Introduction to Predictive Coding

Herbert L. Roitblat, Ph.D. CTO, Chief Scientist, OrcaTec

Predictive coding uses computers and machine learning to reduce the number of documents in large document sets to those that are relevant to the matter. It is a highly effective method for culling data sets to save time, money and effort. Predictive coding learns to categorize documents (for example, as responsive or non-responsive) based on a relatively small sample of example documents.

Predictive coding is not magic. It does not replace all of human review. It does not cure cancer. Predictive coding is mathematical algorithms and applied statistical analysis used to emulate the decisions that an authoritative expert would make, based on the evidence in the documents.

Predictive coding allows one person or a small group of people to effectively review millions of documents in a short period of time, with higher accuracy and consistency, and at a much lower cost than traditional review methods. In predictive coding, a computer is "trained" to distinguish between responsive and non-responsive documents. The system can then use the differences between these two sets of documents to infer how to categorize the remaining documents in the collection.

Rather than having to read every document in a collection, predictive coding allows one to get similar results after reading only a relative few. At its best, predictive coding is like the results one would get if one person read the entire collection of documents with perfect attention and no fatigue – oh, and could read thousands of documents an hour.

There are several ways that systems can get their training examples. These training documents are a sample of all of the documents in the collection. The examples can be selected randomly and categorized, can be provided by expert reviewers, chosen by the computer, or determined by some combination of these.

Predictive coding is a kind of Computer-Assisted Review (CAR) or Technology-Assisted Review (TAR), but it is not the only kind of CAR/TAR. Other types include keyword searching, concept searching, clustering, email threading, more-like-this search, and near duplicates. These other kinds of CAR can be very useful and can reduce the time needed to categorize documents, but they are not predictive coding – they do not predict on the basis of examples which documents are likely to be responsive versus non-responsive.

In predictive coding, the computer uses the decisions made by the expert reviewer(s) to predict how other documents should be categorized. In clustering or the various kinds of searching, the documents are organized into groups and, after the computer has done its work, the reviewers then decide whether each of these groups should be considered responsive or non-responsive. Predictive coding involves what is called in the jargon of machine learning "supervised learning," while the other approach, when it involves machine learning, is called "unsupervised learning." In predictive coding, the authoritative expert reviewer provides feedback or supervision to the predictive coding system.

In this paper we will be concerned with the predictive approach to CAR. Other approaches, though useful, will have to be beyond the scope of the current paper.

Why do we need Computer-Assisted Review?

The three biggest problems in eDiscovery are volume, volume, and volume. The number of electronic documents that must be considered has grown exponentially over the last decade or more. A gigabyte of data was once considered fairly large, but now terabytes are the norm in many kinds of cases. The cost of processing electronic documents has dropped, perhaps 10 or 20 fold (from $3,000 per gigabyte to sometimes less than $300), but the volumes that need to be processed have increased a thousand fold or more.

The cost of processing electronic data has become only a small part of the expense of eDiscovery, typically dwarfed by the cost of review. It can cost a dollar or even several dollars per document just to review for relevance. In fact, the costs of eDiscovery are so large that they threaten to overwhelm the justice system, often exceeding the amount at risk in civil cases. These costs present a severe burden to parties on both sides of the matter, so there is a lot of motivation to find some ways to reduce those costs and to bring them more into proportion to the amounts at risk.

Moreover, traditional linear review, where at least one person reads every document, turns out to be only moderately accurate. For example, Roitblat, Kershaw, and Oot (2010) had two teams of professional reviewers each review a sample of documents. If one team determined that a document was responsive, the odds that the other team would also find it responsive were about 50:50. Other studies have found similar results. Essentially, these studies find that human reviewers are missing almost every other responsive document. Traditional human review is very expensive, and not particularly accurate.

Table 1. Human Review Accuracy from the TREC legal track, 2008 and from Teams A and B from the Verizon study of predictive coding (Roitblat, Kershaw, & Oot, 2010).

Measure	TREC 2008	Verizon Team A	Verizon Team B
Precision	21.0%	19.7%	18.3%
Recall	55.5%	48.8%	53.9%

Some attorneys have tried to reduce this burden by employing search to reduce the number of documents that must be considered by human reviewers. With enough effort and the right search tools, this can be an effective approach, but it is very challenging.

> *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, 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 to tread.*– Judge John Facciola, *United States v. O'Keefe*, No. 06-249 (D.D.C. Feb. 18, 2008).

In the Kleen Products matter (Kleen Products, LLC v. Packaging Corporation of America, No. 10 C 5711 (Northern District of Illinois, Eastern Division, Feb. 21, 2012), one defendant spent 1,400 hours coming up with keyword search terms. We do not yet have a good measure of how successful these search terms were, but they clearly required a great deal of effort. Without that kind of effort, search terms are more like a game of "Go Fish," as Ralph Losey has pointed out, where the parties negotiate over search terms with little direct knowledge of the terms' usefulness. For example, one of the search terms used in the Kleen matter returned about 2/3 of all of the documents in the test set.

So, although keyword searching can be an effective tool under the right conditions, achieving that level of effectiveness appears to require a substantial amount of time, effort, and expense. Without this extended effort, keyword searching tends to be rather ineffective, missing as many as 4 out of 5 responsive documents (for example, in the Blair and Maron, 1985 study). Predictive coding, on the other hand, requires substantially less effort and can be shown to achieve high levels of accuracy for much lower cost.

Predictive coding in theory

The general idea behind predictive coding is to find documents that are similar to those that have been classified by an authoritative source to be responsive or non-responsive. The computer does not insert its own judgments about the responsiveness of documents, but seeks to duplicate the decisions made by the authoritative source. It should be clear, then, that the quality of this authoritative source is critical. Poor training examples will lead to poor results.

There are at least nine technologies in common use today to support predictive coding. Different service providers offer varying combinations of these technologies. Here is a list of these nine technologies and a brief summary of what they contribute to predictive coding.

Predictive coding technologies

1. Latent Semantic Analysis. A mathematical approach that seeks to summarize the meaning of words by looking at the documents that share those words. LSA builds up a mathematical model of how words are related to documents and lets users take advantage of these computed relations to categorize documents.

2. Probabilistic Latent Semantic Analysis. A second mathematical approach that seeks to summarize the meaning of words by looking at the documents that share those words. PLSA builds up a mathematical model of how words are related to documents and lets users take advantage of these computed relations to categorize documents.

3. Support Vector Machine. A mathematical approach that seeks to find a line that separates responsive from non-responsive documents so that, ideally, all of the responsive documents are on one side of the line and all of the non-responsive ones are on the other side.

4. Nearest Neighbor Classifier. A classification system that categorizes documents by finding an already classified example that is very similar (near) to the document being considered. It gives the new document the same category as the most similar trained example.

5. Active Learning. An iterative process that presents for reviewer judgment those documents that are most likely to be misclassified. In conjunction with Support Vector Machines, it presents those documents that are closest to the current position of the separating line. The line is moved if any of the presented documents has been misclassified.

6. Language Modeling. A mathematical approach that seeks to summarize the meaning of words by looking at how they are used in the set of documents. Language modeling in predictive coding builds a model for word occurrence in the responsive and in the non-responsive documents and classifies documents according to the model that best accounts for the words in a document being considered.

7. Relevance Feedback. A computational model that adjusts the criteria for implicitly identifying responsive documents following feedback by a knowledgeable user as to which documents are relevant and which are not.

8. Linguistic Analysis. Linguists examine responsive and non-responsive documents to derive classification rules that maximize the correct classification of documents.

9. Naïve Bayesian Classifier. A system that examines the probability that each word in a new document came from the word distribution derived from trained responsive documents or from trained non-responsive documents. The system is naïve in the sense that it assumes that all words are independent of one another.

All of these approaches involve machine learning, except, typically, Linguistic Analysis (which may or may not include machine learning components). A computational process extracts pertinent information from example documents and builds a mathematical model that allows responsive and non-responsive documents to be distinguished from one another based on the text that they contain.

The accuracy of these systems will depend on the specifics of the implementation and on the quality of the training set used. They may also differ in the amount and type of training that must be conducted, including the level of effort. Other differences among these technologies are beyond the scope of the present paper.

In general, these systems work by extracting “features” from the example documents. Usually these features are words, though they can be word combinations, or mathematical values related to groups of words. The computer learns which features are related to documents in each category, and which distinguish between the categories.

When a new document is presented for classification, the computer compares the features of that document with the features known to distinguish the categories and then assigns the new document to the appropriate category based on its features.

Predictive coding in practice

Actually using predictive coding can be relatively simple in practice. The main effort placed on the producing party is to provide an authoritative reviewer or team of reviewers whose opinion matters. The predictive coding system will depend on the judgments of this authoritative reviewer or expert to determine which documents should be designated responsive and which should be considered non-responsive.

Some systems start with a seed set of example documents. There are several methods for generating that set, including keyword and concept searching, interviews, and others. Whether starting with a seed set or not, most systems then present repeated samples of documents to the authoritative reviewer(s) to classify. These samples may be presented randomly, or they may be selected specifically by the predictive coding algorithm (e.g., active learning). After each sample, the predictive coding system then adjusts its internal mathematical models and presents another set of documents to be reviewed.

Each system has its own methods for knowing when to stop. In addition, legal judgment must figure into this stopping criterion. Some cases merit more rigorous review than others. Some cases justify more effort than others. After a certain amount of training, each additional iteration of document review typically offers diminishing returns. Additional training examples each offer smaller and smaller improvements in the ultimate accuracy of the system. It is largely a legal judgment when it is appropriate to quit providing more training examples. Predictive coding systems differ in how easy it is to make this assessment with respect the current level of accuracy.

Predictive coding systems also differ with respect to the product that comes out of the process. Some of the systems score the documents. Documents with higher scores are those that are predicted to be either more responsive or more likely to be responsive. If the documents are scored, then legal judgment must be used to place the cutoff score – the minimum score, above which the documents will be considered responsive, and produced, and below which, the documents will not be considered responsive and so not produced.

Other predictive coding systems do not score the documents, but simply categorize them into the appropriate categories, taking their cues from the judgment of the authoritative reviewer. If that reviewer is generous and classifies many documents as responsive, the predictive coding system will learn to be generous. If that reviewer is stingy, the computer will learn to be stingy. In these systems the judgment of how to distinguish responsive from non-responsive documents is implicit in the document judgments made, rather than chosen after the fact according to some cutoff score.

What documents should be used for predictive coding?

Predictive coding is designed to categorize documents, typically to separate responsive from non-responsive documents. On the basis of a few categorized documents, even very large sets consisting of millions of documents can be effectively classified. However, many predictive coding service providers charge either by the document or by the gigabyte, so some users try to reduce the volume of documents that are submitted for predictive coding in an attempt to reduce the cost.

The predictive coding algorithms do not care where the documents come from. Predictive coding will learn to make judgments about whatever documents it is trained on and will extend those judgments to whatever documents are in its prediction set. The predictive coding system has nothing to say about any documents not available to it during its training.

Some users, for example, like to use keyword searches to reduce the volume of documents prior to predictive coding. Unless special effort is exerted, however, these keyword searches are subject to all of the known limitations of keyword searching. Typically, keyword searching as a culling tool misses as many as 80% of the responsive documents. Using keyword search to pre-cull documents is using one of the least effective of the CAR methods and therefore limiting the ultimate accuracy of predictive coding. Predictive coding can only work on the few documents that were left after the keyword culling. Predictive coding cannot work to identify documents that were never presented to it, meaning that it cannot be more accurate, relative to the whole data set, than the culling that was used to select the documents that were presented to it.

Moreover, unless one measures explicitly the effectiveness of the keyword culling, it is impossible to know how effective the keyword cull has been. No measure of predictive coding conducted on the post-culling set can tell us about the effectiveness of the keyword cull. It will look like the predictive coding did a good job because it was effective on the documents that were presented to it, but it cannot do anything with the documents that were never presented to it. If the initial culling does not include a large percentage of the responsive documents, predictive coding can do nothing to remedy that situation.

The recommendation against keyword culling does not mean that nothing can be done to reduce the volume of documents before predictive coding. You do not have to check your legal judgment at the door when preparing documents for predictive coding. The same kind of legal judgment that would go into any kind of review can be applied to this selection process. Don't submit for predictive coding files that you know with a high level of confidence are going to be useless. There is no particular value to including documents that can be accurately identified as useless. Predictive coding is not a substitute for legal insight, it is a way to amplify that intelligence.

Document volumes can be safely reduced by the judicious use of date range and custodians to select documents. Excluding system files and other files that do not contain user information also makes a lot of sense. Frequently, certain email domains can be safely eliminated. For example, in many cases, emails to or from domains, such as amazon.com or travelocity.com can be safely excluded. The same kind of legal intelligence that went into selecting document collections before keyword culling or predictive coding came into widespread use can be effective in matters involving predictive coding.

Training predictive coding

There are several ways that a predictive coding system can be trained. One way is to provide a seed set – which is a set of pre-classified documents. Sometimes seed sets are called judgmental samples, but this term has a different meaning in statistics and, I think, is misleading when applied to the creation of the seed set. The seed set may contain only responsive documents or it may contain designated

responsive and designated non-responsive documents along with their appropriate classifications. Systems that use a seed set analyze the content of these documents and begin training of the predictive coding system based on them.

Other systems use a random sample for training. Rather than spending the time to create a seed set, these systems randomly choose documents and present them to the expert for classification. Random sampling provides an unbiased set of documents, which when labeled by an authoritative reviewer provide the training that a system needs. Users, then, do not need a strategy to find responsive documents, they need only to recognize responsive documents when they see them. Some systems allow one to use a combination of a pre-planned seed set and random sampling. Both approaches to creating the training sample require legal judgment concerning the relevance of the documents.

Measurement

One of the side effects of the growing interest in predictive coding is the recognition that the efficacy of eDiscovery can be measured. It is possible to assess that a reasonable search has been conducted. A detailed discussion of measurement is beyond the scope of this article, but there are several measures that can be used, including Agreement, Precision, Recall, and Elusion. All of these measures assess the ability of the predictive coding system to match authoritative classifications like those used to train the system. These measures are derived from a simple contingency table (sometimes called a confusion matrix), that compares the true and correct document classifications (as determined by the authoritative expert) with those predicted by the coding system. Accuracy is the degree to which the computer correctly classifies all the responsive documents (Recall) and only (Precision) the responsive documents as responsive.

Table 2. Contingency Table for Information Retrieval

		Predicted		
		Responsive	Non-responsive	
True	Responsive	A	B	E
	Non-responsive	C	D	F
		G	H	J

- Cell A in this table, for example, shows how often (the frequency) the process correctly predicted that a document was responsive – how often the computer and the expert agreed that a document was responsive.
- Cell D shows how often the process correctly predicted that a document was non-responsive – how often the computer and the expert agreed that a document was non-responsive.
- Cell B shows how often the process missed a responsive document – how often the computer categorized a document as non-responsive when the expert categorized it as responsive.
- Cell C shows how often the coding system called a document responsive when it was not truly responsive.

- Values E and F are the row totals (how often the expert classified documents as responsive or non-responsive respectively).
- Values G and H are the column totals (how often the computer classified documents as responsive or non-responsive respectively).
- Value J is the total number of documents categorized.
- **Prevalence** or **richness** is the proportion of documents that have been found by the expert to be responsive (E / J).
- **Agreement** is the proportion of documents for which the computer and the expert categorization matched ((A + D) / J). Higher agreement corresponds to higher accuracy.
- **Precision** is the proportion of truly responsive documents relative to the total number of documents classified by the system as responsive. Precision refers to the selectivity of the coding process (A / G).
- **Recall** measures the degree to which the predictive coding process finds all of the responsive document (A / E).
- **Elusion** measures the degree to which the system removes all of the responsive documents (B / H). Accuracy is higher when Elusion is lower.

A system that is effective at identifying responsive documents will misclassify very few documents as non-responsive, when they should have been classified as responsive and misclassify very documents as responsive when they should have been classified as non-responsive.

Sampling

Building a contingency table, like that shown in Table 1, requires us to know the true classification of each document. Unfortunately, we do not know the true classification of all of the documents in the collection. If we did, then there would be no need to go through predictive coding because we would already have the answer we are seeking. Predictive coding is designed to reduce the effort needed to separate responsive from non-responsive documents, so it would be unreasonable and counterproductive to also have to review every document in a definitive way.

It is impractical, therefore, to compare the accuracy of our process directly against all of the documents in the collection. We can, however, base our assessment on a representative sample of the documents. With a representative sample, any measurement of the properties of the sample can be extended to the collection as a whole, in the same way that pre-election polling can be used to predict the outcome of an election.

Random sampling is the best way to get a representative sample. Each document in the collection has an equal chance of being selected for the random sample, so any properties we measure of the documents in this sample are likely to be similar to the properties of the collection as a whole. The

proportion of voters in our sample who say that they will vote for candidate A is likely to be similar to the proportion of voters who actually will vote for candidate A.

A random sample can be substantially smaller than the collection as a whole. The accuracy of the sample – that is, the degree to which it approximates the collection – is dependent on the size of the sample, not on the size of the population or on the percentage that the sample represents.

The precision of the sample is measured for a particular confidence level by its confidence interval. The confidence interval is the "margin of error." It is the range of values estimated from the sample that is likely to include the true value of the population from which you have sampled.

For example, if an election poll claims a margin of error of plus or minus 5%, then that means that if the election were held today, the true proportion of voters supporting each candidate would likely be within 5% of the proportion of voters found in the sample. Analogously, if 12% of the documents in a sample are found to be responsive, and the sample size is sufficient for a margin of error or confidence interval of plus or minus 5%, then it means that true proportion of responsive documents in the whole collection is likely to be with 5% of our sample estimate (7% - 17%). How likely these estimates are is given by the confidence level.

A confidence level of 95% means that the true score will fall within the confidence interval 95% of the time. 95 samples out of 100 will find that the true population score falls within the confidence interval or margin of error. One chooses a confidence level (typically 95%) and then measures the confidence interval. A random sample of 400 documents (or voters) has a margin of error or confidence interval of about plus or minus 5%.

Neither the confidence level nor the confidence interval tells you how accurate your review process was or how much confidence you should have in it. Rather, they tell you the likely accuracy of your sample compared to the population or collection as a whole. Accuracy is measured by Precision, Recall, Elusion, and other measure, not by the confidence level. You do not get more accurate predictive coding by increasing you confidence level or the sample size used to measure the process.

When to use predictive coding

Most predictive coding systems require text. Predictive coding generally does not work on non-text documents such as blueprints, CAD drawings, photographs, videos, audio recordings, and so forth, unless they are converted first to text. If you have text documents, then there are five questions you can ask to help you decide whether a matter is appropriate for predictive coding.

- Do you want to find as many of the responsive documents as possible?
- Do you want to review as few of the non-responsive documents as possible?
- Do you want to identify potentially responsive documents as quickly as possible?
- Do you want to minimize the cost of review?
- Do you want to reduce the time needed to review documents?

If the answer to at least one of these five questions is yes, then there is one more question to consider.

- Does your collection contain more than about 5,000 text documents?

Predictive coding does not require a large set of documents, but it's value tends to grow disproportionately as the size of the document collection grows, because the effort typically required to train a system does not grow or does not grow as quickly as the size of the document collection increases. Small collections can require almost the same level of training effort as large collections do.

Share training sets

Of the cases that have been reported on disputes over predictive coding, most of them have involved sharing at least some non-responsive documents with the receiving party. For some attorneys, this is a barrier to using predictive coding. They argue that nothing in the rules requires one to turn over non-responsive documents (see Are Corporations Ready To Be Transparent And Share Irrelevant Documents With Opposing Counsel To Obtain Substantial Cost Savings Through The Use Of Predictive Coding?).

No aspect of predictive coding requires one to share either the seed set or the training set. In fact, turning over seed or training set documents tells the receiving side nothing about the effectiveness of the predictive coding system. They are not indicative of the efficacy of the predictive coding system. They were identified by the attorneys training the system, not by the system itself. They are indicators of what the attorneys training the set thought was important. The responsive ones, provided that they are not privileged, will be turned over any way. The non-responsive training documents tell us nothing about the efficacy of the system because how they were classified was determined as the input to the system, not its output.

Examining these documents can be useful only if one assume that the predictive coding system accurately translated this information into effectively separating responsive from non-responsive documents. If we did not already believe that the system worked, then knowing how it was trained is useless. For example, if I concocted a system that essentially threw the documents down the stairs to score them for predictive coding, it would be of no value whatsoever to know which documents I read before I threw them.

Viewing the seed or training documents allows the receiving party to influence the predictive coding process, but cannot assess its accuracy. I'll leave it to lawyers to decide when this sharing is appropriate, but it is not technically necessary. It is about trust and control, not about predictive coding, the technology, or any other form of review. The fact that it has appeared in a number of predictive coding protocols is a byproduct, I think, of resolving trust issues, not an essential part of predictive coding or its evaluation.

A predictive coding protocol

The following is an outline of a basic, effective predictive coding protocol. It addresses the technological issues involved in using predictive coding, while recognizing that there will also be legal / strategic issues that must be considered. This protocol is only one of many that may be appropriate to a particular situation.

1. **Meet and Confer**. The parties meet to determine the parameters of eDiscovery, including preservation, collection, selected custodians, time ranges, topics, concepts, and other pertinent issues. Repeat as necessary as the case evolves. Although limiting the documents to be considered by date ranges and custodian makes some sense, it may not be advisable to try to limit the documents by keywords, because of the difficulty in guessing the right keywords.

2. **Exploratory Data Analysis.** The producing party, recognizing its obligation to produce responsive documents, begins document analysis. The technology does not require sharing training documents or seed sets with the receiving party. Sharing these documents assumes that the technology works as expected, but that the producing party requires "guidance" to identify the correct documents to be produced. There are many ways to provide this guidance without having to share non-responsive documents. Legal and strategic concerns should govern whether these documents should be shared, it is not an intrinsic part of the predictive coding process.

3. **Estimate Prevalence**. The producing party samples the document set to get an estimate of prevalence. How rare / frequent are responsive documents? Prevalence is important because special steps may be needed to make predictive coding training efficient if responsive documents are extremely rare (e.g., less than 1% of the documents are responsive). Prevalence sampling may be part of the process of training the predictive coding system.

4. **Predictive Coding Training**. The producing party begins predictive coding training. The producing party may report accuracy statistics along the way, or, if training is brief, at the end of training. Not all predictive coding tools yield meaningful statistics during the course of training. Some require small enough amounts of training that reporting in the course of training may be too disruptive.

5. **Predictive Coding**. When predictive coding training is complete, the remaining documents in the collection are coded by the computer.

6. **Evaluation**. A sample of documents is reviewed by the producing party for responsiveness to measure the effectiveness of the predictive coding. There are several different ways to perform the sampling. The exact sampling method should be agreed to by the parties. Use the smallest sample necessary to achieve the desired confidence interval. Choose a confidence interval that is consequential. A confidence interval of, say, plus or minus 5% is usually sufficient. Keep in mind that values in the center of the confidence interval are much more likely than values at the edges of the confidence interval.

   There are several ways that an evaluation can be conducted following predictive coding.

   a. After the documents have been categorized by the system, review can be continued on newly generated random samples of documents. That is, the same expert continues to evaluate random samples of documents until a sample size the parties agree is adequate has been obtained. The system's efficacy on this sample is taken as a measure of its performance.

b. A separate random sample of documents designated by the predictive coding system as non-responsive can be evaluated to compute the Elusion measure. Elusion is the proportion of documents classified as putatively non-responsive that should have been classified as responsive. Ideally, only a small proportion of the documents in the putatively non-responsive set will be found to be responsive. In practice, the proportion of responsive documents in the putatively non-responsive set should be only a small fraction of the prevalence of responsive documents. Elusion, therefore, needs to be compared to the original estimate of responsive document prevalence. The size of this sample will depend on the required confidence level and confidence interval.

c. A set of putatively responsive and a set of putatively non-responsive documents could be evaluated. Ideally, all of the putatively responsive documents will, in fact, be found to be responsive and none of the putatively non-responsive documents will, in fact, be found to be responsive. In practice, most of the putatively responsive documents should be found to be responsive and few of the putatively non-responsive documents should be found to be responsive. This information can be combined with other available information to give an estimate of Precision and Recall.

7. **Privilege Review**. The documents designated responsive by the predictive coding system are reviewed by the producing party for privilege. The privileged documents in this set may be withheld, and the non-privileged ones produced.

8. **Dispute Resolution**. If there are disagreements about the produced documents that cannot be resolved by conferring, then a special master may be appointed to examine a sample of the documents and their computer-generated coding.

Conclusion

Predictive coding is a powerful tool in the arsenal of eDiscovery. When used correctly, it can substantially reduce the volume of documents that must be considered for production or for evaluation of responsiveness. Predictive coding is not a substitute for legal judgment, but an amplifier of it, bringing higher levels of consistency, efficacy, accuracy, and efficiency. For the producing party, it promises to return more focused documents more economically. For the requesting party, it promises to return more complete and focused documents in a shorter period of time. In many cases, predictive coding provides an all-around win, moving litigation to the merits of the case, addressing Rule 1 of the Federal Rules of Civil Procedure "to secure the just, speedy, and inexpensive determination of every action and proceeding."

Glossary

Active learning – a form of supervised machine learning that presents for review or human categorization the documents with the highest current uncertainty, those documents that will be most informative about how to update the learning process.

Bayesian categorizer—an information retrieval tool that computes the probability that a document is a member of a category from the probability that each word is indicative of each category. These estimates are derived from example documents. Uses the probability of each word given each category to compute the probability of each category given each word. Also called a naïve Bayesian Categorizer.

CAR – Computer assisted review. Any of a number of technologies that use computers to facilitate the review of documents for discovery. See TAR.

Collection – A group of documents. These can be documents gathered for a particular matter or purpose. Information retrieval scientists tend to use several well-known document collections (e.g., RCV1) for testing and comparison purposes.

Confidence interval – the expected range of results. If you drew repeated samples from the same population, you would expect the result to be within the confidence interval about the proportion of times given by the confidence level. For example, in an election poll, the difference in the proportion of people favoring each candidate is described as being within a range of, say, plus or minus 5%. All other things being equal, the smaller the confidence interval, the larger the sample size needs to be. Said another way, the larger the sample size, the smaller the confidence interval.

Confidence level –how often we would achieve a similar result if we repeated the same process many times. If we did the same kind of test from the same population more than once, the confidence level would tell us how often we would get a result that is within a certain range (the confidence interval) of the true value for the population. Most scientific studies employ a minimum confidence level of 0.95, meaning that 95 percent of the time when you repeated the experiment you would find a similar result. The higher the confidence level the larger the sample size that is required. Technically, it is the proportion of times when the true population value would be included in within the confidence interval.

Contingency Table – a table of the four response states in a categorization task. The rows of the table may correspond to the correct or true category values and the columns may correspond to the choices made by system. For example, the top row may be the truly positive category (e.g. truly responsive documents) and the second row may be the truly negative category (e.g., truly non-responsive documents). The columns then represent the positive decisions made by the system (e.g., putatively responsive) and the negative decisions made by the system (e.g., putatively non-responsive). The entries in these cells are the counts of documents corresponding to each response state (e.g., true positives, false negatives, false positives, true negatives). Contingency tables are often displayed along with the totals for each row and for each column. Sometimes the rows and columns are reversed, so the columns reflect the true values and the rows reflect the choices.

Elusion – an information retrieval measure of the proportion of responsive documents that have been missed. Most often used as a quality assurance measure in which a sample of non-retrieved documents is evaluated to determine whether a review has met reasonable criteria for completeness.

Judgmental sampling – a sampling process where the objects are selected on the basis of some person's judgments about their relative importance rather than on a random basis. Judgmental sampling sometimes refers to the use of a seed set or preselected documents used to train predictive coding systems. Unlike random samples, judgmental samples are not typically representative of the collection or population from which they are drawn. It is not possible to extrapolate from the characteristics of a judgmental sample to the characteristics of the population or collection.

Language modeling—computing a model of the relationships among words in a collection. Language modeling is used in speech recognition to predict what the next word will be based on the pattern of preceding words. Language modeling is used in information retrieval and predictive coding to represent the meaning of words in the context of other words in a document or paragraph.

Latent Semantic Analysis—(LSA) a statistical method for finding the underlying dimensions of correlated terms. For example, words like law, lawyer, attorney, lawsuit, etc. All share some meaning. The presence of any one of them in a document could be recognized as indicating something consistent about the topic of the document. Latent Semantic Analysis uses statistics to allow the system to exploit these correlations for concept searching and clustering.

Latent Semantic Indexing—(LSI) the use of latent semantic analysis to index a collection of documents.

Machine learning—a branch of computer science that deals with designing computer programs to extract information from examples. For example, properties that distinguish between responsive and nonresponsive documents may be extracted from example documents in each category. The goal is to predict the correct category for future untagged examples based on the knowledge extracted from the previously classified examples. Example approaches include neural networks, support vector machines, Bayesian classifiers and others.

Nearest neighbor classification—a statistical procedure that classifies objects, such as documents, according to the most similar item that has already been assigned a category label. This approach uses a set of labeled examples to classify subsequent unlabeled items, by choosing the category assigned to the most similar labeled example (its nearest neighbor) or examples. K-nearest neighbor classification uses the k most similar classified objects to determine the classification of an unknown object.

Population – the universe of things about which we are trying to infer with our samples. For example, the population may be the set of documents that we want to classify as putatively responsive or putatively non-responsive. The group from which we pull our samples. Also called the sampling frame.

Precision – the proportion of retrieved documents that are responsive. See also recall.

Predictive coding – a group of machine learning technologies that predict which documents are and are not responsive based on the decisions applied by a subject matter expert to a small sample of documents.

Prevalence – the richness or proportion of responsive documents in a collection. More broadly, the prevalence refers to the proportion of one kind of item in a population of items.

Probabilistic Latent Semantic Analysis—a statistical procedure for finding the underlying dimensions of correlated terms. Like Latent Semantic Analysis, this procedure attempts to capture the meaning shared by multiple terms to provide a concept search capability. It differs some from LSA in that it involves a different statistical model. Also called probabilistic latent semantic indexing.

Random – unpredictable. Random selection means that each item has an equal chance of being selected and there is no systematic bias to select one item rather than another. Coin flips are random. Knowing that one coin flip came up heads does not change the likelihood that the next coin flip will come up heads (these coin flips are said to be independent).

Random sampling—the statistical process of choosing objects randomly, meaning that each object has an equal chance of being selected. Random sampling can be used to train predictive coding systems and to evaluate their efficacy.

Recall –the proportion of responsive documents in the entire collection that have been retrieved.

Relevance feedback—a class of machine learning techniques where users indicate the relevance of items that have been retrieved for them and the machine learns thereby to improve the quality of its recommendations.

Richness – the proportion of responsive documents in a collection.

Sampling – the process of selecting a subset of items from a population and inferring from the characteristics of the sample what the characteristics of the population are likely to be. Often refers to a simple random sample, in which each item in the population has an equal chance of being selected in the sample.

Seed set – a collection of pre-categorized documents that is used as the initial training for a predictive coding system.

Support vector machine (SVM) – a machine-learning approach used for categorizing data. The goal of the SVM is to learn the boundaries that separate two or more classes of objects. Given a set of already categorized training examples, an SVM training algorithm identifies the differences between the examples of each training category and can then apply similar criteria to distinguishing future examples.

TAR – Technology Assisted Review. Any of a number of technologies that use technology, usually computer technology, to facilitate the review of documents for discovery. See CAR.

Cited

Blair D. C. & Maron, M. E. (1985). “An evaluation of retrieval effectiveness for a full-text document-retrieval system,” *Communications of the ACM*, 28, 289-299.

Roitblat, H. L., Kershaw, A. & Oot, P. (2010). Document Categorization in Legal Electronic Discovery: Computer Classification vs. Manual Review. *Journal of the American Society for Information Science and Technology*, 61(1):70-80.

