JITP 2011: The Future of Computational Social Science

The Journal of Information Technology and Politics Annual Conference

2011

Tradeoffs in Accuracy and Efficiency in Supervised Learning Methods

Loren Collingwood
University of Washington, lorenc2@u.washington.edu

John Wilkerson
University of Washington

Collingwood, Loren and Wilkerson, John, "Tradeoffs in Accuracy and Efficiency in Supervised Learning Methods" (2011). *JITP 2011: The Future of Computational Social Science*. 4.
Retrieved from https://scholarworks.umass.edu/jitpc2011/4

Tradeoffs in Accuracy and Efficiency in Supervised Learning Methods

Loren Collingwood and John Wilkerson
University of Washington

February 25, 2011

Abstract

Text is becoming a central source of data for social science research. With advances in digitization and open records practices, the central challenge has in large part shifted away from availability to usability. Automated text classification methodologies are becoming increasingly important within political science because they hold the promise of substantially reducing the costs of converting text to data for a variety of tasks. In this paper, we consider a number of questions of interest to prospective users of supervised learning methods, which are appropriate to classification tasks where known categories are applied. For the right task, supervised learning methods can dramatically lower the costs associated with labeling large volumes of textual data while maintaining high reliability and accuracy. Information science researchers devote considerable attention to comparing the performance of supervised learning algorithms and different feature representations, but the questions posed are often less directly relevant to the practical concerns of social science researchers. The first question prospective social science users are likely to ask is — how well do such methods work? The second is likely to be — how much do they cost in terms of human labeling effort? Relatedly, how much do marginal improvements in performance cost? We address these questions in the context of a particular dataset — the Congressional Bills Project — which includes more than 400,000 labeled bill titles (19 policy topics). This corpus also provides opportunities to experiment with varying sample sizes and sampling methodologies. We are ultimately able to locate an accuracy/efficiency sweet spot of sorts for this dataset by leveraging results generated by an ensemble of supervised learning algorithms.

1 Introduction

These days, it seems as though classification algorithms are being applied to almost everything. Some of the most sophisticated and visible are internet search algorithms, which are constantly updated based on user queries and clicks. However, classification algorithms are also used to identify geographical features, potential medical problems, people, and of course text. With the growth of the internet and the wealth of new data possibilities, interest in automated techniques is growing within political science (Cardie and Wilkerson, 2008; Hillard et al., 2008; Hopkins and King, 2010; King and Lowe, 2003; Laver et al., 2003; Lazer et al., 2009; Monroe and Schrodt, 2009). Researchers have classified newspaper articles or internet stories to measure sentiment towards political candidates, and have studied mentions in blog posts and tweets to track public opinion or even happiness (Dodds and Danforth, 2009; O'Connor et al., 2010).

There are many different approaches to automated classification and no single approach is superior to all others. Instead, different approaches have unique advantages and disadvantages. To set the stage for our own work, we will briefly compare dictionary, unsupervised learning, and supervised learning approaches. Dictionary or keyword based approaches take an axiomatic approach to classification. The researcher designates that a specific keyword or combination of keywords implies that an event is of a particular class (Schrodt et al., 1994). Thus, there is never any question about whether an event has been correctly classified. On the other hand, for some tasks, dictionary approaches can be costly because the dictionary must include a mapping for every relevant permutation of the data.

Machine learning approaches to automated classification do not depend on pre-defined rules. *Unsupervised* machine learning methods rely exclusively on data patterns to generate category membership (Grimmer, 2010; Quinn et al., 2010). They let the data do the talking. There are many different unsupervised learning algorithms that make differing assumptions about the underlying data structure. One important benefit of unsupervised learning methods is that they can be used as a discovery tool (Grimmer and King, 2010). A second, not insignificant, advantage is that unsupervised learning methods will categorize a dataset at relatively low cost compared to manual, dictionary and supervised machine learning approaches. Two potentially important limitations, however, are that the resulting categories are empirically rather than theoretically derived (which can raise questions about their validity); and such an approach cannot be used to code an existing classification system to new data.

Supervised learning methods automatically apply an existing classification system to new data. In the first "training" phase, the goal is to build a model that best distinguishes cases that have already been assigned to different classes. A researcher will partition a training dataset so that one portion is used to "train" the algorithm, while another is set aside to "test" its performance. For example, every record in the Congressional Bills Project (www.congressionalbills.org) is a bill's title, and each title is labeled for policy topic (19 major topics). A researcher uses an off the shelf algorithm to build a model to predict the class (topic) of bills based on their "features" (e.g. similarities and differences in words or characters contained in their titles). This training stage is where the heavy lifting is done. The researcher will experiment with different algorithms, different sampling approaches, and different "feature representations" — all with the goal of improving the model's ability to predict the cases in the test set. When performance is acceptable, the model is then used to "classify" other virgin (unclassified) cases (i.e., bill titles).

Unlike unsupervised learning methods, the accuracy of specific labels assigned by a supervised learning model can be validated using the existing labels contained in the training set. This "gold standard" is usually (though not always) a label assigned by a human. In this respect, it is a less objective validation standard than the more rigorous dictionary-based approach of using specific keywords. At the same time, a supervised learning approach may yield high quality results at substantially lower cost, when compared to the costs of developing comprehensive dictionaries.

2 Research Objectives

The primary purpose of this paper is to investigate supervised learning methods as a tool for scholars interested in studying text as data. Although information science researchers devote considerable attention to assessing the performance of different supervised learning algorithms and approaches to feature representation, the questions posed are often less directly relevant to the practical concerns of political scientists. In particular, as these methods become more accessible, we assume that potential users will be particularly interested in the cost/accuracy tradeoffs involved in using such methods when compared to manual approaches. Specifically, supervised learning methods required labeled examples for training purposes. How many labeled examples are required to yield acceptable performance? How much difference in performance can be expected across algorithms? Across topics? Given that labeling of training data is costly, are there more efficient approaches to constructing training datasets besides random sampling? If overall accuracy is not sufficiently high, is it possible to identify the cases that have been classified with high accuracy?

Unfortunately, though perhaps not surprisingly, these questions lack simple answers. Many factors have the potential to affect accuracy besides the number of training examples. In the pages that follow, we first discuss the corpus used in our experiments — the Congressional Bills Project corpus. Second, we briefly describe the four supervised machine learning algorithms used, as well as the standard pre-processing steps of word stemming and stopword removal. We then turn to the cost versus accuracy equation.

In Part I one of the analysis, we control for potentially confounding variables unrelated to the algorithms or sample size: duplicate records and unequal training samples across topics. Many datasets include duplicate records to varying degrees; we therefore de-duplicate the training data prior to the analysis. In addition, we stratify the training sample so that in each experiment, there are an equal number of training examples for each topic. We then compare the performance of the 4 algorithms for different sample sizes. While we fully expect labeling accuracy to improve as sample size increases, important questions about baseline accuracy, the marginal benefits of larger samples, and variations in performance among the algorithms remain to be answered.

Researchers using supervised learning methods often have target performance levels that exceed what a particular algorithm is able to achieve. For example, overall algorithm accuracy for a dataset may be 70 percent, whereas human inter-rater reliability is 85-90 percent. One response to such a deficit is to increase the size of the training sample on the assumption that overall accuracy will improve. However, an alternative approach is to leverage information from the ensemble of algorithms to set aside the cases that have been labeled with high accuracy, and have humans label the remainder. Part I concludes with a set of experiments that illustrate how an ensemble approach can substantially reduce labor costs while maintaining high standards of labeling accuracy.

In Part II of the analysis, we relax the two sampling restrictions imposed earlier. A random sampling approach that allows for duplicate records and variations in training sample sizes across topics is a more realistic reflection of the typical project. Does it matter for the results?

3 Congressional Bills Corpus

Our experiments draw on the Congressional Bills Project (www.congressionalbills.org). The size of the entire corpus is approximately 400,000 bills beginning in the year 1947. We select bills from the 90th-106th Congresses for data management purposes, which yields a total of 229,037 bills. Each bill title in the dataset is assigned one of 20 major topic codes (including "private bill"), and one of 226 subtopic codes drawn from the Policy Agendas Project (www.policyagendas.org). Thus, a bill "To amend the Clean Air Act of 1970" falls into major topic 7 (environment) and subtopic of 705 (air pollution). The coding process is intensive. Undergraduates commit to a year of coding as part of an undergraduate research capstone seminar. During the first academic quarter, 4-8 students label 100 bills for topic each week. The following week, they compare their results to those of the master coder (a graduate student or faculty member intimately involved with the project). Discrepancies are discussed with the goal of further clarifying the intent of the respective topic categories. This process is repeated for approximately 8 weeks while inter-rater reliability between each student and the master coder is monitored.

Importantly, discrepancies do not necessarily imply labeling errors. Two (or more) coders can legitimately disagree about the proper placement of a bill. For example, a bill ending Don't Ask, Don't Tell is arguably related to defense personnel issues (1618) and to civil rights (207). Unlike a dictionary approach, the boundaries between the topics are not objectively defined. Coders must sometimes make a judgment based on general coding principles and model examples.

The general target for the Congressional Bills Project is 85-90 percent inter-rater reliability at the major topic level and 70-80 percent at the subtopic level. Most students achieve these targets by the end of the first quarter. In the second and third quarters they are then given independent coding assignments of about 200 bills per week, while the master coder continues to conduct spot checks to ensure high quality results. (We find other work for students who do not make the targets by the end of the first quarter.) This system has worked well, but it is obviously labor intensive. For this reason we began experimenting with supervised learning methods several years ago. We now rely on these methods to code a large proportion of the 10,000 or so new bills introduced each Congress at levels of reliability approaching that of our human coders.

4 Algorithms and Preprocessing Steps

One of the challenges for many new users of such methods is the availability of accessible tools. Statistical analysis packages such as SPSS and SAS have been instrumental in promoting quantitative research in the social sciences. Similar user friendly packages for classification tasks have lagged behind but this situation is rapidly changing. Open source projects such as `R` and `Python` now include valuable suites of classification tools and Google has announced that it will soon offer 1-day classification services at very low cost (starting at $10 per month for up to 10,000 records) using its highly sophisticated algorithms.

The results reported here are based on the `Rtexttools` package, an `R` wrapper developed for the `Texttools` program, which includes basic pre-processing functionality and 4 machine learning algorithms (Collingwood, 2010). The `Rtexttools` package is designed to lower the costs of using these methods and is publicly available.

`Rtexttools` provides four machine learning algorithms, and pre-processing capabilities. Machine learning algorithms use various optimization techniques to build a training model from the textual documents and then evaluate that model on untrained "test" data. The four algorithms are support vector machine (SVM), maximum entropy, naïve bayes, and ling pipe. We briefly review these algorithms, but for further review see Boser et al. (1992); Hsu et al. (2003) and Cortes and Vapnik (1995).

SVM creates different hyper-planes to divide the data into different groups, or categories. With a training set of document-label pairs $(x_i, y_i), i = 1, \ldots, l$ where $x_i \in R^n$ and $y \in \{1, -1\}^l$, SVM requires the solution to this optimization problem:

$$\min \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i$$

subject to:

$$\begin{aligned} y_i(w^T \phi(x_i) + b) &\geq 1 - \xi_i, \\ \xi_i &\geq 0. \end{aligned}$$

Maximum entropy is an exponential model with a simple intuition: assume nothing about unknown attributes of the data but model all the known attributes. That is, given a set of facts, pick a model that is consistent with those facts, with all remaining information modeled as uniform (Berger et al., 1996). The maximum entropy model has the following exponential form:

$$p(x) = exp\left(\frac{\theta^T f(x)}{Z\theta}\right)$$

with a real parameter vector theta of the exact length as the f(x) feature statistic (Cover et al., 1991).

The naïve bayes classification method uses a similar approach as maximum entropy, however it relies on a naive bayesian assumption of independence (Han and Kamber, 2006). That is, it assumes, probably incorrectly, that all features are independent of one another. However, without this assumption, naive bayes classification would not be possible. Regardless, empirical tests have shown that this classification method is nevertheless effective. If a document, or instance, is described with n attributes (a_i), where i goes from 1 to n, then that instance is classified to a class υ from a set of possible V classes:

$$\upsilon = arg \max_{\upsilon_i \in V} P(\upsilon_j) \prod_{i=1}^{n} +(a_i | \upsilon_j)$$

Lingpipe's classification is based off of an n-gram character language model. These types of models define probability distributions over strings drawn from a fixed set of characters. Probabilities are normalized over strings of a fixed length. The maximum likelihood estimator for this model is:

$$\hat{p}_{ml}(c|\sigma) = count(\sigma c)/extCount(\sigma)$$

where count(σ) is the corpus count of the string σ and $extCount(\sigma) = \Sigma_c count(\sigma c)$ is the count of single character extensions of σ (Carpenter, 2007).

Prior to any automated content analysis, words are normalized by changing all letters to lower case and stripping affixes, a procedure known as stemming. For instance "walking" would be stemmed to "walk". In this way, the algorithms will treat "walking" and "walk" the same, which improves the labeling accuracy of the text. We use the common Porter Stemmer because it supports alternative forms of words and is known to work well in a variety of capacities (Loper and Bird, 2002). Finally, we remove all stopwords from each bill title. Stopwords are high frequency words such as "the" and "also" that have little lexical content and do not help distinguish documents from one another (Loper and Bird, 2002).

5 Analysis Part I

As discussed, supervised machine learning entails a two step process. The first step is an iterative process of training the algorithm using pre-labeled examples (bill titles coded for topic), assessing performance against a set aside test set of (2000) pre-labeled examples, and tweaking the process to improve model performance (for example by increasing the size of the training sample). Once performance has achieved an acceptable level, the next step is to apply that model to cases that have not been labeled. We train the four algorithms, test their baseline performance, and then test how sample size impacts that performance. We also show how the confusion matrix can be used to learn more about where a particular algorithm is performing well and less well. The initial deduped database includes over 150,000 bills, giving us considerable flexibility in experimenting with different sample sizes. We therefore begin by constructing training and test samples of $n = 100$, $n = 200$, $n = 400$, and $n = 1000$ for each of the 20 major topics, which yield total training and test sets of $n = 2000$, $n = 4000$, $n = 8000$, and $n = 20000$, respectively.

5.1 Average Algorithm Accuracy

In terms of overall performance, SVM, naïve bayes, and max ent perform similarly. Using 100 examples for each topic produces 65 percent overall agreement with the human labeled topics in the test set. Ling pipe performs considerably worse for this particular dataset and sample size (54 percent accuracy). As expected, larger samples sizes increase accuracy. Figure 1 illustrates the marginal improvements from the baseline as the sample size increases from 100 examples per topic to 1000. Notably, lingpipe sees the greatest improvement, but its overall performance is still substantially lower than that of the other algorithms.

	n=100	n=200	n=400	n=1000	Difference
Ling Pipe	0.54	0.57	0.62	0.68	0.14
Support Vector Machine	0.68	0.72	0.76	0.79	0.11
Naive Bayes	0.64	0.69	0.72	0.75	0.11
Maximum Entropy	0.68	0.71	0.75	0.79	0.11

Table 1: *On average, the SVM and maximum entropy algorithms outperform the naïve bayes and Lingpipe algorithms, regardless of sample size. However, Ling Pipe improves the most as sample size increases.*

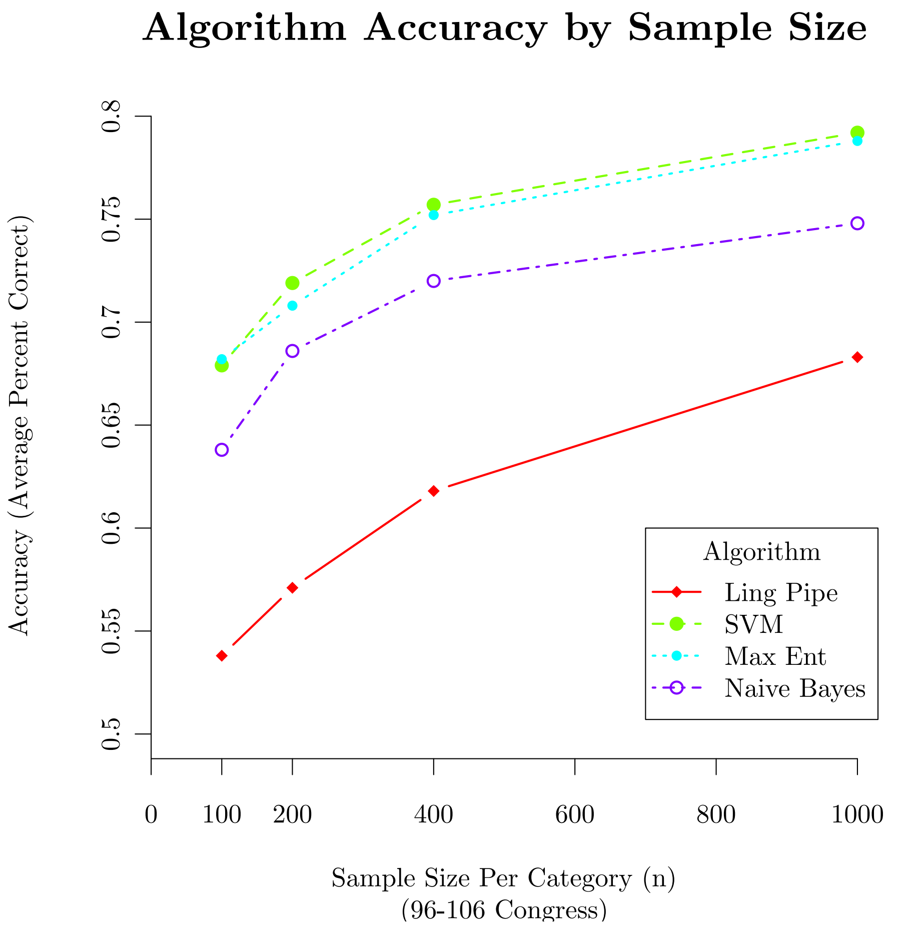

Figure 1: *Normalized samples reveal that the support vector machine and maximum entropy algorithms outperform naïve bayes and lingpipe algorithms, regardless of sample size. For each algorithm, as sample size increases, accuracy improves*

5.2 Category Accuracy by Algorithm

Overall accuracy may conceal considerable variations across topics. Table 2 and Figure 2 illustrate this point. Here, we examine machine recall—as opposed to precision—by individual category. Recall is defined as the percent of cases that are correctly predicted, whereas precision is the percent of predicted cases that are correct. Table 2 looks at accuracy recall across topics for the ling pipe algorithm and reveals that it predicts some topics much better than others (compare banking and finance with private bills). At the same time, the worst performing topics also tend to be the ones where additional training examples yield the greatest improvements. The differences in performance across topics with an $n = 1000$ training sample are much more similar than with an $n = 100$ training sample. Figure 2 illustrates these differences graphically for all four of the algorithms (the figure for ling pipe simply graphs the information contained in Table 2).

	n=100	n=200	n=400	n=1000	Difference
International Affairs	0.42	0.58	0.65	0.69	0.27
Civil Rights	0.49	0.62	0.63	0.72	0.23
Banking and Finance	0.35	0.50	0.50	0.58	0.23
Health	0.48	0.62	0.63	0.70	0.22
Energy	0.52	0.63	0.61	0.73	0.21
Labor	0.42	0.47	0.57	0.63	0.21
Housing	0.51	0.64	0.61	0.71	0.20
Environment	0.55	0.52	0.65	0.70	0.15
Social Welfare	0.55	0.58	0.65	0.70	0.15
Defense	0.51	0.52	0.58	0.66	0.15
Macroeconomics	0.47	0.41	0.55	0.61	0.14
Education	0.55	0.50	0.60	0.67	0.12
Agriculture	0.58	0.65	0.60	0.70	0.12
Law and Crime	0.56	0.52	0.57	0.66	0.10
Public Lands	0.59	0.61	0.70	0.69	0.10
Science and Tech	0.63	0.57	0.62	0.72	0.09
Federal Gov't Ops	0.45	0.38	0.43	0.54	0.09
Foreign Trade	0.63	0.67	0.66	0.68	0.05
Transportation	0.62	0.59	0.65	0.65	0.03
Private Bills	0.87	0.83	0.87	0.89	0.02

Table 2: *The lingpipe algorithm shows dramatic improvement by sample size for a variety of categories including most notably International Affairs, Civil Rights, Banking and Finance, and Health.*

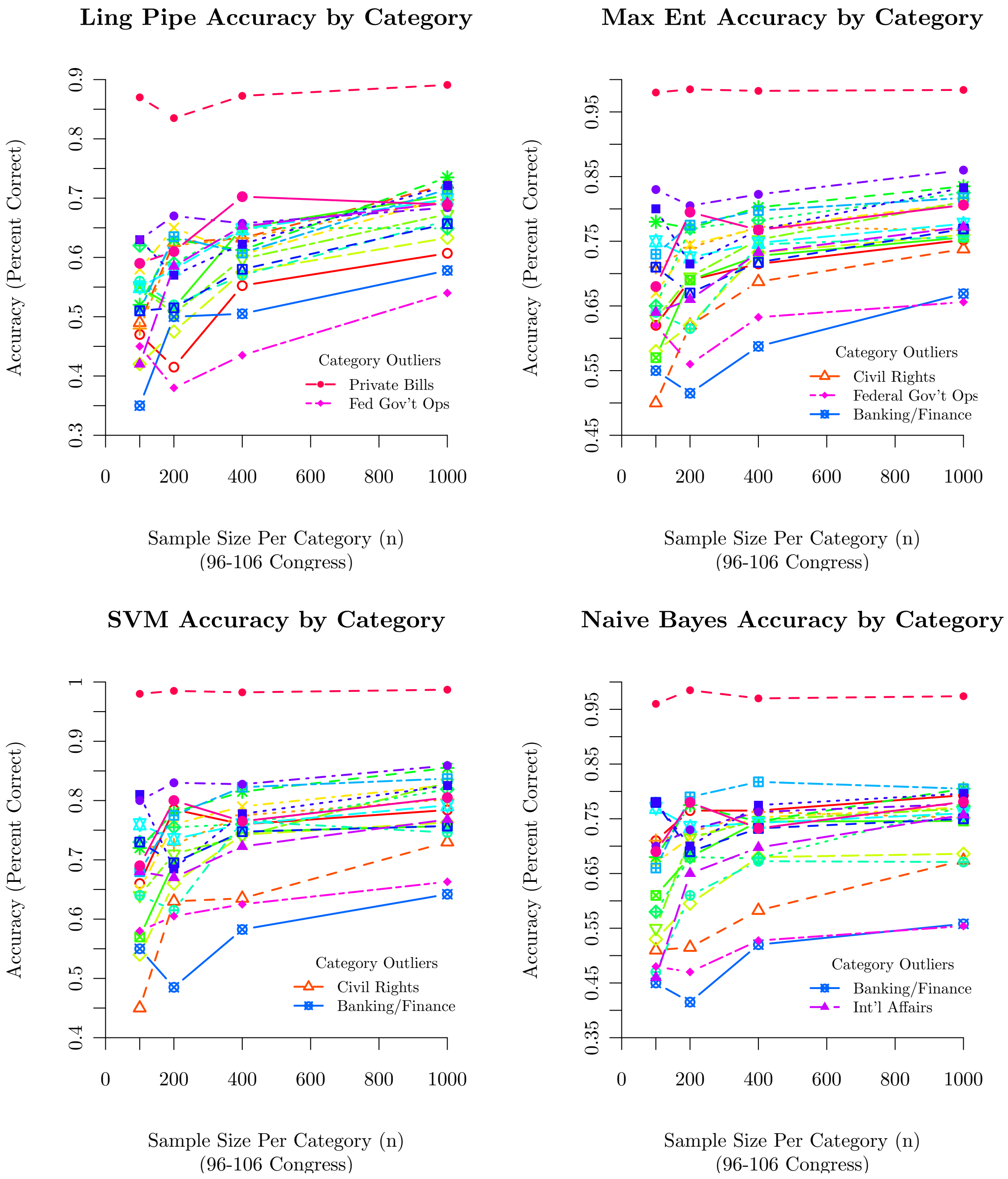

Figure 2: *Overall, when the sample size reaches n=* 1000 *per category, category accuracy tends to converge around 75%, with the exception of lingpipe. The "Private Bills" category is predicted correctly 90-100% of the time by all algorithms. The "Civil Rights" category shows dramatic improvement as sample size improves for the max ent and SVM algorithms. Finally, the "International Affairs" category shows strong improvement with the naïve bayes algorithm.*

5.3 Confusion Matrices

The Confusion matrix adds another diagnostic dimension by providing an opportunity to not only assess recall accuracy but also precision accuracy across categories (Olson and Delen, 2008). This allows us to not only examine error rates, but the specifics of those errors. If the machine mislabels, do the errors tend to be randomly distributed or concentrated?

The difference between precision and recall can be important depending on the goals of a project. In a nutshell, recall assesses how many of the true cases were correctly predicted by the algorithm, whereas precision assesses how many of the predicted cases are actually true cases. Most researchers care about precision, but recall may be of importance for projects where type II errors (false negatives) are especially problematic. For example, in Table 3, SVM predicts that 173 of the 2000 cases in the test set address Economics. In reality, there are 100 “true” Economics bills in the test set. The algorithm correctly “recalls” 66 of those 100 cases. In terms of precision, 66 of the 173 cases that SVM predicts to be about Economics are actually about economics (overall precision of 38 percent).

In terms of how Economics (e.g.) related errors are distributed, we discover (examining the Economics row values) that when true Economics bills are labeled as something else, they tend to be labeled as Banking bills. Examining the Economics column values, we discover that the algorithm is most likely to label Labor and Banking bills as primarily about economics. As shown in Table 4, as the sample size increases, so does an algorithmŠs accuracy. However, the misclassification patterns related to Economics, Labor and Banking persist, suggesting a systemic challenge in terms of the differentiability of these topics.

	SVM Algorithm (Training Set Size per Category = 100																					
Hand Code	Econ	CR	Health	Ag	Labor	Educ	Env	Energy	Tran	LC	SW	Hous	Bank	Defense	Science	FT	Intl	Govt	Lands	PB	n	Pct Right
Economics	66	0	0	0	1	3	0	2	0	0	3	1	11	1	2	0	5	4	1	0	100	66
Civil Rights	3	45	0	1	6	2	2	0	1	8	1	1	7	2	4	0	3	13	1	0	100	45
Health	2	0	72	2	2	0	0	0	0	4	2	0	3	9	0	0	1	3	0	0	100	72
Ag.	9	1	1	65	0	0	0	0	0	1	1	2	5	1	1	4	4	4	1	0	100	65
Labor	16	2	2	2	54	1	0	0	2	6	5	1	0	1	0	0	8	0	0	0	100	54
Education	7	3	1	0	2	64	1	0	0	2	1	1	2	8	5	0	1	1	1	0	100	64
Environment	5	0	1	4	0	1	57	3	3	5	1	0	2	4	1	0	7	0	6	0	100	57
Energy	4	0	0	0	0	0	1	72	1	1	1	0	6	1	5	1	1	0	3	3	100	72
Transportation	6	1	0	0	0	1	0	1	73	3	1	1	1	2	1	1	5	0	3	0	100	73
Law/Crime	4	2	2	0	5	0	0	0	2	64	2	1	2	8	4	0	2	2	0	0	100	64
Social Welfare	5	1	1	1	4	1	0	2	1	1	76	1	0	1	2	0	2	1	0	0	100	76
Housing	7	0	3	2	2	2	1	0	0	1	1	68	3	6	2	0	1	0	1	0	100	68
Banking	17	3	1	2	1	0	1	0	1	3	0	2	55	2	2	1	2	4	2	1	100	55
Defense	2	1	1	0	1	0	1	4	1	3	0	1	0	73	3	2	1	2	4	0	100	73
Science	5	2	0	0	1	0	1	1	0	1	0	0	2	1	81	0	0	3	2	0	100	81
Foreign Trade	2	0	0	1	1	0	0	2	3	1	0	2	3	1	0	80	4	0	0	0	100	80
Int'l Affairs	5	4	1	0	1	0	1	0	1	1	1	0	4	4	1	2	68	3	1	2	100	68
Gov't Ops	5	4	0	1	4	0	0	1	4	4	2	0	2	9	2	0	2	58	1	1	100	58
Lands	3	0	2	0	1	0	10	0	0	3	0	0	1	3	2	0	2	4	69	0	100	69
Private Bills	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	98	100	98
n	173	69	88	81	86	75	76	88	93	113	98	82	109	137	118	91	120	102	96	105		
Pct. Right	38	65	82	80	63	85	75	82	78	57	78	83	50	53	69	88	57	57	72	93		

Table 3: The SVM confusion matrix for $n = 100$ per category reveals both precision and recall error. Precision reads down the column, recall reads across.

	SVM Algorithm (Training Set Size per Category = 1000)																					
Hand Code	Econ	CR	Health	Ag	Labor	Educ	Env	Energy	Tran	LC	SW	Hous	Bank	Defense	Science	FT	Intl	Govt	Lands	PB	n	Pct Right
Economics	784	7	5	6	30	2	2	6	2	6	16	28	36	4	7	14	6	35	3	1	1000	78
Civil Rights	14	730	13	5	18	22	1	2	11	33	20	10	19	11	21	2	18	41	9	0	1000	73
Health	22	3	802	14	14	8	6	0	0	28	34	6	7	32	9	0	0	11	4	0	1000	80
Agriculture	21	1	9	826	7	2	19	3	10	9	5	13	20	2	3	27	8	5	7	3	1000	83
Labor	47	6	19	6	763	18	5	3	7	13	31	8	10	12	5	4	14	24	2	3	1000	76
Education	13	19	11	0	9	830	1	1	5	7	12	6	12	16	13	0	8	26	11	0	1000	83
Environment	17	4	9	20	2	9	757	16	19	8	2	9	18	2	16	9	22	10	50	1	1000	76
Energy	21	2	1	1	1	1	27	855	19	3	2	6	9	5	10	14	4	3	16	0	1000	86
Transportation	20	5	4	1	12	1	21	7	819	14	3	21	14	10	5	8	9	6	18	2	1000	82
Law/Crime	36	29	20	2	21	9	2	1	9	746	19	3	8	11	4	3	21	45	8	3	1000	75
Social Welfare	29	7	37	9	31	16	4	4	4	11	792	12	6	9	5	0	3	14	7	0	1000	79
Housing	26	5	5	13	13	3	8	3	1	9	17	837	17	14	2	0	6	8	11	2	1000	84
Banking	106	17	7	18	14	8	6	17	17	18	5	38	642	4	17	19	14	16	12	5	1000	64
Defense	13	14	16	0	20	16	4	8	11	22	4	8	3	757	2	6	33	33	24	6	1000	76
Science	19	25	4	1	2	16	8	6	8	17	1	2	19	3	825	6	17	14	7	0	1000	82
Foreign Trade	18	5	0	17	4	2	5	7	10	7	1	0	14	2	2	859	32	10	3	2	1000	86
Int'l Affairs	17	17	4	14	7	8	18	6	11	19	2	4	13	17	8	32	768	21	10	4	1000	77
Gov't Ops	53	38	14	3	19	12	7	6	5	44	8	14	12	24	10	4	20	663	17	27	1000	66
Public Lands	13	3	3	7	5	4	40	15	15	11	1	12	9	12	5	1	10	28	805	1	1000	80
Private Bills	0	0	0	0	1	1	0	0	1	0	0	0	1	4	0	0	0	5	0	987	1000	99
n	1289	937	983	963	993	988	941	966	984	1025	975	1037	889	951	969	1008	1013	1018	1024	1047		
Pct. Right	61	78	82	86	77	84	80	89	83	73	81	81	72	80	85	85	76	65	79	94		

Table 4: The SVM confusion matrix for $n = 1000$ per category reveals both precision and recall error. Precision reads down the column, recall reads across. Relative to the smaller training set presented, the data here show much less recall and precision error.

5.4 Accuracy and Coverage Tradeoff with Ensemble Agreement

The results of the earlier experiments could be used to select the highest performing algorithm (SVM) to label the virgin texts. However, with an $n = 1000$ (per category) sample, the overall accuracy of the newly classified cases is predicted to be about 78 percent, which may be lower than our target level of accuracy. Another option is to leverage differences in the algorithms to differentiate bills that are labeled with high accuracy. Ensemble agreement simply refers to whether multiple algorithms make the same prediction concerning the class of an event. If the plurality position of the ensemble corresponds to better predictions, we can use ensemble agreement to infer whether virgin bills have been labeled with high accuracy. Those that have can be set aside, while those that have not can be manually coded. The question, of course, is how many bills can be set aside using this method?

Our experiments are based on respective training and test sets of $n = 20,000$ total bills. The x axis in Figure 3 corresponds to the number of algorithms in agreement (1 = two algorithms agree, two disagree; 4 = svm, maxent, ling pipe, naïve bayes all agree)[1] . The y axis indicates (dashed line) the percent correctly predicted for different levels of ensemble agreement, and (solid line) the percentage of total cases correctly predicted at that level or above.

When one pair of algorithms agree and the other pair of algorithms disagree (1), the average percent of cases correctly predicted (averaging across all 4 algorithms) is just 45 percent while the cases covered equals 99 percent. When two pairs of algorithms make different predictions, the average percent of correctly predicted cases is also 45 percent (and about 92 percent of the cases have ensemble agreement of 2 agree or better). When three algorithms agree, average accuracy improves to 71 percent while coverage declines to about 85 percent. Finally, average accuracy for cases when all algorithms agree on the label is 92 percent, while the number of cases labeled at this level of accuracy declines to about 61 percent.

[1] We do not include a point for no agreement, since labeling is entirely arbitrary in that case

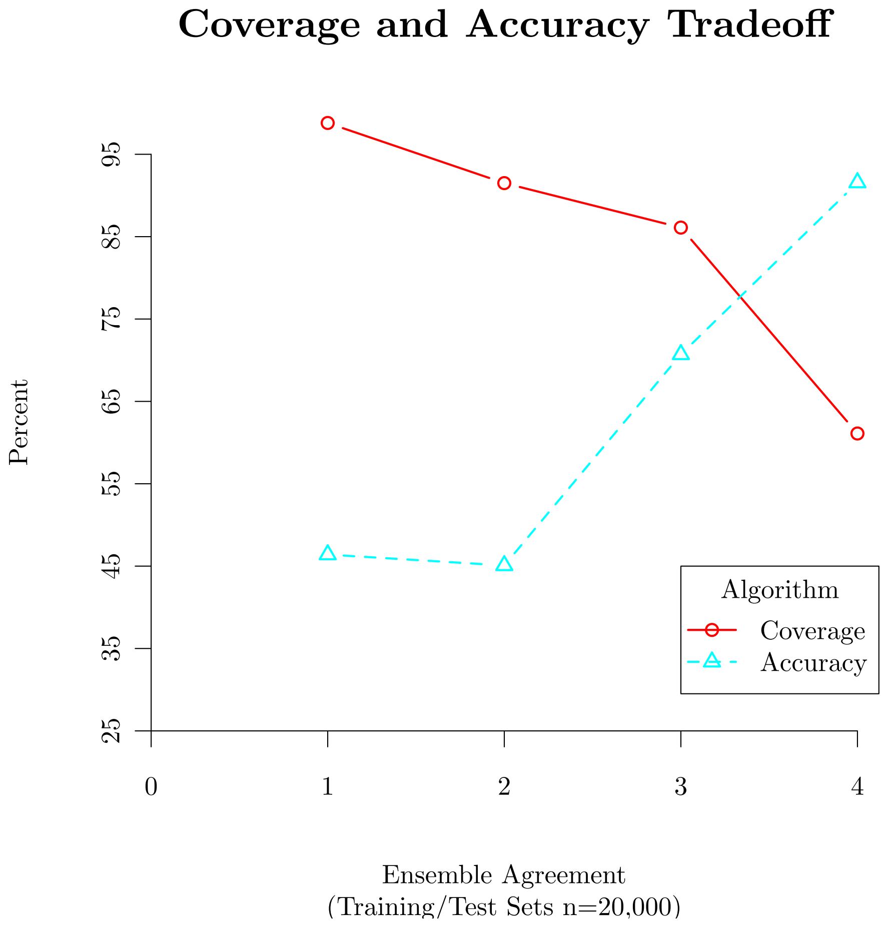

Figure 3: *Ensemble agreement demonstrates that supervised learning accuracy varies depending upon the level of algorithm agreement chosen by the researcher.*

]

Although the four agree cases have the highest accuracy, the percentage of bills that can be set aside using this standard is low (61 percent). However, if we accept bills where at least three algorithms agree (Figure 4), then we can expect 86 percent average agreement, and about 85 percent coverage. This agreement level is substantially higher that what we could expect by relying on a single algorithm (closer to 70 percent). Although some bills will still need to be manually labeled (the ensemble will tell us which these are), we will be able to automatically label about 85 percent of our virgin bills Ű- a substantial savings of time and effort.

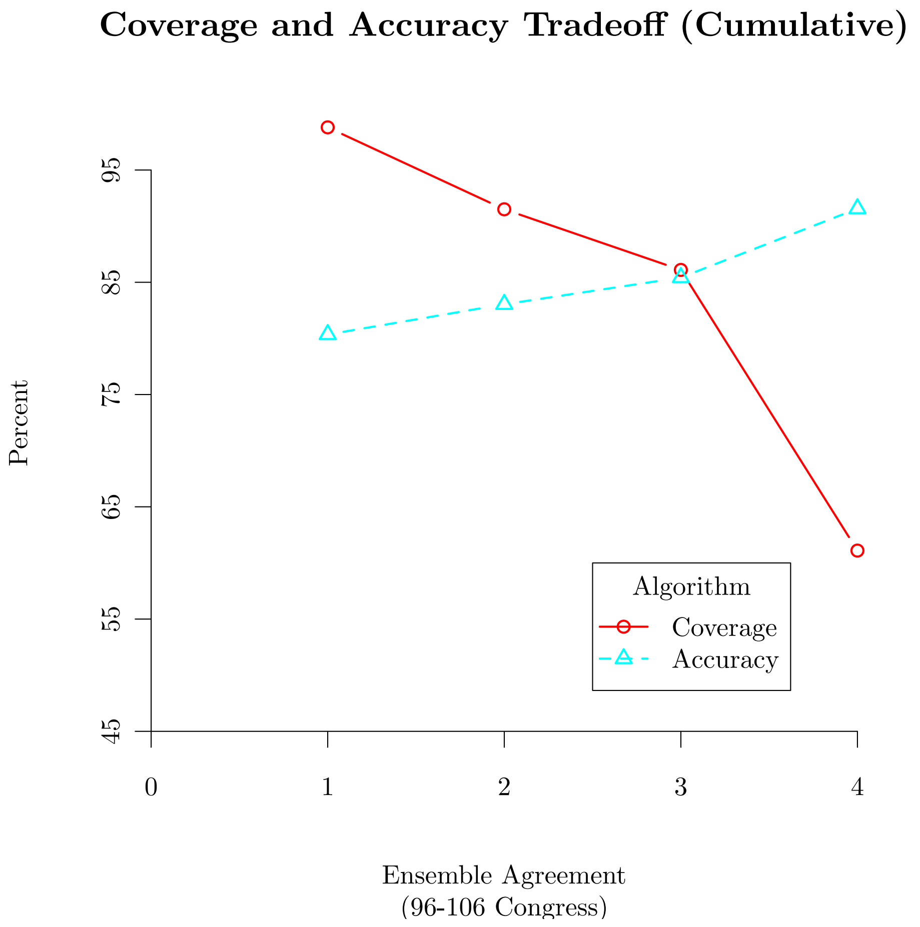

Figure 4: *Ensemble agreement demonstrates that supervised learning accuracy varies depending upon the level of algorithm agreement chosen by the researcher. However, given the tradeoff evinced by agreement and coverage, a three level algorithm agreement is proposed in the current setup.*

6 Analysis Part II

In this section we return to the sampling issues discussed earlier to investigate whether the presence of duplicate bills and variations in training examples across topics impacts algorithm accuracy in the context of the Congressional Bills corpus.

6.1 The Problem of Duplicate Bills and Duplicate Text

In Part I, we controlled for duplicate bills by de-duplicating the database of 229,037 bills, producing 151,819 (66 percent) uniquely titled bills prior to drawing the training and test sets. It remains unclear whether de-duplicating the database leads to improved or depreciated algorithmic prediction. On the one hand, the duplicate problem would leave a non-zero probability of placing the exact same bill in the respective training and test sets. This should lead to overall better prediction within a specific train and test scenario. On the other hand, more duplicate bills included in the training and test sets may reduce the overall diversity of the training set, and therefore the algorithms may have a harder time accurately predicting "rarer" types of bills. Fortunately, this is an empirical question that we test here using $n = 8,000$ training and test sets of bills that have not been de-duplicated.[2]

Figure 5 clearly shows that whether the training set is de-duplicated weighs little on the overall predictive performance of the algorithms. In the non-de-duplicated version, SVM obtains an accuracy rate of 77.41 percent, ling-pipe 66.18 percent, maximum entropy 76.84 percent, and naïve bayes 73.35 percent. For the de-duplicated dataset, the percent correctly predicted are consistently lower, respectively, 75.7 percent, 61.8 percent, 75.20 percent, and 72.00 percent. Thus there is little here to indicate that de-duping is a valuable strategy for improving performance. While perhaps surprising, this is a welcome finding.

[2] We controlled for sample size in this experiment, that is, we used stratified random sampling to obtain our training and tests sets, with a fixed rate of 400 bills per category.

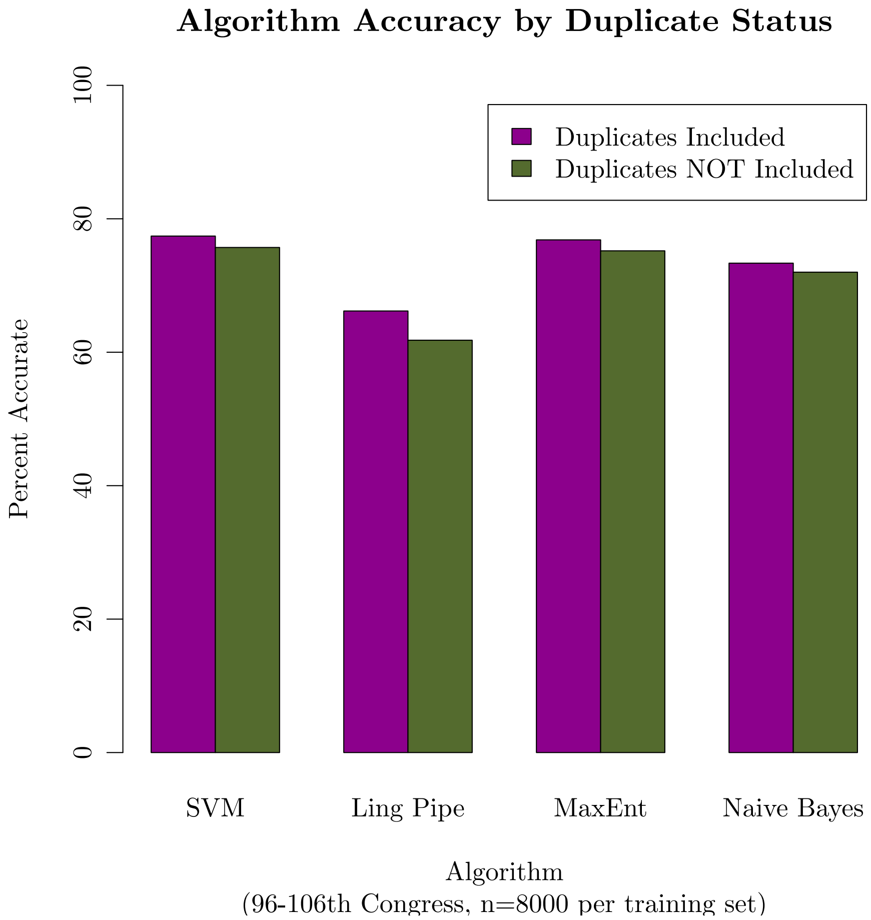

Figure 5: *A comparison of de-duplicated and duplicated datasets yield insignificant differences across the methodology. While the duplicated training sets slightly outperform the de-duplicated training sets, regardless of algorithm, the differences are minimal.*

6.2 Differences in Sampling Methods

In Part I, we stratified the training set to control for the possibility that accuracy might be affected by differences in training sample sizes across topics that would result from a random sampling approach. In this section we relax that restriction to examine accuracy when the training set is randomly drawn — the approach we assume is typical of most projects. Once again, we utilize n = 8000 training and test samples. Figure 6 indicates that a simple random sampling method outperforms a stratified random sampling method across all four algorithms. When the sample is drawn via simple random sampling, the SVM algorithm achieves 82.71 percent accuracy versus 75.7 percent when the sample is stratified. Likewise, lingpipeŠs ratio is 74.29 percent to 61.8 percent, max ent is 83.01 percent to 75.20 percent, and naïve bayes is 76.71 percent to 72.00 percent.

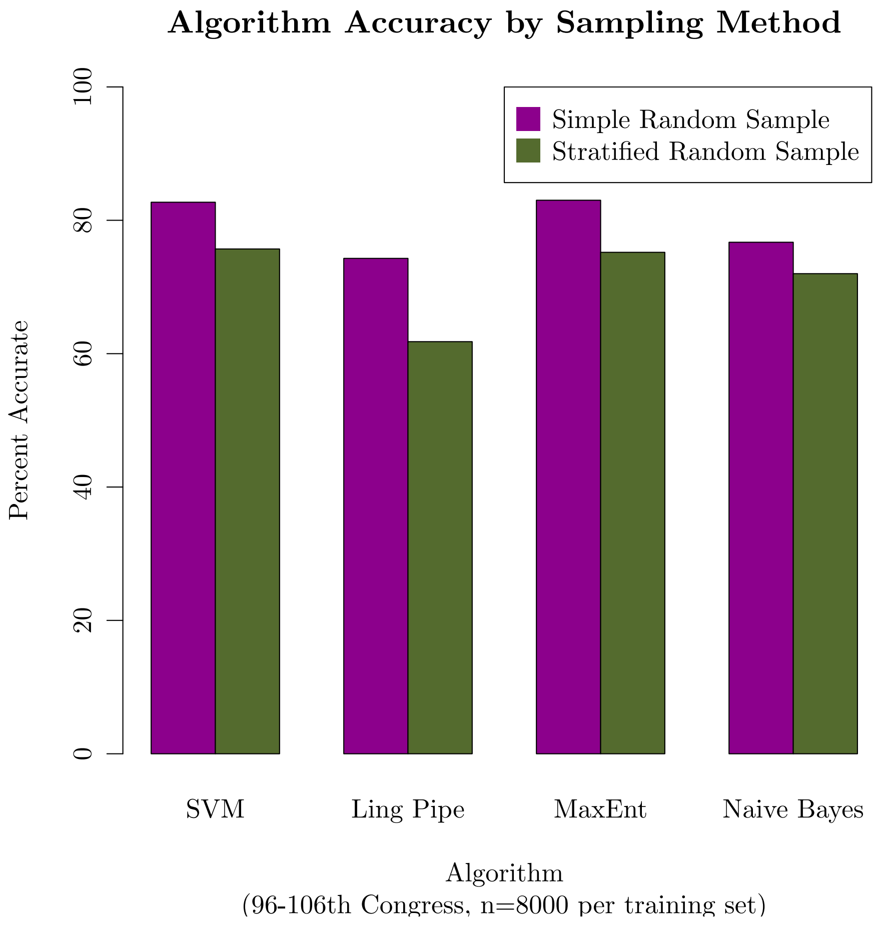

Figure 6: *A comparison of bills sampled via simple random sampling and stratified random sampling where each category is normalized to a set count shows that simple random sampling is the preferred technique presumably due to it's more accurate representation of the data.*

A more precise comparison of the benefits of a random sampling approach can be seen in Figure 7, which plots category precision percentages (for SVM) for the stratified random sample against the simple random sample. The dots above the diagonal line (red dots) indicate that the simple random sample is more precise Ű which is the case for most of the topics. A stratified sampling may still outperform simple random sampling methods for another corpus. Yet a simple random sampling approach performs better in this case. To the extent that these findings are generalizable this also constitute welcome results from an efficiency perspective.[3]

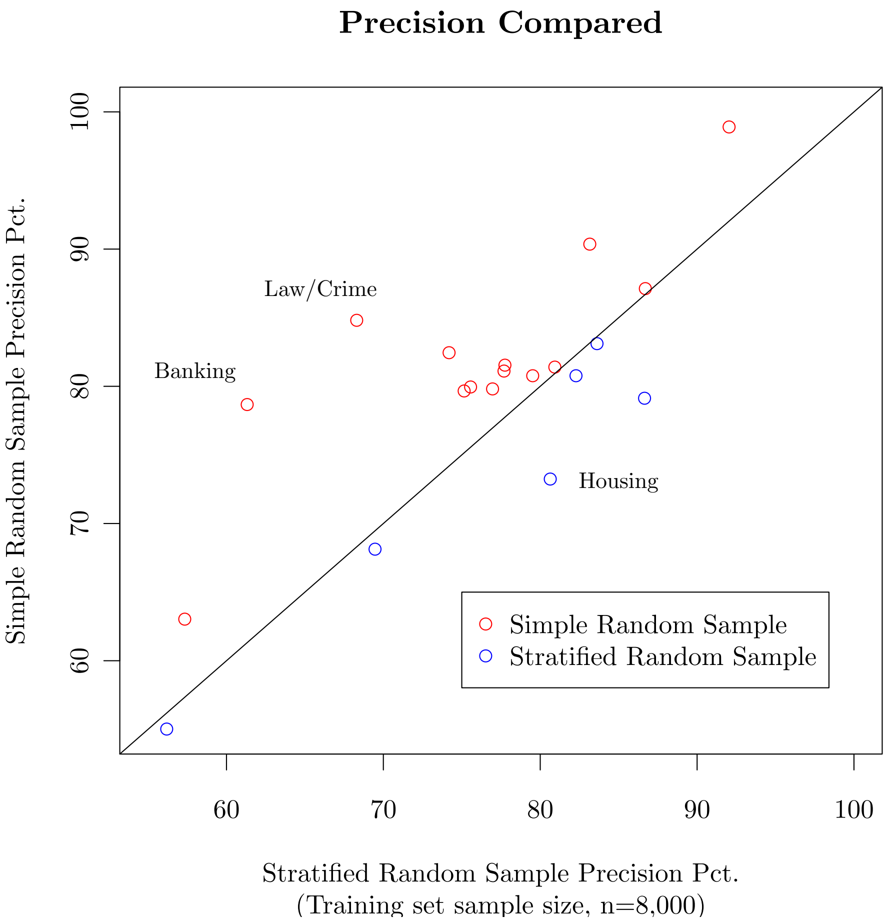

Figure 7: *A comparison of precision rates by category for the SVM algorithm with a training set sample size of* $n = 8000$ *shows that simple random sampling outperforms stratified random sampling.*

[3] See appendix for confusion matrices. To examine how the distribution of the training set categories specifically influences accuracy, we compare confusion matrices of simple random and stratified random sampling runs. In both matrices, the total observation size is $n = 8,000$.

7 Discussion

Information and computer scientists have long used algorithms to classify text and other data. However, until recently, most of these tools and techniques have remained esoteric to social scientists and are only now emerging into the political scientist's methodological toolkit. This is appropriate timing given the availability of text online and other sources, and the vast reduction in labor automated techniques bring.

Supervised learning methods are one way of many that political scientists can use to automate the coding of textual documents. We used the Congressional Bills corpus to evaluate the accuracy and efficiency in supervised learning methods because we are familiar with this database. However, the approach taken here should be evaluated on other existing labeled corpora. How well do these methods apply to newspaper labeling, sentence specific labeling, code frames with fewer codes, and data with perhaps more sentiment than congressional bill titles?

By using ensemble agreement methods on training and test sets of $n = 20,000$ we are able to code approximately 85 percent of all bills with an 85 percent accuracy rate. The tradeoff in accuracy (down from 92 percent) appears worth it given the increase in coverage (61 percent versus 85 percent). Whereas under the second condition, humans must manually code about 40 percent of the corpus, under the first condition humans only have to code 15 percent of all bills to match levels of accuracy observed for highly trained humans. Clearly, this a considerable savings in labor costs.

Finally, we found that de-duplicating the database made no impact upon algorithmic performance. We did find, however, that simple random sampling outperforms stratified random sampling as a method for drawing training sets. Indeed, it seems that the simple random sample more accurately represents the distribution of data, which leads to overall better algorithmic performance. This is welcome news considering that a simple random sample and duplicated databases may be more likely scenarios for researchers.

References

A.L. Berger, V.J.D. Pietra, and S.A.D. Pietra. A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71, 1996. ISSN 0891-2017.

B.E. Boser, I.M. Guyon, and V.N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992. ISBN 089791497X.

C. Cardie and J. Wilkerson. Guest EditorsâĂŹ Introduction: Text Annotation for Political Science Research. *Journal of Information Technology & Politics*, 5(1):1–6, 2008.

B. Carpenter. LingPipe for 99.99% recall of gene mentions. In *Proceedings of the Second BioCreative Challenge Evaluation Workshop*, pages 307–309. Citeseer, 2007.

Loren Collingwood. *Rtexttools: Classifies textual documents via automated content analysis*, 2010. R package version 1.0.

C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995. ISSN 0885-6125.

T.M. Cover, J.A. Thomas, and J. Wiley. *Elements of information theory*, volume 1. Wiley Online Library, 1991.

P.S. Dodds and C.M. Danforth. Measuring the happiness of large-scale written expression: Songs, blogs, and presidents. *Journal of Happiness Studies*, pages 1–16, 2009. ISSN 1389-4978.

J. Grimmer. A Bayesian Hierarchical Topic Model for Political Texts: Measuring Expressed Agendas in Senate Press Releases. *Political Analysis*, 18(1):1, 2010. ISSN 1047-1987.

J. Grimmer and G. King. Quantitative Discovery from Qualitative Information: A General-Purpose Document Clustering Methodology. 2010.

J. Han and M. Kamber. *Data mining: concepts and techniques*. Morgan Kaufmann, 2006. ISBN 1558609016.

D. Hillard, S. Purpura, and J. Wilkerson. Computer-assisted topic classification for mixed-methods social science research. *Journal of Information Technology & Politics*, 4(4):31–46, 2008. ISSN 1933-1681.

D.J. Hopkins and G. King. A method of automated nonparametric content analysis for social science. *American Journal of Political Science*, 54(1):229–247, 2010. ISSN 1540-5907.

C.W. Hsu, C.C. Chang, C.J. Lin, et al. A practical guide to support vector classification, 2003.

G. King and W. Lowe. An automated information extraction tool for international conflict data with performance as good as human coders: A rare events evaluation design. *International Organization*, 57(03):617–642, 2003. ISSN 0020-8183.

M. Laver, K. Benoit, and J. Garry. Extracting policy positions from political texts using words as data. *American Political Science Review*, 97(02):311–331, 2003. ISSN 0003-0554.

D. Lazer, A.S. Pentland, L. Adamic, S. Aral, A.L. Barabasi, D. Brewer, N. Christakis, N. Contractor, J. Fowler, M. Gutmann, et al. Life in the network: the coming age of computational social science. *Science (New York, NY)*, 323(5915):721, 2009.

E. Loper and S. Bird. NLTK: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1*, pages 63–70. Association for Computational Linguistics, 2002.

B.L. Monroe and P.A. Schrodt. Introduction to the Special Issue: The Statistical Analysis of Political Text. *Political Analysis*, 2009. ISSN 1047-1987.

B. O'Connor, R. Balasubramanyan, B.R. Routledge, and N.A. Smith. From Tweets to polls: Linking text sentiment to public opinion time series. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*, pages 122–129, 2010.

D.L. Olson and D. Delen. *Advanced data mining techniques*. Springer Verlag, 2008. ISBN 3540769161.

K.M. Quinn, B.L. Monroe, M. Colaresi, M.H. Crespin, and D.R. Radev. How to analyze political attention with minimal assumptions and costs. *American Journal of Political Science*, 54(1): 209–228, 2010. ISSN 1540-5907.

P.A. Schrodt, S.G. Davis, and J.L. Weddle. Political science: KEDSâĂŤa program for the machine coding of event data. *Social Science Computer Review*, 12(4):561, 1994. ISSN 0894-4393.

Appendix A

SVM

	n=100	n=200	n=400	n=1000	Difference
Civil Rights	0.45	0.63	0.64	0.73	0.28
Labor	0.54	0.66	0.74	0.76	0.22
Education	0.64	0.71	0.74	0.83	0.19
Environment	0.57	0.69	0.75	0.76	0.19
Agriculture	0.65	0.76	0.79	0.83	0.18
Housing	0.68	0.78	0.82	0.84	0.16
Energy	0.72	0.78	0.81	0.85	0.14
Macroeconomics	0.66	0.78	0.76	0.78	0.12
Public Lands	0.69	0.80	0.76	0.80	0.12
Law and Crime	0.64	0.61	0.76	0.75	0.11
Banking and Finance	0.55	0.48	0.58	0.64	0.09
Transportation	0.73	0.76	0.76	0.82	0.09
International Affairs	0.68	0.67	0.72	0.77	0.09
Federal Gov't Ops	0.58	0.60	0.62	0.66	0.08
Health	0.72	0.72	0.78	0.80	0.08
Foreign Trade	0.80	0.83	0.83	0.86	0.06
Social Welfare	0.76	0.73	0.76	0.79	0.03
Defense	0.73	0.69	0.75	0.76	0.03
Science and Tech	0.81	0.68	0.78	0.82	0.01
Private Bills	0.98	0.98	0.98	0.99	0.01

Table 5: The SVM Algorithm shows dramatic improvement by sample size for a variety of categories including most notably International Affairs, Civil Rights, Banking and Finance, and Health.

Max Ent

	n=100	n=200	n=400	n=1000	Difference
Civil Rights	0.50	0.62	0.69	0.74	0.24
Environment	0.57	0.69	0.73	0.76	0.19
Education	0.63	0.69	0.75	0.81	0.18
Labor	0.58	0.62	0.73	0.76	0.18
Transportation	0.65	0.77	0.78	0.82	0.17
Agriculture	0.67	0.74	0.77	0.81	0.14
International Affairs	0.64	0.66	0.73	0.77	0.13
Macroeconomics	0.62	0.69	0.71	0.75	0.13
Public Lands	0.68	0.80	0.77	0.81	0.13
Banking and Finance	0.55	0.52	0.59	0.67	0.12
Law and Crime	0.64	0.61	0.74	0.76	0.12
Housing	0.73	0.78	0.80	0.82	0.09
Defense	0.71	0.67	0.72	0.77	0.06
Health	0.71	0.74	0.77	0.77	0.06
Energy	0.78	0.77	0.80	0.83	0.05
Federal Gov't Ops	0.62	0.56	0.63	0.66	0.04
Science and Tech	0.80	0.71	0.77	0.83	0.03
Foreign Trade	0.83	0.80	0.82	0.86	0.03
Social Welfare	0.75	0.72	0.75	0.78	0.03
Private Bills	0.98	0.98	0.98	0.98	0.00

Table 6: The Max Ent Algorithm shows improvement by sample size for a variety of categories including most notably Civil Rights, Environment, and Education.

Naive Bayes

	n=100	n=200	n=400	n=1000	Difference
International Affairs	0.46	0.65	0.70	0.76	0.30
Education	0.55	0.71	0.75	0.77	0.22
Law and Crime	0.47	0.61	0.67	0.67	0.20
Transportation	0.58	0.68	0.68	0.77	0.19
Civil Rights	0.51	0.52	0.58	0.67	0.16
Labor	0.53	0.59	0.68	0.69	0.16
Housing	0.66	0.79	0.82	0.80	0.14
Environment	0.61	0.68	0.74	0.75	0.14
Energy	0.68	0.78	0.74	0.80	0.12
Banking and Finance	0.45	0.41	0.52	0.56	0.11
Agriculture	0.67	0.71	0.75	0.77	0.10
Public Lands	0.69	0.78	0.73	0.78	0.09
Macroeconomics	0.71	0.76	0.76	0.79	0.08
Foreign Trade	0.70	0.73	0.76	0.78	0.08
Federal Gov't Ops	0.48	0.47	0.53	0.55	0.07
Health	0.71	0.72	0.76	0.75	0.04
Science and Tech	0.78	0.70	0.78	0.80	0.02
Private Bills	0.96	0.98	0.97	0.97	0.01
Social Welfare	0.77	0.73	0.74	0.76	-0.01
Defense	0.78	0.69	0.73	0.75	-0.03

Table 7: The Naive Bayes Algorithm shows improvement by sample size for a variety of categories including most notably International Affairs, Education, Law and Crime, and Transportation.

Appendix B: Simple Random Sampling Versus Stratified Random Sampling Confusion Matrices

	SVM Random Sample Algorithm (Training Set Size = 8000)																					
Hand Code	Econ	CR	Health	Ag	Labor	Educ	Env	Energy	Tran	LC	SW	Hous	Bank	Defense	Science	FT	Intl	Govt	Lands	PB	n	Pct Right
Economics	181	0	2	0	6	2	0	0	0	1	5	3	9	0	0	1	1	40	1	0	252	72
Civil Rights	2	47	0	0	2	4	0	0	1	5	3	0	1	4	0	0	0	24	1	0	94	50
Health	10	0	262	1	1	4	5	0	0	5	14	2	0	15	0	0	0	12	7	0	338	78
Agriculture	7	0	8	150	0	0	1	1	4	1	1	4	12	0	0	6	1	10	3	0	209	72
Labor	21	0	0	0	197	4	0	1	5	3	16	1	1	6	1	0	3	29	2	1	291	68
Education	8	1	2	1	2	182	0	0	4	0	6	1	1	15	0	0	0	7	4	0	234	78
Environment	5	0	2	3	0	4	189	4	13	0	0	0	3	1	1	3	1	10	34	0	273	69
Energy	9	0	0	0	0	0	1	115	4	0	0	2	8	0	1	3	0	8	6	0	157	73
Transportation	8	0	1	1	3	1	6	1	336	4	0	1	2	3	0	1	3	26	20	1	418	80
Law/Crime	6	3	15	0	3	2	2	0	4	240	4	3	4	7	1	0	1	53	5	3	356	67
Social Welfare	13	0	14	0	6	2	0	0	9	4	265	1	0	3	0	0	0	12	3	0	332	80
Housing	3	0	0	0	1	0	1	1	1	1	1	104	5	6	1	0	0	8	4	0	137	76
Banking	15	0	7	6	5	0	4	1	16	4	0	8	225	3	2	2	3	38	6	1	346	65
Defense	4	1	6	0	4	4	1	2	3	2	2	4	0	418	3	0	1	38	12	4	509	82
Science	3	1	0	0	0	0	1	0	1	0	1	0	1	1	63	0	1	6	5	0	84	75
Foreign Trade	5	0	0	3	1	0	1	3	3	0	0	0	9	0	0	91	5	7	4	3	135	67
Int'l Affairs	1	0	1	1	1	1	2	1	1	1	0	0	0	1	0	1	62	18	5	1	98	63
Gov't Ops	25	6	3	0	8	6	7	1	8	8	4	5	4	16	4	0	5	658	22	10	800	82
Public Lands	3	0	0	0	1	3	13	1	8	3	3	3	1	5	1	3	3	29	586	1	667	88
Private Bills	0	0	0	0	1	0	0	0	0	1	0	0	0	3	0	4	1	11	3	2246	2270	99
n	329	59	323	166	242	219	234	132	421	283	325	142	286	507	78	115	91	1044	733	2271		
Pct. Right	55	80	81	90	81	83	81	87	80	85	82	73	79	82	81	79	68	63	80	99		

Table 8: The SVM confusion matrix for $n = 8,000$ total reveals both precision and recall error. Precision reads down the column, recall reads across.

	SVM Stratified Random Sample Algorithm (Training Set Size = 8000)																					
Hand Code	Econ	CR	Health	Ag	Labor	Educ	Env	Energy	Tran	LC	SW	Hous	Bank	Defense	Science	FT	Intl	Govt	Lands	PB	n	Pct Right
Economics	304	2	5	3	9	2	1	0	7	3	4	7	22	0	2	2	4	21	1	1	400	76
Civil Rights	5	254	11	0	7	7	0	1	8	21	9	3	13	12	12	0	7	29	1	0	400	64
Health	10	2	310	14	1	3	0	0	0	9	12	4	2	19	5	0	2	4	2	1	400	78
Agriculture	14	1	6	316	1	0	11	0	3	3	3	3	12	0	2	10	6	2	6	1	400	79
Labor	13	5	11	3	297	2	0	0	5	10	14	3	6	7	1	1	6	11	2	3	400	74
Education	14	15	3	0	6	296	2	0	3	5	15	3	3	13	6	0	7	4	3	2	400	74
Environment	4	6	3	9	0	1	299	12	8	5	0	1	10	0	1	2	10	5	23	1	400	75
Energy	14	0	0	2	1	0	12	326	8	2	0	2	4	1	6	3	1	11	5	2	400	82
Transportation	18	2	1	0	5	5	1	2	304	5	1	3	9	5	1	6	9	9	14	0	400	76
Law/Crime	13	12	4	0	4	4	2	0	4	306	6	3	5	4	2	3	1	26	1	0	400	76
Social Welfare	22	2	21	5	9	2	1	2	5	3	304	7	3	2	0	0	3	6	3	0	400	76
Housing	14	0	1	4	1	3	1	1	3	3	13	329	8	6	2	0	3	3	3	2	400	82
Banking	41	7	2	4	7	5	7	8	10	14	1	12	233	7	6	7	12	11	3	3	400	58
Defense	9	3	8	0	3	7	3	6	1	10	0	10	2	299	6	0	10	7	12	4	400	75
Science	9	8	0	0	2	2	7	2	6	14	2	1	11	2	311	4	6	6	7	0	400	78
Foreign Trade	9	0	1	8	0	0	2	3	1	2	0	1	13	1	2	331	21	4	0	1	400	83
Int'l Affairs	9	3	2	6	2	2	13	3	7	11	2	3	5	8	4	13	289	11	2	5	400	72
Gov't Ops	16	12	9	3	9	5	3	4	6	17	2	5	11	11	8	0	10	250	11	8	400	62
Public Lands	2	4	1	3	3	8	11	6	6	5	3	8	7	5	1	0	8	13	306	0	400	76
Private Bills	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	3	0	393	400	98
n	541	338	399	380	367	354	376	376	395	448	391	408	380	403	378	382	416	436	405	427		
Pct. Right	56	75	78	83	81	84	80	87	77	68	78	81	61	74	82	87	69	57	76	92		

Table 9: The SVM confusion matrix for $n = 400$ per category reveals both precision and recall error. Precision reads down the column, recall reads across.