computerized databases of case law such as Lexis and Westlaw); *Boolean Searching*, LEXISNEXIS WIKI, http://wiki.lexisnexis.com/academic/index.php?title=Boolean_Searching (last visited April 22, 2016).

[203] *See* JURAFSKY & MARTIN, *supra* note 106, at 767.

[204] *See* MANNING ET AL., *supra* note 84, at 1-6 (discussing Boolean retrieval).

[205] *See id.* at 1-6, 29-30 (providing overview of searching and tokenization); MARMANIS & BABENKO, *supra* note 168, at 30-31 (giving brief searching discussion).

[206] *See* MANNING ET AL., *supra* note 84, at 2 ("Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)."); *The Grossman-Cormack Glossary*, *supra* note 4, at 19 (describing "Information Retrieval" as "[t]he science of how to find information to meet an Information Need.").

[207] *See* JURAFSKY & MARTIN, *supra* note 106, at 767.

necessarily the contextual meaning, denote relevance and thus matches.[208] Implicitly, search technologies assume that the attorney has some specific *a priori* knowledge of the relevant terms to craft the search phrase that returns documents matching those terms.[209]

"Search" tools use Boolean operators such as AND, OR, and NOT to aggregate individual word terms (keywords) into more complex search criteria.[210] The aggregate result is a search phrase.[211] The Boolean operators control how the search algorithm determines whether a result "matches" the desired search criteria and the result set can be visually represented by Venn diagrams.[212] For example, the Boolean search "eDiscovery AND predictive AND coding" retrieves only documents in an indexed corpus that contain all three terms. In contrast, the Boolean search "eDiscovery OR predictive OR coding" retrieves all documents that contain at least one of the words—probably a much larger search result than the previous example. Attorneys craft the search phrases to include or exclude terms using the Boolean connectors.

While still relevant and important when used properly and within their domain, search systems suffer from well-known limitations and problems—as attorneys continue to find when trying to use exclusively search systems to meet eDiscovery obligations.[213] The challenge arises when lawyers try to use search, sometimes with the encouragement of

---

[208] *See id.*

[209] *See supra* Section 4.1 (comparing search with TAR and predictive coding).

[210] *See* MANNING ET AL., *supra* note 84, at 15 ("Boolean queries are precise: a document either matches the query or it does not."); JURAFSKY & MARTIN, *supra* note 106, at 767 (explaining search tools); *The Grossman-Cormack Glossary*, *supra* note 4, at 10 (defining "Boolean Search" as "[a] Keyword Search in which the Keywords are combined using operators such as "AND," "OR," and "[BUT] NOT." The result of a Boolean Search is precisely determined by the words contained in the Documents.").

[211] *See* Grossman & Sweeney, *supra* note 197; *The Grossman-Cormack Glossary*, *supra* note 4, at 10 (defining Boolean Search).

[212] *See* Wei Zhou, Neil R. Smalheiser & Clement Yu, *A Tutorial on Information Retrieval: Basic Terms and Concepts*, 1 J. BIOMED DISCOVERY COLLABORATION 2 (2006), http://www.j-biomed-discovery.com/content/1/1/2 (discussing methodology of how search is conducted by query).

[213] *See* Oard et al., *supra* note 182, at 353 ("As [information retrieval] researchers have long known, and recent legal scholarship has recognized, text retrieval systems suffer from a variety of limitations. . . ."); Grossman & Sweeney, *supra* note 197 (providing more general overview); *see also* BLAIR & MARON, *supra* note 130, at 289-91 (addressing limitations of early, full-text, document retrieval systems). Decades of research have attempted to improve keyword search. *See* Scott Deerwester et al., *Improving Information Retrieval with Latent Semantic Indexing*, 25 PROCEEDINGS OF THE 51ST ANN. MEETING OF THE AM. SOC'Y FOR INFO. SCI. 36 (1988) [hereinafter Deerwester et al., *Improving Information Retrieval*] (providing early article on this topic of improving keyword search by latent semantic indexing). One method to improve keyword search looks at the latent relationships between the words in a document to derive a topical meaning from the document—as opposed to matching keywords by rote. *See id.*

industry vendors,[214] when *classification* tasks are instead necessary.[215] Magistrate Judge Andrew Peck insightfully quipped in *Da Silva Moore v. Publicis Groupe*[216] that information retrieval methods, or so-called keyword search, result in an eDiscovery equivalent of the children's game of Go Fish![217]

Attorneys must understand that keyword search assumes *a priori* knowledge of the document corpus in order to know what to search for or what index terms will yield relevant results.[218] Professor Daniel Jurafsky noted the same general problem with search because search systems necessarily rely on the premise that all meaning exists "solely in the set of words [the document] contains."[219] The *a priori* knowledge prerequisite results in the Go Fish! Dilemma because the attorney cannot always know the precise terms or contextual references to craft a comprehensive search using simply terms.[220]

Nevertheless, search and Boolean systems still play a core role, albeit somewhat limited, in eDiscovery contexts when search is correctly used to return term-specific results such as so-called "smoking gun" documents.[221] But trying to use search tools to classify documents into

---

[214] *See* Tim Leehealey, *The Machine Learning/Predictive Coding Silver Bullet*, EDISCOVERY INSIGHT BLOG (Sept. 24, 2012), http://ediscoveryinsight.com/2012/09/the-machine-learningpredictive-coding-silver-bullet. In a blog post, Tim Leehealey insightfully addresses the over-promising by vendors. *See id.* Leehealy notes, "predictive coding is a powerful tool, but users need to understand it before it can be of any real value and it seems clear that the majority of e-discovery vendors are intent on hiding behind confusion rather than shedding light on both the strengths and weaknesses of the technology." *Id.*

[215] *See* Grossman & Sweeney, *supra* note 197. Grossman and Sweeney hint at this issue in a 2010 general article on eDiscovery. *See id.* But, even this frequently cited article conflates the important distinctions between classification tasks and search tasks.

[216] 287 F.R.D. 2d 137, 182 (S.D.N.Y. 2012).

[217] *Id.* at 191 n.13 (citing to work by Ralph C. Losey); *see also* Lisa Holton, *A Front-Row Seat*, LAW TECH. NEWS, Aug. 5, 2013, at 48, *available at* https://advance.lexis.com/api/permalink/7c6ec2b9-e2ca-4c41-8818-451c8dc85ac3/?context=1000516 (naming Magistrate Judge Peck as an eDiscovery pioneer).

[218] *See* Blair & Maron, *supra* note 130, at 289, 295. As early as 1985, David C. Blair and M.E. Maron recognized this problem with document (information) retrieval systems. *See id.*

[219] JURAFSKY & MARTIN, *supra* note 106, at 767.

[220] *See* Blair & Maron, *supra* note 130, at 289. Again, this long-known problem was cited by David C. Blair and M.E. Maron where they remarked on attempts by lawyers to find all documents about "accidents" but failed to recognize that the rote use of the term "accident" missed documents that referred to accidents as "'event[s]', 'incident[s]', 'situation[s]', 'problem[s]', . . . 'difficult[ies]' . . . 'unfortunate situation[s]', [or] 'what happened last week'." *Id.* at 294-95; *see also* William A. Gross Constr. Assocs., Inc. v. Am. Mfrs. Mut. Ins. Co., 256 F.R.D. 134, 135-36 (S.D.N.Y. 2009) (mentioning keyword searches failing to address document context).

[221] *See* Grossman & Sweeney, *supra* note 197. *But see* MANNING ET AL., *supra* note 84, at 15 (cautioning about Boolean search precision, "Boolean queries are precise: a document either matches the query or it does not.").

categories, such as relevant or not relevant, results in the Go Fish! Dilemma. Furthermore, although some have predicted the end of search tools,[222] the reality remains that two different tasks arise—classification (sorting) and specific document retrieval (search)—and search will continue to play a role in legal analysis.

Finally, search systems share some characteristics with more advanced TAR systems, but search uses distinct terminology for key characteristics. For example, search systems typically rely on an index of terms in the document corpus.[223] An index cross-references terms, or words, with the corresponding documents where those terms appear.[224] The Boolean search phrase then matches against the index, but not against the actual documents, to generate results. Thus, understanding indexing provides some insights into search.

4.3 Predictive Coding: Machine Learning, Probability & Natural Language Processing Systems

The legal community fixated on the term "predictive coding" to describe systems which somehow go beyond keyword search. Many also use "predictive coding" as a synonym for the more general category of technology assisted review (TAR) or computer assisted review (CAR) technologies.[225] Others also incorrectly assume that predictive coding describes a single type of technology—as if predictive coding was

---

[222] Magistrate Judge Andrew Peck was quoted, "courts will likely say that TAR must be used instead of keywords." Sean Doherty, *Myth-Busting Predictive Coding: Are Keywords Dead?*, LAW TECH. NEWS, Dec. 2014, at 34-35, *available at* https://advance.lexis.com/search?crid=afac0d39-73f4-4d95-855e-321d5a516b9e&pdsearchterms=LNSDUID-ALM-LAWTNW-1202674375517&pdmfid=1000516&pdisurlapi=true. But contextually, the discussion simply seems to latently realize that search is simply the wrong tool for the task. *See also* Baron, *supra* note 197, at 36 (providing earlier discussion of this topic).

[223] Technically, information retrieval systems continue to converge with natural language processing and machine learning techniques to reduce the mechanical reliance on just indexes. *See* Lecture, Daniel Jurafsky & Christopher Manning, *Natural Language Processing: Week 7-Ranked Information Retrieval, Week 8-Question Answering*, STANFORD U. (Apr. 2012), https://class.coursera.org/nlp/lecture.

[224] *See The Grossman-Cormack Glossary*, *supra* note 4, at 19 (defining "Index").

[225] *See* Christine Hutcheson, *Much Ado About (predictive coding) Definitions*, RECOMMIND: THE CORE PERSPECTIVE BLOG (Mar. 28, 2012), http://www.recommind.com/blog/much-ado-about-predictive-coding-definitions. A better term remains computer augmented legal analysis (CALA) or technology augmented legal analysis (TALA). As attorneys begin to directly understand the technical aspects of predictive coding, attorneys quickly realize that TAR and CAR are quintessentially modern law practice and an aspect of legal analysis—not "merely" technical issues ripe for non-lawyer direction.

somehow a single concept.[226] And still others incorrectly, and perilously, conflate search technologies with predictive coding technologies because they do not understand the essential, technical distinction between *search* systems and *classification* systems.[227] Attorneys must understand that search and classification technologies fulfill very different roles in the eDiscovery process even though attorneys imprecisely use the general term, search, to denote both objectives.

Predictive coding encompasses several technologies deriving primarily from machine learning, neural networks, artificial intelligence, and statistical probability model research. Generally speaking, such systems analyze a subset of data, analyze the classifications of items in the subset by an attorney expert, develop a mathematical model representing the classification schema, and then predictively apply the model to other data to predict the classification of other data.[228] The essence of such systems is the predictive capacity—and thus the "predictive" in predictive coding. How these general systems actually develop the models and the logistics of each type of system vary by implementation, the algorithms used, and vendor biases and objectives. Also, research shows that claims that one system is consistently superior to another seem premature as application in the legal field varies widely—a system that seems superior for a high-volume project of an Enron scale might perform poorly, or not at all, with common, day-to-day cases typically facing inside counsel.[229]

---

[226] *Compare* Doherty, *supra* note 222, at 34-35 (conflating predictive analytics algorithms), *with* Charles Skamser, *The New Generation of eDiscovery Search*, EDISCOVERY TIMES (Feb. 12, 2009), http://ediscoverytimes.com/the-new-generation-of-ediscovery-search/ (acknowledging three "concept search" methods); *see also* sources cited *supra* note 197 (providing additional examples).

[227] *See* Peck, *supra* note 41, at 29 (providing approachable discussion of distinction between search and classification systems).

[228] *See, e.g.*, Da Silva Moore v. Publicis Groupe, 287 F.R.D. 287 F.R.D. 182, 184-87 (2012) (explaining use of predictive coding); Lynne Bernabei et al., *Electronic Discovery Problems in Employment Litigation*, *in* AM. LAW INST. CLE, ADVANCED EMPLOYMENT LAW AND LITIGATION 1182-84 (2014) (providing in-depth analysis of *Da Silva Moore*); Peck, *supra* note 41, at 29.

[229] *See* Cormack & Grossman, *Evaluation of Machine-Learning*, *supra* note 4, at 153-61 (indicating that even TAR requires further research into active learning models); Grossman & Cormack, *Technology-Assisted Review in E-Discovery*, *supra* note 4, at 15-16 (indicating that TAR may be at least as effective as manual review and probably better).

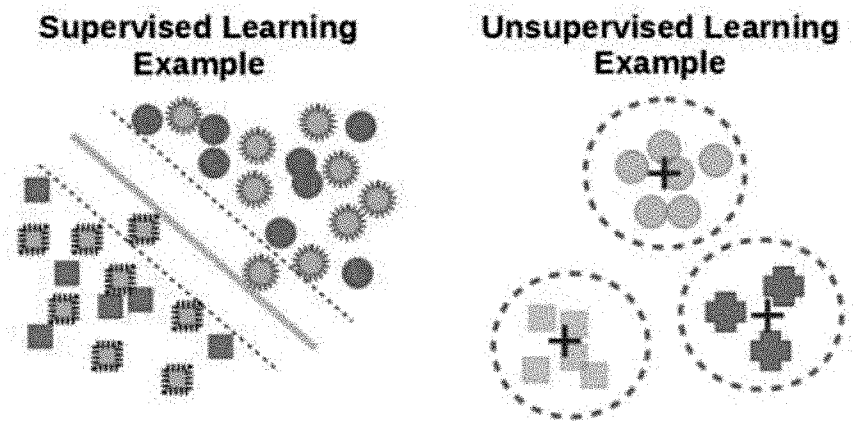

Figure 2: Supervised vs. Unsupervised Learning

TAR algorithms generally facilitate multi-class classifications.[230] However, many current TAR vendors artificially limit classifications to binary situations—such as relevant or not-relevant.[231] Unfortunately, this artificial binary limitation is being tacitly condoned by the few articles in the legal community available about TAR.[232]

Before moving on to exploring TAR algorithms, a summary of some core concepts related to TAR is warranted. First, some attorneys still fearfully view these new systems as black boxes (or perhaps even as black magic). That is why gaining a basic understanding of the algorithms is so important because it strips the fear of the unknown. These systems are in ready use in many other industries and have decades of research behind them—although somewhat new to the legal profession.[233] Second, TAR systems do not replace every other type of tool in the attorney's eDiscovery toolbox—they are not a magic bullet.[234] Instead, TAR systems and

---

[230] *See* ALPAYDIN, *supra* note 108, at 327-28.

[231] *See* discussion *supra* Section 2.1 and note 95 (noting *Glossary* definition artificially limits TAR systems to binary classification).

[232] The *Grossman-Cormack Glossary* definitions, although very helpful otherwise, make no allowance for multi-class classification systems. *See The Grossman-Cormack Glossary*, *supra* note 4, at 19, 26, 32 (defining "Information Need," "Predictive Coding," and "Technology Assisted Review"). The latest research on TAR systems likewise assumes binary, relevant and non-relevant, classifications. *See* Cormack & Grossman, *Evaluation of Machine-Learning*, *supra* note 4, at 153-61.

[233] *See* discussion *supra* note 102 (noting predictive coding has applications in many areas of science).

[234] *But see* William P. Butterfield, Conor R. Crowley & Jeannine Kenney, *Reality Bites: Why TAR's Promises Have Yet to be Fulfilled* 1, 4-8 (2013), http://www.umiacs.umd.edu/~oard/desi5/additional/Butterfield.pdf (providing contrarian view of new TAR systems). The article argues that despite the technical efficacy of the new TAR tools,

algorithms make the common task of classifying or sorting documents into general categories vastly more efficient in appropriate cases—efficient in both time and cost.[235] But other tools, including the maligned keyword search, remain important for some tasks.[236] Attorneys need to think in terms of toolboxes when approaching eDiscovery rather than grasping for magic, all-in-one solutions—much as you use a screwdriver to drive a screw and a hammer to hammer nails. Selecting the right tool for the job becomes a lawyer duty and skill.[237] Third, courts may slowly accept, and even advocate for, the legitimacy of the TAR systems in appropriate cases.[238] In *Da Silva Moore*, Magistrate Judge Peck permitted the first, open use predictive coding in a federal case—but correctly recognized that predictive coding might not apply in every case.[239] Discussion in the legal community reflects the increasing acceptance of these, to the legal community, seemingly exotic new systems.[240] With this context, we can dive into discussing TAR and predictive coding.

4.3.1 Supervised vs. Unsupervised Learning

A discussion of predictive coding must first make a distinction between supervised learning systems and unsupervised learning systems. Most predictive coding systems incorporate some form of supervised learning where an attorney-expert makes legally material decisions when reviewing materials and the system learns from the attorney's analysis.[241] In contrast, and generally not applicable to the legal field at this point, unsupervised

---

traditional litigation strategies (obstructionism and stalling) unique to the legal community may preclude widespread use of TAR—not every party wants an efficient means to analyze data. *See id.* Thus, the writers do not necessarily question the technical aspects of TAR but question the use of TAR in context of legal strategy. *See id.* This is an example of why TAR cannot be driven by non-lawyer experts or delegated to non-lawyer experts who often do not understand the strategic legal issues.

[235] *See* LESKOVEC ET AL., *supra* note 120, at 10-11 (outlining methods and strategies for efficient data mining).

[236] *See* Blair & Maron, *supra* note 130, at 289, 295 (discussing search tools).

[237] *See* discussion *supra* Section 1 (explaining legal practitioner's duty to know technology).

[238] *See* discussion *supra* Section 1.2 and sources cited *supra* note 43 (noting federal cases where court sent "wake up" call).

[239] *See* Da Silva Moore v. Publicis Groupe, 868 F. Supp. 2d 137, *passim* (S.D.N.Y. 2012) (discussing predictive coding as significant issue in opinion and addressing whether predictive coding was appropriate).

[240] *See* Holton, *supra* note 217, at 48 ("It's time for the slow-to-adapt legal infrastructure to sign on [to predictive coding and newer technologies rather than relying on keyword search alone.]").

[241] *See* OWEN ET AL., *supra* note 192, at 229-30, 238-39 (concisely describing differences between classification, recommenders, and clustering and illustrating the supervised-unsupervised distinction); ALPAYDIN, *supra* note 108, at 4-14 (providing more rigorous academic approach).

systems attempt to identify patterns in data without, or with little, human expert intervention.[242] Most current eDiscovery predictive analytics software involves *supervised learning*, but some use probabilistic models consistent with unsupervised or semi-unsupervised learning.[243]

The following chart illustrates some of the differences between supervised and unsupervised machine learning.

	Supervised Learning	**Unsupervised Learning**
Typical general purpose	classification, categorization	detecting latent structure or density patterns in the data, aka "clustering"
Source of teaching/learning	human-expert supervisor	in theory, the data (latent structure of the data as determined by algorithm)
Human-expert input required for learning	necessary, high level, closely analyze seed set to create classifications	optional, minimal level, define target clusters
Generally requires a "seed set"	yes	not necessarily (although feedback regarding cluster centroids can be helpful)
Representative algorithms	support vector machines (SVM) logistic regression naïve Bayes neural networks decision trees	K-means/fuzzy K-means neural networks Hidden Markov Models (HMM)[244]

Fundamentally, supervised learning algorithms use human-reviewed and labeled input to learn general rules or patterns to develop a learning model.[245] Those general rules or patterns (or model) are then used to

---

[242] *See* Ghahramani, *supra* note 83, at 72-75.

[243] *See id.* at 74-77; *see also* CLUSTIFY, http://www.cluster-text.com (last visited Mar. 26, 2016) (clustering-based TAR software tool by Bill Dimm, PhD).

[244] Ghahramani, *supra* note 83, at 82.

[245] *See* OWEN ET AL., *supra* note 192, at 238-39.

predict an outcome on previously unseen-to-the-computer examples.[246] The human-reviewed and labeled input acts as a "seed set" for the supervised learning algorithm.[247] Thus, supervised learning algorithms augment a person's detailed analysis and require significant expert (attorney) input.

In contrast, unsupervised learning tries to sense patterns or relationships in data without any (or much) human input.[248] Unsupervised learning might be what people first think of when they think about machine learning—the computer somehow mysteriously knows how to segment documents into similar "clusters."[249] However, unsupervised learning is not quite that sophisticated (yet). Clustering, a form of unsupervised learning, might provide benefits for some parts of document analysis but does not have adequate or proven capacity to completely replace attorney-experts at this time. Nevertheless, attorneys might encounter clustering in some predictive analytics applications.

Supervised and unsupervised machine learning can be combined into hybrid models. In hybrid models, an unsupervised clustering algorithm might first attempt to place documents into general clusters; a human analyzes the documents in the general clusters and applies classification labels to each item; and the labeled a classified items are then analyzed using a supervised machine learning algorithm.

4.3.2 Logistic Regression

Logistic regression represents a basic, predictive coding type method for classifying documents.[250] The basic concepts in logistic regression may help lawyers understand more complex algorithms and thus serves as a good beginning for lawyers exploring predictive coding.

---

[246] *See id.* at 234 (explaining how classification algorithms learn).

[247] *See id.* at 239-54; HARRINGTON, *supra* note 169, at 101-106 (providing approachable, lay-person treatment).

[248] *See* ALPAYDIN, *supra* note 108, at 11-13; Ghahramani, *supra* note 83, at 72-76.

[249] *See* Ghahramani, *supra* note 83, at 72, 77; HARRINGTON, *supra* note 169, at 217-22.

[250] *See The Grossman-Cormack Glossary*, *supra* note 4, at 22 (defining "Logistical Regression" as "[a] state-of-the-art Supervised Learning Algorithm that estimates the Probability that a Document is Relevant, based on the Features it contains.").

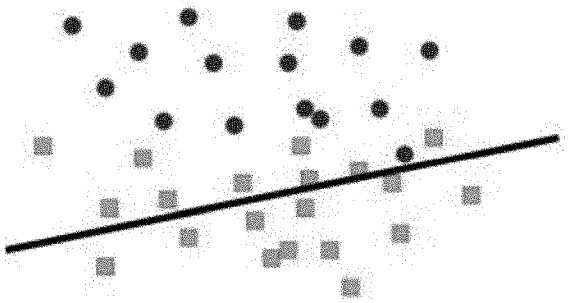

Figure 3: Logistic Regression with Poor Fit

In simplest form, logistic regression attempts to draw a "best-fit line" separating distinct classes of "points."[251] The algorithm defines a linear equation to describe the "best fit line."[252] Once the algorithm establishes the "best fit line," then the same equation can be used to predict the classification of new documents (where documents, via preprocessing, are "points").[253]

---

[251] *See* HARRINGTON, *supra* note 169, at 84; ALPAYDIN, *supra* note 108, at 218-28 (offering more rigorous treatment).

[252] *See* HARRINGTON, *supra* note 174, at 82-83.

[253] *See id.*

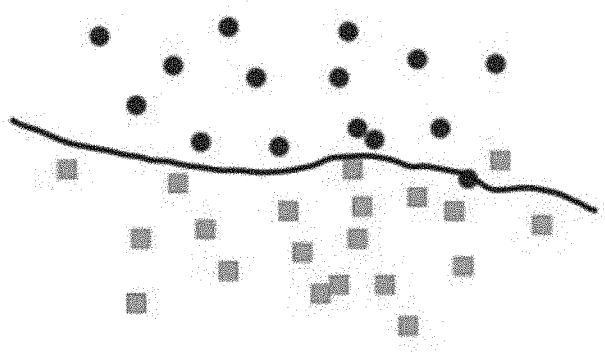

Figure 4: Logistic Regression Complex Line

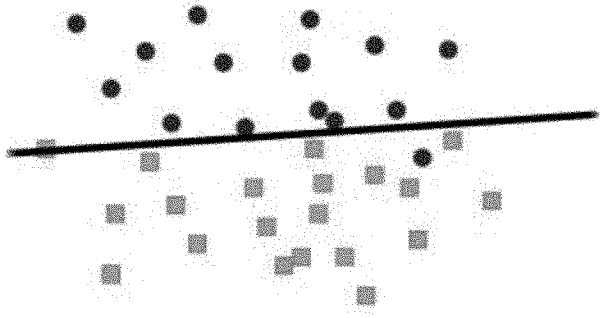

Figure 5: Logistic Regression with Better Fit

Look at a toy example in Figure 3. This extremely simple example shows the "best fit line" attempting to separate the two classes represented as circles or squares.[254] However, this best fit line gets quite a few items wrong—that is, nine of the squares fall on the circle-class side of the line.[255] Now look at toy example Figure 4. After additional optimization, a better equation provides a better separating line—the line now has only two possible errors.[256] Note that the "best fit line" does not need to be "straight."[257] With further optimization, the line might resemble that in Figure 5. The examples simply illustrate how logistic regression might separate two classes.

Technically, logistic regression uses a sigmoid, step-function to classify items—see Figure 6.[258] A sigmoid, or logistic, function provides a

---

[254] *See supra* Figure 3.

[255] *See id.*

[256] *See supra* Figure 4.

[257] *See id.*

[258] *See* Rasmussen & Williams, *supra* note 85, at 37 n.7. The authors describe the sigmoid

decision point.[259] The algorithm assigns documents with sigmoid computer values near 1.0 (the "top") to one class and values near 0.0 (the "bottom") to another class.[260] While conceptually simple, logistic regression can be quite powerful and computationally efficient.[261] Besides being computationally efficient, logistic regression can handle multi-class classification tasks.[262]

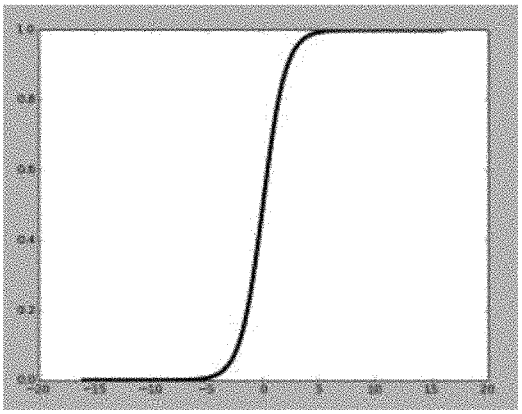

Figure 6: Sample Sigmoid or Logistic Function

Logistic regression also typically uses some form of loss function, commonly a variation on the gradient descent algorithm, to optimize for best-fit line.[263] While gradient descent goes far beyond the scope of this article, gradient descent essentially "sneaks-up-on" an optimal solution by mathematically changing the coefficients very slightly through multiple iterations of computation until the computations minimize the classification error on the training set.[264] Conceptually, gradient descent, in part, explains the reason for the predicate training data (with expert assigned classifications) as identified in Figure 7 because the training classifications from the lawyer-expert provide the correct values so the optimization algorithm can test and see how well the best fit line fits the training data.

One final issue requires explanation. As hinted before in Data Vectors and Data Matrices above, real predictive coding problems involve far more complex data dimensionality.[265] The toy examples only provide

---

function as S-shaped with the top of the S being one class and the bottom of the S being another. *See id.*

[259] *See id.* at 36-37.

[260] *See* HARRINGTON, *supra* note 169, at 85.

[261] *See* OWEN ET AL., *supra* note 192, at 252 (arguing logistic regression allows derivation of efficient learning algorithms).

[262] *See* ALPAYDIN, *supra* note 108, at 224-28.

[263] *See, e.g.*, ALPAYDIN, *supra* note 108, at 218-220; HARRINGTON, *supra* note 169, at 86-93; MOHRI ET AL., *supra* note 173, at 351-57. Essentially, gradient descent (or gradient ascent) slowly changes a set of coefficients to the equation describing the separating hyperplane until those coefficients reach the best possible hyperplane under the conditions—or fails entirely. *See* HARRINGTON, *supra* note 169, at 86-93. But, gradient descent does not guarantee an optimal hyperplane—especially if the separating line is non-linear. *See id.*

[264] *See* ALPAYDIN, *supra* note 108, at 219.

[265] *See supra* Section 3.8 (discussing Data Vectors and Data Matrices).

simple illustrations using extremely simplified, two-dimensional examples—equal to, two, data attributes.[266] Real predictive coding problems easily might operate in 90-dimesional, 560-dimensional, or 38,567-dimensional space.[267]

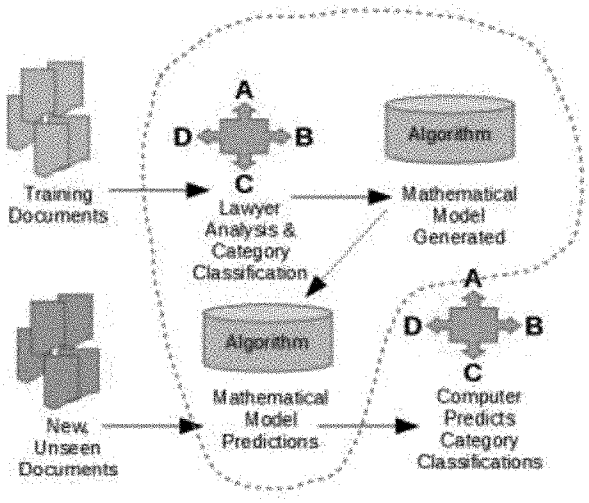

Figure 7: Typical Predictive Coding Training Process

Now look at Figure 7.[268] As with many predictive coding algorithms, the lawyer loads some sample documents into the predictive coding system, analyzes a subsample of the documents, and assigns a classification category to each item in the subsample. A predictive coding algorithm "learns" from the lawyer's analysis and develops the mathematical equation that best separates the different classification categories of documents.[269] Then, the process runs "backwards" and uses the mathematical equation to predict the classification category of previously unseen items—thus, the "predictive" in predictive coding.[270] See Figure 7 for a graphic representation of the process. The dotted line in the graphic outlines the core aspects of predictive coding systems: some type of subset legal analysis leading to

---

[266] *See supra* Figure 3-Figure 5.

[267] *See* MOHRI ET AL., *supra* note 173, at 41-48 (providing rigorous discussion of dimensionality); ALPAYDIN, *supra* note 108, at 109-120 (providing general discussion of dimensionality reduction).

[268] *See supra* Figure 7; HARRINGTON, *supra* note 169, at 233 (providing academic rigor to diagram).

[269] *See* Cormack & Grossman, *Evaluation of Machine-Learning Protocols*, *supra* note 4, at 153-54 (describing learning as interplay of human and machine); *see also* ALPAYDIN, *supra* note 108, at 220-29, 311-17 (applying classification discrimination to logistic regression and describing technical nature of optimal separating hyperplane in kernel machines).

[270] *See generally* ALPAYDIN, supra note 108, at 220-29, 311-317 (applying classification discrimination to logistic regression and describing technical nature of optimal separating hyperplane in kernel machines); Cormack & Grossman, *Evaluation of Machine-Learning Protocols*, *supra* note 4, at 153 (discussing machine learning protocols); *infra* Section 4.3.3 (discussing similar process for Support Vector Machines (SVM)).

category classification (e.g., relevant/non-relevant), the predictive coding algorithm creating a mathematical model.[271]

Applying the general method in Figure 7 to the simple logistic regression examples above and with new data added (depicted as open circles or open squares), Figure 8 shows that the logistic regression function in the example fared well when additional data was added.[272] The mathematical model predicted the location of the open circles or open squares.[273]

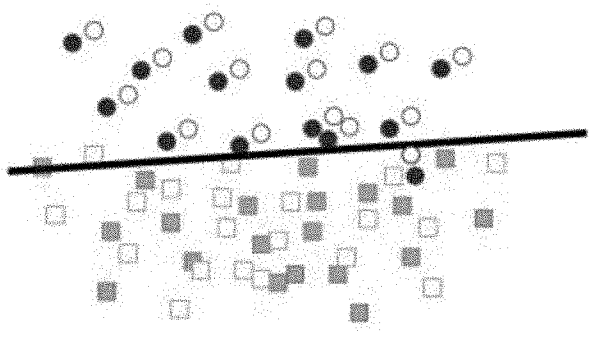

Figure 8: Toy Logistic Regression with Predicted Values

Logistic regression, while perhaps not widely used any more in predictive coding systems, explains the fundamentals important to other types of predictive coding algorithms.

4.3.3 Support Vector Machines (SVM)

Support Vector Machines (SVMs) provide powerful and reliable classifications. Originally developed as a workhorse method to address binary decisions, modifications permit SVMs to also reliably handle multi-class decisions.[274]

The SVM depends on a kernel function which creates a soft margin hyperplane using the "kernel trick."[275]

Fundamentally, a SVM operates on the principle of a separating hyperplane *with margins*.[276] What this means (in unscientific terms) is that the SVM "draws a line," based on the features analyzed in the training set,

---

[271] *See supra* Figure 7.

[272] *See infra* Figure 8.

[273] *See id.*

[274] *See* ALPAYDIN, *supra* note 108, at 327.

[275] *See* Schölkopf & Smola, *supra* note 137, at 16 (discussing support vector machine classification techniques).

[276] *See id.*

to best separate the two classes while considering the "buffer" provided by the margin. The margin distinguishes a SVM from other machine learning methods such as logistic regression or Bayesian systems.

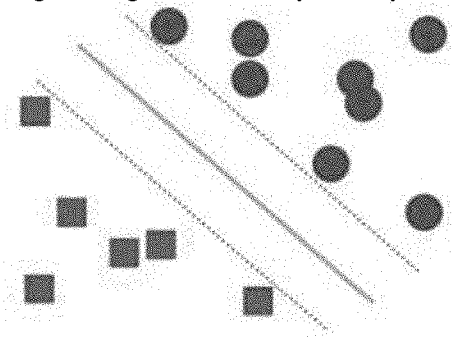

Figure 9: SVM Visualization of Training Set and Hyperplane with Margins

Figure 9 depicts the output from a very simple, binary (two-class), SVM.[277] Note how the solid slanted line, the hyperplane, separates the Boxes Class items from the Circle Class items.[278] But also carefully note the dotted line representing the margins on each "side" of the solid hyperplane line.[279] The margins provide some "wiggle-room," "slack," or "buffer"[280] which provides additional power when applying the hyperplane developed from the training set to the entire document corpus. Otherwise, an infinite number of hyperplanes exist—see Figure 10.

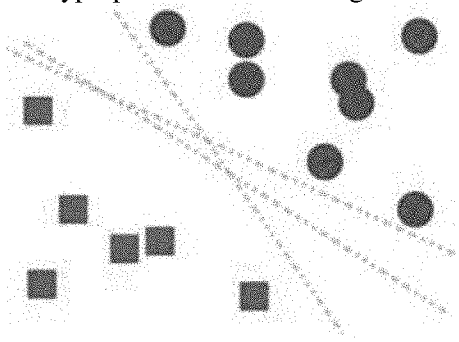

Figure 10: Infinite Separating Hyperplanes (the Long-Dotted Lines) Exist Without Margins

---

[277] *See supra* Figure 9.
[278] *See id.*
[279] *See id.*
[280] *See generally* MOHRI ET AL., *supra* note 173, at 64-83.

Figure 9 and Figure 10 demonstrate the separation of items found only in the *training set*.[281] Now, we add predicted items. The predicted items in each class have a fuzzy circle or fuzzy square with a "N" character.[282] Once you add predicted items, see Figure 11, there is really no means to evaluate which of the infinite hyperplanes best separates the Circle Class items from Boxes Class Items nor an intuitive means of evaluating errors.

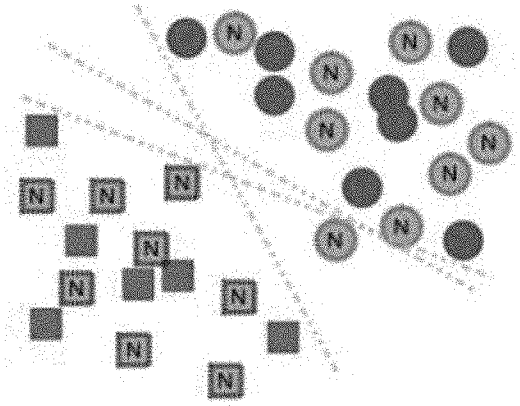

Figure 11: SVM Non-margin Hyperplanes with Predictions and Errors

Figure 12, also adds some predicted items to the diagram, but importantly, adds margins.[283] Note how most items correctly fall on the proper side of the hyperplane *and* also correctly fall on the outside of the respective margins.[284] Even though two items fall within the margins, those two items still fall on the correct side of the hyperplane—due largely to the buffer created by the margins proper positioning the hyperplane in the correct "direction."[285] If instead only one of the myriad potential hyperplanes in Figure 11 was used, the predicted items begin to develop errors, especially within what should be the margins.[286] Thus, the toy examples show how margins may help provide a better overall predictive capacity.

---

[281] *See supra* Figure 9 and Figure 10.
[282] *See infra* Figure 11 and Figure 12.
[283] *See infra* Figure 12.
[284] *See id.*
[285] *See id.*
[286] *Compare supra* Figure 11 and *infra* Figure 12.

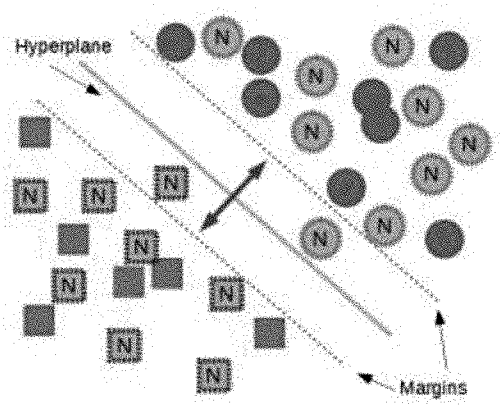

Figure 12: SVM Optimal Hyperplane with Predictions and Errors

The simple examples, however, omit the complexity of the real models generated by a SVM. First, the examples use "straight-line" hyperplanes. However, a real SVM may develop a non-linear model with a complexly curved hyperplane.

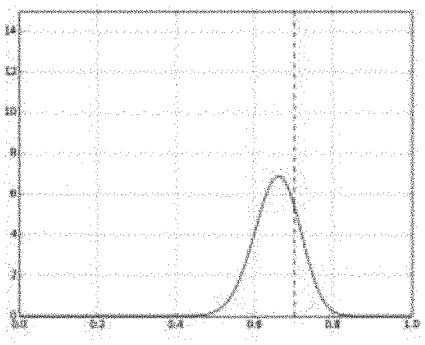

Figure 13: Computing a Normal Distribution (Bell Curve)

Second, the simple examples shown exist in just two-dimensional space. A real SVM might work in 5,000, 25,000, 50,000, or more dimensional space—impossible to depict visually.[287] Thus, the problem of creating a separating hyperplane in two-dimensions may seem fairly simple—even eye-balling might result in a decent plane. But, when scaling even to three dimensions the problem gets harder, and at four dimensions, humans cannot even visualize the space needed to create the hyperplane. At 5,000 dimensions, the hyperplane

[287] Whole areas of intensive research seek to reduce dimensionality to only those essential features necessary to optimally separate the items. *See* MOHRI ET AL., *supra* note 173, at 281-90; Schölkopf & Smola, *supra* note 137, at 427-565 (seminal text on SVMs including dimensionality reduction). Even today, high dimensionality can be very computationally intensive (can take a long time). *See* Schölkopf & Smola, *supra* note 137, at 427-565.

becomes a complex and extremely mathematically and computationally intensive operation.[288]

4.3.4 Bayesian Decision Systems & Naïve Bayes

Bayesian[289] decision systems rely on statistical probability theory.[290] A probability estimates the likelihood that a random event might occur.[291] For example, the probability that a (standard) coin will land with its head-up is about 50%.[292] Far more sophisticated algorithms can look at the joint probability of an outcome or the conditional probability of an outcome.[293] A joint probability describes the aggregate probability of multiple conditions occurring "at the same time."[294] For example, the joint probability of clouds, C, with the probability of rain, R, might be 12.5% assuming the incidence of cloudy days might be about 50% and the incidence of rain might be fairly low, assume 25%.[295] A concise equation represents the above text as: $P(C, R) = 12.5\%$.[296]

Conditional probabilities use a more complex calculation based on Bayes Theorem.[297] Bayes Theorem computes the conditional probability of event A given event B occurred[298] as: the probability of A[299] times the conditional probability of B if A first occurred all divided by the

---

[288] *See supra* note 287 and accompanying text.

[289] Bayesian systems derive from the 18th Century work of Thomas Bayes. Bayes' most famous work is Bayes theorem and provides a method to test hypotheses using conditional probabilities. *See* CATHERINE A. GORINI, MASTER MATH: PROBABILITY 120 (2012) (reviewing Bayes' formula for conditional probability). Bayes Theorem holds that the P(A|B) = P(A) * P(B | A) / P(B). *See* RASMUSSEN & WILLIAMS, *supra* note 85, at 200.

[290] *See* Grossman & Sweeney, *supra* note 197.

[291] *See* GORINI, *supra* note 289, at 11.

[292] *See id.* at 12-13 (providing probability examples).

[293] *See id.* at 110-11 (explaining conditional and joint probability).

[294] *See id.* at 111 (defining joint probability).

[295] .50 * .25 = .125

[296] *See* GORINI, *supra* note 289, at 111 (providing equation). The treatment here, by necessity, merely skims the surface of some of the core theory behind Bayesian decision systems. The point here is simply to illustrate some of the core concepts and terminology to provide a basic understanding of such systems.

[297] *See* GORINI, *supra* note 289, at 120 (reviewing Bayes' formula for conditional probability).

[298] In this example, the P(A|B), the probability of A given B, is a "posterior probability because it gives the probability of an event A after event B has already happened." GORINI, *supra* note 289, at 129.

[299] In this example, the P(A) is the "prior probability because it gives us the probability of an event A before anything else has happened." GORINI, *supra* note 289, at 129.

probability of B—or in an equation form, $P(A|B) = \frac{P(A)*P(B|A)}{P(B)}$.[300]

Advanced Bayesian decision systems can get extremely complex and may combine conditional and joint probabilities. For example, $P(A|B,C) = \frac{P(A)*P(B,C|A)}{P(B,C)}$ [301] meaning: the probability of event A given the prior occurrence of the joint probability of B and C equals the probability of A times the conditional probability of the joint probability of B and C given A already occurred all divided by the joint probability of B and C.[302]

Naïve Bayes decision systems essentially use the latter equation to calculate the probabilities from the feature vectors,[303] Naïve Bayes calculates a probability that a document belongs to Class1 or Class2 based on Bayes Theorem and looking at the joint probability of the specific features present in the document. Adapting the complex equation from earlier, Naïve Bayes compares whether the $P(Class\ 1|B,C) = \frac{P(Class\ 1)*P(B,C|Class\ 1)}{P(B,C)} > P(Class\ 2|B,C) = \frac{P(Class\ 2)*P(B,C|Class\ 2)}{P(B,C)}$.[304] If true, then the algorithm assigns the document to Class1. If false, then the algorithm assigns the document to Class 2. While seemingly a simple comparison, generating the predicate probabilities can be quite complex[305] and computationally intensive.

The probability computations require the predicate calculation of the overall incidence of specific features (words) in the training set documents (or in the document corpus).[306] Essentially, the probabilities derive from the feature set by counting the incidence of words and keeping track, via a postings list or "dictionary," of the words, counts, and documents where the words appear. The Naïve Bayes algorithm can then lookup the word counts and compute the probability of the word in the overall corpus and then compute the conditional or joint probabilities as needed using the postings list as a cross reference.

---

[300] RASMUSSEN & WILLIAMS, *supra* note 85, at 200; *see* GORINI, *supra* note 289, at 120-26.

[301] *See* HARRINGTON, *supra* note 169, at 65.

[302] This lengthy, text description should illustrate why lawyer familiarity with basic mathematical syntax becomes essential when venturing into machine learning analysis.

[303] *See* HARRINGTON, *supra* note 169, at 61-82.

[304] *See* ALPAYDIN, *supra* note 108, at 49; *see also* Rasmussen & Williams, *supra* note 85, at 33-35 (discussing classification problems in probability context).

[305] *See* HARRINGTON, *supra* note 169, at 61-82 (providing simple treatment of the computations).

[306] *See id.* at 69. The calculations noted here somewhat mirror similar calculations used in information retrieval or natural language processing. *See* JURAFSKY & MARTIN, *supra* note 106, at 661-67; RASMUSSEN & WILLIAMS, *supra* note 85, at 33-35.

Simple Postings List Example		
Word	Document ID	Word Term Count
the	1001	5
foggy	1001	2
glen	1001	1
the	1002	15
foggy	1002	1
mind	1002	1
. . .	. . .	. . .

While the Naïve Bayes algorithm description may appear complex, Naïve Bayes represents a fairly simple algorithm. Far more complex algorithms use probabilities such as the general class of Gaussian Process kernel machines,[307] Markov Decision Processes,[308] or Hidden Markov Models.[309]

4.3.4.1 Bayesian Systems Issues & Limits

Most machine learning algorithms make a "hard" classification decision: assigning a prediction to Class A or Class B.[310] Bayesian decisions instead make a "softer" guess regarding the classification of an item and then assign a probability estimate to the guess.[311] In other words, the system guesses the classification of a document based on the features present and reports a calculated probability that the document indeed belongs to the class.

However, probabilities can sometimes mislead because a probability can seem like much more than it is.[312] Fundamentally, the algorithm simply makes a best guess where the guess relies on a probability calculation. Importantly, the probability calculation relies on *a priori* knowledge about both the dataset and the training set.[313] That is, the

---

[307] RASMUSSEN & WILLIAMS, *supra* note 85, at xiv.

[308] *See* GORINI, *supra* note 289, at 285-305 (discussing Markov processes); MOHRI ET AL., *supra* note 173, at 313-19 (citing Markov decision processes as forms of reinforcement learning using probabilities).

[309] *See* ALPAYDIN, *supra* note 108, at 398-400.

[310] *See* HARRINGTON, *supra* note 169, at 33-76.

[311] *See id.* at 61.

[312] Some marketing implies that probabilities are somehow more accurate or more effective than other machine learning algorithms simply because a probability estimate occurs. This marketing may mislead lawyers unware of the limitations of Bayesian decisions—limitations similar to the limits of any machine learning algorithm.

[313] This effectively mirrors the traditional training requirement as seen in other algorithms.

probability estimate is only as good, like any other machine learning algorithm, as the training set.

Furthermore, lawyers must keep in mind that probabilities merely *estimate* the *potential* of an outcome.[314] Probabilities do not mandate a specific outcome and do not provide information on whether a specific observation actually is what the probability implies. That is, the probability provides no more accuracy or precision than most other machine learning algorithms. The probability simply estimates the likelihood and does so expressly—nothing more and nothing less.

4.3.5 Clustering

Some TAR systems employ variations on clustering algorithms. Clustering algorithms, as perhaps obvious, cluster documents into similar groups but typically use unsupervised (or semi-unsupervised) machine learning methods.

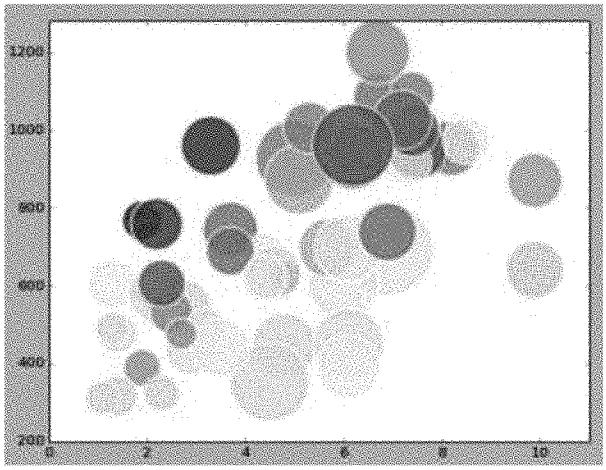

Figure 14: Toy Example of K-Means With Numerous Clusters

Clustering algorithms, see an example of output in Figure 14, traditionally measure the similarity of documents by using a geometric distance calculation and then cluster documents that are of geometrically similar distance.[315] In other words, the clustering algorithm selects several representative documents (called centroids) to serve as the anchors for each

---

[314] *See The Grossman-Cormack Glossary*, *supra* note 4, at 26 (defining "Probability" as "[t]he fraction (proportion) of times that a particular outcome would occur, should the same action be repeated under the same conditions an infinite number of times.")

[315] *See* OWEN ET AL., *supra* note 192, at 118-20; *supra* Figure 14.

cluster and then measures the vector distance of all other documents to the selected centroids to group documents with a similar distance measure in similar clusters. General clustering algorithms include k-Means[316] or k-Nearest Neighbors (kNN).[317]

k-Means algorithms (see Figure 15) are represented by small squares in the middle of each cluster.[318] Figure 15 might reflect a clustering task with two clusters—for example, relevant or not-relevant.[319]

---

[316] *See generally* MARMANIS & BABENKO, *supra* note 168, at 142-46; HARRINGTON, *supra* note 169, at 207-223.

[317] *See* OWEN ET AL., *supra* note 192, at 64-69 (defining kNN and its application); HARRINGTON, *supra* note 169, at 18-36.

[318] *See infra* Figure 15.

[319] *See id.*

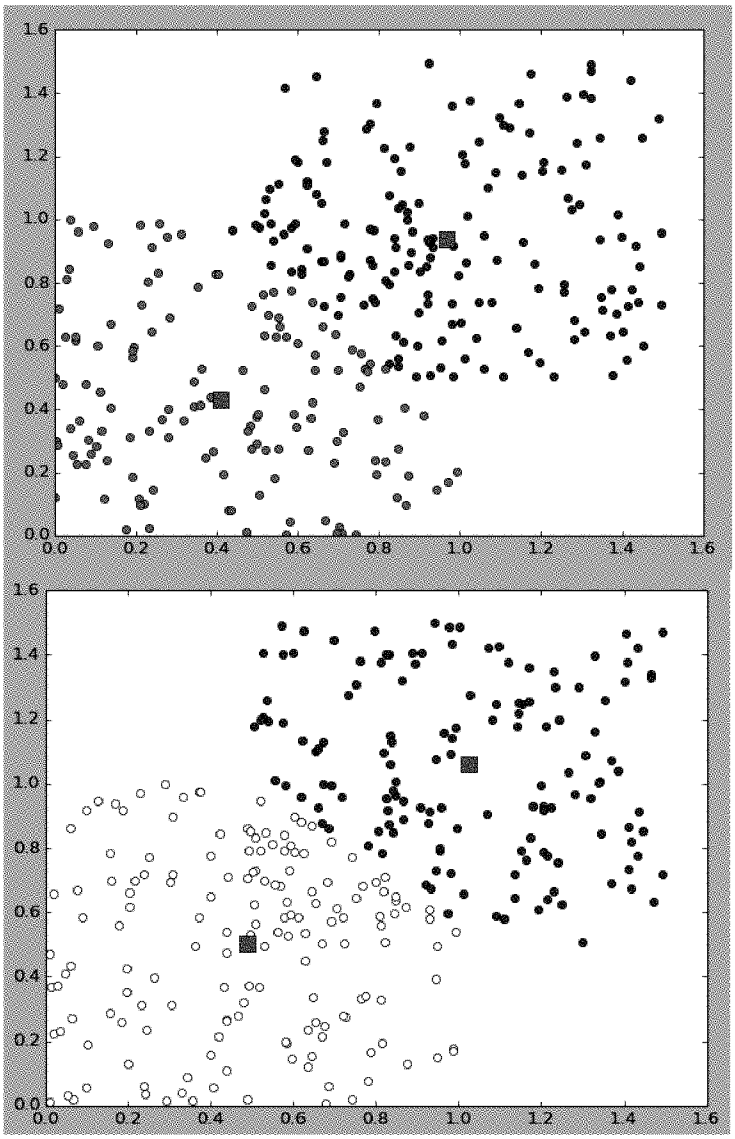

Figure 15: Toy Example of K-Means

Figure 14 shows the results of a similar k-Means algorithm but with a large number of centroids and thus a corresponding large number of clusters.[320] Attorney-expert input would determine whether the clusters fulfill the task or whether additional clusters or centroids are needed.

---

[320] *See supra* Figure 14.

k-Nearest Neighbors (kNN) (see Figure 16) serves a different clustering function.[321] kNN permits "more-like-this" features which show additional documents purportedly related to the representative document.[322]

The clustering algorithms can use several methods to measure the distance or overlap between documents—Euclidian distance, Manhattan distance, cosine distance, Levenshtein Distance, or Jaccard Similarity.[323] In other words, the clustering algorithm analyzes the terms in each document and then constructs a vector-representation of the document. The distance measures typically measure the vector distance from the selected centroids (representative documents).[324] The distance measure can have a notable effect on the clusters because each distance or similarity measure may alter the items included in each cluster. The optimal distance measure depends on the project, the document set, and the overall objectives—and is beyond the scope of this article.

But how do you select the correct representative documents? How do you know the optimal cluster numbers? How do you know which distance measure to use? These represent potential issues with unsupervised clustering and indicate that unsupervised typically requires some type of lawyer-expert input to make these types of decisions in practice.[325] Some systems allow attorney-experts to select representative documents and others allow the attorney-expert to analyze documents, much like seed set review, and tag a subset of documents with the proper cluster. These expert-tagged documents then

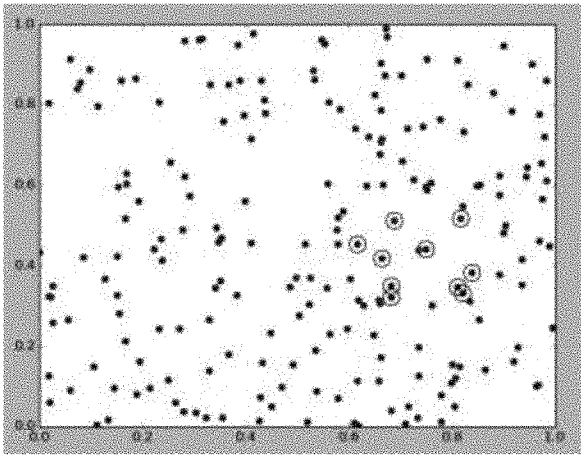

Figure 16: Toy Example of k-Nearest Neighbors

---

[321] *See infra* Figure 16.

[322] *See* OWEN ET AL., *supra* note 192, at 64-69 (defining kNN and its application)

[323] *See* JURAFSKY & MARTIN, *supra* note 106, at 661-67 (discussing vector similarity measures); OWEN ET AL., *supra* note 192, at 125-29. Similarity measures underlie several disciplines including information retrieval, machine learning, and NLP.

[324] ALPAYDIN, *supra* note 108, at 145-48.

[325] *See id.* at 155-57.

serve as the representative documents, or centroids, and the clustering algorithm adjusts the assignments to clusters based on the lawyer-expert input.

The results of clustering often closely mimic the results of classification algorithms (discussed above). But, the key distinction between clustering and classification systems is how the algorithms approach the grouping task—grouping in clusters in clustering algorithms and classes in classification algorithms. Clustering measures distance to specified centroids (or degree of overlap or similarity between documents as based on the occurrence of similar word features) while classification systems use the seed-set to develop an optimal separating hyperplane in high dimensional space that separates the respective documents based on primary features—typically words or word-phrases.[326]

4.4 Natural Language Processing and Latent Semantic Indexing Methods

Attorneys may confuse natural language processing and latent semantic indexing *techniques* with TAR *algorithms* and TAR *systems*.[327] More accurately, natural language processing (NLP) and latent semantic indexing (LSI) largely represent methods or techniques used to analyze and process the human language such as language in a document.[328] The legal community has mentioned these techniques and thus mentioning them here, to place them into context, is necessary.[329] Note, due to the breadth and

---

[326] *See* OWEN ET AL., *supra* note 192, at 118-20 (defining clustering).

[327] *See* Henry, *supra* note 83, at 39-40 (implying that NLP or LSI systems will replace predictive coding in eDiscovery). Professor Henry argues that the NLP and LSI systems "understand" language and thus obviate or minimize the need for attorney-expert input. *Id.* However, this misunderstands the roles of TAR and NLP-and-LSI. NLP and LSI serve as adjuncts to TAR algorithms in an eDiscovery context. *Id.* NLP and LSI systems may replace TAR, but not for the reasons Professor Henry implies. Information governance needs, not eDiscovery, will largely drive research into these types of more advanced systems because information governance requires companies or entities to proactively detect and prevent problems evident in data and thus may preemptively reduce lawsuits and subsequent eDiscovery. Such systems no longer exist as legal-community science fiction. *See* Tam Harbert, *IBM's Watson Coming to a Firm Near You*, LAW TECH. NEWS, Dec. 1, 2014, *available at* https://advance.lexis.com/api/permalink/b2f02840-cf12-4152-bf14-8310c7d40c13/?context=1000516 (introducing IBM's Watson cognitive platform); Mariella Moon, *IBM's Watson Supercomputer Has a New Job, As a Lawyer*, DIGITAL TRENDS (Mar. 13, 2013), http://www.digitaltrends.com/computing/watson-usc-competition/ (recounting competition utilizing IBM's Watson predictively estimating evidence probability).

[328] *See* LESKOVEC ET AL., *supra* note 120, at iv, 418-36 (discussing latent semantic indexing and application in single-value decomposition techniques to optimize algorithms); JURAFSKY & MARTIN, *supra* note 106, *passim* (discussing natural language processing as ability for computers to process human language).

[329] *See* Nat'l Day Laborer v. U.S. Immigration & Customs Enforcement Agency, 877 F. Supp. 2d 87, 109 (S.D.N.Y. 2012) ("[B]eyond the use of keyword search, parties can (and

complexity of these concepts, the purpose of introducing these techniques in this article is to simply distinguish *techniques* from TAR *algorithms* and *systems*.

For the purpose of this article, NLP and LSI techniques are used to optimize a wide array of TAR systems rather than these techniques somehow representing discrete systems in themselves. In other words, at this point, one does not necessarily buy a NLP system or a LSI system although TAR systems may use NLP or LSI elements.

In an early article on document retrieval systems used in legal matters, David C. Blair and M.E. Maron summarize natural language processing:

> The basic idea of [natural language processing] is that one can use the formal aspects of text to predict its meaning or subject content: formal aspects such as the occurrence, location, and frequency of words . . . and syntactic structure of word phrases . . . . It was hoped that . . . one could get the computer to deal with text in a 'comprehending way.[330]

Thus, Natural Language Processing (NLP) uses the knowledge of language itself to analyze language—either written or spoken.[331] While significant overlap exists between disciplines, natural language processing stands apart because it specifically focuses on the language itself as the subject of study.[332] The overlap can cause confusion in the legal community because natural language processing incorporates many technical techniques and analysis methods, including machine learning, statistical, predictive coding, and TAR concepts, rather than relying on any one, distinct technology.[333]

Some may recognize nascent NLP techniques from the 1960s era ELIZA-type computer programs that attempted to mimic human responses

---

frequently should) rely on latent semantic indexing, statistical probability models, and machine learning tools to find responsive documents.").

[330] BLAIR & MARON, *supra* note 130, at 289, 295. As Blair and Maron also note, natural language processing concepts in a legal context were not new in 1985 and predated the 1985 article by over 25 years. *See id.*; David C. Blair and M.E. Maron, *An Evaluation of Retrieval Effectiveness for a Full-Text Document-Retrieval System*, 28 COMMUNICATIONS OF THE ACM 3, 289, 295 (Mar. 1985)

[331] *See* JURAFSKY & MARTIN, *supra* note 106, at 1-4, 6-9.

[332] Natural language processing adapts more generalized tools for use in the study of language. Thus, natural language processing may use information retrieval techniques, decision trees, machine learning, artificial intelligence, neural networks, support vector machines, Gaussian models, hidden Markov models, and other tools but specifically tuned for studying language. *See* JURAFSKY & MARTIN, *supra* note 106, at 6-9, 10-14.

[333] *See generally* JURAFSKY & MARTIN, *supra* note 106, at 6.

to questions posed to the computer.[334] In 2011, the IBM Watson computer famously defeated Jeopardy! champions using NLP.[335] NLP also underlies voice dictation software, Apple's Siri, Microsoft's Cortana, and even optical character recognition (OCR) systems—long used in the legal community for scanning documents to text. As all of these examples illustrate, NLP fundamentally addresses the complex issue of computer recognition of speech communications and text.

4.4.1 Latent Semantic Indexing

Latent semantic indexing (LSI) gained some recognition in the legal community.[336] Latent semantic indexing typically enhances keyword search.[337] The *Grossman-Cormack Glossary* defines LSI as a "feature engineering algorithm" that groups correlated terms.[338] But what does that mean?

In simple terms, latent semantic indexing techniques associate related words into conceptual groups.[339] For example, consider the groupings:

1. "snow, boots, sled, cold" and

---

[334] *See id.* In such a Turing Test, a computer attempts to emulate thinking. *See id.* A video depiction of an early ELIZA-like program is available via YouTube. *See Eliza for the Commodore PET\Commodore CBM*, YOUTUBE (Dec. 18, 2011), https://www.youtube.com/watch?v=praF_0WbuQI. In 2011, the IBM Watson computer famously defeated Jeopardy! champions and was hailed as approaching a successful Turing Test. *See* John Markhoff, *Computer Wins on 'Jeopardy!': Trivial It's Not*, N.Y. TIMES, Feb. 16, 2011, at A1, *available at* http://www.nytimes.com/2011/02/17/science/17jeopardy-watson.html?pagewanted=all.

[335] *See* Markhoff, *supra* note 334, at A1 (recounting IBM's Watson Jeopardy! Experience); *see also What is Watson?*, IBM, http://www.ibm.com/smarterplanet/us/en/ibmwatson/what-is-watson.html (last visited Feb. 1, 2016) (explaining IBM Watson technology platform).

[336] *See* Nat'l Day Laborer Org. Network v. U.S. Immigration & Customs Enforcement Agency, 877 F. Supp. 2d 87, 109 (S.D.N.Y. 2012) (citing Grossman & Sweeney, *supra* note 197); *see generally* Grossman & Sweeney, *supra* note 197 (citing latent semantic indexing but nevertheless defining LSI somewhat inaccurately). *But see* Bothwell v. Brennan, No. 13-cv-05439-JSC 2, slip op. at 6-8 (N.D. Cal. 2015) (holding *National Day Laborer* does not mean keyword searches cannot be used if reasonable and search terms disclosed).

[337] *See* Deerwester et al., *Indexing by Latent Semantic Analysis*, *supra* note 151, at 391-96. Tim Leehealey insightfully notes that LSI is simply an augmented keyword search technique despite bold claims by vendors. *See* Leehealey, *supra* note 214.

[338] *The Grossman-Cormack Glossary*, *supra* note 4, at 22 (defining "Latent Semantic Indexing"). The definition correctly cites LSI as an algorithm, but also carefully notes that LSI is not a TAR algorithm. *See id.* Thus, LSI techniques might apply in a number of analytical contexts rather than being a stand-alone TAR system. *See id.*

[339] *See* Deerwester et al., *Indexing by Latent Semantic Analysis*, *supra* note 151, at 391-96 (providing comprehensive overview of latent semantic structure analysis); *see also The Grossman-Cormack Glossary*, *supra* note 4, at 11, 14, 17, 22 (defining latent semantic indexing, feature engineering, dimensionality reduction, and concept search).

2. "snow, betrayed, charlatan, scoundrel."

In Group 1, the pattern of words latently, to an English-language speaker, suggests a meaning for "snow" as in weather or precipitation. But in Group 2, the same term, "snow," suggests a very different meaning based on the other words in the group—suggesting a meaning such as nefarious, dishonest, of deceptive. With nothing else, one can derive a reasonable meaning of a target word based on the relationship of that word with the other words in the grouping.

In very simple conceptual terms, the groupings reduce the complexity of determining the meaning of target terms within a document because the conceptual groupings reduce the universe of potential meanings of a specific word.[340] In other words, the contextual grouping narrows the possible meanings. This reduction can also help to reduce ambiguity.

Even in early research, LSI showed significant improvement over simple keyword search matching.[341] However, as the early research attests and as the examples demonstrate, claims today that LSI techniques replace TAR systems remain somewhat questionable.[342]

4.4.2 Natural Language Processing

Natural language processing resembles LSI but goes far beyond just conceptual groupings. NLP describes a whole discipline that addresses fundamental issues of the computational processing of human languages. Essentially, as opposed to plain TAR algorithms, which apply to any subject matter, NLP focuses on human languages—both speech and written language.[343] That is, NLP focuses on understanding language itself—including sentence parsing, word frequencies, parts-of-speech parsing,

---

[340] *See* LESKOVEC ET AL., *supra* note 120, at 418-28.

[341] *See* Deerwester et al., *Indexing by Latent Semantic Analysis*, *supra* note 151, at 402; Deerwester et al., *Improving Information Retrieval*, *supra* note 213, at 36.

[342] *See* Henry, *supra* note 83, at 39-40. *National Day Laborer* mentions the technique in a context suggesting that LSI was a newer type system and potentially replacing keyword search. *See* Nat'l Day Laborer Org. Network v. U.S. Immigration & Customs Enforcement Agency, 877 F. Supp. 2d 87, 109 (S.D.N.Y. 2012). Again, even the glossary definition carefully notes that LSI is a technique and not a stand-alone TAR algorithm. *See The Grossman-Cormack Glossary*, *supra* note 4, at 22 (defining "Latent Semantic Indexing").

[343] *See generally* JURAFSKY & MARTIN, *supra* note 106, at 6. Interestingly, NLP techniques arose from cryptoanalysis during World War II and the Cold War. *See* DAVID KAHN, THE CODE BREAKERS 380-83, 759-61 (Scribner 1996) (1967). Claude Shannon wrote the seminal paper on computational analysis of human language (speech) and information theory just after World War II. *See* Claude Shannon, *A Mathematical Theory of Communication*, 27 BELL SYS. TECH. J. 379, 379 (1948).

syntax, word phonology, word roots, and many other aspects of human language that we often take for granted.

Section 3.5, above, mentions n-grams in the context of preprocessing.[344] However, n-grams also illustrate what it means to analyze language.[345]

Consider the following sentence fragment:

"The fish _____ downstream."

An English-language speaker would likely fill the blank with the word *swim*, *swims*, or *swam*. Yet, an English-language speaker is unlikely to insert *RC Cola®*, *bear*, *hiccupped*, *but*, *deposed*, or *res ipsa loquitor* into the blank.[346] But why? Because the context of the sentence, that is the relationship of the other words, largely determines the likely word candidates for the blank. A machine learning scientist or statistician might call this the "likelihood" of a particular combination and try to measure the likelihood with a "probability"—for example, "based on the corpus of all sentences, the probability of *swim*, *swims*, or *swam* given this sentence context is estimated at 95%." Once pointed out, n-grams seem intuitive—of course the other words in a sentence determine the context. Nevertheless, this is a powerful concept and becomes even more powerful when applying to the context of paragraphs, documents, or even document sets.[347]

N-grams have contributed, for example, significantly to the effectiveness of real-time natural language processing applications such as speech-to-text (dictation), smartphone digital assistants (Siri or Cortana), machine-based language translation, and optical character recognition (OCR).[348]

While greatly simplifying a highly complex and emerging research field, analysis of n-grams, for example, permits a software application to use a richer context to predictively determine the most likely "meaning" of a sentence based on the words in the sentence, based on paragraphs, or based on the entire article—not just to fill-in-the-blank, but to potentially extract overall and more complex contextual meaning of a sentence, paragraph, or article.[349]

---

[344] *See supra* Section 3.5 (discussing n-grams).

[345] However, understand that n-grams are simply one possible method of performing NLP analysis.

[346] However, one cannot rule out idiomatic insertions such as `leapt`, `sailed`, `floated`, or `catapulted`. A true NLP system, properly trained, should have probabilities for these types of idiomatic usages because presumably such idioms occur in the corpus of potential documents.

[347] *See infra* note 129 (showing photovoltaic conveys abstract renewable energy meaning).

[348] *See* Microsoft Research, *Natural Language Processing*, http://research.microsoft.com/en-us/groups/nlp/ (last visited Feb. 1, 2016).

[349] *See* Adam Lally & Paul Fodor, *Natural Language Processing with Prolog in the IBM*

5 CONCLUSION

As should be evident, attorneys must understand basic information about the technologies being deployed to address eDiscovery issues. Not only is there an ethical obligation to understand the technologies, courts (and clients) increasingly demand technology knowledge as part of a duty to efficiently address discovery-related matters arising from ESI. Simply put, under current procedural rules, the only practical way to efficiently address the volume of ESI is technology.

Foremost, a fundamental difference exists between traditional keyword search technologies and newer TAR, including so-called predictive coding, technologies. Both coexist in the attorney's toolbox. Keyword search remains relevant when used to retrieve specific documents based on specific and relevant terms. However, keyword search does not work well for general classification or categorization tasks.

TAR systems, in their many variants, address the fundamental weakness of keyword systems. TAR systems work best to classify documents into general categories such as relevant or not-relevant; privileged or not-privileged; and email, medical reports, or memoranda. This classification capability, sorting documents into general categories, distinguishes TAR systems from keyword search. TAR systems can employ classification, clustering, or hybrid algorithms.

Because many legal tasks require analyzing documents and human language, natural language processing techniques play a key role in legal applications. Rather than being a specific algorithm, NLP represents research and methods for analyzing language for meaning or context. NLP techniques can optimize TAR algorithms—and even plays a role in enhancing keyword search systems such as latent semantic indexing.

Lawyer awareness of how systems preprocess documents prior to applying TAR algorithms provides key insights into an often overlooked area of eDiscovery. Preprocessing techniques determine what the algorithms see and thus may affect the results. All known TAR algorithms require some form of preprocessing, thus, the issue is not abandoning preprocessing but assuring that attorneys play a key role in assessing preprocessing decisions.

Watson System, ASSOC. FOR LOGIC PROGRAMMING, 1, 1-2 (2011), http://www.cs.nmsu.edu/ALP/wp-content/uploads/2011/03/PrologAndWatson1.pdf (describing how Watson processed queries). The IBM Watson Computer System, famous for playing the television game Jeopardy!, demonstrates the enormous potential for natural language systems. *See* Quentin Hardy, *IBM to Announce More Powerful Watson Via Internet*, N.Y. TIMES, Nov. 13, 2013, at B1, *available at http://www.nytimes.com/2013/11/14/technology/ibm-to-announce-more-powerful-watson-via-the-internet.html?_r=0.*

While understanding technologies may challenge some attorneys, the changing nature of evidence in legal matters demands significant changes in the methods and tools used by attorneys. Attorneys must now understand the technologies themselves to understand if-how-and-when to deploy the technologies. These decisions flow from the attorney's core duties to clients and now represent the norms of contemporary law practice. Therefore, from within the legal profession, the black boxes must become at least a little greyer, certainly less frightening, and properly managed.[350]

---

[350] *See* Asimov, *supra* note 1, at 68-111. Asimov's science fiction tale traces the fictional development of "machines." *See id.* The machines gain some autonomy following a set of three, simple rules developed to protect humans. *See id.* at 37. As the story progresses, the "machines" get out of control. *See id.* The cautionary tale emphasizes that retaining human control and input, rather than delegating machine control to scientists or the machines themselves, is required to manage the startling "thinking capacity" of the machines. *See id.* Likewise, attorneys delegating TAR tasks risk similar consequences where the machines, and experts, get out of control and replace the legal community. *See id.* at 198-224.