Break up the Family: Protocols for Efficient Recall-Oriented Retrieval Under Legally-Necessitated Dual Constraints

Jeremy Pickens
Catalyst Repository Systems
Denver, USA
jpickens@catalystsecure.com

Thomas C. Gricks, Esq.
Catalyst Repository Systems
Denver, USA
tgricks@catalystsecure.com

Andrew Bye
Catalyst Repository Systems
Denver, USA
abye@catalystsecure.com

Abstract—In the legal domain, the primary objective of eDiscovery is to retrieve, from within a document population, as many individual relevant documents as possible given the expenditure of a reasonable review effort. The production of relevant documents to opposing counsel, however, is typically made on a family basis (an email and all its attachments), such that the entire family is produced as a collective unit. Counsel's ethical responsibility to prevent the disclosure of privileged information necessitates review of every document in every family being produced. Consequently, it is standard industry practice to batch entire families for review together. This work examines the overall review ordering and efficiency of alternative techniques for managing review leading to the production of relevant document families while fully protecting privilege. Batching, document sequencing, and therefore, training of the core supervised machine learning (AI) algorithm differ among each protocol, and lead to different overall efficiencies. Our empirical results support two conclusions. First, break up the family. In every instance, broken family retrieval protocols are more efficient than full family protocols. Second, a carefully designed and implemented dual phase workflow incorporating an initial, expedited relevance review can be more efficient than a single phase workflow, even if some documents are reviewed twice.

Index Terms—eDiscovery, legal information retrieval, document review protocols, privileged data protection

I. Introduction

Electronic discovery (eDiscovery) encompasses one of the principal information retrieval activities in the legal sector. But eDiscovery suffers from certain peculiarities that render straightforward document ranking and review for a single objective untenable. Certainly the primary objective is "to find as nearly all of the relevant documents in a collection as possible, with reasonable effort" [3]. But the process does not end at relevance; it continues through the comprehensive review of document families, including non-relevant documents within those families.[1] This is typically a necessary step in the ultimate production of documents in litigation.

The need to review document families is a function of two criteria: (1) the obligation to produce documents as they are maintained in the ordinary course, and (2) the responsibility to prevent the disclosure of privileged documents. In an eDiscovery matter, relevance is prescribed by a request for production, for which the responding party is required to conduct a review and produce relevant, non-privileged documents. Since electronic documents are typically maintained as a family unit (an email is stored together with all attachments), the typical practice is to request, and produce, the entire family. Moreover, it is ethically incumbent upon counsel to ensure that none of the documents in the families being produced contain privileged information. The practical result of these dual (relevance and privilege) constraints is that the scope of a litigation production review generally extends beyond identification of relevant documents to the eyes-on review of every document in every family that is being produced.[2]

A review process involving supervised machine learning algorithms trained on the explicit document relevance judgments made by the lawyers engaged in the review is known within the legal industry as technology-assisted review, or TAR. There are several ways to manage a document family review, and each process will have a different impact on TAR efficiency. The typical practice is to batch documents for review together as families. Documents are still coded on an individual level, but every document suggested by the TAR algorithm is presented to a reviewer together with all family members. As a result, (1) there reviewer has to fully code both relevant and non-relevant families and (2) the TAR algorithm is trained using all the documents in a family (including non-relevant families) rather than only the individual documents selected by the TAR algorithm.

In order to avoid reviewing entirely non-relevant families, this paper examines a number of different techniques

[1] A family is defined as an email and all of its attachments (Word, PDF, Excel docs, et cetera).

[2] There are procedural protections that may obviate the need to review every single production document for privilege. For example, Federal Rule of Civil Procedure 502(d) limits the circumstances under which production of a privileged document will effect a waiver of the privilege. However, privileged information, once disclosed, is known to the adversary regardless of waiver– the proverbial bell that cannot be unrung. For that reason, the common practice is to review every document that is being produced for privilege.

for batching documents on an individual level, without family members.[3] Family members of only relevant documents can then be reviewed, either contemporaneously or subsequently, which will again impact the training of the algorithm and the order in which the documents are reviewed. In this paper, we empirically evaluate four techniques for managing families in a TAR review (one baseline and three improvements on that baseline) and the impact of each technique on the relationship between recall and number of documents reviewed (effort).

II. Background and Related Work

By far the two most common types of problems studied in Information Retrieval are the retrieval model or learning algorithm, and feature selection to support the algorithm. These problems are often studied in relation to precision-oriented retrieval tasks such navigational, transactional, or informational [2]. Far less common is the study of recall-oriented tasks such as eDiscovery [10]. The length of time spent satisfying a precision-oriented information need typically lasts a few minutes, or at most a few hours over a few multi-day sessions (such as when planning a vacation). In contrast, a single eDiscovery information need commonly lasts weeks, if not months, with ongoing review conducted by multi-person teams occurring eight hours a day, forty hours a week. Thus, recall-oriented retrieval is more concerned with longer term efficiencies than shorter term gains.

While both feature extraction and algorithm selection are important, recall-orientation opens another aspect of systems design that is typically not available in precision-oriented domains: Process. Indeed, [1] notes that "success in using any automated method of [eDiscovery] technology will be enhanced by a well thought out process with substantial human input on the front end". Recognizing this, processes for managing recall-oriented tasks, specifically as they relate to the peculiarities of the dual constraints of relevance and privilege, are the focus of this paper.

In recent years, the work by Cormack & Grossman [3], [8] is the prime example of the effect of process on recall-oriented retrieval, at least under a single (relevance) constraint. All aspects of the system are fixed; both supervised learning algorithm and feature extraction do not change. Instead, the effect on recall efficiency of different training document selection protocols is examined. Their conclusion is that a continuous, active (relevance feedback) approach, or CAL, is the most efficient process among those studied.

Other process-based research includes how to initiate (seed) the task [4], [7], [12]. These works examine whether one needs a large sample (either random or user-specified) in order to achieve high recall in the continuous process, or whether high recall can be achieved starting with only one relevant document or search query. [5] examines the effect of process on aspectual recall, i.e. on whether certain processes are able to achieve high recall on all (not just some) facets when reasonably high generic relevance recall levels have been hit.

[3]It has been suggested that relevance decisions require the context of all attendant family members. In reality, the vast majority of relevance decisions can be made regardless of family context, and modern eDiscovery tools make family members accessible in an ad hoc manner.

III. Experiment Setup

A. Test Collections

For our experiments there are a total of eight eDiscovery matters. The data and ground truth for these eight matters is from actual litigation. All families were reviewed in full in each of these matters, so ground truth is complete. Table I shows the basic statistics for each of the matters. The collection size is given in the first column, ranging from a very small 4,706 documents to more typical collections in the hundreds of thousands. The document count and richness of two different views of relevance are also given: Actual and Producible. Actual is the number of individual documents that are tagged as relevant, and Producible adds non-relevant family members to arrive at the total number of documents that need to be assessed for privilege before they are produced to opposing counsel.

B. Simulation Process

The basic protocol utilized in this paper is the continuous active learning (CAL) protocol of [3]. First, a set of starting, or seed, documents are chosen. These documents (and only these documents) are then labeled for relevance using the ground truth and fed into a supervised learning algorithm. The learning algorithm ranks the as-yet simulatedly unreviewed documents by their likelihood of being relevant. The next documents to review are chosen from the top of this ranking, the ground truth labels are applied to these documents and the cycle is repeated until the entire collection has been simulatedly reviewed.

In these experiments, every simulation variant on a given matter is seeded with the same 100 document random sample. The batch update rate k, i.e. the number of documents and/or families (depending on the protocol) selected for review before the supervised machine learning algorithm is retrained, is universally set to 100. The features used in the supervised learning algorithm are word n-grams. The learning algorithm used in these experiments is proprietary, but competitive with state-of-the-art results on other recall-oriented tasks [9]. Indeed, everything is held constant except for the manner in which family selection is integrated into the CAL protocol. Thus, just as [3] only varied the protocol, and not the feature selection or algorithm, so to is this the approach that we have taken.

The CAL protocol was developed for the single (relevance-only) constraint scenario and our main contribution is to adapt it to legally-necessitated, more task-

	Collection	Actual		Producible		Seeds		
	Size	Relevant	Richness	Relevant	Richness	Total	Relevant	Richness
Matter 1	28856	1608	5.6%	2331	8.1%	100	5	5.0%
Matter 2	118137	7012	5.9%	10519	8.9%	100	5	5.0%
Matter 3	44761	4002	8.9%	4705	10.5%	100	6	6.0%
Matter 4	4706	442	9.4%	549	11.7%	100	7	7.0%
Matter 5	176251	21419	12.2%	25254	14.3%	100	18	18.0%
Matter 6	196471	32034	16.3%	55210	28.1%	100	16	16.0%
Matter 7	131282	32261	24.6%	42454	32.3%	100	25	25.0%
Matter 8	235325	73408	31.2%	93164	39.6%	100	39	39.0%

TABLE I: Matter collection sizes and relevance statistics for both actual and producible (family-inclusive) documents. Seed counts and sample richness also shown.

realistic dual (relevance and privilege) constraints by altering the manner in which document selection from the top of a supervised learning ranking is done. In the following sections we describe in detail the manner in which this is done.

IV. Full vs Broken Family Protocols

In the first experiment, we look at the effect of reviewing in full family order two various broken family orderings. In each of these protocols, when a document is reviewed, it is reviewed wholly for both relevance and privilege.

A. Full Family (FF-CAL) Protocol

The Full Family protocol is state-of-the-art industry practice so it will be the first baseline. In this protocol, the moment any document gets surfaced by the retrieval (machine learning ranking) process, the remaining documents in the family are brought immediately into the review queue, whether or not they are highly ranked. This is illustrated by the SelecdtFF(·) function inside of the Algorithm 1 simulation. In practice, a system implementing FF-CAL would cease the ranking and selection loop inside of the Main(·) function at the point that a high recall level is hit. For our experiments, we continue ranking until the entire collection has been (simulatedly) reviewed to show overall efficiency.

Experience shows that lawyers engaged eDiscovery are very seldomly willing to entertain approaches other than full family; industry attachment to full family review is strong. Indeed, there are a few hypothetical reasons why full family review might already be maximally effective. First, full family review may retrieve documents that turn out to not be relevant, but that have relevant family members that are otherwise difficult to predict (e.g. Excel spreadsheets, documents in a language different than the current predictive model, etc.). Second, because those documents get used as training earlier in the process, they might help retrieve more difficult to predict documents, increasing efficiency even further.

B. Positive Family (PF-CAL) Protocol

The Positive Family protocol is the first broken family process variant to be tested. A positive family review proceeds in normal CAL manner, with documents being retrieved and reviewed in continuously updating relevance-predicted order. When a document is marked non-relevant, the next document in the relevance queue is sent for review. However, if a document is marked relevant, its remaining as-yet unreviewed family members are sent for review. Because of the vagaries of machine learning and unstructured data, it may very well be that, in a particular family, a non-relevant family member was predicted and reviewed before the first relevant family member was found. Thus, when a positive family member is found,

Algorithm 1 Full Family [FF-CAL] Simulation

```
1:  procedure CreateGainCurve(simorder)
2:      plotpoints = [ ]
3:      for d in simorder do
4:          x ←x + 1
5:          if isRelevant(d) then
6:              y ←y + 1
7:          plotpoints ←[plotpoints, (x, y)]
8:      return plotpoints
9:
10: procedure SelectFF(ranking, simorder, k)
11:     selected = [ ]
12:     repeat
13:         seen ←simorder ⋃ selected
14:         d ←d | d = top(ranking) ⋀ d ∉ seen
15:         fam = [d′ | d′ ∈ family(d) ⋀ d′ ∉ seen]
16:         selected ←[selected, fam]
17:     until k iterations have elapsed
18:     return selected
19:
20: procedure Main(initialseeds, k)
21:     simorder ←[initialseeds]
22:     loop
23:         ML.train(simorder)
24:         ranking ←ML.rank
25:         selected = SelectFF(ranking, simorder, k)
26:         simorder ←[simorder, selected]
27:         if size(simorder) = size(collection) then
28:             return createGainCurve(simorder)
```

Algorithm 2 Positive Family [PF-CAL] Simulation

```
procedure CreateGainCurve(simorder)
    Same as in Algorithm 1

procedure SelectPF(ranking, simorder, k)
    selected = [ ]
    repeat
        seen ←simorder ⋃ selected
        d ←d | d = top(ranking) ⋀ d ∉ seen
        if isRelevant(d) then
            fam = [d′ | d′ ∈ family(d) ⋀ d′ ∉ seen]
            selected ←[selected, fam]
        else
            selected ←[selected, d]
    until k iterations have elapsed
    return selected

procedure Main(initialseeds, k)
    Same as in Algorithm 1 except with SelectPF(·) as
            the selection mechanism
```

any previously reviewed non-relevant family members are not added to the review queue. No document is reviewed twice. Algorithm 2 demonstrates this protocol, with the SelectPF(·) function being the primary difference from the FF-CAL protocol.

C. Individual Padded (IP-CAL) Protocol

The Individual Padded protocol begins in the same as would a standard CAL review, with documents being retrieved and reviewed individually with no awareness of family membership. However, once a given recall level is achieved, all remaining unreviewed family members of families with at least one relevant document are added to the review queue, and this set of documents is reviewed linearly in its entirety. The main conceptual difference is that PF-CAL retrieves (and therefore trains on) members of relevant families immediately, whereas IP-CAL retrieves (and therefore trains on) only the highest predicted relevant documents. IP-CAL's post hoc padding process ensures that all producible family members are reviewed, as per the dual requirements of the task, but does not use them for training.

Essentially, the Main(·) loop and the SelectIndiv(·) selection function in Algorithm 3 are the same as in the standard CAL protocol. The post-hoc padding is done via the CreatePaddedGainCurve(·) function. We note that because IP-CAL's final padded result is dependent on the pre-padding stopping point, CreatePaddedGainCurve produces an amalgamation of having stopped at every possible stopping point.

Algorithm 3 Individual Padded [IP-CAL] Simulation

```
procedure CreatePaddedGainCurve(simorder)
    plotpoints = [ ]
    simorder′ = [ ]
    for d in simorder do
        simorder′ ←[simorder′, d]
        padded = [ ]
        for d′ in simorder′ do
            if isRelevant(d′) then
                fam = [d″ | d″ ∈ family(d′) ⋀ d″ ∉ simorder′]
                padded ←[padded, fam]
        x ←size(simorder′) + size(padded)
        y ←relCount(simorder′) + relCount(padded)
        plotpoints ←[plotpoints, (x, y)]
    return plotpoints

procedure SelectIndiv(ranking, simorder, k)
    selected = [ ]
    repeat
        seen ←simorder ⋃ selected
        d ←d | d = top(ranking) ⋀ d ∉ seen
        selected ←[selected, d]
    until k iterations have elapsed
    return selected

procedure Main(initialseeds, k)
    simorder ←[initialseeds]
    loop
        ML.train(simorder)
        ranking ←ML.rank
        selected = SelectIndiv(ranking, simorder, k)
        simorder ←[simorder, selected]
        if size(simorder) = size(collection) then
            return createPaddedGainCurve(simorder)
```

D. Results and Discussion

Results are found in Table II and Figure 1. Table II shows (1) the amount of effort (percentage of the collection) needed to achieve a target recall, (2) absolute percentage reduction in effort of the PF-CAL and IP-CAL protocols over FF-CAL, and (3) residual percentage reduction over FF-CAL. Absolute reduction takes into account all effort to get to a given recall point, whereas residual only counts wasted effort. That is, in a collection with (say) 5,000 relevant documents, at least 3,750 relevant documents need to be reviewed to achieve (say) 75% recall, no matter which protocol is followed. The residual therefore removes those 3,750 documents from the calculation, and only looks at the number of additional documents – the residual – that needed to be reviewed to hit that many relevant documents.

Matter	Recall	Collection % Review Effort to Recall k			% Effort Reduction Over FF			
					Absolute		Residual	
		FF-CAL	PF-CAL	IP-CAL	PF-CAL	IP-CAL	PF-CAL	IP-CAL
Matter 1	75	7.76	7.18	7.21	7.5	7.1	34.4	32.5
Matter 2		11.51	9.16	9.43	20.5	18.1	48.7	43.0
Matter 3		10.36	9.35	9.59	9.7	7.5	40.6	31.3
Matter 4		17.84	13.03	13.94	27.0	21.9	53.0	42.9
Matter 5		23.46	16.62	16.38	29.2	30.2	53.8	55.7
Matter 6		35.29	26.40	26.80	25.2	24.1	62.6	59.7
Matter 7		31.33	28.49	28.39	9.1	9.4	40.2	41.6
Matter 8		37.12	34.92	33.46	5.9	9.8	29.6	49.2
Matter 1	90	11.07	8.78	8.70	20.7	21.4	60.4	62.3
Matter 2		18.77	12.26	19.06	34.7	-1.5	60.5	-2.7
Matter 3		14.24	13.56	12.90	4.8	9.4	14.2	28.0
Matter 4		25.96	19.76	19.00	23.9	26.8	40.1	45.0
Matter 5		39.03	29.09	27.82	25.5	28.7	38.1	42.9
Matter 6		54.80	39.78	45.30	27.4	17.3	50.9	32.2
Matter 7		42.65	37.95	36.57	11.0	14.3	34.7	44.9
Matter 8		51.06	48.19	46.33	5.6	9.3	18.6	30.7

TABLE II: Full Family (FF-CAL) versus Broken Family (PF-CAL, IP-CAL) results at 75% and 90% recall levels. Raw effort (left), absolute and residual percentage reduction in raw effort relative to FF-CAL (right).

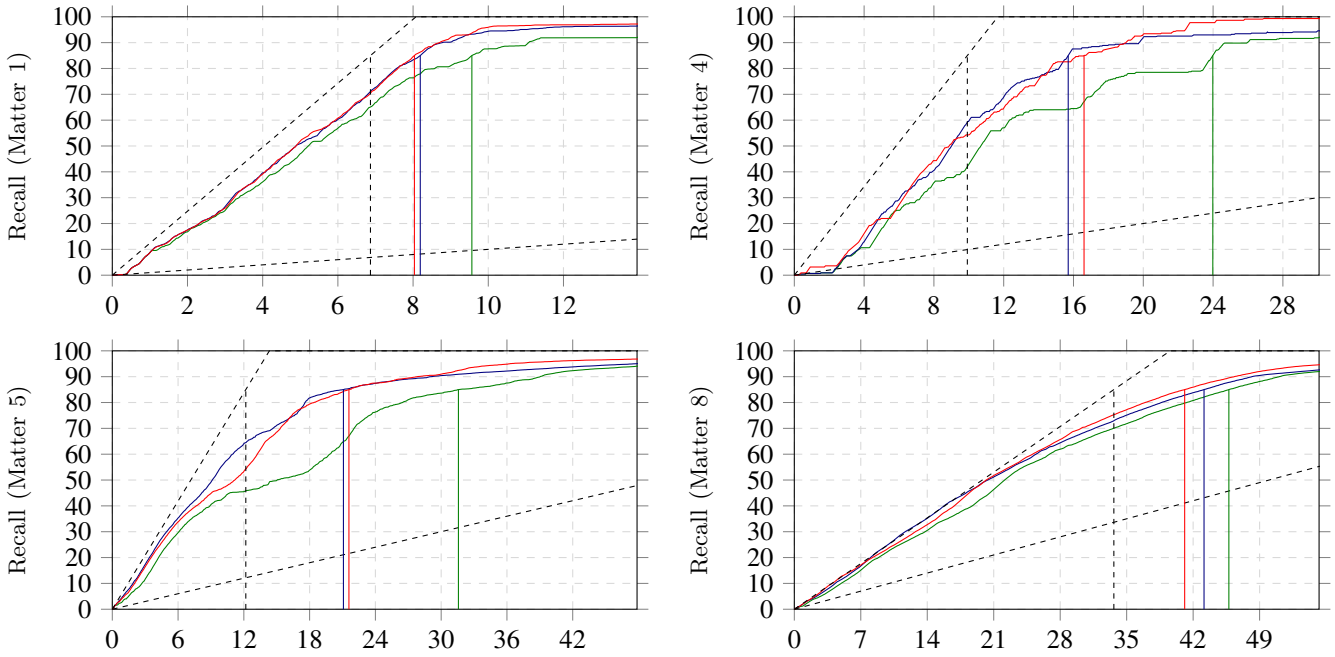

Fig. 1: Document-based gain curves for FF-CAL=Green, PF-CAL=Blue, IP-CAL=Red. Dashed black lines indicate (producible) perfect and linear. X-axis is collection percentage.

For the eDiscovery task, a minimum of 75% recall is typically expected by the courts, though achieving higher recall lowers the risk of being challenged, so Table II gives values at both 75% and 90% recall. Figure 1 shows a few gain curves for these matters so as to get an overall sense of the performance of each protocol, and not just the effort at the 75% and 90% points. Purely in the interest of space, only matters 1, 4, 5, and 8 are shown in Figure 1, which matters were chosen to demonstrate a range of case size, richness, and effectiveness levels. Vertical lines are drawn at 85% recall (happy medium between 75% and 90% recall) to give a visual sense of how much more effort is needed under the various protocols.

Both broken family protocols significantly outperform a full family review and do so at every given recall level except for one case: the 90% recall level for Matter 2. In that case, PF-CAL still beats FF-CAL, but IP-CAL is 1.5% worse in absolute terms (2.7% worse residually), which is a relatively small difference. These results are not only statistically significant (16 for 16, $p < 0.0001$ at 75% and 80% recall, and 15 for 16, p=0.0003 at 90% recall) but the magnitudes of the differences are large enough to have meaningful effect on the time and money it takes to review for production. The recommendation from these results is clear: Break up the family.

The results are less clear when it comes to which broken family protocol, PF-CAL or IP-CAL, is more effective. At 75% recall, PF-CAL bests IP-CAL 5 of 8 times, essentially a coin flip. At 90% recall, IP-CAL bests PF-CAL 6 of 8 times, which is better but with a sign test p-value of only

0.14, i.e. not statistically significant. This opens up the possibility for the design of a third broken family protocol: Phased Review.

V. Joint vs Phased Protocols

Every protocol in the previous section shares one commonality. When a document is reviewed, it is reviewed jointly for both relevance and privilege, and no document is ever reviewed twice. There is an alternative: Some documents may be reviewed twice, but only for relevance the first time.

The intuition that currently guides legal industry practice posits that repeated document review is always less effective. However, two insights allow for the possibility of beating that intuition. First, a protocol which delays padding the review with unreviewed positive family members until the end (IP-CAL) is no worse than a protocol in which positive family members are immediately reviewed and used for training (PF-CAL). Second, relevance only review can be much faster than full privilege review. When reviewing and tagging a document for relevance, the reviewer need only determine if there is something in the document that makes it "about" the legal matter, at which point the remainder of the document does not need to be reviewed. Privileged information, on the other hand, may reside anywhere in the document, so just because the first half of the document does not contain privileged information does not mean that the second half will not. Context is more important, too, and documents are privileged not only for what they contain (sensitive information specifically addressed to one's attorney), but for what they don't contain (communication copies to a third party which might invalidate claims of privilege). Privilege reviews take longer.

A. Phased (PH-CAL) Protocol

We combine these two insights into a new broken family protocol: Phased review, or PH-CAL. Phase one is a standard CAL review with purely individual document level selection, with two twists. First, documents are only reviewed for relevance and never for privilege (or potential privilege), meaning that it is not required that the entire document be examined. Review of a document is cut short the moment anything that makes that document relevant is seen. Second, once a single member of a family is identified as relevant, phase one immediately suppresses the review of all remaining family members. If a document in a family is marked non-relevant (and no other family members have previously been marked relevant), then remaining family members are not suppressed, allowing another potentially relevant family member to be retrieved at a later point. Finally in phase two, every family with at least one relevant member is reviewed in full for both relevance and privilege, including relevant family members already reviewed in phase one, as these have not yet been reviewed for privilege. This protocol effectively generates the producible population very quickly, and the more comprehensive review is limited to that producible population. Suppression reduces the number of documents (and time) that need to be reviewed in phase one, but also reduces available training data, which is why simulations of entire review sequence effectiveness are necessary.

It should be explicitly noted that phased review should not be confused with another occasional legal industry practice: two-stage review. In two stage review, less

Algorithm 4 Phased [PH-CAL] Simulation

```
1:  procedure CreatePhasedGainCurve(simorder, rate)
2:      plotpoints = [ ]
3:      phaseone = [ ]
4:      phasetwo = [ ]
5:      for d in simorder do
6:          phaseone ←[phaseone, d]
7:          if isRelevant(d) then
8:              phasetwo ←[phasetwo, family(d)]
9:          x ←size(phaseone) * rate + size(phasetwo) * 1.0
10:         y ←relCount(phasetwo)
11:         plotpoints ←[plotpoints, (x, y)]
12:     return plotpoints
13:
14: procedure SelectPhaseOne(ranking, simorder, k)
15:     selected = [ ]
16:     suppressed = ∅
17:     for d in simorder do
18:         if isRelevant(d) then
19:             suppressed ←suppressed ⋃ family(d)
20:         else
21:             suppressed ←suppressed ⋃ d
22:     repeat
23:         d ←d | d = top(ranking) ⋀ d ∉ suppressed
24:         if isRelevant(d) then
25:             suppressed ←suppressed ⋃ family(d)
26:         else
27:             suppressed ←suppressed ⋃ d
28:         selected ←[selected, d]
29:     until k iterations have elapsed
30:     return selected
31:
32: procedure Main(initialseeds, k, rate)
33:     simorder ←[initialseeds]
34:     loop
35:         ML.train(simorder)
36:         ranking ←ML.rank
37:         selected = SelectPhaseOne(ranking, simorder, k)
38:         simorder ←[ simorder, selected ]
39:         if size(simorder) = size(collection) then
40:             return createPhasedGainCurve(simorder, rate)
```

expensive contract attorneys do full FF-CAL relevance and (potential) privilege review of a collection in the first stage, followed by more expensive senior attorneys who do a "proofing" review in the second stage. The PH-CAL protocol, like the FF-CAL, PF-CAL, and IP-CAL protocols, all operate strictly within what would be the first stage of such a practice.

B. Results and Discussion

In Section IV results were presented in the form of document-based (x-axis) gain curves. In practice, however, the number of documents reviewed matters much less than the time that it takes to review those documents. Reviewers are paid by the hour, not by the document. When all documents are reviewed for both relevance and privilege this doesn't matter, as document-based gain curves differ only as a scalar to time-based gain curves. However, in a phased review, the time to review a document only for relevance in phase one differs from the time to review for both relevance and privilege in phase two. We therefore switch to time-based gain curves [11], with time on the x-axis rather than document count.

The authors have experience with document review up to five times faster when doing only quick assessments of relevance. Nevertheless, as actual rates are reviewer and matter dependent, rather than assuming that all matters will have a 5x relative review rate, we do a sensitivity analysis, showing the relative efficiencies of various rates. Relevance-only document coding will certainly be more than 1x faster than combined relevance+privilege coding, but may not always achieve 5x speedup, either.

Each of our experimental runs shows the combined time to do both phases of the review, i.e. phase one at a rate from 1x to 5x, and phase two always at a rate of 1x. We use an industry standard rate of an average of one document per minute for the 1x (combined relevance and privileged review) speed. Results are found in Table III and Figure 2. Table III shows exact values as raw savings (in minutes) at 90% recall over both IP-CAL and FF-CAL, as well as percentage differences relative to each baseline (respectively). Figure 2 shows a visual representation of the time-based gain curve for both IP-CAL and the various PH-CAL speeds, with vertical markers at 90% recall. Again in the interest of space, only matters 1, 4, 5, and 8 are shown in Figure 2.

As expected, PH-CAL is not enough to be more efficient than IP-CAL at a 1x phase one review rate due to the repeated review of documents, but is surprisingly still more efficient than FF-CAL on three matters. At 5x phase one review rate PH-CAL outperforms even the strongest IP-CAL baseline for all matters by large margins. While 5x increases are not guaranteed, 3x increase are not unreasonable. And at 3x, PH-CAL has negligible differences relative to IP-CAL on two matters (226 minutes worse on one, 253 and 317 minutes better on the others), but saves from 9 to 48 days (4,550 to 22,932 minutes) on the other five matters. The savings of PH-CAL at 3x relative to industry standard FF-CAL are universally positive, and range between approximately 1 day at the low end, to 86 days at the high. Note that the percentage improvements shown in Table III are in terms of the full review effort required, akin to the absolute improvements in Table II. We do not show the residual improvements, but those would be even higher.

In summary, PH-CAL under reasonable assumptions is capable of beating not only the FF-CAL industry state-of-the-art baseline, but our stronger IP-CAL baseline as well. The savings are material to the practice of law and the ability of lawyers to cost-effectively meet their obligations to produce documents responsive to a matter.

VI. Conclusion

The overall conclusion of this paper is simple: Break up the family, but not all at once; do it in phases. Our results show that all three proposed broken and phased family protocols outperform the industry standard full family protocol by wide margins on every matter. We did not find a significant difference between a protocol in which unreviewed family members are immediately reviewed inline with a newly found relevant document (PF-CAL) versus a protocol in which the review (and therefore training of the supervised learning algorithm using those family members) is delayed to the end (IP-CAL), though there was slight evidence that the latter performs better at higher recall. A sensitivity analysis on the phased approach further found that if relevance-only review can be done at 3x faster than a dual relevance and privilege review, the phased protocol yields additional significant improvement over IP-CAL. For recall-oriented search tasks such as eDiscovery, process design is important. Even with state-of-the-art supervised machine learning algorithms driving the review, carefully crafted process design can translate into hours, days, even weeks of increased review efficiency.

VII. Future Work

The success of the various review protocols suggests that process design should continue beyond family protocols. For example, in eDiscovery we note that document review is often undertaken by multi-person teams of reviewers. This means that processes could be designed to take advantage of different reviewer skills or roles and route different documents to different reviewers to optimize global effectiveness. User studies and document content analyses could help predict more accurate speedup rates for relevance-only phased review. Finally, the question of review stopping point has been studied within a single (relevance-only) constraint CAL context [6], but it is an open question whether the same techniques (such as the knee-finding method) work as well in a PH-CAL review, not to mention in the other broken family protocols as well.

		1x	2x	3x	4x	5x	1x	2x	3x	4x	5x
	Recall	PH-CAL Time Reduction over IP-CAL					PH-CAL Time Reduction over FF-CAL				
Matter 1	90	-1392	-522	-226	-87	0	-563	303	597	736	822
Matter 2		-851	6277	8701	9841	10554	-981	6131	8550	9688	10399
Matter 3		-2167	-313	317	614	799	-1432	419	1049	1345	1531
Matter 4		-83	168	253	294	319	431	682	767	807	832
Matter 5		-11052	7905	14350	17383	19279	8802	27742	34181	37212	39106
Matter 6		-12108	14041	22932	27116	29730	6750	32874	41756	45936	48549
Matter 7		-23818	-1823	5655	9175	11374	-10139	11856	19334	22854	25053
Matter 8		-39050	-6512	4550	9756	13010	-27744	4747	15794	20992	24241
	Recall	PH-CAL % Improvement over IP-CAL					PH-CAL % Improvement over FF-CAL				
Matter 1	90	-54.9	-20.7	-8.9	-3.5	0.0	-22.2	12.0	23.5	29.7	34.0
Matter 2		-5.9	43.4	60.0	68.1	72.9	-6.8	42.4	58.9	67.0	71.9
Matter 3		-35.7	-5.2	5.2	10.1	13.2	-23.6	7.0	17.3	22.2	25.2
Matter 4		-8.8	18.1	27.2	31.5	34.2	45.8	73.6	82.6	86.6	89.3
Matter 5		-21.6	15.4	28.0	33.9	37.6	17.2	54.1	66.8	72.7	76.4
Matter 6		-15.5	17.8	29.3	34.8	38.1	8.6	41.7	53.4	58.9	62.1
Matter 7		-47.8	-3.7	11.5	18.4	22.9	-20.4	23.9	39.2	45.9	50.4
Matter 8		-34.4	-5.8	4.0	8.6	11.5	-24.4	4.2	13.8	18.5	21.4

TABLE III: Sensitivity analysis of review time for PH-CAL. Raw time reduction (total minutes saved) and relative time reduction (percentage improvement) for PH-CAL over the IP-CAL (top) and FF-CAL (bottom) baselines, at simulated 1x through 5x phase one review speeds, shown at 90% recall.

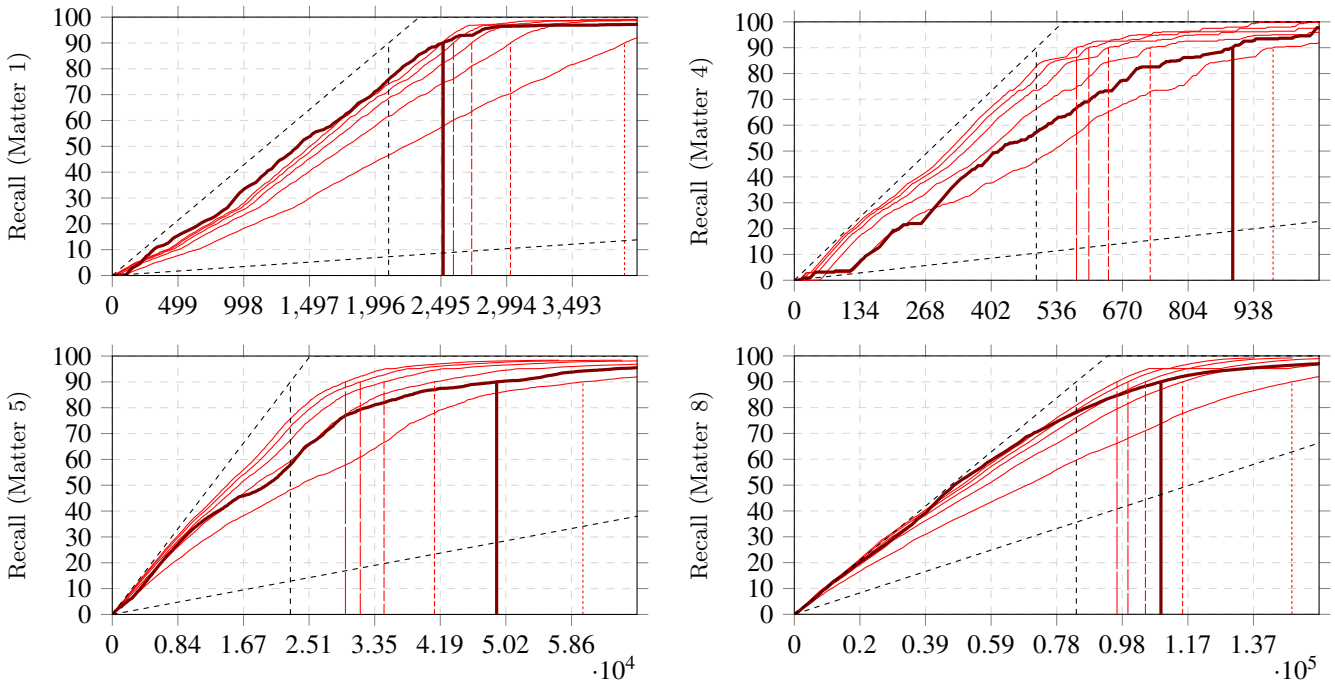

Fig. 2: Time-based gain curves for IP-CAL review protocol (thick red) versus PH-CAL review protocol at assumed 1x (shortest dotted red) through 5x (longest dotted red) relevance-only review speed. Vertical indicators at 90% recall. Dashed black lines indicate (producible) perfect and linear. X-axis is time (in minutes).

References

[1] J. R. Baron. Panning for gold in e-discovery: What every information scientist should know about the way lawyers search for electronic evidence. http://videolectures.net/cikm08_baron_pfgied/?q=jason%20baron, 2008.

[2] A. Broder. A taxonomy of web search. SIGIR Forum, 36, 2002.

[3] G. V. Cormack and M. R. Grossman. Evaluation of machine learning protocols for technology-assisted review in electronic discovery. In Proceedings of the ACM SIGIR Conference, Gold Coast, Australia, 6-11 July 2014, Gold Coast, Australia, 2014.

[4] G. V. Cormack and M. R. Grossman. Autonomy and reliability of continuous active learning for technology-assisted review, April 2015.

[5] G. V. Cormack and M. R. Grossman. Multi-faceted recall of continuous active learning for technology-assisted review. In Proceedings of SIGIR'15), Santiago, Chile, August 2015.

[6] G. V. Cormack and M. R. Grossman. Waterloo (cormack) participation in the trec 2015 total recall track. In Proceedings of the TREC'15 Conference, Gaithersburg, MD, 2015.

[7] B. Dimm. The single seed hypothesis. https://blog.cluster-text.com/tag/single-seed/, 2015. published: 2015-04-25; accessed: 2018-01-18.

[8] M. R. Grossman and G. V. Cormack. Comments on "the implications of rule 26(g) on the use of technology-assisted review". Federal Courts Law Review, 7, 2014.

[9] M. R. Grossman, G. V. Cormack, and A. Roegiest. Trec 2016 total recall track overview. In Proceedings of the TREC'16 Conference, Gaithersburg, MD, 2016.

[10] D. W. Oard and W. Webber. Information retrieval for e-discovery. Foundations and Trends in Information Retrieval, 7:99–237, 2013.

[11] M. D. Smucker and C. L. A. Clarke. Time-based calibration of effectiveness measures. In Proceedings of the ACM SIGIR'12 Conference, Portland, Oregon, 2012.

[12] J. Tredennick, J. Pickens, and J. Eidelman. Predictive coding 2.0: New and better approaches to non-linear review, January 2012. LegalTech Presentation.