Proceedings of the Twenty-Seventh International Florida Artificial Intelligence Research Society Conference

SMART Electronic Legal Discovery via Topic Modeling

Clint P. George, Sahil Puri,
Daisy Zhe Wang, and **Joseph N. Wilson**
Computer and Information Science and Engineering
University of Florida

William F. Hamilton
Executive Director
UF Law E-Discovery Project
Levin College of Law
University of Florida

Abstract

Electronic discovery is an interesting sub problem of information retrieval in which one identifies documents that are potentially relevant to issues and facts of a legal case from an electronically stored document collection (a corpus). In this paper, we consider representing documents in a topic space using the well-known topic models such as latent Dirichlet allocation and latent semantic indexing, and solving the information retrieval problem via finding document similarities in the topic space rather doing it in the corpus vocabulary space. We also develop an iterative SMART ranking and categorization framework including human-in-the-loop to label a set of seed (training) documents and using them to build a semi-supervised binary document classification model based on Support Vector Machines. To improve this model, we propose a method for choosing seed documents from the whole population via an active learning strategy. We report the results of our experiments on a real dataset in the electronic discovery domain.

1 Introduction and Background

Electronic-discovery (e-discovery) is the process of collecting and reviewing electronic documents — which may be in plain text or converted into plain text using methods such as Optical Character Recognition (OCR) — to identify their relevance to a legal case (Berry, Esau, and Keifer 2012). The amount of data to be dealt with in any single case can be enormous, making the manual reviewing process cumbersome and expensive. For example, the study conducted at kCura (kCura 2013) comparing the 100 largest cases found that the median case size grew from 2.2 million documents in 2010 to 7.5 million in 2011. Sometimes even with expert reviewers the results of manual review are inconsistent (Lewis 2011). Litigation costs are increasing and as a result, are removing the public dispute resolution process from reach of the average citizen and medium-sized company. Thus, legal professionals have sought to employ information retrieval and machine learning methods to reduce manual labor and increase accuracy.

In a typical computer assisted review (CAR) setting — a.k.a., technology-assisted review (TAR) or predictive coding — one trains a computer to categorize documents based on relevancy to a legal case using a set of seed documents labeled by expert reviewers or lawyers. CAR has three main components — a domain expert, an analytics or categorization engine, and a method for validating results. A domain expert is a well trained human reviewer, e.g., a lawyer, who can identify and label *relevant* and *irrelevant* documents from the available document collection of a legal case. The categorization engine propagates the domain expert's knowledge to the whole document collection via varying indexing, relevance-ranking, and classification methods. Finally, a validation method such as statistical sampling (Cochran 2007) helps lawyers to validate whether the system's results are the results desired by the review team. For more discussion about the CAR process and e-discovery, the readers are recommended to see, e.g., the Computer Assisted Review Reference Model[1], Relativity (kCura 2013), etc. We follow the CAR model as a baseline for building our e-discovery retrieval model.

A critical task associated with the categorization of documents is ranking their relevance to a given query. In a relevance-ranking framework, users typically list topic-specific keywords or phrases. For example, when searching for *computers*, a search string such as *computer* or *PC* or *laptop* or *RAM* might be formulated. The software searches for documents containing the *keywords* (or variants thereof if the software has more advanced fuzzy logic, stemming and other capabilities), ranks them using a *similarity score* (e.g., cosine similarity) and displays the results to users. Such keyword ranking methods are flawed as they are limited by the parameters and search terms employed by the user. Typically, when we search for documents we look for their concepts or topics rather than their keywords. This line of thinking lead us to build a hybrid document retrieval system that uses the topics underneath a keyword-query along with existing keyword-search strategies, e.g., (Lucene 2013).

Background

In a typical document retrieval, where incoming user queries are compared with stored or indexed text documents, a major task is to represent entities, i.e., keyword queries and documents, in an indexing space where similar entities lie near each other and dissimilar ones are far apart. Vector space

[1] `http://www.edrm.net`

modeling (Salton, Wong, and Yang 1975) is an indexing method which maps a corpus that consists of D documents and V vocabulary words into a *term-document* matrix. Each column in the matrix represents a document in the corpus with each column element represents the weight of a term, e.g., the relative frequency of a term, in the document. The matrix is then translated into vectors in a vector space where one vector is assigned to each document in the corpus. One can consider a user's keyword-query as as a document and easily map it to a vector in the vector space and compute a similarity score such as cosine similarity between the query vector and document vectors.

Latent semantic indexing (LSI) (Dumais et al. 1995), a well-known method in the field of information retrieval, can group together words and phrases that have similar meanings (i.e., synonymy). One can use these groups or concepts to represent the documents in a collection and keyword queries, and perform a "concept search" to retrieve relevant documents by defining a similarity score on the new representative domain. LSI typically performs matrix factorization over a *term-frequency inverse-document-frequency* (TF-IDF) (Jones 1972) matrix, using the concepts of eigenvalue decomposition, and identifies patterns in the relationships between the document terms and concepts or topics. One drawback of this model is that it is not a probabilistic model. Probabilistic models can be easily extended to more complicated models and generalized to include newly encountered documents. In addition, LSI's parameters are linear in the number of documents in the training corpus.

Probabilistic topic modeling allows us to represent the properties of a corpus with a small collection of topics or concepts, far fewer than the vocabulary size of a corpus. Latent Dirichlet allocation (LDA) (Blei, Ng, and Jordan 2003) is a probabilistic topic model, which is most easily described by its generative processes, the random process by which the model assumes the documents created. It assumes that the corpus has a vocabulary of terms, and each document in the corpus is described by a mixture of topics. A topic is represented by a distribution of words in the vocabulary. For example, the *sports* topic has words about sports, e.g., football, soccer, etc., with high probability. Each observed word in a document is chosen from a topic distribution that is assigned to the word by a random process which samples a topic (that is latent or hidden) for the word from the document topic distribution. LDA enables us to infer the values of these latent topic variables and the topic structure of each document in the corpus. Topic modeling is also known as a method to handle word polysemy (i.e., words that have more than one distinct meaning) and synonymy of words that causes poor precision and recall in keyword-based searches.

Batch-based document classification, which deals with large static document collections, is usually performed using a supervised learning algorithm such as a support vector machine (SVM), neural network, or naïve Bayes classifier. One historical example of this type of system is the DolphinSearch tool (Berry, Esau, and Keifer 2012), which supports electronic document discovery solutions. It splits the data into a training set and test set, and learns a classification function which maps the input documents to the corresponding labels using the training set. Then one analyzes the quality of the classification function by testing it on the test set, and uses the classification model for newly encountered unlabeled documents. These methods require a sufficiently large set of manually labeled documents. To overcome this, we have built a system that uses both semi-supervised and active learning strategies to discover relevant documents, which are represented in terms of identified topics of the corpus, for a legal case. In a semi-supervised learning framework, we assume that there are some existing searches or keyword-documents from which to form a sufficient sample set (seed documents) of user queries. The system can build a semi-supervised model based on these already labeled documents. To further improve this model, the seed documents are selected carefully via certain active learning strategies, and presented to the domain experts for labeling.

In summary, we exploit the use of topic models such as LSI and LDA to improve the state of the art CAR process for e-discovery by (a) using topic modeling methods to represent documents and keyword-queries providing better power than commonly employed methods such as keyword search and TF-IDF, (b) using identified topics for document categorization and ranking their relevance to a given user query, and (c) improving the seed document selection by involving human-in-the-loop, active learning methods. This paper is organized as follows. Section 2 describes the design and work-flow of our system. In Section 3, we describe the experiments performed to choose the best classification and ranking method for our e-discovery model using a labeled e-discovery dataset. Section 4 concludes this paper with a discussion of our future plans.

2 The Approach

This section describes the ongoing research and proposed approach to develop a SMART retrieval model that will assist the search and review process of lawyers and other stakeholders in e-discovery. Figure 1 shows the graphical representation of the proposed SMART electronic document discovery model's work-flow.

Data Preprocessing and Metadata Extraction First, we parse all documents in different formats such as PDF, plain text, and email via a metadata extraction module. For example, from plain text formatted emails, we can extract metadata such as sender, receiver, subject, date, and email body. We assume that all documents are represented in plain text format so that we can tokenize them with any available tokenizer. We use the python Natural Language Processing Toolkit (Bird, Klein, and Loper 2009) and predefined regular expressions for tokenizing plain text. Second, we remove stop-words (e.g., a, an, the, I, you, etc.) to remove noise and apply two available methods —(a) stemming and (b) lemmatization— to reduce a word to a common base form. Stemming heuristically removes word endings without the knowledge of a context, e.g., *fish* is the stem for *fishing*, *fished*, *fish*, and *fisher*. On the other hand, lemmatization does full morphological analysis (using a dictionary such as

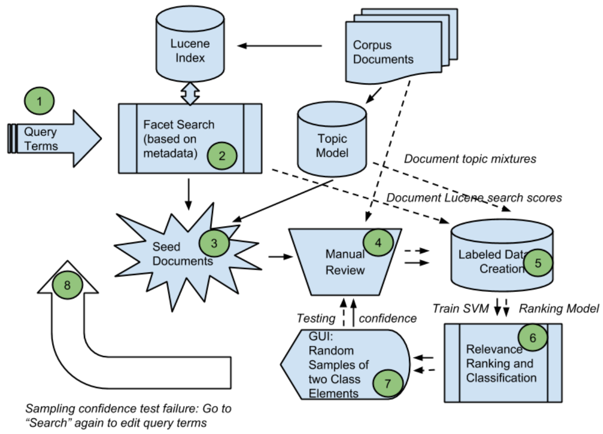

Figure 1: SMART Electronic Legal Discovery work-flow: Circled numbers represent each step in the electronic discovery work-flow.

WordNet) to determine the lemma for each word, e.g., *good* is a lemma for *better*. We also discard words that appear only once in the corpus. Finally, we represent each document in a bag-of-words format after building a vocabulary of the remaining words in the corpus.

Keyword-based and Topic-based Document Indexing

As an initial step, we use the well-known Apache Lucene search algorithm (Lucene 2013) as our keyword-search method. Lucene enables us to index documents' metadata (e.g., file modified date) and data fields (e.g., email subject) and search on them using a keyword-query represented in the Lucene query format (Lucene 2013). Lucene has several algorithms to rank documents given a query based on similarity scores such as cosine similarity and returns relevant documents given a query along with their ranking scores. We also use the Lucene results as a baseline for analyzing our proposed classification and ranking models.

A challenging problem in document classification and ranking is the choice of features. Considering individual words in documents as features as in TF-IDF models may yield a rich but very large feature set and cause computational difficulties. A much better approach is to analyze documents in the reduced topic space extracted by a topic model such as LSI and LDA.

We use the scalable implementations of LSI and LDA algorithms by (Řehůřek and Sojka 2010) for topic modeling. We assume we know the number of topics K in a corpus for running LDA and LSI. The LSI implementation performs a scalable singular value decomposition of the TF-IDF matrix of a corpus, and projects documents represented in the TF-IDF matrix into the LSI (semantic) space. The LDA implementation is based on a variational Bayes (VB) algorithm, which reduces any document in the corpus to a fixed set of real valued features—the variational posterior Dirichlet parameters θ_d^* associated with each document d in the corpus. Here, θ_d^* is computed (using the VB algorithm) as an estimate of θ_d, i.e., document d's distribution on the topics. We also use θ_d^* to represent the projected document d in the LSI space to simplify the notation.

For the topic modeling-based document retrieval, we consider each keyword-based query as a document in the corpus and identify its representation θ^*_{query} in the topic or semantic space using a pre-identified LDA or LSI model of the corpus.

Dynamic Seed Document Selection via Active Learning

As mentioned, we are interested in reducing manual labor and increasing accuracy in the CAR process. A major input to the CAR framework is the seed documents to be labeled for building the categorization and ranking model. The seed documents are usually chosen randomly or from initial ranking results from any keyword search engine. We follow an active learning (Settles 2009) strategy to chose the best subset of documents available in the collection. This method emerges from the concept of "stratified sampling from the whole population" in which, we employ a distance-based clustering algorithm such as k-Means clustering (Hartigan and Wong 1979) on the document feature vectors $x_d \in \mathcal{R}^{K+1}$ to find out their membership clusters. The vector x_d is a combined vector of document d's K-dimensional θ_d^* from a learned topic model and ranking score computed by Lucene search engine given a keyword-query. Finally, we generate a representative sample from each of the learned clusters and combine them to form a seed set of documents to be reviewed and labeled by domain experts (See circles #3 and #4 in Figure 1 for reference).

SMART Electronic Legal Discovery

The SMART Electronic Legal Discovery system works as follows. The user enters a query or combination of multiple queries on various document metadata fields (i.e., Facet Search). The system generates a set of seed documents based on the method described above and display it to the domain experts such as lawyers. They can review each document in the seed set based on its relevancy to a given query and mark it as *relevant* or *irrelevant*.

Our retrieval system has two main parts — (a) classifying documents as *relevant* and *irrelevant* given a case and (b) computing document relevancy ranking scores in each of those classes. We use the well-known support vector machines (Vapnik 1995) to train the document classifier. For SVM training, each document's feature vector $x_d \in \mathcal{R}^{K+1}$ is a combined vector of K-dimensional θ_d^* from a learned topic model and ranking score computed by Lucene metadata search engine given a keyword-query, and desired output is the expert annotated label. The trained SVM model is used for classifying other unlabeled documents in the collection. In our experiments, we used the SVM implementation given in LIBSVM (Chang and Lin 2011) radial basis function (RBF) as the kernel function. The SVM parameters (i.e., the penalty parameter $C > 0$ of the error term and the RBF parameter $\gamma > 0$) are chosen by performing grid-search on C and γ using cross-validation accuracy on the training set (see the LIBSVM guide for more details).

In an e-discovery process, it's also important to know how relevant a document is to a legal case quantitatively, because this helps the lawyers to decide the review budget and the limit on the number of documents to be reviewed. We consider a number of different methods to identify an optimal ranking for the documents including:

- Lucene: We present the query keywords to the Lucene search algorithm (Lucene 2013) and use its relevance response as each document's relevance index. This method is essentially the type of keyword search done in state-of-the-art e-discovery software.
- LSI with keywords (Keyword-LSI): We use LSI to identify topics within the document collection and map documents and keyword queries to vectors—θ^*_ds and θ^*_{query}s—in the LSI space, then for each document d in the corpus compute cosine similarity, $\cos(\theta^*_d, \theta^*_{\text{query}})$, to identify its relevance index.
- LDA with keywords (Keyword-LDA): We use LDA to identify topics within the document collection and map documents and keyword queries to vectors—θ^*_ds and θ^*_{query}s—in the LDA topic space, then for each document d in the corpus compute cosine similarity, $\cos(\theta^*_d, \theta^*_{\text{query}})$, to identify its relevance index.
- LDA with keyword topics (Topic-LDA): We use LDA to identify topics within the corpus, and then identify the top-K topics most relevant to the query from the query-topic-distribution θ^*_{query} that is obtained from the learned LDA model. Finally, we use the combined relevancy score of the identified topics in any document's estimated topic distribution θ^*_d as the document's relevance index as follows.

  Let $\mathcal{K}$ represent the indexes of topics in the corpus and $\mathcal{T} \subset \mathcal{K}$ represents the indexes of dominant topics in a keyword query, based on the probability values in query's θ^*_{query}. Then, for each document $d = 1, 2, \ldots, D$ in the corpus, we calculate (George et al. 2012)

  $$\text{sim}(d) = \sum_{j \in \mathcal{T}} \ln \theta^*_{dj} + \sum_{j \notin \mathcal{T}} \ln(1 - \theta^*_{dj}) \tag{1}$$

  Note that high values of $\text{sim}(d)$ indicates the topics indexed in $\mathcal{T}$ are prominent in document d.

Once we have the relevancy scores and class labels for documents we display the results to the user. The amount of data available for each class can be enormous making manual verification of the classification results intractable. A typical quality control method used in the e-discovery community is to generate random samples from the set of relevant and irrelevant documents, and evaluate the quality of both of these sets by manual review of the samples. The sample size is determined by given confidence intervals (CI) and confidence levels (CL), see, e.g., (Israel 1992). If the sampling test is passed the user can proceed to generate reports, otherwise, the user can go back and edit the query keywords and continue the ranking process in an iterative fashion.

3 Experiments and Preliminary Results

Our preliminary experimental results using topic-learning methods provide the evidence that topic-learning may be able to improve automatic detection of relevant document sets and can be employed to rank documents by their relevance to a topic. This experiment is conducted on the data employed in the TREC 2010 Legal Learning Track (Cormack et al. 2010). The dataset is composed of emails and their attachments from the well-known Enron dataset[2]. TREC has annotated a subset of this dataset against seven sample queries as relevant, irrelevant, and not assessed. We use these annotated subsets after removing the email attachments and non-assessed documents. For example, Query #201 was defined to identify,

> All documents or communications that describe, discuss, refer to, report on, or relate to the Company's engagement in structured commodity transactions known as *prepay transactions*.

Query (#)	Keywords	Relevant (#)	Total (#)	Vocabulary (#)
201	prepay transactions	66	278	12,117
202	FAS, transaction, swap, trust, Transferor, Transferee	191	392	8,693
207	football, Eric Bass	69	222	13,543

Table 1: TREC-2010 Legal Track query dataset statistics, after removing email attachments and non-assessed documents. The query keywords are formed based on (Tomlinson 2010). The vocabularies are created using raw word tokens.

As discussed, we consider a number of different methods to identify an optimal ranking for documents based on their ability to classify them as relevant or irrelevant documents. We evaluate the performance of binary classification of documents using receiver operating characteristic (ROC) curve analysis (Swets 1996). A traditional ROC curve requires a set of data points (in this case documents), some of these are positive data points displaying a property of interest (relevance to the query keywords) and others are negative data points. Each data point is assigned a scalar valued confidence that it is positive. A curve is constructed by varying the confidence value c from its greatest to least value and plotting a curve showing the fraction of true positives (relevant documents with a confidence value greater than c) and the fraction of false positives (non-relevant documents with a confidence value greater than c). A perfect discriminator will yield a curve that goes from the bottom left corner $(0, 0)$ to the top left $(0, 1)$, then to the top right $(1, 1)$. The worst case of detection referred to as the chance diagonal, a straight line from the bottom left corner to the top right. One can use the area under a ROC curve, i.e., AUC, as an estimate of the relative performance of labeling methods.

Figure 2 shows the performance of the ranking methods described above on Query #201 dataset (topic modeling is performed on the raw document words). In this ex-

[2] `http://trec-legal.umiacs.umd.edu`

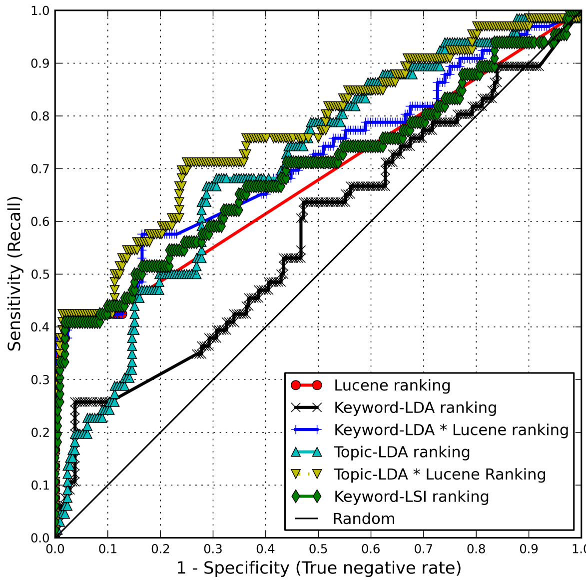

Figure 2: The ROC curves of several ranking methods based on raw word tokens for Lucene indexing and topic modeling on the Query #201 dataset. We use each document's ranking score as the classifier's confidence value to plot ROCs.

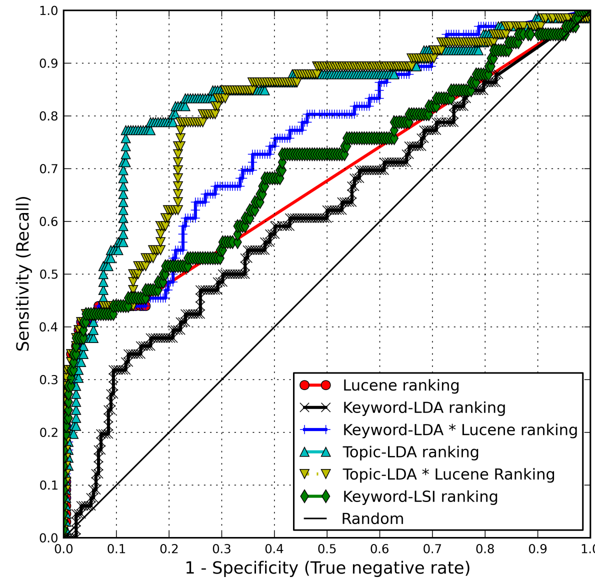

Figure 3: The ROC curves of several ranking methods based on stemmed and lemmatized word tokens for Lucene indexing and topic modeling on the Query #201 dataset.

periment, the configurable parameters, the number of topics T of LDA is set to 30 and the number of components of LSI is set to 200. From Query #201's description, we form a Boolean query —`all_fields:(prepay transactions)`— for Lucene, which will search the keywords "prepay" and "transactions" in all indexed fields in Lucene index. It is clear that Lucene, which uses query-keywords alone, performs relatively poorly. It achieves about a 40% true positive rate (TPR, Recall) with a very small false negative rate (FNR) of about 2%, but learns very little after that. It does not exceed about 42% TPR before it labels every succeeding document with an identical confidence (as shown by the diagonal line leading to the upper right corner). The Keyword-LDA method performs better, achieving a TPR of more TPR 55% with a 10% FNR before making essentially random confidence assignments. The Topic-LDA method does not perform significantly better than Lucene.

It is often the case that different discrimination algorithm will exploit complementary information and we can combine the results of multiple algorithms to yield a better result. We found that using the geometric mean of the Lucene and Keyword-LDA method confidence assignments yields a curve which achieves a 70% TPR with only a 30% FNR. This method also dominates the Lucene approach, and it never displays a lower TPR for any given FNR. The fusion of Topic-LDA with Lucene ranking shows mixed performance, yielding higher TPRs in some parts of the curve and lower TPRs at others.

Figure 3 shows the performance of the methods described above on the same dataset except performing *stemming* and *lemmatization* of document words. We notice an increase in the performance of Topic-LDA and the fusion of Topic-LDA with Lucene ranking ROCs as compared to these ranking methods' ROCs on raw document words. We can also see a decline in performance when we fuse Topic-LDA and Lucene ranking scores, which is due to low Lucene ranking scores. Lastly, we found that the Keyword-LSI method did not perform well in either of these approaches.

To evaluate the performance of the proposed document classifier, which is trained on the manually labeled seed documents, we compute its accuracy on the random samples of the document classification results, i.e., the predicted classes of documents. In this experiment, we use the query datasets described in Table 1. To simulate the manual labeling of seed documents, we create the seed sets by pseudo-randomly selecting documents from a labeled query-dataset. We train SVM models using the TRUE-labeled seed document sets, and apply them on the whole data population to separate them as relevant and irrelevant document sets. Table 2 shows the preliminary results of this experiment.

Query (#)	Accuracy (%)		
	On Seed	Irrelevant Sample	Relevant Sample
201	88.88	89.66	83.72
202	78.78	71.76	61.69
207	88.88	73.17	60.53

Table 2: SVM-based document classifier's results evaluated on the SVM training (seed documents) and testing (the samples of the predicted relevant and irrelevant documents) sets. The sample sizes are computed using the configurations $CI = 2\%$ and $CL = 95\%$.

4 Conclusion and Future Work

In this paper, we describe a computer assisted review workflow to retrieve relevant documents for a legal case based on the principles of topic modeling methods such as LDA and LSI and active learning. We found that ranking models developed based on the documents which are represented in a topic space created via the LDA algorithm give better ranking scores than using the typical keyword-based ranking method Lucene alone, by a study conducted on a small, labeled e-discovery dataset. In addition, we noticed that fusing keyword-based ranking scores with topic-modeling-based ranking scores gives better performances in certain cases. Lastly, the preliminary results on the document classifier models, which are trained using randomly created seed documents, show promising research prospects in active learning-based seed document selection.

Having seen the experimental results we plan to address the following in future: Investigate automated relevance ranking strategies combining Boolean search with topic-learning methods and incorporate these into the ranking model; evaluate the current active learning strategy to select seed documents, capture human ranking feedback, and employ it to further improve our model. Automated relevance ranking involves testing different methods that may help the topic modeling process. Our current approaches of either using the query keywords as a document or selecting top probable topics from its distribution are relatively weak and should be improved.

5 Acknowledgments

We would like to thank ICAIR, the International Center for Automated Research at the University of Florida Levin College of Law, for the generous support for this project.

References

Berry, M. W.; Esau, R.; and Keifer, B. 2012. *The Use of Text Mining Techniques in Electronic Discovery for Legal Matters*. IGI Global. chapter 8, 174–190.

Bird, S.; Klein, E.; and Loper, E. 2009. *Natural Language Processing with Python*. O'Reilly Media.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research* 3:993–1022.

Chang, C.-C., and Lin, C.-J. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2:27:1–27:27.

Cochran, W. G. 2007. *Sampling techniques*. John Wiley & Sons.

Cormack, G. V.; Grossman, M. R.; Hedin, B.; and Oard, D. W. 2010. Overview of the trec 2010 legal track. In *TREC 2010 Notebook*, TREC 2010. USA: TREC.

Dumais, S.; Furnas, G.; Landauer, T.; Deerwester, S.; Deerwester, S.; et al. 1995. Latent semantic indexing. In *Proceedings of the Text Retrieval Conference*.

George, C. P.; Wang, D. Z.; Wilson, J. N.; Epstein, L. M.; Garland, P.; and Suh, A. 2012. A machine learning based topic exploration and categorization on surveys. In *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*, volume 2, 7–12. IEEE.

Hartigan, J. A., and Wong, M. A. 1979. Algorithm as 136: A k-means clustering algorithm. *Applied statistics* 100–108.

Israel, G. D. 1992. *Determining sample size*. University of Florida Cooperative Extension Service, Institute of Food and Agriculture Sciences, EDIS.

Jones, K. S. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* 28:11–21.

kCura. 2013. Workflow for computer-assisted review in relativity. In *EDRM: White Paper Series*. EDRM. –.

Lewis, D. D. 2011. Machine learning for discovery in legal cases.

Lucene, A. 2013. The Lucene search engine.

Řehůřek, R., and Sojka, P. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 45–50. Valletta, Malta: ELRA.

Salton, G.; Wong, A.; and Yang, C. S. 1975. A vector space model for automatic indexing. *Commun. ACM* 18(11):613–620.

Settles, B. 2009. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Swets, J. A. 1996. *Signal detection theory and ROC analysis in psychology and diagnostics: Collected papers.* Lawrence Erlbaum Associates, Inc.

Tomlinson, S. 2010. Learning task experiments in the trec 2010 legal track. In *TREC*.

Vapnik, V. 1995. *The Nature of Statistical Learning Theory*. Springer.