Vocabulary Reduction, Text Excision, and Procedural-Context Features in Judicial Document Analytics

L. Karl Branting
The MITRE Corporation
7515 Colshire Drive
McLean, Virginia 22102
lbranting@mitre.org

ABSTRACT

Collections of documents filed in courts are potentially a rich source of information for citizens, attorneys, and courts, but courts typically lack the ability to interpret them automatically. This paper presents technical approaches to two applications of judicial document interpretation: detection of document filing errors; and matching orders with the motions that they rule on. An empirical evaluation identified several techniques that exploit genre-specific aspects of judicial documents to improve performance on these two tasks, including vocabulary reduction to task-specific terms, excision of the portion of documents unlikely to contain relevant text, and optimizing error detection by separating document classification into two stages: classification of the document's text followed by interpretation of this text classification based on procedural context.

1. INTRODUCTION

The transition from paper to electronic filing in Federal, state, and municipal courts, which began in the late 1990s, has transformed how courts operate and how judges, court staff, attorneys, and the public create, submit, and access court filings. However, despite many advances in judicial access and administration brought about by electronic filing, courts are typically unable to interpret the contents of court filings automatically. Instead, court filings are interpreted only when they are read by an attorney, judge, or court staff member.

Machine interpretation of court filings would open a rich source of information for improving court administration and case management, access to justice, and analysis of the judiciary. However, there are numerous challenges to automating the interpretation of case filings. Courts typically accept documents in the form of PDFs created from scans. Scanned PDFs require optical character recognition (OCR) for text extraction, but this process introduces many errors and does not preserve the document layout, which contains important information about the relationships among text segments in the document. Moreover, the language of court filings is complex and specialized, and the function of a court filing depends not just on its text and format, but also on its procedural context. As a result, successful automation of court filings requires overcoming a combination of technical challenges.

This paper (1) describes the nature of court dockets and databases, (2) describes two classes of representative judicial document analysis tasks: docket error detection; and order/motion matching, and (3) presents technical approaches to each of the tasks together with preliminary empirical evaluations of the effectiveness of each approach.

2. COURT DOCKETS AND DATABASES

A court *docket* is a register of document-triggered litigation events, where a *litigation event* consists of either (1) a pleading, motion, or letter from a litigant, (2) an order, judgment, or other action by a judge, or (3) a record of an administrative action (such as notifying an attorney of a filing error) by a member of the court staff. Contemporary electronic docket systems are typified by CM/ECF [4], which was developed by the Administrative Office of US Court (AO) and is used in all Federal Courts. Each docket event in CM/ECF includes both (1) metadata generated at the time of filing, including both case-specific data (e.g., case number, parties, judge) and event-specific data (e.g., the attorney submitting the document, the intended document type) and (2) a text document in PDF format (except for administrative entries). One typical CM/ECF database for a large federal court contains approximately 420,000 cases involving 1,200,000 litigants, attorneys, and judges, roughly 10,900,000 docket entries, and approximately 4,000,000 documents. The experiments described below were performed on a collection of 267,834 documents that were filed consecutively in 2015 in a large federal district court.

3. DOCKET ERROR DETECTION

There are many kinds of docket errors, including defects in a submitted document (e.g., missing signature, sensitive information in an unsealed document, missing case caption) and mismatches between the content of a document and the context of the case (e.g., wrong parties, case number, or judge; mismatch between the document title and the document type asserted by the user). For attorneys, detection of defects at submission time could prevent the embarrassment of submitting a defective document and the inconvenience and delays of refiling. For court staff, automated filing error detection could reduce the auditing staff required for filing errors, a significant drain of resources in many courts. Automating error detection could significantly reduce both of these problems.

This section focuses on four types of docket errors:

- Event-type errors. Specifying the wrong event type for a document, e.g., submitting a Motion for Summary Judgment as a Counterclaim. In the experiments below, there were 20 event types, such as complaint, transfer, notice, order, service, etc.

- Main-vs-attachment errors. Filing a document, such as an exhibit, that can only be filed as an attachment to another document, as a main document or filing a document, such as a Memorandum in Support of a Motion for Summary Judgment, that should be filed as a main document, as an attachment.

- Show-cause order errors. Only judges are permitted to file show-cause orders; it is an error if an attorney does so.

- Letter-motion errors. In some courts, certain routine motions can be filed as letters, but all other filings must have a formal caption. Recognizing these errors requires distinguishing letters from non-letters.

Each of these error-detection tasks requires classifying a document with respect to the corresponding set of categories (event type, main vs. attachment, show-cause order vs. non-show-cause order, and letter vs. non-letter) and evaluating whether the category is consistent with the metadata generated in CM/ECF by the filer's selections. Event type document classification is particularly challenging both because document types are both numerous (20 in the test dataset) and skewed (roughly power-law frequency distribution in the test set).

3.1 Text Classification

The first set of experiments attempted to identify each of the four docket errors above by classifying document text and determining whether there is a conflict between the apparent text category and the document's metadata. An initial barrier was that OCR errors greatly expand the apparent vocabulary size, making term-vector representations of documents extremely verbose and leading to very slow training and large models. One approach to reducing this verbosity is to classify documents using a language model (LM)[1, 9], which can be trained incrementally. Language-model classification is relatively fast even if the feature sets include n-grams with large n. The experiment in this paper used the lingpipe[1] LMClassifier, which performs joint probability-based classification of token sequences into non-overlapping categories based on language models for each category and a multivariate distribution over categories.

A second approach is to reduce the vocabulary to terms likely to be relevant to the particular domain [7]. A third approach is to excise those portions of documents that contain the least information about the document type. All three approaches were explored in this work.

3.1.1 Vocabulary Reduction and Text Excision

Court filings can be thought of as comprising four distinct sets of terms:

- Procedural words, which describe the intended legal function of the document (e.g., "complaint," "amended," "counsel"

- stop-words (uninformative common words, such as "of" and "the")

- Words unique to the case, such as names, and words expressing the narrative events giving rise to the case; and

[1]http://alias-i.com/lingpipe/

Types (20)		Letter vs other		Show-cause vs other	
united	0.3693	dear	0.2126	show	0.3455
states	0.3651	re	0.2019	cause	0.3210
judge	0.3511	judge	0.1211	ordered	0.1751
complaint	0.3343	letter	0.1053	order	0.1446
motion	0.3228	request	0.1026	heard	0.1346
plaintiff	0.3209	respectfully	0.0889	soon	0.1301
ordered	0.3118	date	0.0720	deemed	0.1180
action	0.3032	extension	0.0470	shall	0.1101
relief	0.2787	submitted	0.0461	thereafter	0.1098
must	0.2636	conference	0.0342	annexed	0.0943

Figure 1: The information gain of the 10 highest-information terms for 3 legal-document classification tasks.

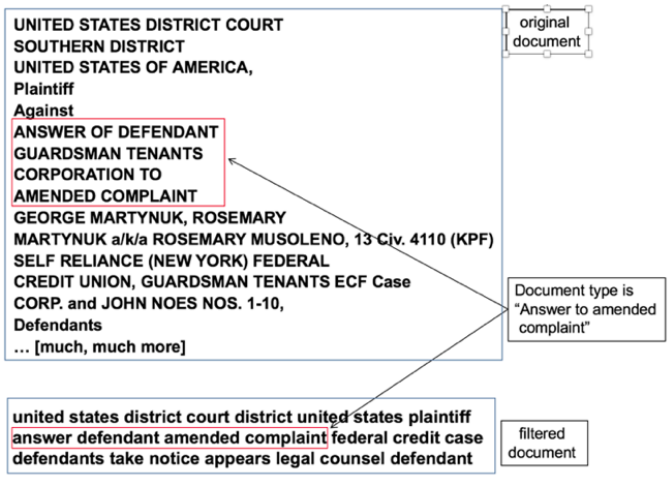

Figure 2: Reduction of a full document to just FRCP terms.

- Substantive (as opposed to procedural) legal terms (e.g., "reasonable care," "intent," "battery").

Terms in the first of these sets–procedural words–carry the most information about the type of the document. These words tend to be concentrated around the beginning of legal documents, often in the case caption, and at the end, where distinctive phrases like "so ordered" may occur.

We experimented with several approaches to vocabulary reduction: two ad hoc and domain-specific and one general and domain-independent. The first approach was to eliminate all terms except non-stopwords that occur in the Federal Rules of Civil Procedure [5]. An alternative approach was to remove all terms except for non-stopwords occurring in "event" (i.e., document) descriptions typed by filers when they submit into CM/ECF. The third approach was to select terms based on their mutual information with each particular text categories [2]. The first lexical set, termed FRCP, contains 2658 terms; the second, termed event, consists of 513 terms. Separate mutual-information sets were created for each classification task, reflecting the fact that the information gain from a term depends on the category distribution of the documents.

For example, Figure 1 shows the 10 highest information terms for

Table 1: Thresholds and size of large and small high information-gain term sets.

	showcause	main_attch	types	letter
ig_small	0.01 (135)	0.025 (262)	0.1 (221)	0.0005 (246)
ig_large	0.0025 (406)	0.0125 (914)	0.05 (689)	0.00001 (390)

three different classification tasks: event-type classification, distinguishing letters from non letters, and show-cause order detection, illustrating that the most informative terms differ widely depending on the classification task.

Figure 2 illustrates the reduction of full document text to just FRCP terms, which typifies the vocabulary-reduction process.

Several approaches to document excision were explored as well. The first was to limit the text to the first l tokens of the document (i.e., excise the remainder of the document). If l is sufficiently large, this is equivalent to including the entire document. A second option is to include the last l tokens of the suffix as well as the prefix. For simplicity, the same l is used for both the prefix and the suffix.

The initial set of experiments using language model (LM) classification evaluated the effect of varying the following parameters:

- Vocabulary reduction: none, FRCP, event, ig_small, ig_large
- Prefix length, l
- Whether l tokens of the suffix are included, in addition to the prefix
- The maximum n-gram length, n

Two different information-gain thresholds were tested for each classification type, intended to create one small set of very-high information terms (ig_small) and a larger set created using a lower threshold (ig_large). The thresholds and sizes of the large and small high information-gain term sets are set forth in Table 1.The text of each document was obtained by OCR using the open-source program Tesseract [11]. Each text was normalized by removing non-ASCII characters and standardizing case prior to vocabulary reduction, if any.

Figure 3 shows a comparison of four vocabulary alternatives on the four text classification tasks described above. These tests measured mean f-measure in 8-fold cross validation using a 1-gram language mode, 50-token prefix length, and no suffix. In the baseline vocabulary set, normalize, non-ASCII characters, numbers, and punctuation are removed and tokens were lower-cased. The results show that classification accuracy using an unreduced vocabulary was significantly lower than the best reduced vocabulary performance for show-cause order detection and type classification. Choice of vocabulary had little effect on accuracy for the letter and main vs. attachment detection tasks. No reduced-vocabulary set consistently outperformed the others, although ig_large (with a lower information-gain threshold) was consistently slightly better than ig_small (with a higher information-gain threshold).

Figure 4 shows the results of running this same set of experiments with more-expressive 4-gram models. Accuracy rose for all test

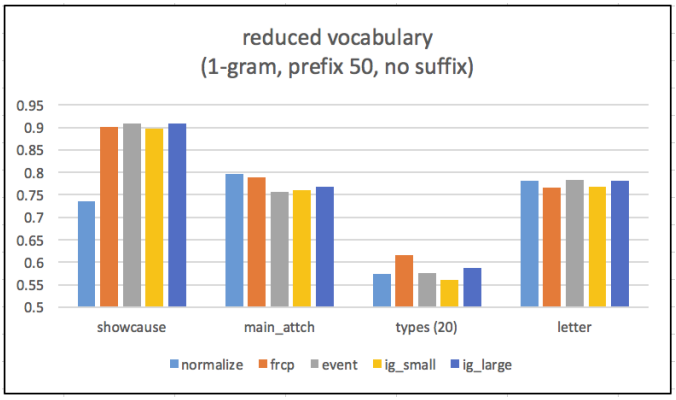

Figure 3: Classification accuracy as a function of reduced vocabulary (8-fold cross validation using a 1-gram language model, 50-token prefix length, and no suffix).

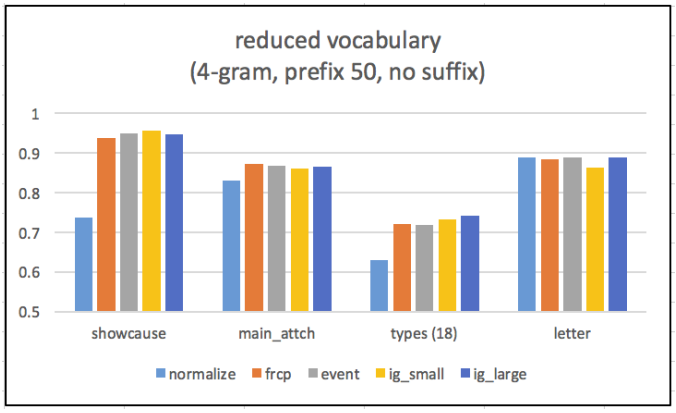

Figure 4: Classification accuracy as a function of reduced vocabulary (8-fold cross validation using a 4-gram language model, 50-token prefix length, and no suffix).

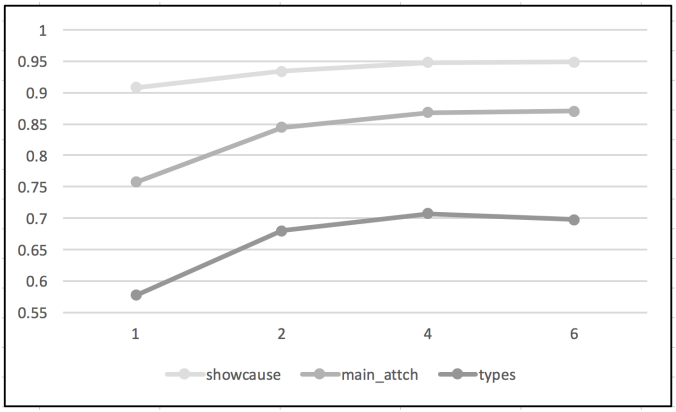

Figure 5: Classification accuracy as a function of maximum n-gram size (8-fold cross validation using event vocabulary, 50-token prefix length, and no suffix).

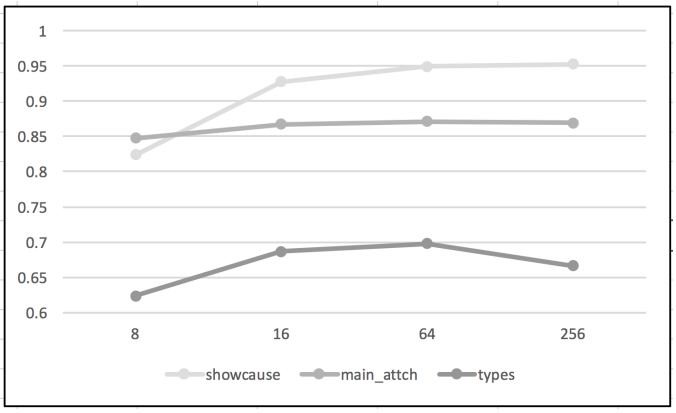

Figure 6: Classification accuracy as a function of prefix length (8-fold cross validation using event vocabulary, maximum n-gram length n = 6, and no suffix).

conditions, but for the show-cause, main vs. attachment, and types tasks, accuracy rose more for the reduced vocabulary than for the unreduced conditions. Once again, there was little consistency in the relative classification accuracy under alternative restricted vocabularies. This indicates that restricted term sets derived through information gain perform roughly as well as those produced using domain-specific information, suggesting that the reduced vocabulary approach is appropriate for situations in which domain-specific term information is unavailable.

Use of a reduced vocabulary makes it feasible to go beyond simple term frequency models by building n-gram language models using a large n. Figure 5 compares the performance of n-gram models on the three text categories as a function of the size of n (event filtering, 50-token prefix length, no suffix). Accuracy appeared to flatten out at n=4, with a slight decrease at n=6 in the type categorization.

Figure 6 shows classification accuracy for the three document categories as a function of the length of the prefix extracted from the text. Accuracy improved with increasing prefix length up to length 64, at which point accuracy began to decrease for event-type classification.

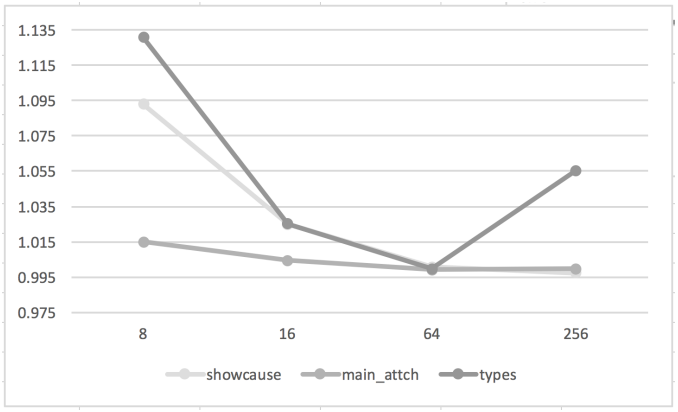

Figure 7: The ratio of classification accuracy using both prefix and suffix to accuracy using prefix only, as a function of prefix and suffix length (8-fold cross validation using event vocabulary, and maximum n-gram length n = 6)

Figure 7 shows the ratio of prefix-only to prefix-plus-suffix as a function of prefix and suffix length. For these classification tasks, the greatest improvement in performance from including the suffix as well as the prefix occurred when the prefix and suffix lengths were quite short (8 tokens), and there was no improvement for l=64.

Summarizing over the tests, the the highest mean f-measure based on text classification alone and the particular combination of parameters that led to this accuracy for each document category were as follows:

1. **Event type: 0.743** (prefix=50, no suffix, max n-gram=4, ig_large vocabulary, 20 categories)
2. **Main-vs-attachment: 0.871** (prefix=256, no suffix, max n-gram=6, event vocabulary)
3. **Show-cause order: 0.957** (prefix=50, no suffix, max n-gram=5, ig_small vocabulary)
4. **Letter-vs-non-letter: 0.889** (prefix=50, no suffix, max n-gram=4, ig_large vocabulary)

3.2 Incorporating Procedural Context Features

The accuracy of event-type detection (f-measure of roughly 0.743 under the best combinations of parameters) is sufficiently low that its utility for many auditing functions may be limited. An analysis of the classification errors produced by the event-type text classification model indicated that a document's event type depends not just on the text of the document but also on its *procedural context*. For example, motions and orders are sometimes extremely similar because judges grant a motion by adding and signing an order stamp to the motion. Since stamps and signatures are seldom accurately OCR'd, the motion and order may be indistinguishable by the text alone under these circumstances. However, orders can be issued only by a judge, and judges never file motions, so the two cases can be distinguished by knowing the filer. In addition, attachments have the same event type as the main document in CM/ECF. So, for example, a memorandum of law is ordinarily a main document, but an already-filed memorandum can sometimes be filed as an attachment, in which case its event type is the same as that of the main document. So, determining the event type of a document

requires knowing both (1) whether it was filed as a main document or as an attachment and (2) the filer.

The first and simplest step to incorporating procedural-context information is to assign to each document filed as an attachment the type of the main document to which it is attached. This reduces the event-type classification problem to classifying documents that are filed as main documents. The f-measure for LM classification restricted to main documents was 0.809, as compared to .707 for main documents and attachments combined (event vocabulary, prefix length 25, max n-gram size 6, suffix not included).

Four approaches were explored for adding procedural-contextual information to improve event-type classification for main documents. The first was the highly *ad hoc* procedure of prepending procedural-context features as unique tokens to the document text to which the LM classifier was applied. Using filer (crt,aty,rep,unk,pty,jud) document page length, the first token of the filer's document summary description, and the nature of the suit as procedural-context features raised the f-measure to 0.847.

To avoid the unnatural procedure of adding context features to the input into a language model, the next two approaches represented documents as collections of term features together with context features. Using the event vocabulary yields 513 term features. Two types of term features were evaluated: term frequency; and inverse first-occurrence position (IFOP). The rationale for IFOP is that it is typically the first occurrence of a term that indicates its procedural function, e.g., the first time "motion" occurs is likely to be in the case caption or in the introductory sentence of a letter, in either case indicating the purpose of the document. Subsequent occurrences are likely to refer to other documents. Inverse first position weights early occurrences more heavily than later occurrences. Applying the J48 decision tree classifier from WEKA [6] yielded f-measures of 0.847 (term frequency) and 0.870 (IFOP).

The fourth approach was to use the classification predicted by an LM classifier as a feature combined with metadata features (filer, nature of suit, page count), yielding an f-measure of 0.964. These results are summarized in Table 2.

Table 2: Event-type classification results incorporating different term and procedural-context features sets.

LM classifier main and attachments	0.707
LM classifier main only	0.809
LM + features	0.847
J48 features + terms-freq	0.846
J48 features + terms-pos	0.870
LM output + features	0.964

Inspection of the decision tree induced by J48, a fragment of which is shown in Figure 8, reveals that the topmost branch is on the LM classifier feature, e.g., "LMpredicted = motion," with subsequent branches identifying conditions in which the text-based prediction associated with that branch may be in error, e.g., a document for which "filer = crt" is probably an order even if its text standing alone is typical of a motion.

In summary, event-type classification accuracy increased from 0.743 to 0.964 when procedural-context information was added by (1) assigning attachments the event type of the main document to which they are attached, and (2) training a document classifier on filer, page length, and filer's document description features, together with the classification of the text.

```
LMpredicted = motion
|   filer = crt
|   |   pageCount <= 2
|   |   |   pageCount <= 1: order (540.0/92.0)
|   |   |   pageCount > 1
|   |   |   |   natureOfSuit <= -1: order (194.0/93.0)
|   |   |   |   natureOfSuit > -1
|   |   |   |   |   natureOfSuit <= 555
|   |   |   |   |   |   natureOfSuit <= 450: order (69.0/16.0)
|   |   |   |   |   |   natureOfSuit > 450: motion (26.0/9.0)
|   |   |   |   |   natureOfSuit > 555: order (35.0/5.0)
|   |   pageCount > 2
|   |   |   natureOfSuit <= 555: motion (560.0/57.0)
|   |   |   natureOfSuit > 555
|   |   |   |   natureOfSuit <= 710: order (6.0/2.0)
|   |   |   |   natureOfSuit > 710
|   |   |   |   |   natureOfSuit <= 870: motion (28.0/6.0)
|   |   |   |   |   natureOfSuit > 870: order (6.0)
|   filer = aty: motion (16953.0/405.0)
|   filer = rep: motion (0.0)
|   filer = unk: motion (35.0/1.0)
|   filer = pty: motion (43.0)
|   filer = jud: motion (0.0)
```

Figure 8: A fragment of the J48 decision tree for event-type classification using both procedural-context features and LM predictions.

4. ORDER/MOTION MATCHING

In many federal courts, docket clerks are responsible for filing orders executed by judges into CM/ECF, a process that requires the clerk to identify all pending motions to which the order responds and to link the order to those motions. This entails reading all pending motions, a tedious task. If the motions corresponding to an order could be identified automatically, docket clerks would be relieved of this laborious task. Even ranking the motions by their likelihood of being resolved by a motion would decrease the burden on docket clerks. Moreover, order/motion matching is a subtask of a more general issue-chaining problem, which consists of identifying the sequence of preceding and subsequent documents relevant to a given document.

A straightforward approach to this task is to treat order/motion matching as an information-retrieval task, under the hypothesis that an order is likely to have a higher degree of similarity to its corresponding motions than to motions that it does not rule on. An obvious approach is to present pending motions to the clerk in rank order of their TF/IDF[2]-weighted cosine similarity to the order.

The evaluation above showing that vocabulary reduction improves document classification raises the question whether vocabulary reduction might be beneficial for order/motion matching as well. A second question is whether the IDF motion should be trained on an entire corpus of motions and orders or whether acceptable accuracy can be obtained by training just on the order and pending motions.

To evaluate the effectiveness of this approach to order/motion match, a subset of the document set described above was collected consisting of 3,356 groups, each comprising (1) an order, (2) a motion

[2]Term Frequency/Inverse Document Frequency

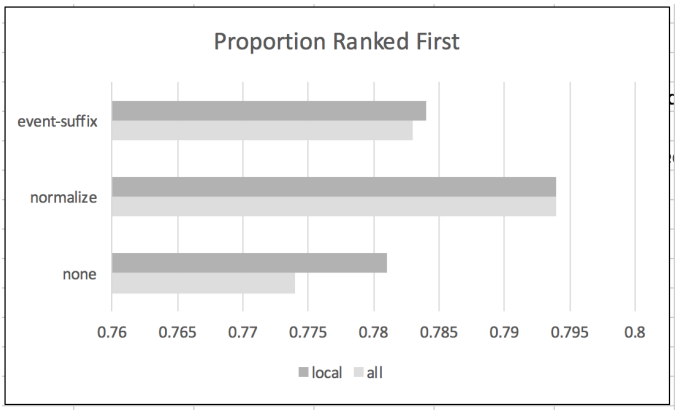

Figure 9: The proportion of groups for which the order is more similar to the triggering motion.

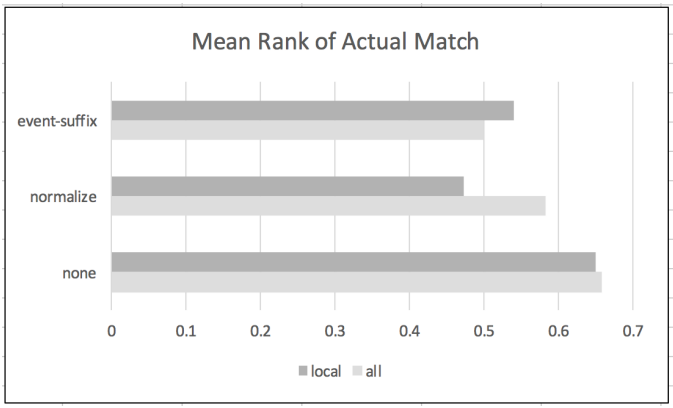

Figure 10: The mean rank of the triggering order among all pending orders, zero-indexed (lower is better).

that the order rules on (a *triggering motion*), and (3) a non-empty set of all motions that were pending at the time of the order but not ruled on by the order (*non-triggering motions*). The mean number of motions per group was 5.87 (i.e., there were on average 4.87 non-triggering motions). For each group, all motions were ranked by similarity to the order under the given metric. The proportion of triggering motions that were ranked first and mean rank of the triggering motion were calculated from each group's ranking.

These groups were evaluated using three vocabulary reduction approaches: the raw document text (which often contains many OCR errors); normalization, as described above; and event terms. The two alternative TF/IDF training models were applied to each of the three vocabulary reduction approaches, for a total of 6 combinations. For each combination, the mean rank of the triggering motion among all the motions was determined.

Figure 9 shows that the highest accuracy, as measured by the proportion of triggering motions that were ranked first among all pending motions, was achieved by normalizing the text but not by vocabulary reduction. Intuitively, reduction to procedurally relevant terms improves the ability to determine what docket event a document performs, but reduces the ability to discern the similarity between pairs of documents. TF/IDF training on just the order and pending motions (*local*) is at least as accurate as training over all orders and motions (*all*). Figure 10 shows the mean rank (zero indexed) of the most similar motion under each of the six conditions. The best (lowest) mean rank was achieved with normalization and local TF/IDF training.

It is not unusual for a single order to rule on multiple pending motions. A more realistic assessment of the utility of pending motion ranking is therefore to determine how many non-triggering motions a clerk would have to consider if the clerk read each motion in rank order until every motion ruled on by the order is found. One way to express this quantity is as mean precision at 100% recall. In the test set described above, using text normalization and local TF/IDF training, mean precision at 100% recall was 0.83, indicating that the number of motions that a clerk would have to be read was significantly reduced.

5. RELATED WORK

There is a long history of applying text classification techniques to legal documents dating back at least to the 1970s [3]. Text classification has been recognized as of particular importance for electronic discovery [10]. Little prior work has addressed classification of docket entries other than Nallapati and Manning [8], which achieved an f-measure of 0.8967 in distinguishing Orders to Show Cause from other document types using a hand-engineered feature set. As shown above, we obtained an f-measure of 0.9573 using the reduced vocabulary approach as well as good performance on other classification tasks and scalability. Thus, the approach described in this paper represents a significant advance over prior work.

6. SUMMARY AND FUTURE WORK

Judicial document collections contain a rich trove of potential information, but analyzing these documents presents many challenges. This paper has demonstrated how vocabulary reduction, text excision, and procedural-context features can be used in combination to improve the accuracy of recognizing the nature of legal documents, including whether the document is a main document or an attachment, the document's event type, and whether the document is a show-cause order. Reduced vocabularies based on domain-specific information–FRCP terms and the document description field of the CM/ECF database–performed with comparable accuracy to reduced vocabularies based on information gain, illustrating that useful reduced term sets can be derived without domain-specific information.

These results demonstrate the feasibility of automating the process of auditing CM/ECF submissions, which is currently a significant drain on court resources. The experiment with order/motion matching demonstrates that while vocabulary reduction may improve accuracy for document classification, it can decrease accuracy for tasks that involve matching based on overall similarity rather than procedural similarity.

Many of the challenges addressed in this work arise from the current inability to reason about the layout of legal documents. For example, many documents have a case caption in which the case title and other standard information fields have standard spatial relationships to one another. We are currently engaged in developing an annotated corpus of court documents for use in training 2-dimensional conditional random fields and other spatially-aware document analysis tools. However, even when these tools are available, in many cases it will remain necessary to reason about the text itself.

No single technology is applicable to all judicial documents, nor is

any approach sufficient for all document analysis tasks. However, each addition to this suite of technologies adds to the capabilities available to the courts, government agencies, and citizens to exploit the deep well of information latent in judicial document corpora.

Acknowledgment

The MITRE Corporation is a not-for-profit Federally Funded Research and Development Center chartered in the public interest. This document is approved for Public Release, Distribution Unlimited, Case Number 16-1541.

7. REFERENCES

[1] J. Bai and J.-Y. Nie. Using language models for text classification. In *Proceedings of Asia Information Retrieval Symposium*, Beijing, China, October 2004.

[2] R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5(4):537–550, Jul 1994.

[3] J. Boreham and B. Niblett. Classification of legal texts by computer. *Information Processing & Management*, 12(2):125 – 132, 1976.

[4] CM/ECF. `https://en.wikipedia.org/wiki/CM/ECF`. Case Management/Electronic Case Files.

[5] L. I. I. Cornell University Law School. The federal rules of civil procedure. `https://www.law.cornell.edu/rules/FRCP`.

[6] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explorations*, 11(1), 2009.

[7] R. E. Madsen, S. Sigurdsson, L. K. Hansen, and J. Larsen. Pruning the vocabulary for better context recognition. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 483–488. IEEE, 2004.

[8] R. Nallapati and C. D. Manning. Legal docket-entry classification: Where machine learning stumbles. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 438–446, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[9] F. Peng, D. Schuurmans, and S. Wang. Augmenting naive bayes classifiers with statistical language models. *Inf. Retr.*, 7(3-4):317–345, Sept. 2004.

[10] H. L. Roitblat, A. Kershaw, and P. Oot. 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, 2010.

[11] Tesseract. `https://en.wikipedia.org/wiki/Tesseract_(software)`.