LJN LAW JOURNAL NEWSLETTERS

The Corporate Counselor®

An ALM Publication

Volume 29, Number 3 • June 2014

Predictive Coding and Keyword Search

Complements or Competitors?

By Adam Beschloss and James K. Jones

Litigation, investigation, and regulatory requests require in-house counsel to manage multivariate issues (legal and business) to effectively mitigate risk involving threats to reputation, finance, and even survivability. This must all be done within the confines of expedience and cost.

Well into the era of electronic discovery, few would argue against the value of technology to assist in this regard. Predictive coding, also referred to as "technology assisted review" or TAR, is the latest attempt at taming the electronic data behemoth that presents itself as millions of pages for review (and represents approximately 70% of discovery costs). Clearly, one cannot apply the same methods or technologies that were established when a matter involved boxes of paper to electronic data volumes that are now counted in terabytes. (This article only considers the review of language-based, unstructured data and not structured data, such as financial information or documents such as schematics.)

In-house counsel are often presented with a multitude of options by both outside counsel and eDiscovery vendors, and it can be difficult to digest this information and choose the best approach for a given matter, especially when faced with the strong opinions of a trusted law firm that has successfully used a technology or process before. While outside counsel will be tasked with actually implementing the process going forward, in house counsel can play a more involved role in the decision making process if they have a grasp of the right questions to consider. Is the use of traditional keyword searching to effectively cull and search a corpus now defunct? Is predictive coding inherently superior? What does predictive coding do differently than keyword searching and how can counsel design the most effective process around predictive coding?

Much of the discussion has forced attorneys and other laypersons (from the perspective of the science involved) to come to terms with concepts such as recall and precision, richness, confidence levels, confidence intervals, acceptable error rates, false positives and negatives, statistical sampling, algorithms, and other rather involved topics. "Acceptable error rate" is a particularly jarring concept for the practice of law, where attorneys strive for nothing less than perfection outside the realm of eDiscovery. This concept, however, is not the sole purview of machines. Human review is subject to these rules as well, as are keyword searching methods.

The advent of TAR with the express purpose of not reviewing certain documents, however, has pushed this topic to the forefront. "Technology-assisted review highlights, and renders eminently calculable (at least from a statistical perspective) … the fact that some number of relevant documents knowingly will not be produced" (Schieneman, Karl, and Thomas C. Gricks III, The Implications of Rule 26(g) on the Use of Technology-Assisted Review. *The Federal Courts Law Review*. no. 1 (2013): 243). It is worth stepping back from the trees of statistics, linguistics, and algorithms to see the forest, particularly as in-house counsel consider how money can best be spent to uncover salient issues and meet production obligations.

Predictive Coding

At a high level, predictive coding works as follows: A set of features is identified that is used to distinguish between relevant (R) and non-relevant (NR) documents. These features often include the frequency of occurrence of individual words, phrases, and sets of words that co-occur in documents. The reader may recognize similarity between what predictive coding is attempting and what a well-constructed keyword query ideally does. A key difference is that predictive coding automates this process. Of course, this automation does not happen by magic. The system must be trained.

As the system is trained, it determines which features best discriminate between R and NR documents. There are a variety of ways this is accomplished, but the primary methods include: 1) calculating the probability that a particular feature is associated with R or NR documents; 2) using the features to determine which documents are most similar to each other; 3) deriving rules from the features to make R/NR classifications; and 4) trying to draw a line that would best separate the R documents from the NR documents if you were to graph them according to their features. (The graph would actually be multi-dimensional and the "line" a hyperplane. We can add hyperplane separation theorem and Euclidean Geometry to the list of concepts that most lay people shouldn't need to talk about.)

Training

All of this is made possible through the training. This training is usually accomplished with "training" and "seed" sets. These sets are composed of documents known to be R or NR, and are used to train the system in identifying other documents like them (based on the features described

Adam Beschloss is director, business development in the New York office of QuisLex, where he designs and implements eDiscovery workflows for litigation matters and investigations. **James K. Jones** heads QuisLex's Legal Technology Group. The authors would like to thank **Phil Algieri**, associate vice president of Legal Services, for his contributions to this article.

above). You can't simply point the predictive coding engine at a document collection and say "do math and hyperplane separation and find me only (and all) the relevant documents."

We must instead turn to the oft-maligned entity, the human mind. The training regimen requires trainer(s) to code documents R and NR to train the system. Multiple trainers may be required due to time constraints and the number of training documents required, but this raises concerns about consistency. (This now adds repeatability, reproducibility, and measurement system analysis to our lexicon of concepts that lay people shouldn't have to talk about.) All of this can present a bit of a conundrum: If one knew where to find R documents to begin with, we wouldn't have to understand the arcane concepts of statistics and linguistics.

Creating a Model

Basically, the system needs a fundamental model of what R and NR documents look like to function. With this model, the system returns documents identified as matching the set of features found in the exemplar documents. The trainer(s) then agrees or disagrees with the identification and returns the results to the system. This continues until the system has the best fit for the model. Again, how do we find those initial "training" documents?

Finding the 'Training' Documents

One approach is to randomly sample documents from the unculled document population, and then have the trainer(s) code them as R or NR. This continues until the system has enough identified R documents that it can form the necessary model. The drawback is that the document population can be extremely large, while the "yield" or "richness" (*i.e.*, the percentage of R documents in the unculled pool) is typically very low. This necessitates a significant amount of manual review before the predictive coding system can perform well. Further, such an approach may not identify enough exemplar documents for the system to fully develop a model. The number of NR documents identified through random sampling is likely to be much higher than the number of R documents delivered to the trainer(s), prolonging the exercise.

Keyword Searches

Another approach involves using keyword searches to increase the likelihood of identifying R documents and consequently provide a richer set of data for initial training. While certain proponents of predictive coding may doubt the efficacy of this process, search terms can perform very well if properly constructed. The key is to not simply create a list of terms via guesswork, but to engage in a process involving custodial interviews, sampling and statistical validation, iterative refinement, intelligent application of Boolean search concepts in collaboration with search experts. In fact, many predictive coding systems allow the trainer to proactively identify particularly relevant documents with which it can supplement the model. Through the intelligent use of search as described above we can specifically target these documents. It may be that the "old" method is what provides efficacy to the "new."

> **"Acceptable error rate" is not the sole purview of machines. Human review is subject to these rules as well.**

An argument against the use of keywords to kick-start predictive coding is the fear of bias. Specifically targeting documents relating to known issues to form the training set may bias the system to recognize as relevant only these known issues while failing to identify important issues you may not have known to search for, but would be uncovered with a large random sample. The fear is that the system may code documents pertaining to the important but unknown issues as NR having never encountered documents of this type.

On the other hand, as responsive or relevant topics are often interrelated, it may be more likely for the trainer(s) to come across these unknown issues as they review the targeted set as opposed to happening upon them in a random sample. In short, even within a predictive coding regimen, a properly constructed keyword search may have real value, particularly when it comes to increasing the degree of relevant material in the initial training and seed sets and in effect, making better use of dollars invested.

Search terms can be used later in the process as well when further investigating the document population that has been "predictively" coded. If you are are only concerned with fulfilling your basic discovery obligations, it may be possible to engage in a review designed to simply validate the system's R/NR decisions. However, the "most relevant" documents as determined by the review technology may be routine, uninteresting documents that need to be produced, but aren't meaningful or case altering (aka, hot). Customized search strings (which can then be further refined by date, time, custodian, file type, etc.), however, can be very effective at examining specific issues. In this sense, keyword search again serves as a powerful tool.

Conclusion

Ultimately, there is nothing inherently best about any particular methodology. A technology or process is only valuable in terms of the ability to solve a particular problem. As effective as technology may be, no single tool (excepting human intellect) is appropriate in all circumstances. Understanding what tools to use, and when, is critical from both a legal and business perspective. Moreover, the ability to comprehend their efficacy relative to the nuances of a particular matter determines whether potential benefits are fully realized.

Technology, such as predictive coding, requires the involvement of highly trained attorneys, statisticians, linguists, and technologists. Effective keyword searching requires no less skill. It is only the practical application of technology (and knowledge) that will provide counsel with the confidence that they have done what is necessary from a strategic vantage point and met their 26(g) obligation. Whether this is affected by a review employing predictive coding, keyword search, or some combination of the two, is a decision that shouldn't be based on the general acceptance that only the newest technology is best, but on the specifics of the matter.