A Legal Citation Recommendation Engine Using Topic Modeling and Semantic Similarity

Talia Schwartz, LL.B., LL.M.  
LL.M. Candidate  
University of California, Berkeley  
School of Law  
taliashwartz@berkeley.edu

Michael Berger, B.Sc., J.D.  
M.I.M.S. Candidate  
University of California, Berkeley  
School of Information  
mjb@ischool.berkeley.edu

Juan Hernandez, B.A.  
M.Sc. Candidate  
Northwestern University  
McCormick School of Engineering  
juanhernandez2016@u.northwestern.edu

ABSTRACT

Topic models are statistical models that detect themes in text corpora. They can be used in information retrieval to find documents that are "similar" to a query, based on the similarity of the themes in the query to the documents in the retrieval database. Applying such models to the domain of legal research might help in improving the efficacy and accuracy of legal research and writing processes that currently rely, to a large extent, on specialized domain knowledge to conduct human-supervised information organization, query formulation, and document retrieval. In this paper, we suggest a novel approach that incorporates automatic content analysis methods into the legal sphere and applies Latent Dirichlet Allocation (LDA) to assist in case retrieval when drafting legal documents. We develop a prototype, built for a popular word processor, that runs on a fixed corpus of sixty-four thousand United States Supreme Court cases. The tool is called while the user is developing a document, using the document itself to formulate a query. The tool detects the user's writing context to automatically formulate a query, and uses topic modeling methods to recommend relevant legal cases for citation and quotation within that context. The paper offers an initial evaluation of the method by testing performance using paragraphs from existing legal cases and their associated citations, showing that our algorithm provides an overall effective recommender system compared with the traditional manual-human legal querying method.

Keywords

Text analysis; Topic modeling; Latent Dirichlet Allocation (LDA); Legal citation; Legal Corpus; United States Supreme Court; Legal research; Information Retrieval; Document Similarity;

1. INTRODUCTION

This paper reports on a novel implementation of a recommendation algorithm that utilizes LDA in the legal context to provide a legal citation recommendations. Using natural language data processing while benefiting from the multi-topic nature of LDA, the proposed tool is well-suited to legal research and is capable of augmenting the abilities of domain-knowledgeable users.

In common law, precedent-based legal systems, case law is a primary source of law. Legal practitioners, academics, and students all routinely conduct legal research and turn to case law in the pursuit of the most relevant set of facts and applicable legal rules in a court decision or other legal document. Once a relevant document is found, it can be cited in support of an argument, or distinguished as inapplicable. The legal recommendation system reported in this paper, was thus motivated by the desire to use text analysis techniques to complement traditional legal research with an effective (accurate and relevant) and efficient (low time and effort) search tool. The extensive amount of digitized legal text available for search presents the problem of an ineffective legal search, a problem with which scholars have been trying to cope for several decades [8]. Another issue that the presented tool is intended to address is the human “tendency toward the familiar”, as was also reported in the legal context [25]. Indeed, the incorporation of text analysis methods into Law and Artificial Intelligence has been recognized as a pressing need [2]. A study conducted in 2007 on the American case law network found a highly skewed distribution of authorities in the Web of Law; 2% of U.S. Supreme Court citation network comprise 56% of all citations. [27]. The ultimate meaning of this “rich get richer phenomenon”, results in legal practitioners analyzing the legal sphere via a highly limited lens [9].

Previous attempts at using automated means for legal research have sometimes been met with difficulty due to technological limitations in both hardware and software. But with recent advances in both hardware capabilities and new algorithmic techniques, the technology is ripe for further exploration of legal search tools. Such tools have the additional potential of normalizing the distribution of case law usage and authority in the Web of Law. The need for an efficient legal citation tool is complemented and reinforced by the existence of advanced data mining and text analysis techniques as well as a more open, digitized environment from which a legal corpus of data can be drawn. The LDA algorithm is of particular interest as it can produce higher-dimension topics/clusters than other topic modeling and clustering algorithms that have been applied to this problem in the past.

To meet this end, a system was developed which recommends a case that is relevant to a text query that has been written in natural language. The described system runs on a data set of the United States Supreme Court decisions, which consists of 63,547 documents.

The rest of the paper unfolds as follows. The second section provides an overview of previous work utilizing text analysis techniques in various field—and in particular, with regard to legal information extraction and retrieval systems and methods. Next, the methods used in our proposed model are described, namely, the application of topic modeling / Latent Dirichlet Allocation to a legal corpus. The fourth part provides our results, including..
results from a novel automated testing method devised by the authors to test the practical performance of the tool built. In the last part, conclusions and further research are discussed.

2. PREVIOUS WORK
The use of probabilistic topic models to identify systematic patterns and commonalities in data has applications in diverse fields, from medicine [1] to political science [12][17]; and is implemented for such purposes as text mining [19], social network analysis [6], customer purchasing behavior [14], language modeling [3], and others. The application of collaborative topic modeling to a recommendation system for scientific articles [31] is a utilization of LDA that shares some features with the approach taken in this paper. The application of text analysis methods in the legal sphere is limited, albeit not novel. Automated content analysis and semantic interpretation of legal text has been implemented for the purposes of case summary [20][21] legal outcome prediction [2]; identification of justices’ ideology [16] or quantifying the “complexity” of a written opinion [22]; determining consensus among justices [23]; attributing authorship in unsigned court opinions [18]; and as a litigation support system in the field of Online Dispute Resolution (ODR) [10].

The presented recommendation system involves several lines of research within Law and AI, namely, knowledge representation, legal retrieval systems, and information extraction using various text analysis methods. The SALOMON system detects relevant legal text by word-frequency techniques and performs retrieval based on similarity measures, while acknowledging the potential of learning topics by the grouping of text [20][21]. Other models extract semantic information and classify legal text by pre-defined concepts [5], or suggest retrieval based on the k-Nearest Neighbors approach [2][24]. Natural Language Processing techniques have been also implemented in automatic classification of legal case factors [2][32] and in extracting ontologies from legal text [15]. A recent work by El Jelali et al. introduces an information retrieval system that uses machine learning and natural language processing techniques to match disputant case descriptions with court decisions [10]. Text mining and analysis have been implemented outside the academic realm as well, in aiding legal work from a practitioner’s perspective. There are several designated Information Retrieval systems in the field of Online Dispute Resolution (ODR)1.

As has been done in more recent work [10], the tool reported here uses natural language inputs rather than pre-defined, constructed, template-based methods, which were more common in older systems. Critically, the model presented here differs from previous work in the application’s methods and goals. To the best of our knowledge, there is no previous published work that utilizes LDA in the context of the legal domain in order to build a legal document recommendation system. The advantage of LDA over many other methods that have been attempted before lies in its modeling of documents as distributed over multiple topics, rather than a single topic, as is assumed in clustering approaches such as K-means and K-medoids. In modeling legal decisions as topic distributions, and topics as distributions over words, LDA accounts for the multi-topic nature of a court case (a single case, or even a single part of a case, may naturally address several legal topics.) Another advantage of LDA is that it learns the relevant item representation in a fully unsupervised, automatic fashion, eliminating the need for manually curated topics or human-enabled feature extraction. Moreover, a significant portion of former work on legal information processing and retrieval focused on semantic inferences that could be gleaned from the legal text. While the suggested topic model algorithm does embody semantic similarity measures [26] rather than a word-based approach [10], in its simplicity, it overlooks the complexity of judicial language. In other words, the proposed tool uses text-analysis techniques that take into account similarity measures while being “agnostic” to words’ linguistic meanings, differently from other work [8][10][15][19][32]. Another feature that distinguishes the work reported here is a practical testing technique, which automatically analyzes 1000 queries extracted from a United States Supreme Court Case database.

3. METHODS
Our aim was to design a tool for one type of use-case that is common among legal researchers—i.e. the task of finding a case to cite that is relevant to a portion of text that has already been written in a legal document. Although this is not the only conceivable use-case for a legal research tool, it is a fairly common one, and has the potential to be modified to support other uses.

The methods we employed are directed toward this use-case, and can be summarized as a development phase and a testing phase: first, we build a recommendation engine using LDA that suggests legal citations that are most relevant to a given query from a user (that query consisting of one or more words of natural language text); and second, we test the recommendation engine using an automatic process that simulates the expected use-case. In this model, “relevance” is defined by topic similarity: the most relevant document is the document that is most similar in topics to the text query. In the first phase, we build several different models by varying certain parameters, such as K (the number of topics) and alpha (a hyperparameter used in LDA, described below).

In this section we outline the data used, the steps used for processing data, the method for approximating LDA, and our testing framework.

3.1 Latent Dirichlet Allocation
LDA is a hierarchical Bayesian generative model, in which documents are represented as a mixture of a limited set of topics, where each topic is characterized by a distribution over words [3]. The model, which allows the representation of documents in a reduced feature space consisting of K dimensions (rather than a larger space that contains as many dimensions as the number of unique words in a given corpus) is thus used as a method of unsupervised machine learning to discover latent topics that are hidden in a text.

LDA can be represented by plate notation to describe the generative process wherein each document in the corpus is generated by picking a multinomial topic distribution \( \theta \sim \text{Dirichlet}() \) for the document \( d \). Each word of the document is assigned a topic from the topic distribution. Given the topic, a word is drawn from the multinomial word distribution \( \beta_k \sim \text{Dirichlet}() \) from the dictionary for that topic.

The method used in our model for approximating the LDA is the Online Learning algorithm, described by Hoffman et al., a variation on the Expectation Maximization approach [13]. The Online Learning algorithm is streamed and runs in constant memory with regard to the number of documents and can make use of a distributed system, which allows it to be implemented on a much larger corpus.

### 3.2 Data

#### 3.2.1 Data Source

Our data set is the entire corpus of United States Supreme Court decisions available on courtlistener.com, which consists of 63,547 court decisions. The data span a period of over two centuries of deliberations by the United States Supreme Court, from 1754 to 2014. For the purpose of creating our models we sampled fifty percent of the mentioned corpus, meaning, about 32,000 documents, to form a modeling corpus. Once a model was learned from the modeling corpus, topics were inferred on the entire corpus used in the training model, and these topic distributions were used to make the recommendations that we then tested, as described further below.

#### 3.2.2 Data Preprocessing

As part of preprocessing, data cleaning for LDA consisted of removing punctuation and stop words, in which each document in the corpus was transformed into a “bag of words” representation. Moreover, we created two different corpora to test whether or not stemming helped to reduce noise and produce LDA models that represented topic distribution more accurately.

### 3.3 Creating and Evaluating Models

The modeling corpus is divided into a training set - consisting of 80% of the corpus selected randomly (approximately 26,000 documents) - and an evaluation set consisting of the remaining 20% of the corpus. To create each model, a set of \( K \) topics are first learned from the training set by applying Latent Dirichlet Allocation [3] to learn topics. Each model was then evaluated for perplexity based on the remaining evaluation set, as described in section 4.2. The LDA algorithm uses two user-determined parameters, \( \alpha \) and \( K \), which alter the sparsity of the topic distribution and the number of topics, respectively. The best model for which we had the computational resources to implement was then chosen using the perplexity measurement described in the Analysis section below. The best model built on the modeling corpus is then applied among the entire corpus of documents to infer the topic distribution of each document therein, representing each document as a vector of length \( K \). This model underlies the recommendation engine - the topic vector of each document, as computed by the model, is stored in a database.

### 3.4 Retrieving Relevant Citations

In order to perform the information retrieval task, the engine receives a query consisting of a text input automatically extracted from the user’s draft document. It then searches for relevant documents as follows: (1) the user’s text is transformed into a \( K \)-dimensional vector representing that query’s topic distribution using the same LDA model learned from the training corpus; (2) the similarity of the query topic vector to the topic vectors of each of the documents in the database is computed, and (3) the documents are retrieved as citation recommendations in descending order of their semantic similarity, as measured by a distance measure between the topic distributions.

We compare four different distance measures: Kullback-Leibler (KL) Divergence,

\[
KL(p, q) = \sum_{i=1}^{T} p_i \log \frac{p_i}{q_i}
\]

Information Radius (IR),

\[
IR(p, q) = \sum_{i=1}^{T} p_i \log \frac{2 \times p_i}{p_i + q_i} + \sum_{i=1}^{T} q_i \log \frac{2 \times q_i}{p_i + q_i}
\]

Hellinger Distance (HD),

\[
HD(p, q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{T} (\sqrt{p_i} - \sqrt{q_i})^2}
\]

and Manhattan Distance (MD),

\[
MD(p, q) = 2 \times (1 - \sum_{i=1}^{T} \min(p_i, q_i))
\]

In all the above formulae, \( p \) and \( q \) are the topic vectors to be compared, and \( p_i \) and \( q_i \) are each of the co-ordinate elements of the respective topic vectors.
These distance measures were used to create a document-to-document similarity coefficient following the example of Dagan et al. (1997) and Rus et al. (2013)

$$\text{SIM}(p, q) = 10^{-\delta R(c,d)}$$

Another version of the retrieval algorithm weighs the similarity coefficient alongside a measure of the retrieved documents' popularity (i.e. the number of times each document has been cited before). The weighting is accomplished by multiplying the similarity by a coefficient of between 0.8 and 1.2, scaled to the total distribution of document citations. The coefficient boundaries were selected empirically to be conservative in order to prevent overweighting—i.e., to prevent the popularity measure of a document from having an excessive effect on the ranking and overpowering the similarity measure.

3.5 Testing the Recommendation Engine

Besides the standard evaluation methods for topic models, we set up the following practical testing scenario as another method to determine the optimal parameters, as well as test the relevant performance of our two\(^1\) corpora: We randomly selected 1000 paragraphs from cases in the US Supreme Court database, each of which contains some text and a citation to another Supreme Court decision. The citations were then separated from the paragraphs, and each of the paragraphs were used as query text to be input into the recommendation engine. Using the above-described algorithm, we retrieve a sorted list of recommended citations for each test paragraph. Of course, we removed any citations that were not dated earlier than the document from which the test paragraph was taken. Our metric for performance was the position of the actual cited document in the recommendation ranking. Ideal performance would be when the number one recommendation in the ranking matches the actual citation that had been originally cited in the paragraph. We compared how well each of our models, similarity measures, and weighting with popularity ranks the citation found in the paragraph. The optimal combination of parameters will be used in the tool implementation.

4. RESULTS AND ANALYSIS

4.1 Model Parameter Choices

We developed two sets of LDA models with 15, 30, 50, 100, 200, 300, 400, 500, 600, 700, 800, and 900 topics, one set using stemming and another set without stemming words. In addition to comparing the number of topics, we tested the difference in results from selecting $\alpha$ as 50/k (a rule of thumb following Griffiths and Steyvers, 2004) and dynamically learning $\alpha$ from the data.

4.2 Model Evaluation

As described in the Methods section above, each model was evaluated in an initial evaluation stage. This was done by calculating the per-word likelihood and perplexity, using the 20% holdout of the modeling corpus as an evaluation set. Intuitively, perplexity is an indicator of the ability of a given model to predict the next word in a given document, and the lower the perplexity, the better the model is at doing so [30]. More formally, perplexity is derived from log-likelihood of unseen documents and is calculated as follows, where $w$ is the set of unseen documents (and $w_d$ refers to each document in the set), $\Phi$ is the topic matrix, and $\alpha$ is Dirichlet prior of the topic distribution:

$$L(w) = \log p(w|\Phi, \alpha) = \sum_d \log p(w_d|\Phi, \alpha).$$

$$\text{perplexity(test set } w) = \exp \left\{ \frac{-L(w)}{\text{count of tokens}} \right\}$$

The figure below shows the results of our evaluation of the perplexity of the models described above. Since automatic alpha parameter setting was computationally more expensive and yielded no relative benefit, we discontinued it.

Although the apparently superior models contained over 500 topics, computing resource restrictions permitted us only to do the next phase of the testing with a 200-topic model by the publication deadline. Additional testing with the models of greater than 200 topics is currently being performed, with more results expected in the coming weeks.

4.3 Recommendation Engine Testing Results

Using the 200-topic model in the engine, we tested the recommendation performance of the engine using the procedure described in the Methods section above. This additional procedure measured the ability of the recommendation engine to retrieve the citation that we knew was in some way relevant to a given text query paragraph (because it had been cited by the Supreme Court in relation to that paragraph). In the graph below, we show how the length of the query paragraph (“Document Length”) affected the ranking of the “correct” citation returned by the recommendation engine (“Minimum Ranking”).

\(^1\) (1) Corpus with stemming, and (2) Corpus without stemming.
The graph above shows that performance stabilizes as the query length exceeds 250 words. These results have implications for recommendation tool design: using a 200-topic model, a recommendation tool built on this engine should extract 250 words of context from the user’s document in order to retrieve better results.

We investigated the outlier points in the boxplot and found that they represented documents from the data set where the parser failed to parse a query paragraph correctly. Including query cases of every length, as well as outliers, our initial overall quantile results were as follows, where the top row shows the percentile and the bottom row shows the minimum ranking of the cited document:

<table>
<thead>
<tr>
<th>Percentile</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
<th>60%</th>
<th>70%</th>
<th>80%</th>
<th>90%</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>2</td>
<td>8</td>
<td>24</td>
<td>53.6</td>
<td>135.5</td>
<td>286.8</td>
<td>632.9</td>
<td>1223.8</td>
<td>2924.9</td>
<td>37854</td>
</tr>
</tbody>
</table>

Focusing on query cases that were of length greater than 250 words, we report the quintile results below:

<table>
<thead>
<tr>
<th>Percentile</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
<th>60%</th>
<th>70%</th>
<th>80%</th>
<th>90%</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>0.6</td>
<td>2</td>
<td>3</td>
<td>6.4</td>
<td>13</td>
<td>27.2</td>
<td>59</td>
<td>174.6</td>
<td>703.2</td>
<td>9970</td>
</tr>
</tbody>
</table>

Thus, for queries of length greater than 250 words, the recommendation engine built on the 200-topic model weighted with popularity returned the original citation associated with that query in its top 3 results 30% of the time, and in the top 59 results at least 70% of the time. These results are very encouraging for an initial test, and we are now iterating on this design by engaging additional computing resources to test the models we created that had a larger number of topics, K.

4.4 Illustrative Query Results

To illustrate more intuitively how the recommendation engine was able to find the matching document to many of the query paragraphs, we provide an example of both a matching and non-matching document to a single query.

The graph below shows the probability distribution of the top 20 topics in an example query (blue), and compares them with the distribution over the same 20 topics in the highest ranked document returned by the engine (pink). As can be seen, the topic distributions are fairly well matched:

Compare this with the next graph, which shows the same query compared with a low-ranked document returned from the engine. As can be seen, the top 20 topics in the distribution for the comparison document (pink) are barely found in the probability distribution for topics in the query (blue):

5. CONCLUSIONS

To the best of our knowledge, this was the first attempt of its kind to create a reliable legal recommendation system based on LDA topic modeling, and the first attempt at testing such a system by automated means. As described above, initial results have been encouraging, and further research and testing is currently being undertaken.

The tool’s admittedly circumscribed ability to detect the cited document among its top recommended results is nevertheless worthy of further exploration, given the limited amount of topics in the model we were able to test with our set of computing resources. Additional models with higher number of topics and lower perplexity are currently being tested to see if they improve performance on the automated test set. While results points to the need for more topics in the models, an optimal number of topics should be learned from additional tests.

Additional modifications to the data set in the pre-processing stage, as well as further changes to the ranking method, are also candidates to be considered for improving the system’s overall performance. Moreover, for the purpose of this research, the data set was intentionally left unfiltered. Further research might
consider filtering out some of the documents based on noise (documents with low to non-alphabetic content) and legal relevance (for example, the corpus consisted of a large number of decisions dismissing writs of certiorari with very little content.) Another route for further research and development could consider using existing documents and their citations to learn (using supervised machine learning) the optimal weights for popularity and other metadata available in our corpus to improve recommendations.

This work also raises additional interesting avenues of research exploration in the domain of testing. Whereas the automated testing techniques described rely on judicial language, additional testing could be performed to evaluate the tool’s performance given text queries generated by laypeople, students, or legal practitioners with different linguistic styles. A user experience study could be conducted among legal practitioners to determine the tool’s efficiency (low time and effort), by comparing the legal research user experience with the aid of the suggested tool versus traditional legal research and writing methods. Furthermore, testing is required among testers with domain specific knowledge to evaluate the effectiveness of the system, in terms of the recommendation accuracy and relevance. By collecting user data, this information could also be used to optimize the recommendation, using supervised learning, in a similar way to the calibration framework suggested above.

Yet another direction of future research could be the substitution of alternative and newer methods of topic modeling, such as Heirarchical Dirichlet Process (HDP) and Nested Chinese Restaurant Process (nCRP) [4][28] in an attempt to contribute to the sophistication of the model and its accuracy, by implementing/learning topic hierarchies, rather than flat lists, within the legal texts.

Finally, from a normative and policy perspective, additional research should consider the implications of the discussed technology on the legal industry and profession, such as its potential narrowing effect on legal research and writing skills, or its potential in widening the distribution of case law networks.

6. ACKNOWLEDGMENTS
The authors wish to thank Brian Carver and the Free Law Project for their generous provision of data. This resource was invaluable in conducting our analysis. We also acknowledge the UC Berkeley DLab, its director Justin McCrary, the Computational Text Analysis Working Group led by Nicholas Adams and Brooks Ambrose, for their excellent support and assistance. Finally, we thank the UC Berkeley School of Information and its Computing & Information Services Unit for their technical and application support, access, and hosting.

7. REFERENCES
Based on the Selection of Representative Objects. *Journal of the American Society for Information Science* 50(2), 151.


