Learning Policy Levers: Toward Automated Policy Classification Using Judicial Corpora

Elliott Ash, Daniel L. Chen, Raúl Delgado, Eduardo Fierro, Shasha Lin *

August 10, 2018

Abstract

To build inputs for end-to-end machine learning estimates of the causal impacts of law, we consider the problem of automatically classifying cases by their policy impact. We propose and implement a semi-supervised multi-class learning model, with the training set being a hand-coded dataset of thousands of cases in over 20 politically salient policy topics. Using opinion text features as a set of predictors, our model can classify labeled cases by topic correctly 91% of the time. We then take the model to the broader set of unlabeled cases and show that it can identify new groups of cases by shared policy impact.

1 Introduction

A vast amount of research on judicial decisions aims at elucidating their cause and impact. In this light, judges are generally not perceived as passive 'oracles of law', as William Blackstone had suggested (see Posner (2011)), but as human beings that are influenced by their ideological beliefs (Sunstein et al., 2006) or strategic preferences (Epstein et al., 2013; Stephenson, 2009). Federal judges do not only interpret the law in this reading, but also create it (Epstein et al., 2013). It is further clear that judicial decisions can have wide ramifications. It has for example been demonstrated that

*Elliott Ash, Assistant Professor of Law, Economics, and Data Science, ETH Zurich, ashe@ethz.ch. Daniel L. Chen, Professor of Economics, University of Toulouse, daniel.chen@iast.fr. We thank Theodor Borrmann for helpful research assistance.

the decisions can affect issues as diverse as housing prices (Belloni et al., 2012), birth outcomes (Currie and MacLeod, 2008), or religious sentiments (Ash and Chen, 2017). To analyze these causes and consequences, it is often necessary to classify and cluster a huge quantity of law cases into different policy categories. In this paper, we present a new method to automatically undertake this task.

It is currently still common practice to classify judicial decisions by hand, which can limit the research range (Edwards and Livermore, 2008; Livermore et al., 2016). As an alternative, we propose and implement a semi-supervised clustering model that uses recent advances in natural language processing (NLP) and machine learning to classify and cluster law cases into multiple known and unknown policy categories. Our method builds on previous research in the area of law and computation which has so far mainly focused on topic modeling (cf. Ash et al. 2018b; Ganglmair and Wardlaw 2017; Livermore et al. 2016).

We take a more targeted approach than topic models that may be more useful for policy research. Rather than grouping cases by topic, we classify them with regards to their policy decisions. The novelty of our approach consists in handling documents from both known and unknown categories. Our model, a Distributed Bag Of Word version of Paragraph Vector (PV-DBOW), first decides whether text documents fit existing policy classes. If they do not match any of them, the model is further applied to the set of unclassified cases and automates the clustering of these cases by policy, using a smart initialization. In our illustration of the method, our semi-supervised clustering method correctly recovers 15 out of 22 labeled policy topics.

To evaluate our model, we use another set of topic labels, external and unseen by the training model, as well as a number of featurizations, including simple bag of words, doc2vec, and word2vec. Our model fares well in classifying documents into specific categories and in creating new ones. Using opinion text features as a set of predictors, our model can correctly classify labeled cases 91% of the time. It thus significantly outperforms another tested approach which our appendix includes for comparison.

This method could be useful for automating policy analysis using legal corpora. Once the clustering process is successfully completed, the researcher can use the random assignment of judges to federal courts as an external instrument to determine the downstream impact of judicial decisions. This instrumentation has already been widely applied to analyse causes and effects of decisions (cf. Ash and Chen, 2017; Belloni et al., 2012; Chen et al., 2016).

The trained policy classifier could in principle be taken to other corpora besides the

one it is trained on. Using policies detected in federal courts, for example, one could try to recover the policy topics in state courts. Similarly, one could detect policies in legislation, or even in non-legal documents such as newspaper articles.

The rest of the paper is organized as follows. Section 2 reviews some of the related literature. Section 3 describes our data sets, while Section 4 describes the feature extraction from those data sets. Section 5 looks at modeling. Section 6 reports our results. Section 7 concludes.

2 Literature Review

The empirical analysis of reasons and results of judicial decisions poses severe methodological challenges, mainly due to the high number of cases and their complexity (Livermore et al., 2016). In the past decade, legal scholars have therefore increasingly drawn on computational methods and have used advances in quantitative text analysis.

The dominant approach in the computational study of law is topic modelling. Topic models like latent dirichlet allocation (LDA) automate the coding of texts by generating probability distributions over the vocabulary ('topics'). In the area of law, topic models have already been widely applied. Livermore et al. (2016), for example, deploy LDA to understand the content of Supreme Court decisions. Leibon et al. (2016) apply a network model to capture the geometry of Supreme Court cases. Similar examples can be found with regards to constitutional archetypes (Law, 2016), constitutional change (Young, 2012), or the effects of electoral systems (Ash et al., 2017b), to name just a few. Figure 1 depicts the confusion matrix of the paper by Ash et al. (2017b), demonstrating the classification success of an ideological topic model. In their paper, the authors use a supervised model to classify 300,000 political speeches into 44 categories, achieving an accuracy of 48.4%. Given the relatively large amount of topics, this number is quite remarkable. In the presented matrix, an even higher accuracy is achieved by merging adjacent categories into larger ones, yielding a total of 19 categories.

Although there is an abundant literature on topic modelling, legal scholars have so far largely neglected NLP techniques. This is unfortunate, as NLP provides an opportunity to deeply delve into the subtleties of human language and to detect preferences and otherwise hidden connections (Ash and Chen, 2018). Word and document embeddings, typical NLP techniques, map texts into a high-dimensional vector space (Le and Mikolov, 2014; Mikolov et al., 2013).

Figure 1: Confusion matrix (Ash et al., 2017b)

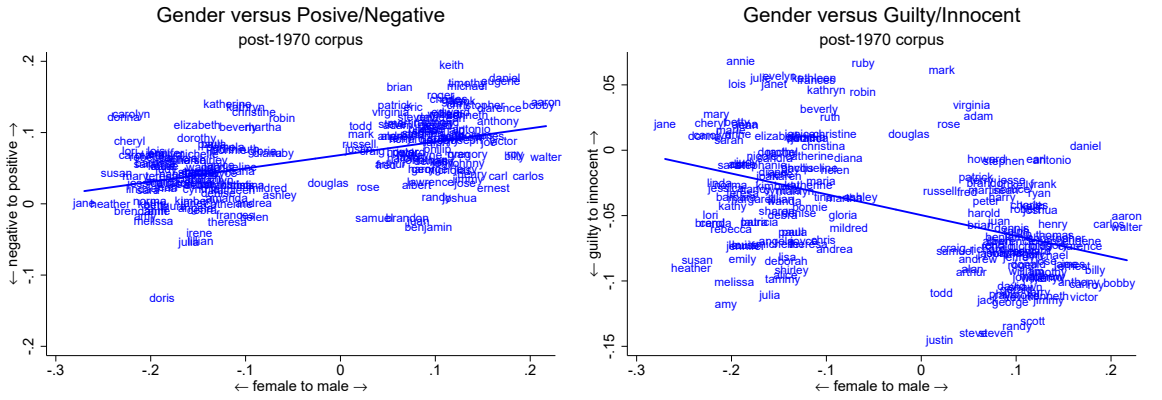
	Admin.	Agri.	Cult.	De-cent.	Econ.	Educ.	Freed.	Intnl.	Labor	Milit.	Natl Way Life	Demos	Other	Party Pol.	Quality	Target	Tech	Morals	Wel-fare	Total True
Adminis-tration	270	5	2	7	77	12	71	7	5	3	4	1	78	11	36	3	11	0	101	**704**
Agricul-ture	5	114	0	3	24	3	7	4	2	0	2	2	23	1	56	0	7	0	22	**275**
Culture	8	3	155	5	22	21	22	4	0	1	19	4	56	3	18	7	17	1	26	**392**
Decentral-ization	27	2	6	73	16	11	24	1	0	1	3	1	32	7	20	0	13	2	22	**261**
Economics	59	12	5	4	715	13	29	16	19	2	11	4	138	30	96	1	49	2	126	**1331**
Education	7	1	9	1	17	461	8	0	0	0	1	4	75	10	10	2	37	3	69	**715**
Freedom	51	0	9	20	41	13	642	19	3	16	12	6	126	34	16	4	14	7	89	**1122**
Interna-tionalism	8	1	4	3	33	2	45	245	2	14	11	1	46	9	19	3	3	0	26	**475**
Labor Groups	11	2	2	0	36	7	12	2	92	0	2	3	20	3	13	0	6	1	40	**252**
Military	7	0	1	0	7	1	24	27	2	118	5	0	32	2	10	0	9	1	13	**259**
Nat'l Way of Life	4	1	5	3	38	3	32	17	2	6	88	6	61	21	19	2	2	4	44	**358**
Non-Econ Demo Grps	7	1	7	0	15	15	30	1	5	2	5	95	36	8	4	5	5	5	104	**350**
Other Topic	38	14	17	13	78	48	86	25	20	12	20	26	1263	39	71	20	48	7	159	**2004**
Party Poli-tics	20	5	5	2	44	2	48	9	4	5	10	0	80	183	25	4	10	2	53	**511**
Quality of Life	21	17	9	4	102	9	13	14	7	3	7	1	112	16	684	0	42	0	34	**1095**
Target Groups	11	2	7	2	7	17	24	2	4	0	8	1	28	4	1	67	8	1	63	**257**
Tech & In-fra	22	8	5	8	57	31	6	6	5	1	4	2	85	1	62	1	413	0	40	**757**
Trad'l Morality	2	0	2	0	6	9	21	2	0	0	8	3	25	4	1	1	0	61	40	**185**
Welfare	33	6	11	3	93	32	62	3	10	1	9	26	142	37	29	11	23	7	1173	**1711**
Total Predicted	**611**	**194**	**261**	**151**	**1428**	**710**	**1206**	**404**	**182**	**185**	**229**	**186**	**2458**	**423**	**1190**	**131**	**717**	**104**	**2244**	

A pioneering paper combining NLP and legal studies is the work by Ash et al. (2017a), who use linguistic features of judicial opinions, combined with features related to the background and education of the judges, to analyze how this impacts the outcome of their decision. The same authors also use NLP to make a deep quantitative analysis of conservative jurisprudence by comparing tokens (words and bigrams) from the judge's opinion and their relative frequency in economic journals and commonplace literature [citation not found - Ash, Chen, and Naidu 2018]. They find, in line with the aforementioned paper, that judges with a language derived from law and economics are more likely to author conservative verdicts.

NLP has also been successfully applied by Ash and MacLeod (2016). Ash and MacLeod seek to measure the effects of changes in judicial electoral processes on judge work quality. They measure quality as an average number of sentences and words per opinion. This index is combined with other measures of quality unrelated to text as data and is used to argue among others that non-partisan elections select better judges.

Yet another research trajectory integrating NLP into legal studies is the work by Ash et al. (2018a) who use word embeddings to measure implicit bias in court decisions. Figure 2 shows for example that judges use a more negative language for female

Figure 2: Implicit gender bias (Ash et al., 2018a)



defendants than they use for male ones, but that they are also biased to assume their innocence. In general, they find that judicial language clearly entails implicit associations between social groups and relevant attributes.

An application of document embeddings, taken from a paper by Ash and Chen (2018), is exemplary presented in Figure 3 . The graph shows document vectors from a data set of US Supreme Court and US circuit court cases ranging from 1887 to 2013. The vectors are represented as dots and are coloured according to their general issue. The plot is centered on the judge interacted with year, and averaged by topic year. As one can see, document embeddings successfully capture linguistic differences across the different areas and are clearly able to distinguish between them.

3 Datasets

We have obtained a deluge of circuit court texts dating from 1880 to 2013, that add up to 326, 554 legal decisions. From these, a small subset of 7, 685 cases have been hand-labeled into 22 legal topics like abortion, racial discrimination, sexual harassment or the first amendment (Randazzo et al., 2010).

It is worth noting that, as seen in Figure 4, the categories "Americans with Disabilities Act" and "Piercing the Corporate Veil" represent a total of 3,778 cases, this is 49% on the whole label sample. The category with the fewest number of cases is "FCC - Chevron/Liberal-conservative", with barely 19 cases, 0.02% of the sample.

Figure 3: Case Vectors by topic (Ash and Chen, 2018)

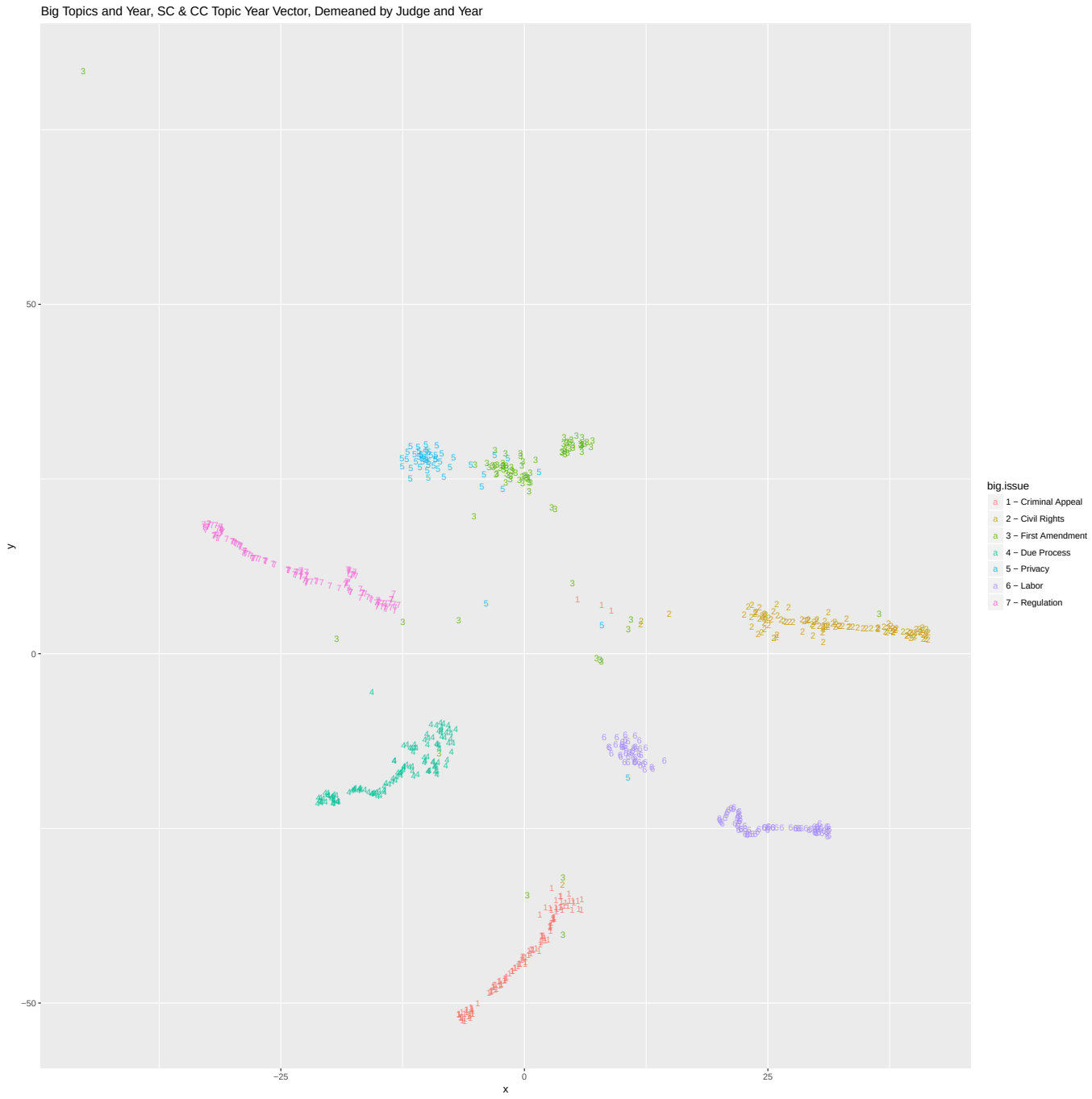
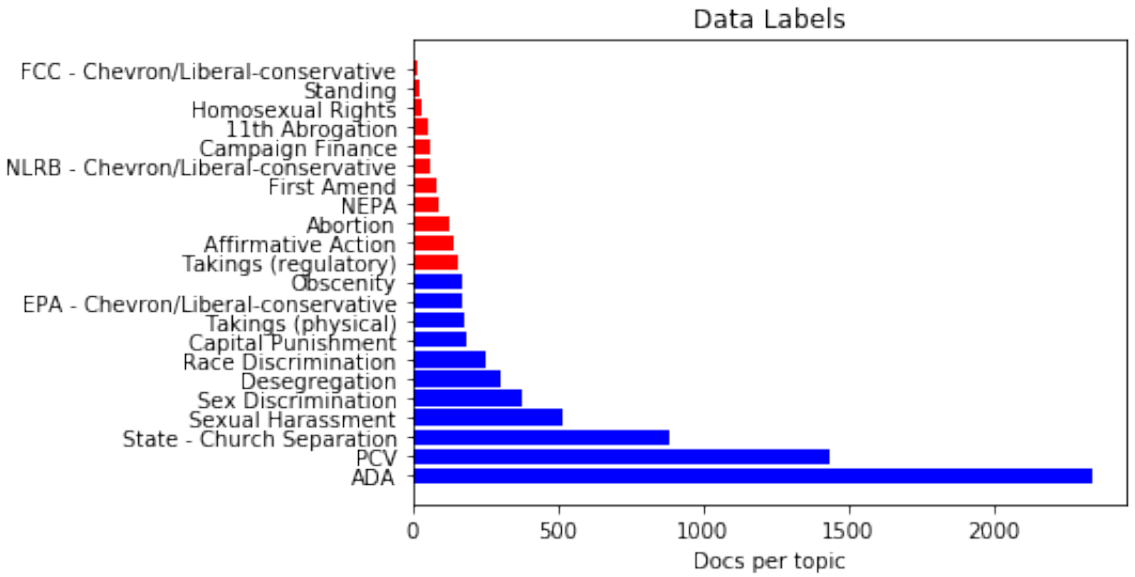


Figure 4: Number of cases by label



This data frame was split into Train, Validation and Test data, making sure the random split process kept at least one case of each category into each split. The final split followed a 60%-20%-20% proportion respectively.

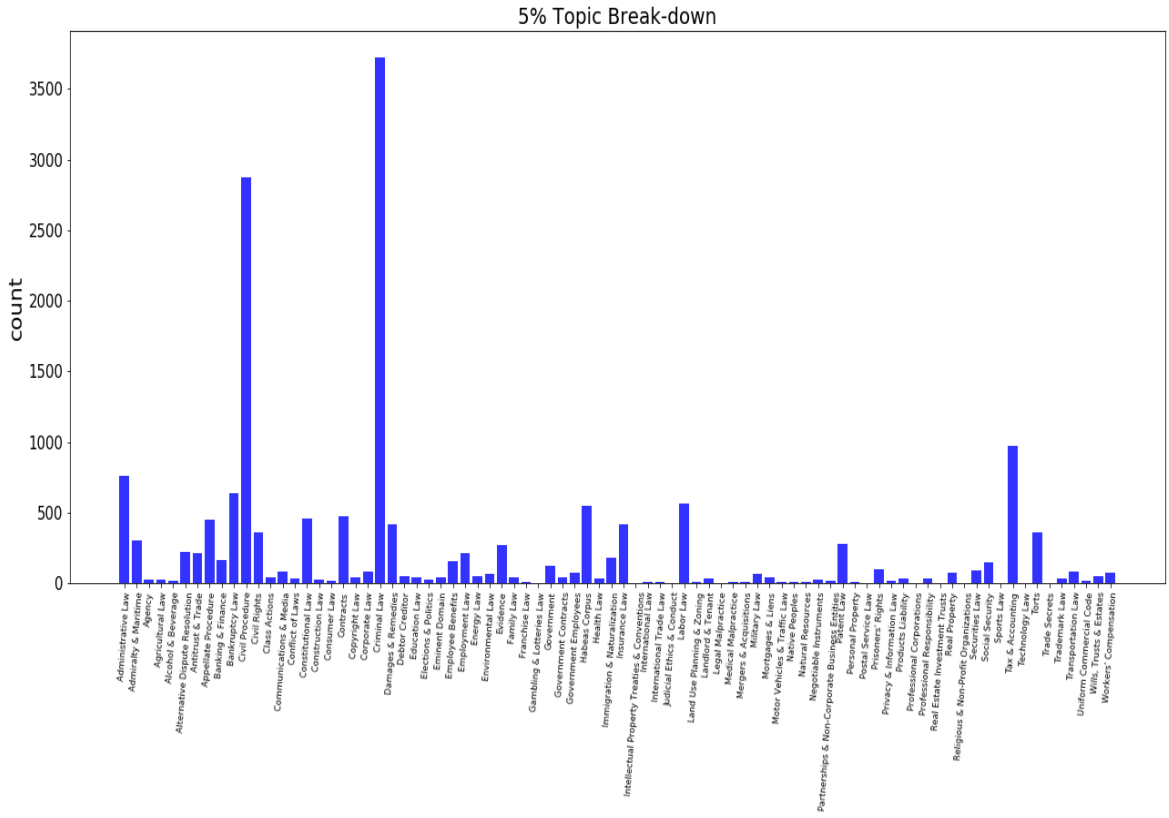
Another source of data, used in this project for the sake of evaluation, is the 5% labeled sample built by Songer, Auburn, and co-authors (Randazzo et al., 2010). This data consists of two different kinds of labels for more than 17,000 cases. We used both the 82 topic labels, which include specific legal areas such as 'Administrative Law', 'Bankruptcy Law', 'Consumer Law' and 'Energy Law' (detailed breakdown shown in Figure 5), as well as nine more general category labels: 'criminal', 'civil rights', 'First Amendment', 'due process', 'privacy', 'labor relations', 'economic activity and regulation', 'miscellaneous' and 'not ascertained' (detailed breakdown shown in Figure 6).[1].

4 Feature Extraction

Three different approaches were tested to extract features from original document texts. We describe each of them below.

[1]General label correspond to the "GENISS" on the original codebook; topic label is provided by Dr. Chen in the updated 5% case coding: Caselevel5Percent_Corrected_Touse.dta

Figure 5: Detailed Topic Label counts of 5% dataset



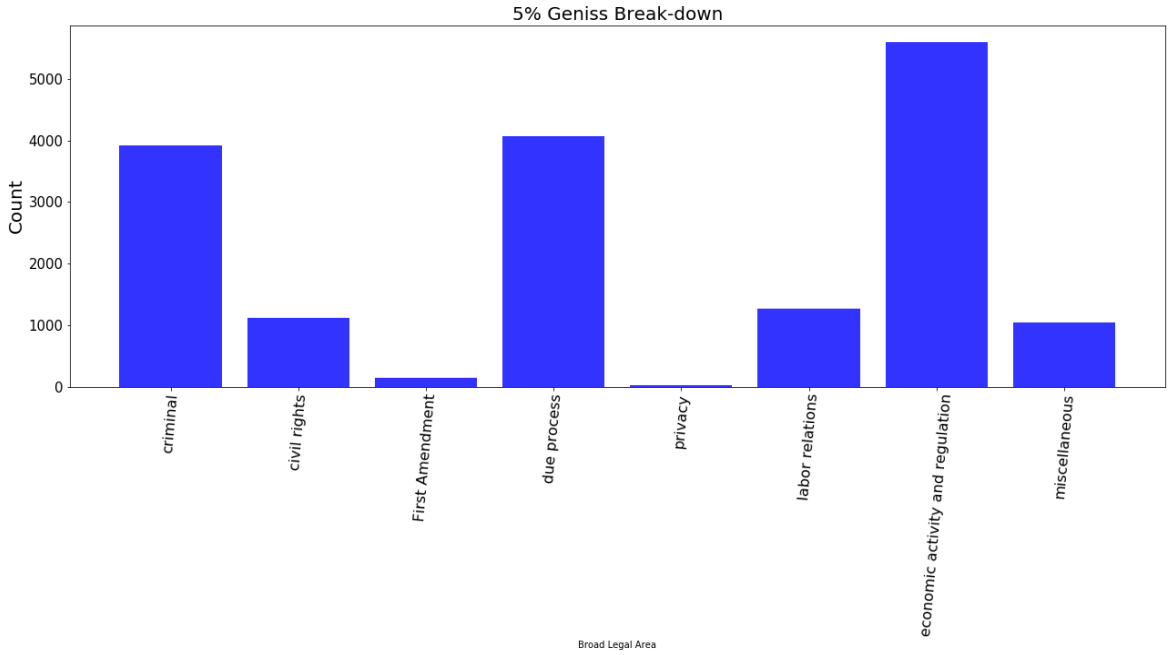
4.1 N-gram Tokenization

Following Ash (2016), we tokenized the texts and represent them as a frequency distribution over the tokens. In this approach, words are standardized (stripped of morphological parts), and n-grams, in this case up to length 2, are extracted. This method, commonly known as Bag of Words (BoW), was used and tested for both the classification and clustering, although this last part with unsuccessful results.

Because of this, a Term Frequency-Inverse Document Frequency (TF-IDF) was tested as a complement of the BoW approach. TD-IDF is a popular feature extraction paradigm for text processing. Inverse Document Frequency (idf) is defined as:$idf(t) = \log \frac{1+n_d}{1+df(d,t)} + 1$, where n_d is the total number of documents, and $df(d,t)$ is number of documents that contain term t. The simplest feature constructed with TF-IDF is the product of raw frequency and idf(t).

After any of these methods, the dataset is transformed into an occurrence matrix,

Figure 6: Detailed Topic Label counts of 5% dataset



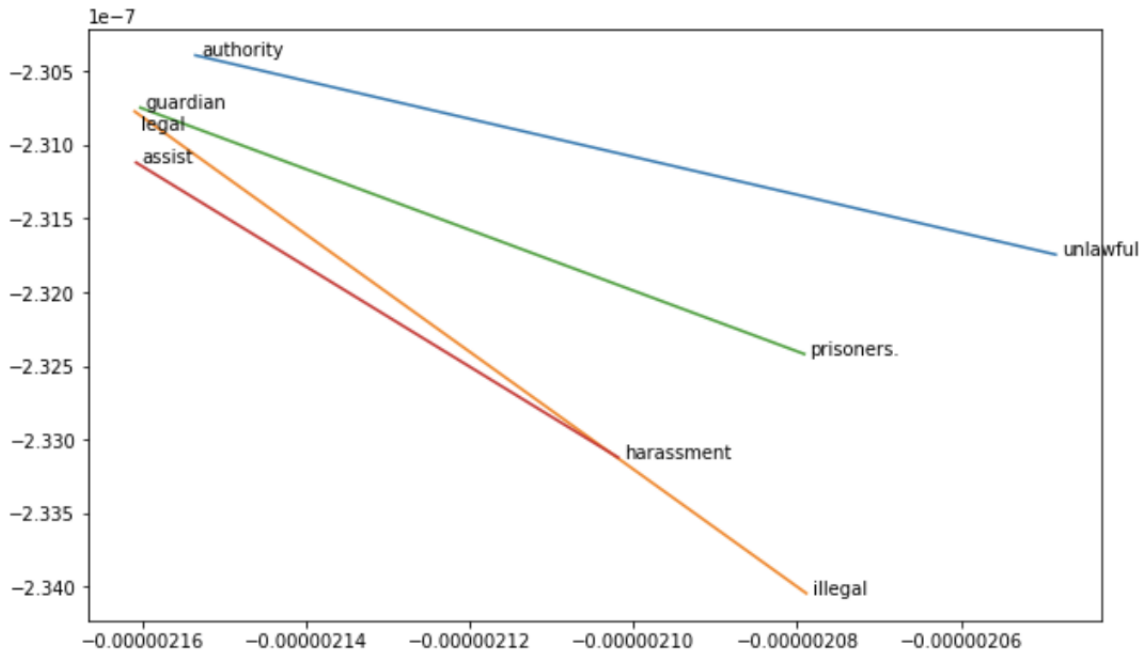
with each legal document as one row, and each 2-gram extracted from above as a column. It is worth noting that, in both cases, the pre-processing strategy was to remove upper cases and stop words.

4.2 Word Embeddings

We used Diffusion Maps over a word co-occurrence matrix to generate dense representations of words in R^{500}. The vocabulary used to build the embeddings were the most common 20,000 words from the training split of the hand-labeled data which included 22 topics. Each word in the vocabulary appeared at least 75 times. In order to build the word co-occurrence matrix we used a symmetric context window of 4 words in each direction. After building the word embeddings, the case embeddings were built by adding up the corresponding vector of each word in the text. We would expect that case embeddings from different labels will point in different directions, allowing a model to classify correctly.. The most time consuming task of this process is the word co-occurrence matrix building; besides that, extracting word embeddings and building case features is relatively fast.

An advantage of this technique is that the embeddings show linguistic similarity, similar words are expected to be associated with similar feature vectors, and semantic structure, simple algebraic operations can be used on word vectors to find analogies. The following plot, Figure 7, shows embeddings from our text sample that have these properties. As one can see, the featurization correctly specifies that 'legal' is to 'illegal' what 'guardian' is to 'prisoners'.

Figure 7: Word embeddings showing semantic structure



4.3 Document Embedding

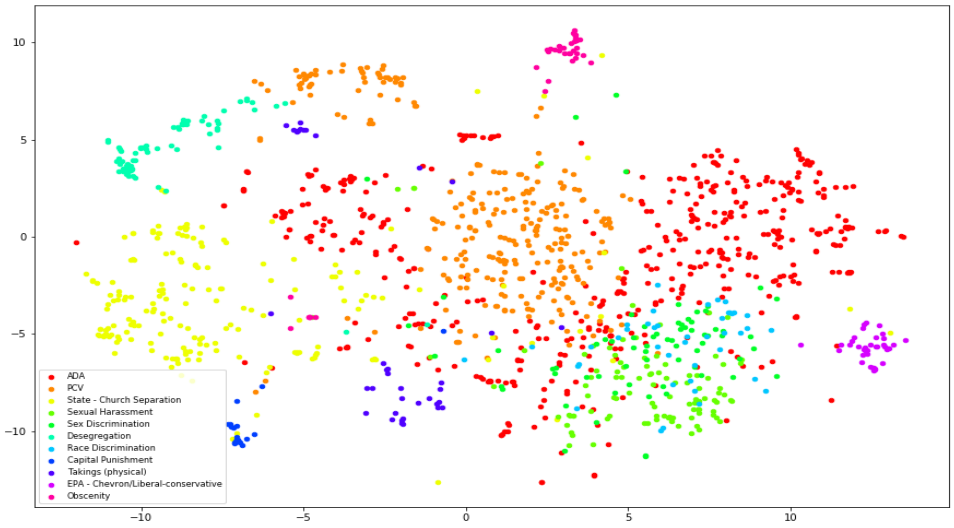
Building up on word embedding, we also used methods proposed in Le and Mikolov (2014) to embed each legal text as a single vector. We tried two variants of neural network based document embedding, per Le and Mikolov (2014).

In the Distributed Memory Model of Paragraph Vectors (PV-DM), a context vector is combined with word vectors as input for a neural network to predict the next word. The context vector is either concatenated or averaged with the word vector.

For the Distributed Bag of Words (PV-DBOW), we randomly chose a word to predict other words in the text window, similar to a Skip-gram model, but with a twist:

the prediction task is conditioned on a paragraph/document vector. Figure 8 shows a two-dimensional visualization of document vectors trained with this method. As we can see, different categories of documents represent distinct groups in the feature space.

Figure 8: Doc2Vec Feature Visualization



5 Model description

As already mentioned, one of the biggest challenges of this project was to create a model that performs an accurate classification, while leaving space for it to create a new cluster, if the cases do not fit sufficiently into one category.

We therefore specify the problem as a two-stage process. First we utilize a part of the training samples and their labels to learn the features, and classification, of some existing categories. Then we cluster the documents, incorporating the features or documents learned in the first stage.

Our semi-supervised approach solves the simple two-steps classification process in the following way:

Supervised classifier. As shown in Figure 4, we merged the least common 11 labels into a single 'Unlabeled' class, and fitted a multiclass (one-vs-all) support vector machine model to classify the "Unlabeled" and the most common 11 categories (making a total of 12 categories). We cross-validated the regularization term and RBF, linear, polynomial kernels for this model, using 80% of 7,638 Chicago Project data points for training and 20% for validation.

Unsupervised clustering. There are two possible approaches to clustering. The first one is hierarchical classification, whereby a spectral clustering algoritm is used to cluster only those instances classified as 'Unlabeled' in the supervised stage. The second approach, the approach we finally adopted for our model, applies smart initialization. For this process, the researcher initializes the centroids/cluster means of unsupervised algorithms (e.g., Gaussian mixture models) with the centroids of labeled data, concatenated with the centers found by kmeans++ algorithm from those classified as unlabeled in the first stage.

We used Doc2Vec features for this approach, and used Bayesian Information Criterion (BIC) as an evaluation method to cross-validate for the best number of clusters. BIC is defined as

$$\log(n)k - 2\log\hat{L}$$

where $\hat{L}$ is the maximized likelihood of the data given the model, n is number of observations in dataset, and k is the number of free parameters. BIC selects the model that maximizes the likelihood of the data but penalizes models with bigger number of free parameters.

We chose the Gaussian mixture (GM) model family for this approach for a very practical reason: scikit-learn has a BIC evaluation implementation for Gaussian mixture models, and allows passing initial model centers. However, BIC can be easily integrated with other clustering techniques, such as k-means, or any model families that fit in the broader Expectation-Minimization paradigm.

The means in the GM models were partially initialized with the mean of each labeled category in the training set. For additional means (number of clusters - number of labeled classes), we used kmeans++ for initialization on the validation data that was predicted to be 'Unlabeled' based on the first stage.

This approach integrates the 1st and 2nd stage of our model in a semi-supervised manner, where the unsupervised model is initialized with information found by the

Table 1: Supervised SVM model performance

Features	**Accuracy (%)**
PV-DM w/ concatenation	46
PV-DM w/ averaging	67.7
PV-DBOW	91
Bag-Of-Words	83.9
Word2Vec	64

supervised classifier.

6 Model evaluation and results

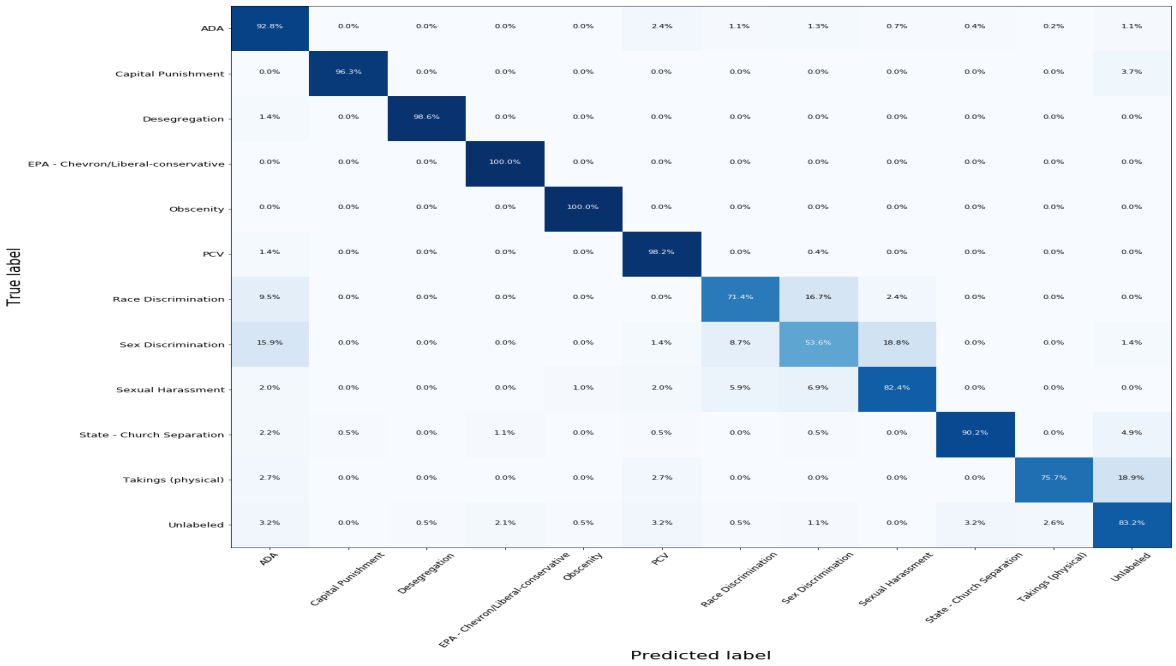
Because of the complexity of the problem, the final model was evaluated by two different datasets, namely the regular test set and the Songer-Auburn 5% sample.

The regular test set comprised 20% of our data which we had deliberately reserved for evaluative purposes. We used the test set to assess both the supervised and unsupervised model. In the supervised stage, our model had to classify cases into 11 policy labels as well an 'unlabeled' category which we later used to test the unsupervised model. All tested models applied support vector machine for classification, but different feature extraction methods, as outlined above in chapter4. As a baseline, simply predicting the majority case would yield 30.5% accuracy.

Table 1 summarizes the performance of different feature sets with Support Vector Machines (SVM) models. PV-DM with concatenation method performed poorly (achieving only 46% accuracy), whereas PV-DM with averaging achieved similar accuracies as simply summing all word embedding vectors for each document as the document vector (accuracy = 67.7%). **PV-DBOW** became the best performer. PV-DBOW uses a window width of 12 words, includes only words that exceed a frequency of 50 in the 7,685 documents, and yields 100-dimensional document vectors. This method outperformed PV-DM, and achieved 91% accuracy in the supervised prediction stage with an SVM model. Word embeddings performed poorly in the supervised stage of the model. A number of models were tested and evaluated on validation data, including Random Forest (Accuracy of 66.73%), Gradient Boosted Trees (69.8%), Ada boost (37%) and SVM (64%). All of them tested in a One-Vs-All multiclass method. As stated before, 2-grams were used in a BoW approach, specifically for the first stage of

the model. The best result achieved, with a one-vs-all SVM model, achieved 83.9% accuracy over the 11 labels and the unlabeled class. In Figure 9, we show the confusion matrix for SVM performance with PV-DBOW features. The diagonal pattern of the figure illustrates the success of the model.

Figure 9: Crosstab for doc2vec SVM prediction & true labels

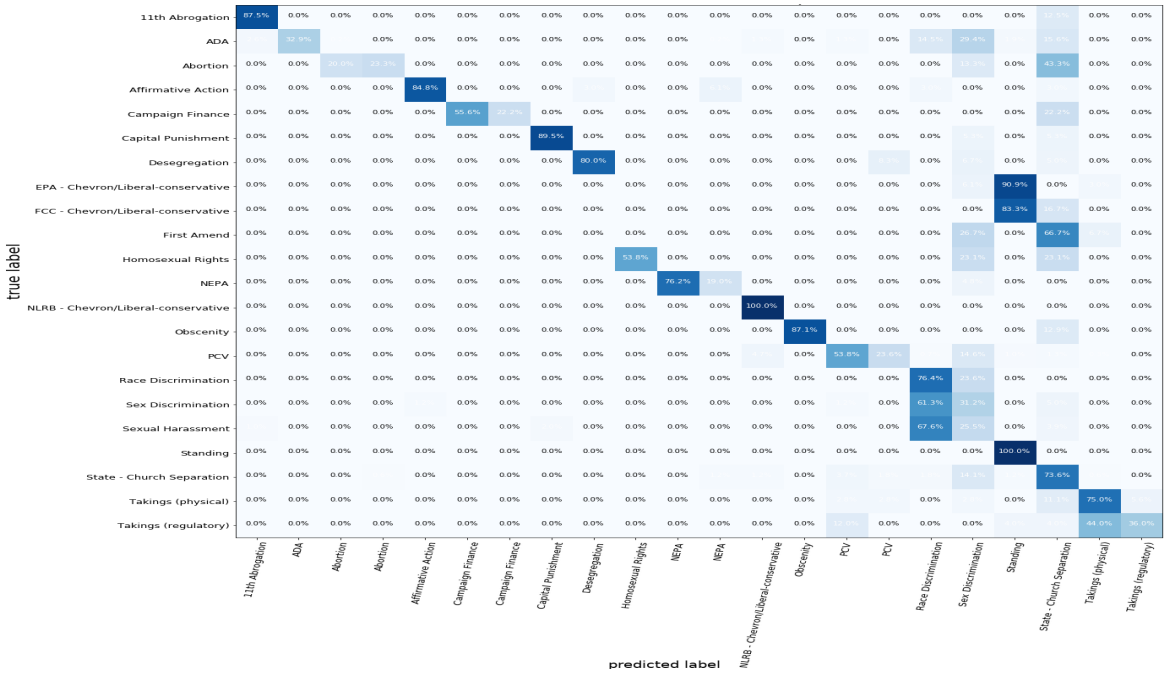


In addition to evaluating model performance with the validation set, we evaluated the clustering results of our model. In the supervised stage, we intentionally lost 11 cases to observe the accuracy of our unsupervised model. We evaluated the clustering of our model by naming the calculated clusters according to the majority class in that label (we used percentage of true label instead of actual count to correct for the unbalanced class size), and by examining the number of true categories (out of 22) that had the most instances fall into the cluster that was labeled with the actual category. Using the validation set with PV-DBOW features, BIC helped us correctly choose 22 as the number of components/clusters. We used the best GM model ('spherical' covariance , 22 components) to predict the categories.

Figure 10 shows the confusion matrix between true labels and GM-predicted labels on the 20% test data. We correctly identified 15 out of 22 clusters. Not only is there a clear diagonal pattern, but the matrix provides some explanation to the mislabeled

categories: we can see that the model often confused race discrimination with sex discrimination and sexual harassment; although the model failed to identify a "First Amendment" class, the majority of First Amendment cases fell in the State-Church Separation class, which is an area under the first amendment to the U.S. constitution. Therefore, both quantitatively and qualitatively, we were assured that the unsupervised model was learning important information about the documents in each legal area.

Figure 10: Crosstab for unsupervised GM performance on test data



In addition to our regular test set, we also used the Songer-Auburn 5% sample to evaluate our model. We trained the best performing models from the regular test evaluation with all of the 7,685 cases from the Chicago Project, and used it to make predictions on the cases classified by the Songer-Auburn 5% sample. Later, a cross tabulation is made between the predicted label and the Songer-Auburn category.

After training an SVM with PV-DBOW features on the whole Chicago Project Data (7638 instances), we initialized the means Gaussian-Mixture models based on SVM result. To get the correct parameters (covariance type, number of components), we used the BIC score to cross-validate. Table 2 presents the parameters we tried — four covariance types and 46 different number of components ($4 \times 46 = 184$) were

Table 2: Cross-validated parameters for Gaussian Mixture Models

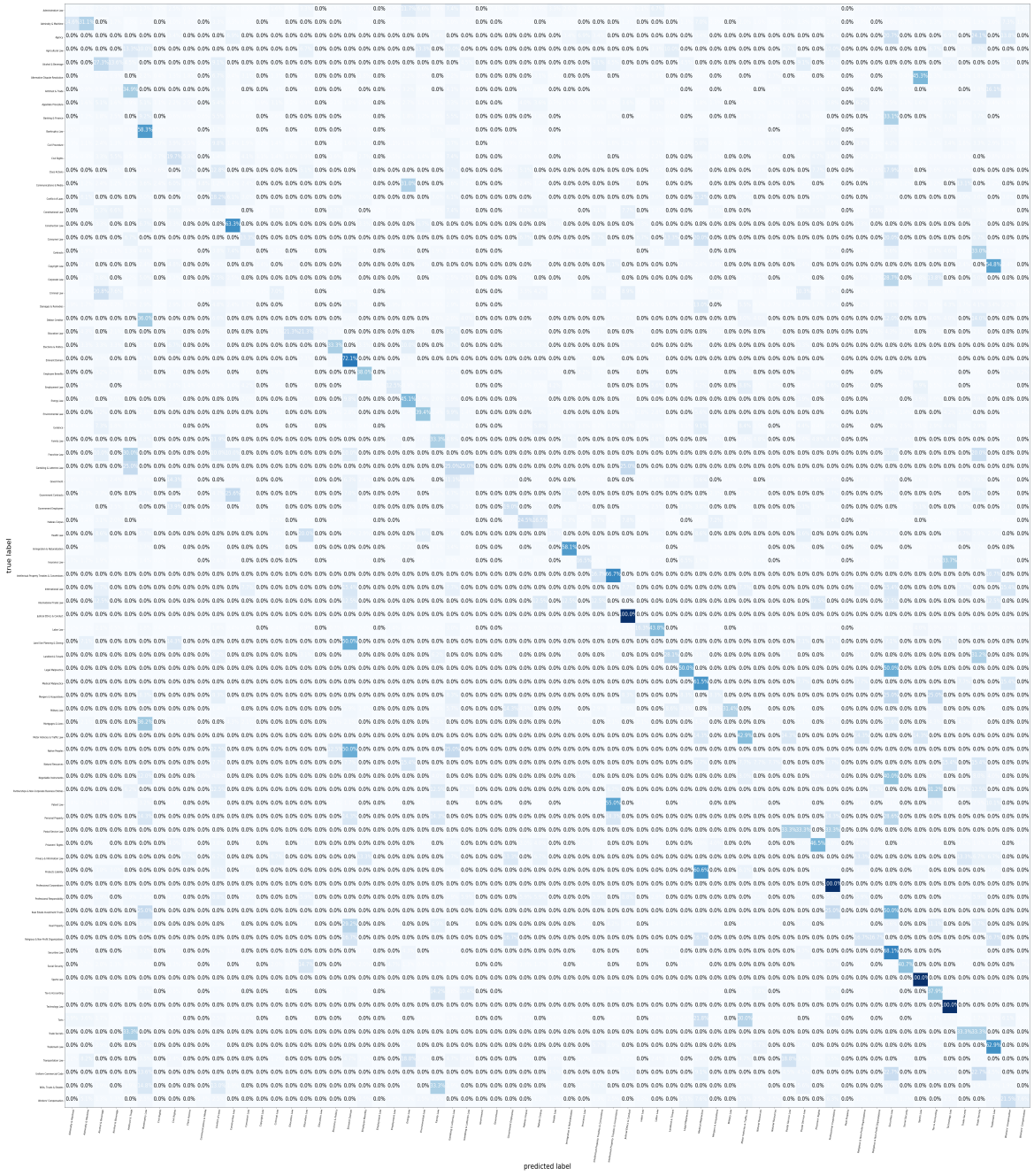
Covariance Type	**No. of Components**
'full' (component has own general covariance matrix)	12, 15, 18, 21, 24
'tied' (components share same covariance matrix)	27, 30, 33, 36, 39
'diag' (component has own diagonal covariance matrix)	42, 45, 48, 51, 54
'spherical' (component has own single variance)	57, 60, 63, 66, 69
	72, 75, 78, 81, 84
	87, 90, 93, 96, 99, 102
	05, 108, 111, 114, 117,
	120, 123, 126, 129, 132
	135, 138, 141, 144, 147

different models attempted. For each Gaussian Mixture model, we used kmeans ++ initialization method to find [number of components - 11] centers for 5% cases that were predicted as 'unlabeled' by the SVM, and concatenated these centers with the 11 centers obtained from the 11 majority labeled cases in Chicago Project Data, making a total of [number of components] centroids. We initialized the Gaussian Mixture model with these centroids as the mean for each Gaussian and trained it on all 5% dataset. Our best performer used 'tied' covariance type and 66 components. For each of the 66 clusters, we named them with the true 'topic' labels that had highest percentage in that cluster. Figure 11 is a cross tabulation between the 'topic' labels and the named clusters. The fact that there is a clear diagonal pattern demonstrates that the model has performed as expected: topic areas were mostly located in the clusters that bear their names. More specifically, 35 out of 82 topic areas had most of their instances fall in the cluster that bear exactly the same name.

7 Conclusions

In this paper, we used document embeddings as a new model for classifying and clustering court decisions with regards to their policies. This semi-supervised model, based on recent advances in machine learning and natural language processing, is able to correctly identify classes known from the Chicago project (Randazzo et al., 2010), and to furthermore discover new clusters whenever documents do not fit into existing ones. Having successfully classified and clustered the cases with the aid of the model, the researcher can hand-label the new clusters and can then use the random assignment of

Figure 11: Crosstab for Gaussian mixture model on 5% data



judges as exogenous instruments to analyze the downstream societal impact of judicial decisions.

While the results from our document embeddings model are promising, a lot of research remains. The model is rather costly to validate and may not be easily generalizable to other courts. Due to computational constraints, we were unable to deploy it on the dataset of 300,000+ cases. Even in evaluating its performance on the 5% Songer-Auburn dataset, we have to keep in mind that the document embedding was only trained on 7,685 documents. This is less than 3% of the documents. We believe that training the embedding on the entire dataset should greatly boost model performance — something we would like to try in the future.

This model is already quite useful for identifying the in-sample policies in other courts. Future work could take the model trained on the U.S. Circuit Courts and apply them to U.S. District Courts, state courts, and possibly other similar legal systems such as Canada, U.K., Australia, or New Zealand. It may also work outside of courts, to look at policies in statutes and regulations. Outside of the law, it might be used to classify the policy content of other types of corpora, such as newspaper articles.

We believe this project sets a new precedent on semi-supervised models, while it proposes a domain-specific way to evaluate unsupervised and semi-supervised models. Further work on it can have promising results into the automatization of categorizing new cases in established categories, leaving space for it to predict new categories when needed.

References

Ash, E. (2016). The political economy of tax laws in the u.s. states. Technical report. 4.1

Ash, E. and Chen, D. L. (2017). Religious freedoms, church-state separation, and religiosity: Evidence from randomly assigned judges. 1

Ash, E. and Chen, D. L. (2018). Case vectors: Spatial representations of the law using document embeddings. 2, 2, 3

Ash, E., Chen, D. L., and Naidu, S. (2017a). The impacts of legal thought. Technical report, NBER. 2

Ash, E., Chen, D. L., and Ornaghi, A. (2018a). Implicit bias in the judiciary: Evidence from judicial language associations. Technical report, Technical report. 4.1, 4.3. 2, 2

Ash, E. and MacLeod, W. B. (2016). The performance of elected officials: Evidence from state supreme courts. Technical report, NBER. 2

Ash, E., MacLeod, W. B., and Naidu, S. (2018b). Optimal contract design in the wild: Rigidity and discretion in collective bargaining. Technical report. 1

Ash, E., Morelli, M., and Osnabrügge, M. (2017b). Electoral Systems and Political Competition: Evidence from the Electoral Reform in New Zealand. 2, 2, 1

Belloni, A., Chen, D., Chernozhukov, V., and Hansen, C. (2012). Sparse models and methods for optimal instruments with an application to eminent domain. *Econometrica*, 80(6):2369–2429. 1

Chen, D. L., Cui, X., Shang, L., and Zheng, J. (2016). What matters: Agreement between u.s. courts of appeals judges. 1

Currie, J. and MacLeod, W. B. (2008). First do no harm? tort reform and birth outcomes. *The Quarterly Journal of Economics*, 123(2):795–830. 1

Edwards, H. T. and Livermore, M. A. (2008). Pitfalls of empirical studies that attempt to understand the factors affecting appellate decisionmaking. *Duke Law Journal*, 58(8):1895–1989. 1

Epstein, L., Landes, W. M., and Posner, R. A. (2013). *The Behavior of Federal Judges*. Harvard University Press. 1

Ganglmair, B. and Wardlaw, M. (2017). Complexity, standardization, and the design of loan agreements. Technical report. 1

Law, D. S. (2016). Constitutional archetypes. *Tex. L. Rev.*, 95:153. 2

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*. 2, 4.3

Leibon, G., Livermore, M., Harder, R., Riddell, A., and Rockmore, D. (2016). Bending the law. 2

Livermore, M. A., Riddell, A., and Rockmore, D. (2016). Agenda formation and the us supreme court: A topic model approach. *Arizona Law Review*. 1, 2

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. 2

Posner, R. A. (2011). Realism about judges. *Nw. UL Rev.*, 105:577. 1

Randazzo, K. A., Songer, D. R., Spaeth, H. J., and Walker, L. D. (2010). U.s. appeals courts database. *The Judicial Research Initiative (JuRI)*. 3, 3, 7

Stephenson, M. C. (2009). Legal realism for economists. *The Journal of Economic Perspectives*, 23(2):pp.191–211. 1

Sunstein, C. R., Schkade, D., Ellman, L. M., and Sawicki, A. (2006). *Are Judges Political?: An Empirical Analysis of the Federal Judiciary*. Brookings Institution Press. 1

Young, D. T. (2012). How do you measure a constitutional moment: Using algorithmic topic modeling to evaluate bruce ackerman's theory of constitutional change. *Yale LJ*, 122:1990. 2

A Classifying with Bag of Words and Spectral Clustering

A.1 Introduction

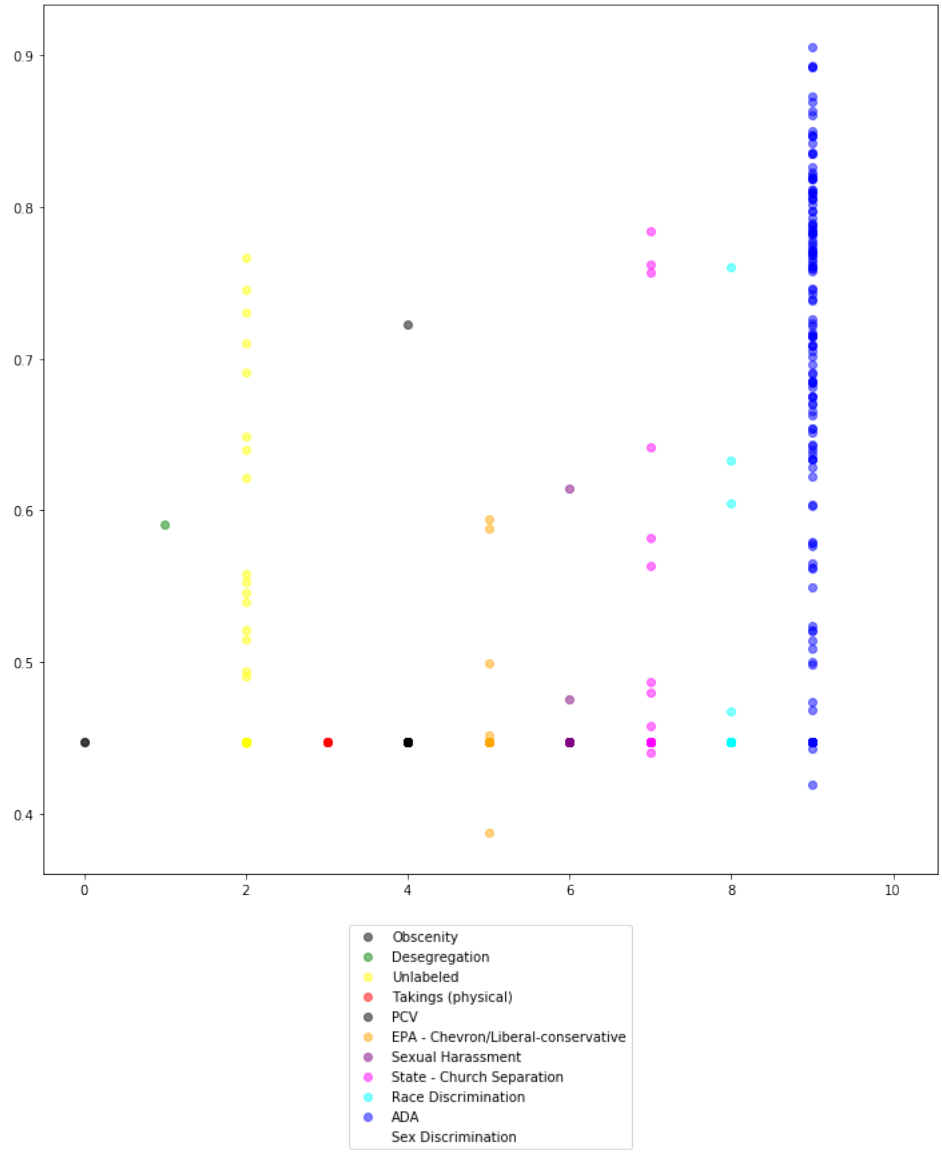
Before we opted for the PV-DBOW algoritm, we tested another model using Bag of Words (BoW) and spectral clustering. In particular, we tried a score or probability classification with the 22 labels of our dataset, and defined a threshold for a case to enter the label. We then performed a spectral cluster analysis using word2vec featurization. Nevertheless, as seen in Figure A.1 , for many of the models tested, there was no clear way to define a label, and most of the predicted cases fell into a very low score.

A.2 Evaluation

In evaluating our model, we used the test set and the Songer-Auburn 5% dataset as well as an evaluation mechanism based on citations.

The results of the test set evaluation were already presented in Table 1 and showed that the BoW approach achieves an accuracy of 83.91%. Even though this is rather

Figure A.1: Multiclass-SVM score for each of the predicted categories on validation data



good, especially when compared with PV-DM or word2vec, it did not perform as well as PV-DBOW.

Using the trained SVM from the Chicago Project Data and the 2-gram featurization, we predicted the labels for all the 17,000 + cases of the Songer-Auburn 5% data. Next, for the predicted "unlabeled" class, we performed a Spectral Cluster using word2vec featurization, specifying the number of clusters according to the `Mean-Shift` algorithm.

From the 17,180 cases used in this evaluation, only 1,233 were classified as "unlabeled". From this, the unsupervised part of the model identified 33 new clusters. As seen in Figure A.2 , the categories "Americans with disabilities Act" and "Piercing Corporate

Figure A.2: Predicted classes and Original Label on 5% sample

	First Amendment	civil rights	class B	criminal	due process	economic activity and regulation	labor relations	miscellaneous	no label	not ascertained	privacy
ADA	32	424	0	1021	48	1779	290	140	7	115	4
Capital Punishment	4	15	0	246	0	42	2	1	0	0	2
Desegregation	14	115	0	401	13	347	149	42	0	4	1
EPA - Chevron/Liberal-conservative	0	5	0	41	2	76	1	2	0	0	0
Obscenity	5	14	0	335	1	21	5	1	0	0	0
PCV	41	208	0	1599	41	5551	437	149	3	12	8
Race Discrimination	4	83	0	12	1	14	9	1	0	0	0
Sex Discrimination	10	150	1	83	7	74	24	14	0	1	1
Sexual Harassment	10	72	0	47	5	37	9	4	0	0	0
State - Church Separation	76	149	0	409	21	264	60	61	0	0	8
Takings (physical)	4	13	0	187	3	174	5	24	0	0	0
Cluster_0	2	5	0	12	0	8	5	1	0	0	0
Cluster_1	2	10	0	10	1	14	3	0	0	0	0
Cluster_2	0	5	0	16	0	1	1	2	0	0	0
Cluster_3	1	7	0	44	0	27	2	6	0	0	1
Cluster_4	0	1	0	0	0	0	1	0	0	0	0
Cluster_5	0	0	0	0	0	1	0	0	0	0	0
Cluster_6	1	1	0	1	0	42	0	0	0	0	0
Cluster_7	2	19	0	0	2	10	3	1	0	0	1
Cluster_8	1	9	0	30	0	11	36	1	0	0	1
Cluster_9	0	10	0	5	3	56	15	7	0	0	1
Cluster_10	1	13	0	0	1	2	2	1	0	0	0
Cluster_11	0	0	0	1	0	10	5	0	0	0	0
Cluster_12	0	0	0	1	0	2	0	0	0	0	0
Cluster_13	0	0	0	0	0	0	17	0	0	0	0
Cluster_14	0	0	0	0	2	26	0	0	0	0	0
Cluster_15	0	0	0	0	0	1	0	0	0	0	0
Cluster_16	0	0	0	2	1	1	1	0	0	0	0
Cluster_17	2	12	0	11	4	14	5	1	0	0	0
Cluster_18	0	3	0	0	1	2	0	0	0	0	1
Cluster_19	1	0	0	0	0	0	0	0	0	0	0
Cluster_20	0	1	0	2	0	8	10	0	0	0	0
Cluster_21	5	1	0	3	5	9	3	4	0	0	1
Cluster_22	0	0	0	0	0	1	16	0	0	0	0
Cluster_23	1	11	0	8	1	9	4	1	0	0	0
Cluster_24	0	0	0	0	0	0	1	0	0	0	0
Cluster_25	3	11	0	10	8	36	11	2	0	0	1
Cluster_26	2	6	0	0	0	0	5	0	0	0	0
Cluster_27	0	1	0	0	0	6	3	0	0	0	0
Cluster_28	13	49	0	48	15	136	79	13	0	0	2
Cluster_29	0	0	0	0	0	1	0	0	0	0	0
Cluster_30	1	1	0	0	1	2	27	0	0	0	0
Cluster_31	4	1	0	2	0	8	4	1	0	0	0
Cluster_32	0	1	0	0	0	9	0	0	0	0	0

Veil" received the most number of cases (69% of all predicted labels). This is, in fact, a result from the uneven distribution of cases. Future investigations should try to down-sample these categories for better predictions. Nevertheless, it is an encouraging fact that the category PCV is mostly composed from the label "economic activity and regulation" from the Songer-Auburn 5% data. In fact, 68.9% of all predicted cases of PCV are originally from it. This is also the majority class for ADA, representing 46% of the cases predicted as "Americans with disabilities Act". It is also encouraging that the four main predicted labels for the "civil rights" category of the Songer-Auburn 5% data fall in ADA, PCV, sex discrimination and State-Church Separation, a result that makes absolute sense.

Another interesting result of this model comes from Cluster 28, which encompasses cases from almost all categories. Our interpretation of this cluster is that it encompasses cases that the model could not fit into a particular class. Other clusters, like cluster

14, receive cases almost exclusively from one category of the Songer-Auburn 5% data, which is also encouraging for further developing this model; or cluster 2, which receives all but one case from the "labor relation" category.

Another approach to evaluate our model was to use the citations of court cases. We have data on the relationship between each case and the previous cases that this case cites. For each case, this data includes a counter object with the citation code (i.e. '342 F.2d 255') and the number of times cited by the original case. Using the full corpus of more that 300 thousand cases, we randomly selected 10,000 cases and predicted the categories according to the BoW and clustering approach described above. This is, 2-gram featurization and a fitted SVM model with the Chicago project labels (11 labels + 1 "unlabeled" class encompassing the 11 least common labels), and a Spectral Cluster with the number of clusters fixed to the algorithm `Mean Shift`.

The model predictions can be seen in Table A.1, 28 clusters were formed and all 11 original categories also receive cases. Again, probably because of the original and uneven case distribution, PCV and ADA receive 67% of all cases.

To evaluate this model, we imported the citations data for all of the 10,000 randomly selected cases and, for each cited case, evaluated in how many categories/cluster each one falls. There are a total of 69,135 cases cited from this subsample of 10,000 cases. Of this 69,135 cases, 18,114 are cited just once (26.2%), and 50% are cited just twice.

From the cases that are cited twice or more, 78.7% fall within a unique cluster/category. Nevertheless, as seen in A.3, this percentage falls fast as the number of cases that include that citation increases. For example, within the citation that are cited only twice, 96.7% fall within the same cluster/category, but among those that are cited three times, this decreases to 45.7%. Although this result can be demotivating at first, it might also be a result that cases that are cited more may also be cases that are referred to by all categories for procedural purposes.

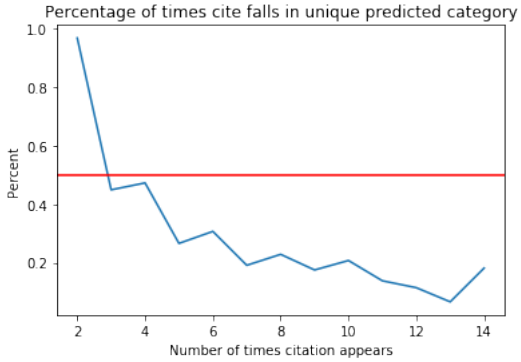
A.3 Model deployment

At the end, we deployed the BoW and Word2Vec model on the whole sample of cases provided. The results of the first stage algorithm (Multiclass SVM) are resumed in Table A.2. The Mean Shift Cluster then found 412 new clusters for the remaining 26,634 cases. After this, the cases classified as 'unlabeled' were clustered using the Spectral Clustering algorithm to fit the 412 clusters. Unfortunately, we were not able to fit this 26,634 into the 412 clusters due to computational restrictions. Future projects

Table A.1: Predictions for the 10,000 randomly selected cases

Prediction	Number of cases		Prediction	Number of cases
PCV	3782		Cluster16	24
ADA	2918		Cluster3	22
State-Church Separation	740		Cluster26	21
Desegregation	405		Cluster9	17
Capital Punishment	303		Cluster0	17
Obscenity	268		Cluster15	16
Sex Discrimination	256		Cluster19	13
Takings (physical)	184		Cluster14	11
Sexual Harassment	136		Cluster7	11
EPA - Chevron	82		Cluster11	10
Race Discrimination	75		Cluster24	10
Cluster6	246		Cluster21	9
Cluster4	70		Cluster17	4
Cluster12	66		Cluster20	2
Cluster13	56		Cluster23	2
Cluster22	54		Cluster2	2
Cluster5	46		Cluster18	1
Cluster25	42		Cluster8	1
Cluster1	40		Cluster27	1
Cluster10	37			

Figure A.3: % of citations that fall within the same category by the number of times it is cited



can look on options to program this in a parallel way, or with computers with greater CPU capacity.

Table A.2: Predictions on the First Stage for the entire dataset (318,869 cases)

Prediction	Number of cases
PCV	122211
ADA	90676
State-Church Separation	22311
Desegregation	14235
Capital Punishment	9727
Obscenity	8880
Sex Discrimination	8239
Takings (physical)	6383
Sexual Harassment	4004
Race Discrimination	2919
EPA - Chevron	2650
UNLABELED (TO CLUSTER)	26634