Legal Issue Spotting - First Phase Legal Analysis Recommendations from Parallel Naïve Bayes Models

John Phillips Stanford University jophilli@stanford.edu

Abstract – Legal analysis is a multistep process that performs the complex tasks of identifying real world activities through the lens of a predefined legal code and sets of laws. Based on a brief understanding of an issue a legal professional must make a recommendation on where to apply limited researching resources.

We employ Naïve Bayes models in a parallel, non-mutually exclusive alignment towards the generic first phase of legal analysis: legal issue spotting. We do this by training separate supervised learning models by first extracting all references to the U.S. Legal Code from each case opinion of the U.S. Federal Reporter, 3rd Series (F.3d) and treating it as a binary label to that U.S.C. section. We then use the same feature set for all binary classification models selected from the most frequently occuring words in the opinions of the U.S. Federal Reporter, 3rd Series (F.3d) after NLP techniques are applied. To test we use 80/20 hold out validation testing and find mixed but not uninspiring results.

Keywords: U.S. Legal Code, law, legal analysis, supersized learning, naïve bayes

INTRODUCTION

Despite efforts in recent years to advance Legal Analysis through various techniques, automated "issue spotting" remains problematic throughout the legal profession. The complex¹ and overlapping implications of legal understandings and a syntactically nuanced legal code create massive challenges for strictly NLP ^{2 3} and syntactic learning mechanisms⁴ and algorithms⁵. Moreover the writing style of different legal professionals (and of clients) present real challenges towards the scaling of any successful model. Despite these challenges, the ambition of advancing legal counsel through automated issue spotting and recommended analysis remains preeminent.

The objective of this project is to create a model (or set of models) capable of ingesting text based scenario descriptions and predicting which areas of the Code of Laws of the United States of America (U.S. Code), (U.S.C.).

We should note that by "issue spotting" we refer to the first step in legal research of identifying possibly legal areas which may have been violated and not the specific logistical aspect of legal research of looking up legal documents - where there is a small number of commercialized catalogues of the U.S. Legal Code which aid researchers once legal areas have been identified for research and where a good number of technological advancements have already occurred.⁶

EXPERIMENTAL DESIGN

A. Background

Prior efforts towards the creation of an algorithm capable of identification of tautological statements within a given text have proved problematic. Nuances of meaning and word associations have prevented unambiguous procedural machine learning efforts to understand a given issue provided its written description. Moreover, when text is compartmented and restricted to registered input values, these associations have proved in some cases overly narrow and thereby restrictive in developing into insight beyond the alternative of direct human observation.

From the standpoint of developing a fully formed legal analysis tool, these results at best could be characterized as having achieved an incomplete mosaic of legal analysis.

Thus, while our initial orientation towards this problem indicated conducting an NLP treatment of the actual U.S. Code itself before applying a machine-learning algorithm against the digested NLP treatment, our investigation of prior research along these lines dissuaded us from continuing down this line of inquiry.

Instead we sought an entirely different dataset which might be distilled into a supervised learning training set(s) and test set(s) and which might help simplify our pre-training data processing. To this end we believe we now have such a dataset, though perhaps more bulky than one would hope.

B. Dataset

Our effort therefore was to find a dataset that highlights a significant number of complex legal topics without losing the contextual flavoring of associated descriptions. In finding such a dataset we aimed to bypass syntactic and referential complexities associated with current impasses in NLP research of the same end state pursuit.

Towards this end we scraped all 491 volumes of opinions of the U.S. Federal Reporter, 3rd Series (F.3d), dating back to 1993 (from this writing in December 2014). In this scraping we found 174,135 opinions enumerated by our target website, with each opinion listing references totaling anywhere from zero to 134 individual (though not unique) references to various U.S.C. sections and subsections.

In total we found 541,311 individual references to 10,735 unique U.S.C. sections and subsections, with the most frequently referenced sections appearing in approximately 10-12% of legal opinions and quickly dropping by the twentieth

most frequently referenced U.S.C. section, to appearing in less than 2% of the legal opinions.

To give a foreshadowing of how we used these references, we might see how the array of references to each of the 10,735 U.S.C. sections can be treated as a binary value vector for a simple Naïve Bayes Multinomial model—developing a unique model for each unique reference destination, with the features coming from word frequencies of the entire corpus of legal opinions.

C. Processing

Before we describe our models, however a more exhaustive description of the original data and processing may be instructive.

Each legal opinion was written to explain the given ruling of a particular case. Paragraphs explain the pertinent events of a given specific set of facts that relate to the case and in some instances these paragraphs are followed by a quick sentence indicating a law and section or sometimes a subsection within the U.S.C. which may be instructive for the reader of the opinion to review. This structure is the heart of constructing our training examples and provides us the type of dataset we sought earlier.

By viewing the words within an opinion as the basis to draw features or 'X' values from, we then may view these short references to U.S. Legal Code sections as destination value or 'Y' mappings.

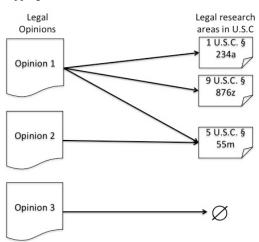


Fig. 1: Opinion to Legal Code law mapping

An example here is illustrative:

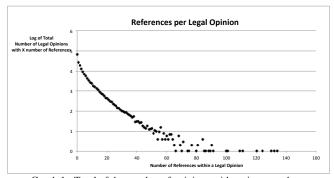
The jury found the defendants guilty of conspiracy to distribute and to possess with intent to distribute cocaine and heroin in violation of 21 U.S.C. Sec. 846 (1988) and possession with intent to distribute and distribution of a controlled substance in violation of 21 U.S.C. Sec. 841(a)(1) (1988). In addition, Thornton and Jones were convicted of participating in a continuing criminal enterprise in violation of 21 U.S.C. Sec. 848 (1988 & Supp. III 1991), and Fields was convicted of using a firearm during a drug trafficking offense in violation of 18 U.S.C. Sec. 924(c)(1) (1988 & Supp. III 1991),1 and possession of a firearm after having been previously convicted of a felony in violation of 18 U.S.C. Sec. 922(g)(1) (1988). All three defendants were

sentenced under the United States Sentencing Guidelines to life imprisonment, and Thornton and Jones were each ordered to forfeit \$6,230,000 to the government pursuant to 21 U.S.C. Sec. 853 (1988). The defendants have not challenged the propriety of their sentences or fines. Nor, significantly, have they alleged that the evidence was insufficient to support the verdicts.⁷

Thus, we can observe that in only one paragraph of one opinion in one volume of opinions we find no less than six distinct laws under the U.S.C., which our hope is to train our model to correctly predict references to.

So by using these individual references from the original opinions and setting each of these 'target' references to be a 'Y' destination for each training example of an opinion we fashion numerous positive examples of each law or sub law referenced in the original opinion.

Moreover, while we may only get a handful of positive examples of a mapping instances to a given law, by applying this method to a large number of opinions and taking care to ensure that do not submit a training example as both a positive and negative example for a given law, then we simultaneously generate numerous negative examples out of each original instance which does not reference the specific legal code section or subsection. We are thus afforded this benefit through submitting each of the examples in parallel for each of the binary decision models for each specific law, which again is dependent on the non-mutually exclusive nature of our recommendation architecture.



Graph 1: Total of the number of opinions with a given number references. Totals are given by log(Total).

Our total number of 10,735 uniquely referenced U.S.C. sections and subsections might then be turned into 10,735 unique models, each using a unique binary value vector based on the references to –or not to- that particular U.S.C. section. (We expand this idea later in our Model description and examine why we chose this over a k-clustering and other models.)

So to address an earlier concern, we can note here that by using the opinions instead of the U.S.C. itself we abstract away from dealing with the NLP issues around the language of various laws and how they interact with given circumstances and rely instead on the word choice of numerous Justices as they write their considered opinions.

D. Concerns

One risk that we will acknowledge here and attempt to address in our conclusion, is that the sum total of opinions – even at its fullest articulation may miss some areas of the U.S. Legal Code to which a user might be concerned.

We know that the U.S.C. is itself rather large and complex: In 2013 the U.S. House Judiciary Committee asked the Congressional Research Service to provide a calculation of the total number of criminal offenses contained in the U.S.C. The CRS responded indicating that they lacked the manpower and resources to provide an update to a number from 2008 of 4,500 total crimes, however the Judiciary Committee Chairman characterized the as growing "at a rapid rate of 500 a decade". So if we allow that the total number of criminal offenses may only make up a subset of U.S.C. sections which an ideal legal analysis function may map to, it is reasonable to be concerned that the overall number of opinions may only generate a subset of training events per U.S.C. section.

While we will attempt to address this concern later, it may be that this weakness of the dataset must simply be endured in it's the models which use it.

MODEL AND FEATURE SELECTION

A. Model Selection

Thus with our dataset given and our concern of one possible risk articulated, we look to select a model which will enable us to make the most use of our data.

With our updated dataset the structure of the binary nature of the mapping and similarity to SPAM filter type problems becomes immediately apparent, and this channels our research towards two clear implementation models: Support Vector Machines (SVMs) and Naïve Bayes 'bucket-of-words'.

The Model we select is a Naïve Bayes word bucket model which a CS229 participant will recognize as the model used for SPAM filtering in the early part of the course. We add one simple change to this model. Instead of selecting a feature set and then training one binary model against that feature set (SPAM/not-SPAM) we train multiple binary recommendation models against the same set of features using separate binary recommendation labels. In theory we would seek to do this for the entire U.S.C. reference list (10,735 unique U.S.C references). However due to time and resource constraints we modeled the top 20 frequently occurring references for this project and this paper.

Additionally, our original ambition was to implement both NB and SVM models and compare the results. However due to the size of the dataset and the constrained working environment of only having one project team member, we had to reduce our goals to accommodate the more immediate task of completing our research on time.

To that end, the Model description is a straightforward Multinomial Naïve Bayes Model, which we used for each of the 20 models we created.

Multinomial Naive Bayes Model:

$$\phi_{k|y=1} = p(x_j = k|y=1) = \frac{\left(\sum_{i=1}^m \sum_{j=1}^{n_i} 1\{x_j^{(i)} = k \text{ and } y^{(i)} = 1\}\right) + 1}{\left(\sum_{i=1}^m 1\{y^{(i)} = 1\}n_i\right) + |V|}$$

$$\begin{split} \phi_{k|y=0} = p(x_j = k|y=0) &= \frac{\left(\sum_{i=1}^m \sum_{j=1}^{n_i} 1\{x_j^{(i)} = k \text{ and } y^{(i)} = 0\}\right) + 1}{\left(\sum_{i=1}^m 1\{y^{(i)} = 1\}n_i\right) + |V|} \\ \phi_y = p(y=1) &= \frac{\sum_{i=1}^m 1\{y^{(i)} = 1\}}{m} \end{split}$$

In theory we would set each of our 10,735 'Y' value laws as separate target filters and allow the collection of the 'Y' values to be non-mutually exclusive, thus creating 10,735 unique models. In practice we use 20 binary value vectors for the top 20 most frequently occurring references for 20 unique models.

Regardless of the number of models, however, we want to enshrine the non-mutual-exclusivity of our dataset and carry that through to our results. We do this with an eye towards our end user as a legal professional and the context of legal research as the user will likely be interested in all types of open legal questions, not simply the highest likelihood given the additive nature of legal complexities. Thus we would not choose a clustering algorithm to evaluate our dataset corpus as the clustering algorithm would simply return the highest likelihood references, not a composite of all likely references.

So with these destination 'Y' values in place, we can now use the entire training set as both positive and negative training instances depending on the nature of the 'Y' mapping for the individual training event and which specific law we are currently training on. And indeed, the complete list of laws would –in theory- be the entire U.S. Legal Code but will -in practice- be the full list of laws referenced in the opinions contained in the full training set.

For training and testing we can see that this amount of data lends itself to a k-fold validation. Though, given the inevitably low number of positive training instances for some referenced laws, we may take note that it make sense to modify our training set, so that our Naïve Bayes model does not become overly negative in its predictions. We will examine this concern more in the results section.

B. Feature Selection

Finally, for our model's feature set we identified the top 2000 most frequently occurring lemmas after we conducted Natural Language Processing to remove punctuation, numbers and uppercase, and after tokenizing and lemmatizing those results.

We made certain to remove numeric values as a simple review of the dataset would provide the concern that that a Naïve Bayes model may identify the actual references of specific legal code section (e.g. '846', '841', '848', '924', '922', and '853' from our earlier example) and attach a higher value on those raw numbers as they might occur more regularly than a randomly occurring number within a random explanation of a legal situation. Moreover this potential problem might have metastasized when we consider non-U.S. Legal Code references that may occur on a regular basis such as individual state laws, or the titles of specific court cases. Both of these appear to occur frequently enough to 'corrupt' our values within our model for those individual words and abbreviation terms. Thus, our removal of integer values from the corpus before obtaining word-bucket frequencies addresses these concerns.

Results table for the top 20 models of the most frequently occuring U.S. Legal Code references within our dataset:

Total Number of Appearances in						
Appealate Course					Test Documents	
Cases Corpus	Legal Code Section	Legal Area (Subjective Description)	Train Set (Pos/Neg)	Test Set (Pos/Neg)	Misclassified	Test Error
21466	42 U.S.C. 1983	Statutue of limitations	25100 (12550/12550)	6276 (3138/3138)	1189	0.1895
19158	21 U.S.C. 841	Prohibited acts A; Drugs	15730 (7865/7865)	3932 (1966/1966)	271	0.0689
18058	28 U.S.C. 1291	Final decisions of district courts	27020 (13510/13510)	6754 (3377/3377)	1554	0.2301
16208	28 U.S.C. 2254	State custody; remedies in Federal courts	12260 (6130/6130)	3066 (1533/1533)	285	0.0930
11354	18 U.S.C. 924	Mandatory Minimums	10878 (5439/5439)	2720 (1360/1360)	230	0.0846
9936	28 U.S.C. 2255	Federal custody; remedies on motion attacking sentence	9996 (4998/4998)	2500 (1250/1250)	680	0.2720
9787	42 U.S.C. 2000	Federal Employment Discrimination; Unlawful employment	7228 (3614/3614)	1808 (904/904)	114	0.0631
9332	18 U.S.C. 922	Unlawful acts; firearms	7544 (3772/3772)	1886 (943/943)	163	0.0864
8391	18 U.S.C. 3553	Imposition of a sentence	5768 (2884/2884)	1442 (721/721)	104	0.0721
5206	21 U.S.C. 846	Attempt and conspiracy	3824 (1912/1912)	956 (478/478)	105	0.1098
5003	8 U.S.C. 1101	Immigration Reform and Control	3542 (1771/1771)	886 (443/443)	15	0.0169
4700	8 U.S.C. 1252	Judicial review of orders of removal	2898 (1449/1449)	724 (362/362)	5	0.0069
4362	18 U.S.C. 2	federal crimes and criminal procedure	4412 (2206/2206)	1104 (552/552)	162	0.1467
4000	28 U.S.C. 1915	Proceedings in forma pauperis	3366 (1683/1683)	842 (421/421)	232	0.2755
3961	18 U.S.C. 3742	Review of a sentence	4480 (2240/2240)	1120 (560/560)	125	0.1534
3699	18 U.S.C. 371	Conspiracy to Defraud the United States	3992 (1996/1996)	898 (449/449)	141	0.1413
3578	28 U.S.C. 1292	Interlocutory decisions	4050 (2025/2025)	1014 (507/507)	128	0.1262
3523	15 U.S.C. 78	Securities Exchange	1636 (818/818)	410 (205/205)	43	0.1049
3490	8 U.S.C. 1326	Reentry of removed aliens	2164 (1082/1082)	542 (271/271)	56	0.1033
3432	29 U.S.C. 1132	Civil enforcement	2266 (1133/1133)	568 (284/284)	23	0.0405

80-20 Hold out on all positve references with negative references obtained randomly to match the positive value total

After this processing we were left with over 48 million individual combinations of an opinion-to-feature word count. For intuition, this set totaled roughly 611MB of data in space delimited .csv file of three integer columns. Thus to process a given models input, we simple reduced the full 611MB to training and test sets of feature data, and paired those sets with their corresponding binary label vectors for the given examples used.

RESULTS

A. Training Difficulty

We originally trained our models by taking the entire dataset and using an 80/20 holdout for each separate training and test set. However, while this method worked for the most frequently occurring referenced models of our twenty models, this quickly presented a problem for the less frequently referenced legal sections' models within our twenty, as they started to predict negative recommendation for all test instances

Considering that our long term goal might be to develop the additional 10,715 models not built for this project and paper, and knowing that all of those models would have a reference level lower than our lowest referenced models, we needed to find a different way to accommodate the training and testing of our low reference models.

To that end we treated the positive references as the rare commodity for each model and created our training and test sets with an eye explicitly towards the number of positive reference values of the entire training set. We would do this by taking the entire positive reference set and splitting it into 80/20 holdout training and test sets, and then matching those positive training events with an equal number of negative training and testing values which we sampled randomly from the entire negative reference sets.

B. Results

Our results were mixed. While no individual model showed a test error greater than .3 many were near (.2755, .2720, .2301). Balancing some model's high error scores, we did have some models score dramatically low with values less than

.05 and in one case below .01. Overall the majority of our error levels fell near .10 .

DISCUSSION

The difference which yielded our spread in test error results might originate from the type of language used within the specific types of case opinions containing a particular reference type. A quick look at the higher scoring models yields the subjects of the proceedings in forma pauperis, ensuring federal custody, and final decisions of district courts. All of these from a subjective vantage point appear to have a highly legal nature to them, which might make them difficult to disambiguate from negative examples. But this intuition may not hold, some of the low error models dealt with topics regarding equally similar legal focus.

Nevertheless, one clear area for future examination is to test and build up a useable stop-word list (possibly through tf-idf processing of the corpus).

CONCLUSION AND FUTURE WORK

Overall the majority of our error levels fell near .10 which, though high for a SPAM filter, might be considered a valuable first contribution given the complexities of legal analysis.

Our initial intuition to use of one feature set for all types of U.S. Legal Code reference models, while valuable from an initial processing standpoint, likely needs to reexamined. It is probably more reasonable to develop frequency lists based off of each positive set of references and with respect to each model (and in combination with a legally focused stop-word list), rather than the entire corpus of Appellate Court ruling opinions.

Nevertheless the use of building multiple models against datasets with multiple parallel labelings seems to hold promise as the variance between our model's error rates implies a shifting detection criteria from model to model, and therefore a confirmation of our intuition that separate reference subjects

elicit word usage of a shifting yet individually model-specific correlative nature.

Moreover as a simple way of 'inverting' the Naïve Bayes 'SPAM' filter from removing negative content to recommending multiple non-mutually exclusive types of content we believe that this type of parallel model usage shows promise.

To revisit a concern briefly touched on earlier, an observer may detect an implicit assumption: that we use the articulated legal opinions of appellate court judges as training data and from the ingested description of events from an unknown source which we will use at test time. In practice this difference may vary considerably. In practice this difference may vary in professionalism, writing ability, and tone thus producing a model bias that is not inherent in the models' SPAM filter cousins which both train on and are applied to equivalent subject emails. More research will be needed to discover if this difference in source type proves vital.

Separately, we likely have some over representation of legal terms in our feature set, which could be removed and replaced with features further down on the frequency distribution. This —as noted earlier—might be accomplished through the use of stop-words during our frequency distribution preprocessing when creating our feature lists.

Finally, to deal with the issue of our dataset not mapping to the entire U.S. Legal Code as a result of an insufficient number of cases applying to that area of the U.S. Code we might attempt to engage an associative clustering algorithm to derive similarity between individuals laws based off of their non-standard terms.

To deal with these low reference values and unmapped laws we might engage a second dataset: the text of the U.S. Laws themselves. With this second data set we might engage an associative clustering algorithm to derive similarity between individuals laws based off of their non-standard terms (a function of tf-idf processing). This output ultimately would feed into a clustering analysis of all U.S. Federal Laws which we use to associate laws to one another. This association of Laws would allow us to thus provide a proximity value for unmapped laws from the first dataset model thereby providing us more laws within our models' 'reach'.

Thus by attaching clustered laws to the output targets from the Machine Learning Model we may be able to map to a much broader segment of the U.S. Legal code.

REFERENCES

- ⁴ Baharudin, Baharum, Lam Hong Lee, and Khairullah Khan. "A review of machine learning algorithms for text-documents classification." Journal of advances in information technology 1.1 (2010): 4-20.
- ⁵ Waterman, Donald A., Jody Paul, and Mark Peterson. "Expert systems for legal decision making." Expert Systems 3.4 (1986): 212-226.
- ⁶ Cormack, Gordon V., and Maura R. Grossman. "Evaluation of Machine-Learning Protocols for Technology-Assisted Review in Electronic Discovery."
- ⁷ United States of America v. Bryan Thornton, A/k/a "moochie", Appellant (d.c. Criminalno, 91-00570-03).united States of America v. Aaron Jones, A/k/a "a", "j", Appellant (d.c. Criminal No.91-00570-01).united States of America v. Bernard Fields, A/k/a "quadir", "q", Appellant (d.c.criminal No. 91-00570-05), 1 F.3d 149 (1993), paragraph 4.

¹ Bommarito II, Michael J., and Daniel M. Katz. "A mathematical approach to the study of the united states code." Physica A: Statistical Mechanics and its Applications 389.19 (2010): 4195-4200.

 $^{^2}$ Lame, Guiraude. "Using NLP techniques to identify legal ontology components: concepts and relations." Law and the Semantic Web. Springer Berlin Heidelberg, 2005. 169-184.

³ Doan, AnHai, et al. "Ontology matching: A machine learning approach." Handbook on ontologies. Springer Berlin Heidelberg, 2004. 385-403