LEGAL IT PROFESSIONALS

April 7, 2014

Has Predictive Coding Made Keyword Search Obsolete?

By Adam Beschloss

Well into the era of electronic discovery, few would argue against the use of technology to assist in document review. Predictive coding is the most recent attempt at taming the electronic data behemoth that presents itself as millions of pages for review. Clearly, one cannot apply the same methods that were established when a matter involved boxes of paper to massive volumes of electronic data. But does this new technology render keyword search obsolete? Is predictive coding inherently superior, or can they serve as complements?

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, 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 legal practitioners who insist on perfection outside of eDiscovery. The advent of predictive coding with the express purpose of not reviewing all documents has pushed this concept to the forefront. "Technology-assisted review highlights…the fact that some number of relevant documents knowingly will not be produced." It is worth stepping back from the trees of statistics and algorithms to see the forest, particularly as many practitioners ponder how best to employ predictive coding technology (if at all).

Firstly, what does predictive coding do differently than keyword searching? At a high level, predictive coding develops a set of features used to distinguish between relevant ("R") and non-relevant ("NR") documents. Features often include the frequency of occurrence of individual words, phrases, and sets of words that co-occur in documents. This is not unlike what a well-constructed keyword search attempts to do. A key difference is that predictive coding automates this process. Of course, this automation does not happen by magic. The system must be taught.

Training enables the system to build a model of features that best discriminate between R and NR documents. Primary calculations include (i) the probability that a particular feature is associated with R or NR documents, (ii) determining which documents are most similar to each other, (iii) deriving rules from the features to make R/NR classifications, and (iv) 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 is actually multi-dimensional and the "line" is actually 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.)

To initiate these calculations, we must turn to the oft maligned technology, the human mind. The training regimen requires a trainer or trainers to code documents R and NR. Multiple trainers may be required due to time constraints and the number of training documents required which may raises concerns regarding consistency. (This now adds repeatability, reproducibility, and measurement system analysis to our lexicon of concepts that lay people shouldn't talk about.) This presents a bit of a conundrum: if one knew where to find R documents to begin with, we wouldn't need predictive coding.

The system needs a basic model of R and NR documents to function. Ideally, once the model is established, only those documents that match the features found in the exemplar documents will be returned for review by the trainer(s) who agrees or disagrees and thusly tunes the system. This process continues until the system calculates the best possible fit for the model. So how do we find effective training documents?

One approach is to randomly sample documents from the unculled document population and have the trainers code them as R/NR until the system has enough identified R documents to form the necessary model. The drawback is that the document population can be extremely large, while the "yield" or "richness" (the percentage of relevant documents in the unculled pool) is typically very low, necessitating a significant amount of manual review before the predictive coding system can perform well, if at all. Further, such an approach may not identify enough exemplar documents for the system to fully develop a model.

Another approach involves using keyword searches to increase the likelihood of identifying R documents and provide a richer set of data for initial training. The key is to not simply create a list of terms via guesswork, but to engage in a thoughtful process involving custodial input, sampling, statistical validation, iterative refinement, intelligent application of Boolean search concepts, and collaboration with linguists and other search experts. Doing so significantly increases the effectiveness of search terms, reducing false hits and improving yield.

Many predictive coding systems also allow the trainer to proactively identify highly relevant documents to supplement its model. The intelligent use of keyword search as described above can specifically target these documents. It is interesting to ponder that the "old" method may be what provides efficacy to the "new."

An argument against the use of keywords to prime 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 as yet unknown issues which presumably the larger purely random sample of an unculled population would uncover. On the other hand, responsive or relevant topics are often interrelated. It may be more likely to come across these unknown issues in a targeted set. In this way, even within a predictive coding regimen, keyword search may add value.

Finally, we can revisit the use of search terms later in the process when further investigating the document population that has been "predictively" coded. If only concerned with fulfilling basic discovery obligations, it may enough to simply validate the system's R/NR decisions. However, as attorneys involved in document review know, the features that determine whether a document is "relevant" are often very different than those that determine whether a document is "important" (aka "hot"). Documents determined by the review technology as relevant may be routine, uninteresting documents that need to be produced, but aren't meaningful or case-altering. Keyword search (which can then be further refined by date, time, custodian, file type, etc.), however, can be very effective at examining specific issues, and serve as a critical complement to predictive coding in achieving substantive objectives.

Ultimately, a technology or process is only valuable in terms of the ability to solve a particular problem. As smart as technology may become, no single tool (excepting human intellect) is appropriate in all circumstances. The ability to comprehend technology's efficacy relative to a particular matter, and the limitations of automation, determines whether potential benefits are fully realized.

Adam Beschloss, Director, Business Development, QuisLex, has over 15 years of experience designing and implementing eDiscovery workflows for litigation matters and investigations. Beschloss received his B.A. from Columbia University and is located in QuisLex's New York City headquarters

Originally published April 7, 2014 by Legal IT Professionals http://www.legalitprofessionals.com