An Active Learning Framework for Classifying Political Text[1]

Prepared for delivery at the 2007 Annual Meeting of the Midwest Political Science Association

Chicago, IL. April 14-17.

Dustin Hillard
Dept of Electrical Engineering - University of Washington
hillard@u.washington.edu

Stephen Purpura
The Program on Networked Governance
John F. Kennedy School of Government - Harvard University
stephen_purpura@ksg07.harvard.edu

John Wilkerson
Department of Political Science - University of Washington
jwilker@u.washington.edu

[1] We acknowledge support from the Program on Networked Governance at Harvard through NSF grant No. 0429452 and the Connecting to Congress Project, and NSF Grants No. 00880066, 0111443, 00880061 for the Congressional Bills Project. The views expressed are those of the authors and not the National Science Foundation. We thank Bryan Jones, Frank Baumgartner, Richard Zeckhauser, Jesse Shapiro, Claire Cardie, Eduard Hovy, David Lazer, Michael Neblo, Kevin Esterling, Aleks Jakulin, Matthew Baum, Jamie Callan, Micah Altman, David King, James Purpura, Alan Gibson and Julianna Rigg for their helpful comments and feedback.

Abstract

We develop a framework and tools for applying a computer-assisted context analysis system and find that it achieves levels of accuracy comparable to humans for about 80% less effort when starting from scratch (no labeled examples). The system is presented using a case study of Congressional bill titles as a proxy for the full text of Congressional bills. We also demonstrate that the system can use information learned from previous experiments to reduce the labeling requirements still further to over 90% savings of the current human effort.

This study assumes that social scientists have a need to locate individual documents in a subject area. To support this need, "Topic classification," where documents are coded according to some organizing framework, is used to facilitate search and summarization. Our proposed framework for effectively employing machine learning methods mitigates the high costs of the standard method of topic classification - human labeling. We scientifically evaluate the efficacy and accuracy of the automated approach using a large corpus of 380,000 human-labeled events and a classification system that includes 20 major policy topics, 226 subtopics, and a demonstrably strong level of human inter-coder agreement.

Introduction

The ability to locate and examine the details of individual documents within a subject area is a requirement for many, if not most of the questions that interest social scientists. Yet, researchers studying digital documents in large datasets encounter a daunting obstacle: how to find the documents that are of particular interest. "Topic classification," where documents are pre-coded according to an organizing framework, is a common approach to facilitating search in these situations. Government databases, such as those maintained by the Library of Congress, rely on human-centered topic classification to help policymakers and citizens locate what they need. Commercial Internet services such as Yahoo! Flickr and Personals also turn to document classification systems to enable users to focus their search efforts on information that most closely matches their own interests. In each case, the task is the same: to restrict the document domain to the things that most interest the researcher. Anyone who has struggled to construct a useful Google search can appreciate the value of this enterprise.

A central drawback of topic classification can be its high cost, which probably helps to explain why the subject of topic classification has received limited attention in Political Science. High quality classification systems have historically depended on trained human coders, whose performance is regularly evaluated for reliability and accuracy. There are good reasons to turn to humans when the objective is to label what a document is primarily about. On the other hand, high costs limit the extent to which topic classifications systems are used and their unique benefits realized.

In this paper, we report on a project that employs machine learning methods to mitigate this central disadvantage of the human labeling approach (its high cost) while preserving its advantages. We show that software-assisted context analysis can enable researchers to investigate complex questions that would otherwise be extremely expensive or intractable. Our method uses a case-based or "learning by example" approach[2] to replicate the ability of humans to classify individual documents. The software trains on human labeled examples before proceeding to label other documents at high levels of accuracy.

This method supports the same statistical evaluations used by researchers employing more traditional human driven classification and labeling processes. How confident is the machine in its estimate? How do we evaluate agreement between the computer and human coders? How do we

[2] This method, called structured learning, has been the subject of more than 1,000 scholarly publications in computational linguistics literature recent years (Mann, Mimno, McCallum, 2006). We intentionally avoid advancing a particular structured learning algorithm, such as Support Vector Machines, and instead propose a general approach to topic classification that allows us to draw on the best available innovations from the natural language processing and machine learning disciplines. Ideally, social scientists will collaborate with computational linguists to advance research in this area. See the proceedings from a NSF funded workshop http://codeshop.wikispaces.com/ (Shulman, Balla, Callan, Cardie, Hovy, Kwon, Mitchell, Purpura, Rothman, Rose, Schrodt, Wiebe, Wilkerson, Xue, Yang, Zavestoski, 2006).

evaluate when the automated classifier is working well for each specific category, and what additional refinements are required to achieve desired levels of accuracy?

We draw on a large corpus of 380,000 human-coded events and an established topic system of 20 major policy topics and 226 subtopics to scientifically evaluate the efficacy and accuracy of an automated approach. A central purpose of the paper is to provide a framework for future applications by non-specialists using tools that we are developing for this purpose. We document key human-computer interaction (HCI) problems uncovered and addressed as part of the research process.

We find that this approach achieves levels of accuracy and intercoder reliability comparable to those of trained human coders. Turning to a substantive question motivating the project, we are able to verify an assumed feature of the existing database – that a congressional bill's title generally reflects it content. Using software to categorize bill titles generates valuable information that speeds the process and improves quality in ways that were not initially envisioned. [3]

Topic Based Search and Retrieval

Social science scholars have not been particularly attentive to the implications of the classification systems used for analysis.[4] For example, keyword or dictionary searches are commonly used to trace changing issue attention in the media or in legislatures. Unfortunately, a keyword search that is too narrow inevitably excludes relevant activities, while one that is too comprehensive generates an unknown number of "false positives." Searching for news stories in Lexis-Nexis for "Iraq" over the past

[3] Other researchers have developed useful information extraction methods for content analyzing political documents. Google relies on simple keyword searches against a Markov chain of the web for finding well-referenced web sites. KEDS/TABARI (Schrodt, Davis, and Weddle, 1994) and similar methods (King and Lowe, 2003) and Wordscores (Laver and Garry, 2003) select 'cases' for learning-by-example in distinctly different ways. With KEDs/TABARI and similar methods, humans create the rules that the software uses to code events. Wordscores uses reference document cases to rank subsequent documents. In our work, humans assign a small number of cases to mutually exclusive topic categories. Using these initial cases as a training set, the software then computationally generates rules for the correct labeling of unlabeled cases. Once the rules have been generated, users feed unlabeled cases to the system for categorization. To test performance, we assess whether the system's predictions for the unlabeled cases accord with those assigned by human coders over a 50 year period. To ensure that our methods aren't unacceptably biased due to agreement-by-chance or high variance within specific categories, our performance metrics discount performance assessment for these factors. For these reasons and others, our methodology is considerably different from the above mentioned approaches as well as unsupervised approaches to detecting patterns in data (Quinn et al., 2006). In computational linguisitics, unsupervised approaches are frequently used as a first step toward developing topic classification systems in the absence of strong priors about the organization of such a system (Hand, Mannila, & Smyth, 2001).

[4] This is not intended to denigrate careful scholarly research on the subject such as that of Schrodt and Gerner (1994) or King and Lowe (2003). Our point is that the perils of keyword searches or retrieval-centered search systems are not generally appreciated.

decade uncovers thousands of stories that include references to Iraq but are not primarily about the war in Iraq.

Other topic classification systems that are designed to address the shortcomings of keyword approaches may also be less than ideal for the purposes of scholarly research. The Library of Congress relies on humans to code congressional legislation for topic. Because its 'customer' is the contemporary user (e.g. a congressional staffer or lobbyist) who is trying to retrieve documents on a hot issue, the topics evolve as the needs and orientations of these users change. Over time, topics have been added so that they number in the thousands. Though effective for contemporary search, this topic system (like many others) should not be used to study changing topic attention over time because similar events are being coded differently at different points in time (Baumgartner, Jones and Wilkerson, 2002). Many valuable political science databases also include subject codes, but rarely do these projects share reports on reliability if they examine the question at all (e.g. Poole and Rosenthal's NOMINATE project (www.voteview.com).

Requirements of a topic system. A valuable topic system for social science facilitates individual level search to support qualitative analysis and statistical validation, as well as reliable investigations of trends over time and across decision-making venues. "Mixed method" investigations that combine case level investigations with broader pattern analyses are also increasingly valued within social research (King, Keohane and Verba, 1994). To serve these research goals, a topic classification must include the following features. First, it must be ***discriminating***. It should permit a researcher to limit search to documents that are primarily about (and not peripherally about) a topic of interest. If a second goal of the system is to compare patterns and trends, then the topic categories must also be mutually exclusive and span the entire agenda. In addition, it should be ***probabilistic***. The system should not only choose the primary topic for each example, but also provide information about the likelihood of other topics. This allows the researcher to expand a search to those documents that are closely related to, but not primarily about, her subject of interest. Third, it should be ***accurate***. The tagged documents should correspond to the tag. Fourth, it should be ***reliable***. The topics or the documents falling within a topic should not drift from one year to the next. Finally, it is ***efficient.*** It can be used to quickly code events coming on line, or large historical datasets, with minimal effort.

The Congressional Bills Project Corpus

The NSF funded Congressional Bills Project has assembled a dataset of all federal public bills introduced since 1947. The project currently contains 380,000 records and includes details about each bill's substance (its title or short description), progress through the legislative process, and sponsor. It is publicly available at www.congressionalbills.org.

Each bill has been coded, by hand, into a mutually exclusive, hierarchical classification scheme originally developed as part of the Policy Agendas Project (www.policyagendas.org). Table 1 lists the 20 major topics of this system. Each major topic has additional partitions, for a total of 226 subtopics. For example, topic 7 (environment) includes 12 subtopics such as 'species and forest protection,' 'recycling,'

and 'drinking water safety.' [5] (Additional details about these topic categories and the coding process can be reviewed online.[6])

It is important to emphasize that this scheme partitions the legislative agenda by issue area rather than by program. Thus, the subject categories remain valid even as programs come and go. Related projects have or are applying the same topic system to executive, judicial, media and public opinion data since WWII, and to U.S. state legislatures, nations in the European Union, and Canada.

Table 1: Major Topics of the Congressional Bills Project
1 Macroeconomics
2 Civil Rights, Minority Issues, Civil Liberties
3 Health
4 Agriculture
5 Labor, Employment, and Immigration
6 Education
7 Environment
8 Energy
10 Transportation
12 Law, Crime, and Family Issues
13 Social Welfare
14 Community Development and Housing Issues
15 Banking, Finance, Domestic Commerce
16 Defense
17 Space, Science, Technology, Communications
18 Foreign Trade
19 International Affairs and Foreign Aid
20 Government Operations
21 Public Lands and Water Management
99 Private Legislation

Many hours of coding by trained graduate and undergraduate students have been invested in the project, with observed levels of Cohen's Kappa (Cohen, 1968) inter-coder agreement approaching 0.9 at the major topic level and 0.8 at the subtopic level. [7] The resulting database is of high quality and used by researchers, instructors, and students to study policy trends over 50 years, or to "drill down" to

[5] http://www.policyagendas.org/codebooks/topicindex.html#7

[6] http://www.policyagendas.org/codebooks/topicindex.html

[7] See http://www.congressionalbills.org/BillsReliability.pdf

the individual cases that make up those trends.[8] For example, a researcher who learns that 48 bills of the bills introduced in the 89th Congress (1967-68) were primarily about 'Air Pollution, Global Warming and Noise Pollution" has the ability to also inspect each of those bills to learn more about their specifics and their sponsors.

It is difficult to overstate the value of the case specific topic information that this approach to coding provides. For example, scholars studying issue sponsorship or co-sponsorship in legislatures must have the ability to trace issue specific legislative activity at the level of the individual member, as Sulkin does in her study of the effects of elections on incumbent policy behavior (2005). In other cases, this topic system is used to narrow what would otherwise be an overwhelming search task. Adler and Wilkerson (forthcoming) studied the impact of congressional reforms on committee jurisdictions. To do this, they needed to ask whether a reform altered where bills falling within a given jurisdiction were being referred. Visually inspecting 100,000 bills was impractical, and the authors saved considerable time by limiting their inspections to the subtopics that contained jurisdictionally relevant bills. Even scholars with interests limited to trend data need to be attentive to how the substance of activity within a topic area is changing from one year to the next. This is only possible when they also have access to the underlying source data that make up those trends.

More generally, multiple methods that combine qualitative and quantitative approaches are becoming the standard in social research, and linked examples are often critical to persuading readers to accept interpretations of statistical findings (e.g. Baum 2003). Baum's work is especially compelling as an example. In it he conducts content analysis in diverse corpora with different linguistic patterns and vocabulary norms (the transcripts of Entertainment Tonight, the jokes of John Stewart, and news program transcripts). The ability to explore the details of this approach was critical to its acceptance.

In the next section, we begin to lay out the features of the automated classification approach developed in this paper. This includes a detailed discussion of our methods and evaluation metrics, as well as the human challenges associated with building tools for general benefit. In the section that follows, we report findings from a series of experiments that partition an existing corpus so that segments of labeled bills are used as training samples for predicting unlabeled cases. We devote much of this attention to demonstrating the benefits of Active Learning methods for addressing concerns associated with labeling new events (e.g. bills introduced in the current congress). We conclude that this approach represents a feasible approach to achieving the five requirements of a valued topic classification system presented above.

[8] For an example, see http://www.policyagendas.org/datatools/toolbox/analysis.asp

Automated Text Classification for Political Science

Structured learning systems are widely accepted in many fields, but have received less attention within Political Science. In Purpura & Hillard (2006), we used a common machine learning technique (support vector machines or SVMs[9]) and cross-validation to demonstrate that a complex topic classification scheme could be automatically applied to congressional bill titles with high fidelity results. Bill titles average only 15 words, yet the system was able to find enough signals to nearly duplicate the findings of human researchers.

However, that study was of limited value for the Congressional Bills Project team's goal of lowering the costs of coding contemporary data. Today's bills are not a representative sample of the bills that have been introduced in the past. The language of bills addressing the same topic changes over time. Policy programs are created anew, or existing problems are framed in new ways. An efficient automated system must be able to recognize and adapt to such changes. Although others have suggested that high levels of precision and recall cannot be achieved in the absence of coherence between topic classification scheme and corpus (Hand, Mannila, & Smyth, 2001), we wanted to specifically investigate this subject because such a system would substantially reduce updating costs through automated classification.

The machine learning algorithm used here (and described below) has been extensively modified from what was reported in our earlier project. Interestingly, the changes were motivated not by a desire for increased performance of the system itself, but by a desire to mitigate Human-Computer Interaction (HCI) issues detected in that first generation tool. Users weren't able to quickly assemble sufficient data on "edge" cases to learn whether over-sampling, under-sampling, or drift were serious concerns. The upgrade presented here provides improved feedback about which cases are the most valuable for labeling first in successive iterations of an Active Learning process. In addition, we did not employ stratified sampling (by Congress) for the development of the training set in our earlier work.[10]

[9] Computer science reviewers of our work have rightly criticized us for a lack of thoroughness by failing to document our approach compared to all simple classifiers, such as kNN. We appreciate this criticism, and we have a simple response: SVMs were a tool that was available and familiar. It is inexpensive, well tested, well supported, and widely used. If a user of our framework desired, they could replace any of the machine learning algorithms with kNN or more advanced solutions which investigate opinion analysis. We do not claim that others cannot find a more efficient solution, nor do we claim that other hierarchical methods are less efficient. Our work is not about classifier efficiency as much as it is about end-to-end system efficiency. It is intended to show what can be accomplished with off-the-shelf technology produced by natural language (NLP) researchers and readily available for consumption. In fact, we count on the increasingly useful tools produced by NLP researchers to conduct more advanced research in later work.

[10] A stratified sample is non-random to account for the fact that a corpus is lopsided in some way. The distribution of cases by category may be uneven or the summarization features do not efficiently discriminate among categories. "Investigating Unsupervised Learning for Text Categorization" by Gliozzo, Strapparava, and Dagan suggests the use of selected samples by keywords to bootstrap text categorization. "An Efficient SVM Classifier for

Once again, we build on existing automatic text classification research in the Computer Science and Computational Linguistics literatures. A relatively comprehensive analysis (Yang and Liu, 1999) finds that support vector machines are usually the best performing model. The most typical feature representation first applies Porter stemming to reduce word variants to a common form (Porter, 1980) before computing term frequency in a sample divided by the inverse document frequency (to capture how often a word occurs across all documents in the corpus) (Papineni, 2001). A list of common words (stop words) may also be omitted from each text sample.

When unlabeled in-domain data are available, active learning approaches can lead to faster improvements in classifier accuracy compared to random selection and other approaches (Cohn et al., 1994). Active Learning is well referenced in the computer science literature and (as a term) it refers to proactively choosing cases for humans to label by estimating which cases will most improve system performance. The goal is to have humans labeling the cases which have the maximum impact while letting the computer label the easier, repetitive, cases.

Early active learning methods predicted hypothesis categories over the unlabeled data before selecting those cases for which the classifier had the lowest confidence (Lewis and Catlett, 1994). Subsequent approaches have sampled based on committees of classifiers (Dagan and Engelson, 1995; Freund et al., 1997; Liere and Tadepalli, 1997). Recently, Muslea (2006) has found better performance through Ensemble learning, or the use of multiple models to estimate an answer. We adopt a similar approach. When 3 different models fail to agree on topic assignment for any given case, the ensemble learning system flags the case before the Active Learning system prioritizes it for review by human coders.

Machine Learning Methods

The main purpose of automated text classification is to replicate and assist the performance of human labelers. In this case, the classification task consists of 226 categories. We exploit the natural hierarchy of the categories by first building a classification system to determine the major category, and then building a child system for each of the major categories that decides among the subcategories within the major class that is decided by the first level of classification. This is the simplification approach advocated by Koller and Sahami (1997), although more complex approaches exist that also take relationships among categories and their children into account (McCallum et al., 1998; Dekel et al., 2004).

Existing research indicates that combining the decisions of multiple statistical systems (a.k.a. ensemble learning) usually improves final results (Brill and Wu, 1998; Dietterich, 2000; Curran, 2002). We implement three modeling approaches that are freely available to the research community. We use

---

Lopsided Corpora" suggests that using the weighting capabilities of many off-the-shelf SVM toolkits is an easy solution for the use of a stratified sample. Thanks to Aleks Jakulin at Columbia for his assistance.

SVMlight for SVM classification (Joachims, 1998); the Bow toolkit for maximum entropy models (McCallum, 1996); and the Boostexter tool for the AdaBoost.MH algorithm (Schapire and Singer, 2000). The following three sections describe how each of our three classifiers is trained.

The SVM Model

The SVM system builds on binary one vs. one classifiers between each pair of categories, and selects a final category by choosing the category that is selected most often by the one vs. one classifiers. Other approaches are also common (such as a committee of one vs. all classifiers), but we found our approach to be more time efficient, with equal or greater performance. We use a linear kernel, Porter stemming, and a feature value that is slightly more detailed than the typical inverse document frequency feature. In addition, we prune those words in each bill that occur less often than the corpus average. Further details and previous results of the system are described in (Purpura and Hillard, 2006).

Boostexter Model

The Boostexter model can easily accommodate multi-category tasks, so only one model need be learned, which then decides among the candidate categories. The Boostexter tool allows for features of a similar form to the SVM, where a word can be associated with a score for each particular text example. We use the same feature computation as for the SVM model, and likewise remove those words that occur less than often than the corpus average. Under this scenario, the weak learner for each iteration of AdaBoost training consists of a simple question that asks whether the score for a particular word is above or below a certain threshold.

The Maxent Model

For the last model type, the Maxent model, we use the rainbow toolkit. The toolkit provides a cross validation feature, which we use to select the optimal number of iterations. Here, we provide just the raw words to rainbow, and let it run word stemming and compute the feature values.

Active Learning

About 10,000 bills are introduced in every two year Congress. As attractive as automated approaches may seem for a large corpus such as this one, social scientists must first be convinced that the potential adverse effects of topic drift on labeling reliability and precision are limited (Soroka et al. 2006; Baumgartner, Jones and Wilkerson 2002). The implications of topic drift exclude any approach that lacks systemic validation. Managing topic drift in the Congressional Bills Project will require some human annotation during each successive period.[11] However, how much human labeling is required to

[11] Other time decay based methods are also available. These methods did not work for our purposes and we will discuss the issues in future research. For the Congressional Bills team, we currently recommend erring on labeling

achieve accepted levels of accuracy is unknown at this point. Do they need to hand label 10% or 90% of the bills in a new congress?

Topic drift is just one type of bias, or perceived bias that automated topic labeling systems may introduce (Sugiyama and Ogawa, 1999). For better and worse, computers make systematic decisions. They cannot judge, they can only apply. We contend that the best resolution is to develop frameworks for managing bias that give humans maximum control and feedback from the process. In general, we also assume that available computing cycles are a sunk fixed cost, and we proceed influenced by dynamic budgeting models from operations management research (Pollack and Zeckhauser, 1996) for managing our resources under constraint. Doctors in an HMO are given a fixed set of resources and told to maximize expected benefit to society. They must dynamically calculate thresholds of patient illness and resource expenditures such that society's benefit is maximized. Patients entering an HMO at different times of the budget year may be treated differently based on prior resource consumption and the general level of illness in society. In effect, the doctor calculates the option value of withholding treatment to one patient in the current period to be able to apply treatment to 'sicker' patients in subsequent periods.

Similarly, our framework seeks to maximize performance subject to a multi-period budget constraint, through selective sampling of the cases to be reviewed by humans. The social scientist (as the budget gatekeeper) is given authority to expend a certain amount of money within a labeling cycle. She must decide which bills to label during each cycle to maximize her expected benefit from the process, while also controlling for reliability.

For our Active Learning experiments, we consider only the previous and current Congress as our sets of possible training data. Though we are moving towards the Pollack and Zeckhauser dynamic approach, here we describe a simpler method which replaces several dynamic decisions with static decisions for ease of explanation. The "previous" Congress is the hand labeled training set. To evaluate the effects of Active Learning sampling techniques for performance, we reserve half of the "current" Congress as our unlabeled test set and sample from the other half.

We sample blocks of 15 previously labeled bills as if they were selected and labeled by the Congressional Bills team using their tight quality control methods to achieve high levels of reliability and validity. The goal of the active learning system is to select the best cases for them to code. Our methodology alternates between periods of discriminating and random selection. Discrimination prioritizes cases with the least agreement among the three types of classifiers. Random selection sampling draws from the remaining bills (without replacement)[12].

---

more records than is actually necessary to obtain convergence with the system until issues with stratified sampling and time decay validation of topic drift management are resolved to cautious scientific satisfaction.

[12] Note that we are not attempting to generate random samples IID. It isn't necessary for the approach, but obtaining a pseudo-random sample is useful to inform the observer about convergence.

After selecting and simulating the annotation process,[13] we feed the updated data back into the system, generate new predictions and reevaluate performance. This production system allows researchers to annotate during the day based on active learning feedback, while a nightly batch job incorporates the updates, generates new results, including a new priority list for additional annotation. Accuracy estimates can be generated by cross-validating the samples labeled in each of the random iterations. This estimate will improve as the number of iterations increases (and thus the size of the available randomly sampled and labeled bills from the new Congress). When used in an operational system, this allows for calculation of the marginal benefit of continuation of the active learning coding process.

Evaluation Methods and Metrics

In our experiments, evaluation is straightforward because high quality information about "the ground truth" is available. To support our research, we assess performance against human labelers using a variety of experiments designed to triangulate support for our analysis across 50 years of legislative activity. Each of the experiments examines inter-coder agreement in common scenarios before examining case examples to get to the details of the performance failures. In addition, we also generate the Recall statistics for clarity.

The Recall statistic has the same definition in our paper as it does in information retrieval. It represents the percentage of human-assigned categories that the system also produced. In contrast, Precision is the percentage of machine assigned categories that also appeared in the human assigned labels. These statistics make the simplistic (and frequently incorrect) assumption that the human labels are always correct.

Instead of simply adding up instances of agreement between the machine generated predictions and the human team's estimate, our analysis methods discount performance for "confusion" and for the probability of agreement by chance. With so many topic categories, of course, the probability of agreement by chance is quite low. But if these methods were used for a dataset where the number of categories was small, the probability of agreement by chance would deserve scrutiny because it would be substantially greater. Our methods also discount for small record numbers, which is another potential problem that is not an actual problem in the current analysis.

Our performance metrics are commonly used in topic spotting and clustering analysis. Performance is measured by the system's ability to achieve high levels of inter-coder agreement with the well-trained human coders. We use inter-coder agreement statistics common in the topic

[13] In our case we are using the already labeled data to simulate what would happen if humans attentively labeled the data in real-time. We realize this isn't a perfect experiment because there are interaction effects from labeling.

classification literature to facilitate comparisons with other systems - Cohen's Kappa (Cohen, 1968) and AC1 (Gwet, 2002).

Cohen's Kappa statistic is a standard metric used to assess inter-coder reliability between two sets of results. Usually, the technique is used to assess agreement between two human coders, but the computational linguistic field also uses it to assess agreement between human and machine coders. Cohen's Kappa statistic is defined as:

$$\kappa = \frac{p(A) - p(E)}{1 - p(E)} \quad (1)$$

In the equation, *p(A)* is the probability of the observed agreement between the two assessments:

$$p(A) = \frac{1}{N}\sum_{n=1}^{N} I(Human_n == Computer_n) \quad (2)$$

Where *N* is the number of examples, and *I()* is an indicator function that is equal to one when the two annotations (human and computer) agree on a particular example. *P(E)* is the probability of the agreement expected by chance:

$$p(E) = \frac{1}{N^2}\sum_{c=1}^{C} (HumanTotal_c \times ComputerTotal_c) \quad (3)$$

Where *N* is again the total number of examples and the argument of the sum is a multiplication of the marginal totals for each category. For example, for category 3, health, the argument would be the total number of bills a human coder marked as category 3, times the total number of bills the computer system marked as category 3. This multiplication is computed for each category, summed, and then normalized by N^2.

Due to bias under certain constraint conditions (Gwet, 2002), computational linguists also use another standard metric named the AC1 statistic to assess inter-coder reliability. The AC1 statistic corrects for the bias of Cohen's Kappa by calculating the agreement by chance in a different manner. It has similar form:

$$AC1 = \frac{p(A) - p(E)}{1 - p(E)} \quad (6)$$

But the *p(E)* component is calculated differently:

$$p(E) = \frac{1}{C-1}\sum_{c=1}^{C} (\pi_c (1 - \pi_c)) \quad (7)$$

Where *C* is the number of categories, and π_c is the approximate chance that a bill is classified as category *c*.

$$\pi_c = \frac{(HumanTotal_c + ComputerTotal_c)/2}{N} \quad (8)$$

In this paper, we report both Cohen's Kappa and AC1 because the two statistics provide consistency with topic spotting research and most other research in the field. For coding tasks of this level of complexity, a Cohen's Kappa or AC1 statistic of 0.70 or higher is considered to be very good agreement between coders.

Human Computer Interfaces

Ultimately, a useful classification tool needs to be accessible to non-expert users. The first step in this research process involved Wilkerson providing two sets of bills to Purpura & Hillard for testing. The first set – the training set – had a topic associated with each bill title (or short description for bills introduced prior to 1973). The second set – the testing set – had its topics removed so that only Wilkerson would know the human decision for each bill. The first experiments were conducted on slightly less than 10,000 bills and completed in a few days. Not only had the predictions been fairly accurate but, like a spellchecker, sometimes the human was right and sometimes the computer was right. Automated classification looked to be an inexpensive approach to simple validity checking[14].

After this initial experiment, our assessments became much more formalized. However, Wilkerson didn't know how to interpret the confidence estimates (in sigmoid curves) and couldn't easily format data for use by the NLP tools. In response, Purpura & Hillard created a "tool harness" with wrapper functions that require simple comma delimited files and Excel spreadsheets as inputs and outputs, and reconfigured performance indicators from point estimates on a sigmoid curve to topic rankings corresponding to the system's confidence estimates. They also introduced confusion tables and explained their use. [15]

[14] It's also interesting to note that the use of bill titles was a convenience which happened to validate that the computer could pick up on the same types of cues that humans were seeing in the titles. We didn't know until much later when we extracted the full text of bills from THOMAS for analysis that bills titles would actually be a less noisy signature for bills than the full text.

[15] Confusion tables are so well used in the computational linguistic literature to examine category performance of systems (Hand, Mannila, & Smyth, 2001) that researchers don't even cite their use any longer. Especially since the precision, recall, AC1, and F1 statistics incorporate confusion measures within the standard metrics the discipline

Two different implementations of the system are now maintained. Purpura & Hillard maintain the original version, used for this paper, which is still used for bulk processing because it is integrated into a Condor cluster in Purpura's home. Wilkerson (with the aid of Alan Gibson) has constructed an on-line tool to facilitate text annotation projects (by humans and/or machines) that includes assessment tools and active learning features. Users of this tool can either apply previously trained algorithms such as the Policy Agendas topic system to new data, or alternatively upload their own codebooks, data and algorithms.

Some HCI challenges remain of course. Coders must be well trained in how to apply their particular topic coding system. They must learn how to manage data, interpret output (including summary scores and the confusion tables), and make adjustments to take advantage of the information provided by the NLP system. Although this paper sheds some light on this subject, we leave the most of this discussion for a future paper.

Experiments and Findings

We first test the accuracy of our classifiers on the entire labeled corpus by splitting the data into two randomly selected halves (i.e. bills are not divided temporally or by any other systematic distinction). Thus, about 190,000 labeled samples are used to predict to about 190,000 unlabeled cases. With 20 major topics and 226 subtopics, a random assignment of bills to topics and subtopics would produce very low levels of accuracy. It is therefore very encouraging to find (Table 2) high levels of prediction accuracy across the different algorithms -- and especially for the combined system that is at the center of the Active Learning system developed in this paper. The overall improvement over the single SVM system is statistically significant at the .01 level of confidence.[16] And, as mentioned, not all of these errors should be attributed to failures of the automated systems, since many are undoubtedly human generated.

---

uses for performance assessment (see Hand, Mannila & Smyth, 2001 or Yang, Callan, and Shulman, 2006 for examples). When researchers don't use these metrics and fail to summarize confusion effectively, they hide important details of the performance of their work.

[16] The next version of this paper will include the n-fold cross validation of the experiments. The n-fold results do not change the conclusions in this paper, but they inform the systemic weights that we will give to selecting stratified training examples. Weighting the number of training examples by category is a standard operation documented in the SVMlight manual and in the papers of Thorsten Joachims. One example of the use of weighting to normalize training across categories occurs in "An Efficient SVM Classifier for Lopsided Corpora" by (Zhang et al., 2006).

Table 2: Bill Title Prediction Recall for Three Model Types				
	SVM	**Maxent**	**Boostexter**	**Combined**
Major topic N=20	87.4%	85.5.%	83.7%	87.7%
Subtopic N=226	79.1%	77%	74.4%	
Results are based on using approximately 190,000 human labeled samples to predict approximately 190,000 unlabeled cases				

Table 3 provides additional details about alternative sampling scenarios and their effects for inter-coder agreement when just using the SVM system alone. Training on the same Congress's bills achieves a median inter-coder agreement of 0.95 at the major topic level and 0.92 at the subtopic level. This scenario 'cheats' because it is completely unfair to test the system using the same training and testing samples. At the same time, it also demonstrates that the system does not achieve perfect inter-coder agreement even under ideal conditions.

Table 3 also provides insight into the effectiveness of using House bills to predict Senate bill topics (and vice versa). Median inter-coder agreement levels in this experiment are 0.84 and 0.74, at the major and subtopic levels respectively.

Training on the previous 5 congresses to predict the current congress yields average inter-coder agreement of 0.77 and 0.62. If we use just the previous congress to predict the next congress, on average the system achieves inter-coder agreement of 0.79 at the major topic level and 0.66 at the subtopic level – close to the inter-coder reliability levels reported for human annotators.[17]

[17] http://www.congressionalbills.org/BillsReliability.pdf

Table 3. Prediction Results for Different Training Sets (80-105th Congresses) -- SVM Model Only							
	Recall Major	**Recall Subtopic**	**S | M[18]**	**AC1 Major**	**AC1 Subtopic**	**Kappa Major**	**Kappa Subtopic**
Train on Self, Test on Self (n=2)							
Median	0.954	0.924	0.971	0.954	0.924	0.951	0.922
Average	0.955	0.924	0.969	0.954	0.922	0.951	0.921
S.D.	0.009	0.015	0.008	0.008	0.015	0.009	0.015
Train on Random Half of Entire Set to Predict Other Half (n=10)							
Median	0.874	0.791	0.904	0.869	0.790	0.865	0.787
Average	0.874	0.791	0.904	0.869	0.790	0.865	0.787
S.D.	0.000	0.001	0.001	0.000	0.000	0.001	0.001
Train on House to Predict Senate (n=2)							
Median	0.851	0.753	0.885	0.843	0.740	0.836	0.741
Average	0.851	0.753	0.885	0.843	0.740	0.836	0.741
S.D.	0.012	0.019	0.011	0.013	0.021	0.011	0.019
Train on Previous 5 Congresses to Predict Current (n=20)							
Median	0.778	0.626	0.803	0.767	0.606	0.762	0.621
Average	0.775	0.637	0.819	0.764	0.618	0.756	0.625
S.D.	0.057	0.081	0.048	0.060	0.085	0.057	0.072

[18] The "S| M" heading reads, "Subtopic precision given that a major topic is correct" and it means the precision of the subtopic category, given that we've predicted the major topic correctly.

Bill Titles as a Proxy for Bill Substance

The original decision to use the bill title for coding purposes was justified by a feature of the American Congress. The Rules of the House of Representatives encourage the parliamentarian to focus on a bill's title when making referral decisions. This in turn encourages sponsors to choose descriptive titles. However, no validity check has ever been conducted to see whether this assumption holds up in practice, and prior research has highlighted legislators' incentives to manipulate the bill referral process, perhaps by drafting misleading bill titles (King 1997).

As a practical matter, the Congressional Bills Project team was interested in tracing legislative activity during the post WWII time period, whereas the full texts of bills are only available digitally from 1989 to the present.

Do the titles of bills accurately reflect their substance? To find out, we extracted a sampling of the full text of public bills from the Library of Congress THOMAS website for the 1991-98 time period. We first sampled bill numbers (without replacement) from the Congressional Bills database. We then "scraped" the complete text of these bills by HTTP automation using tools developed by Purpura, and pre-processed them to eliminate irrelevant HTML tags using software developed by Purpura, Mitchell, and Hillard. [19] The process of eliminating irrelevant HTML tags also divided the full-text into segments with section boundaries which are used as an additional feature for the SVM system.[20] The 3,000 full-text bills were then divided in half for cross-validation and processed using only the SVM system.

Training on 1500 bill texts to predict to 1500 other bill texts yielded performance very similar to that reported for bill titles (0.74 major topic; 0.65 subtopic). In our view, these findings strongly support the working assumption that bill titles are in fact reliable approximations of bill substance.

[19] Thanks to Darren Mitchell, the Connecting To Congress Project, the Congressional Management Foundation, and David Lazer at Harvard for writing and supporting the software which diced the THOMAS HTML extract.

[20] Processing with just the section boundaries yields greater than 0.5 AC1 inter-coder reliability, which indicates that Congressional legislation is fairly similar in terms of the boiler plate language in documents in the same category. This presents an interesting avenue for future research, to integrate the data with our use of bill titles in general. However, this step is not currently feasible because the section boundaries are not available for bills which are not digitized.

Predicting the Future: How much human intervention is enough?

As discussed, a central goal of the Congressional Bills Project is to draw on information from previous congresses to automatically label bills in subsequent congresses. Our next set of experiments evaluated the proposed system's ability to draw on previous research to label the 100th through the 105th Congresses, as if each was being newly presented to the coding team for annotation. This is a solid test, because the machine learner performance for the 100th through the 105th Congresses is (in each case) at the median or lower in comparison to the other congresses.

Table 4. Train on Previous Congress to Predict Current (n=25) e.g. train on 80th to Predict 81st through train on 104th to Predict 105th Congress – SVM model only

	Recall Major	Recall Subtopic	S | M[21]	AC1 Major	AC1 Subtopic	Kappa Major	Kappa Subtopic
Median	0.799	0.665	0.833	0.788	0.647	0.784	0.660
Average	0.799	0.671	0.837	0.791	0.655	0.778	0.655
S.D.	0.045	0.081	0.057	0.049	0.088	0.043	0.070

Step 1: Use prior results as the training set, predict on the unlabeled

The first step is to assess the baseline for the level of inter-coder agreement the system is able to obtain when the human coding team does zero work. For reasons documented prior, we don't recommend that the team do this, but it is useful to calculate the baseline for performance assessment. The results are presented in column 3 of Table 5 (Before Active Learning).

Step 2: Use Active Learning to simulate the performance of the human team.

The second step simulates system performance by adding (to the data used in Step 1) "correctly" labeled samples selected by the Active Learning system. The way to think about this is that the Active Learning system generates a work list each day for the coding team, the team labels each bill on the

[21] The "S| M" heading reads, "Subtopic precision given that a major topic is correct" and it means the precision of the subtopic category, given that we've predicted the major topic correctly.

work list, and the next day the system tells them to stop coding or provides another work list. The results are presented in column 4 of Table 5 ("After Active Learning").

The results presented in Table 5 for the 105th Congress show nearly an 8% improvement in inter-coder agreement between the machine and the human team. When compared against just labeling records using random sampling, the gain is about 3% and it is significant, when tested using a t-test. For the purposes of this paper, our selection of 15 points in 30 alternating periods was "optimized" for both the 105th Congress and explanatory compactness. In real scenarios, we expect to calculate the distribution of random and Active Learning points per day using our version of the Pollack & Zeckhauser system.

Table 5: Inter coder agreement for major category before and after introduction of the Active Learning method. (Ensemble Learning System)[22]

Initial Training Data from:	Predicted Congress	Before Active Learning: AC1 for Major Topics	After Active Learning: AC1 for Major Topics
99th Congress	100th Congress	0.78	0.80
100th Congress	101th Congress	0.80	0.84
101th Congress	102th Congress	0.78	0.80
102th Congress	103th Congress	0.80	0.83
103th Congress	104th Congress	0.77	0.80
104th Congress	105th Congress	0.71	0.79

When Step 2 is finished, the coding team will have reviewed all of the high priority bills selected by the Active Learning system. In addition, they will have labeled a pseudo-random sample of the bills. During each Active Learning iteration (one day in our story line), the inter-coder agreement of the pseudo-random sample is calculated at the end of the day and the result is inserted into a distribution. The effect is a snapshot of the performance of the system for that iteration. By itself, this score is not interesting, but when performance is evaluated for convergence of the standard deviation of the distribution of these scores, it is insightful. A wide variance represents unmanaged issues with heterogeneity in the categories. A thread of small variance cycles represents convergence of the model.

[22] The "After Active Learning" results are measured as-if Active Learning were to continue for 1,500 records

Step 3: Analyze the Daily Reports (confusion tables, disagreement reports, and rankings) to monitor improvement and make adjustments

Step 3 is managed by the coding team leader. The Daily Reports compares the machine learner's predictions to the actual inputs of the human coders. In our experiments, when the system reaches a convergence state, one of the "top 3 major categories" predicted by the machine is same as the human labeled category with probability in excess of 0.9. When we correctly estimate the major category of the bill, the probability of the human label matching the subtopic prediction approaches 0.9.

The Daily Report of Disagreement *(e.g. for bills predicted to be in Category 20)****.*** The following report is from the actual experiment used to predict the codes of the bills of the 105th Congress. This is a snippet of the full report, which also lists subtopics, the bill's title, and confidence estimates. "Major Actual" is the input value from humans. For all but one of these cases of first choice disagreement, the human code matches the second or third predictions of the classifier.

Bill ID	Major Actual	Ranked Alternatives for Major Category Disagreement
395134	10	20 **10** 7 5 3 21 12 15 2 1 16 19 18 17 6 8 4 13
396499	17	20 **17** 12 21 10 15 19 1 3 7 5 16 18 8 2 14 6 4
396114	15	20 12 **15** 21 7 10 3 5 16 1 19 2 18 4 17 6 8 13
393942	3	20 **3** 5 7 15 21 12 1 2 16 10 19 17 18 14 8 13 4
395201	10	20 **10** 21 15 12 7 3 5 1 16 14 19 2 18 4 8 17 13
361419	15	20 10 **15** 12 7 21 16 3 5 1 19 18 2 8 4 17 13 6
395090	8	20 7 15 5 12 3 21 1 **8** 10 19 16 18 2 17 6 13 4

The report of disagreement is a more efficient way for the master coder to oversee the work of others without examining the large proportion of entries that are correct. These disagreements are also valuable for catching human mistakes, because the machine is often correct while the human coders are wrong. Although we don't track these occurrences in software, inspections by master coders suggest that human errors account for perhaps 50% of all disagreements. Part of the explanation is that many bills have ambiguous titles, such as *"A bill to improve the economic security of workers, and for other*

purposes." Ideally, a coder will research such bills in order to gain a clearer sense of their purposes. The machine learner, in contrast, never forgets.

Budgeting Perspective

The system also allows us to estimate the marginal cost of additional improvements in performance (Figures 1 and 2). We discuss the graphs in more depth in future research, but for now we note that the information they contain can be used to decide when labeling of additional data points using our Active Learning process is no longer cost effective in terms of improved accuracy, and the user may want to switch to alternative methods.

In the current system, "alternative methods" are left to the devices of the coding team. For example, information gleaned from confusion tables (discussed below) may lead it to conduct in depth analyses of entire categories or sub-categories. The goal of the system is to provide users with the information and tools they need to conduct quantitative analyses as well as qualitative reviews of records and category record sets.

Figure 1: This graph shows that cumulative gains attributed to points selected by Active Learning outperform those selected by random sampling. In our example, approximately 600 bills are labeled by the process to complete the Active Learning process. By smartly choosing the bills to label in this manner, the user could expect to see at least a few percentage points of increased quality for the same cost.

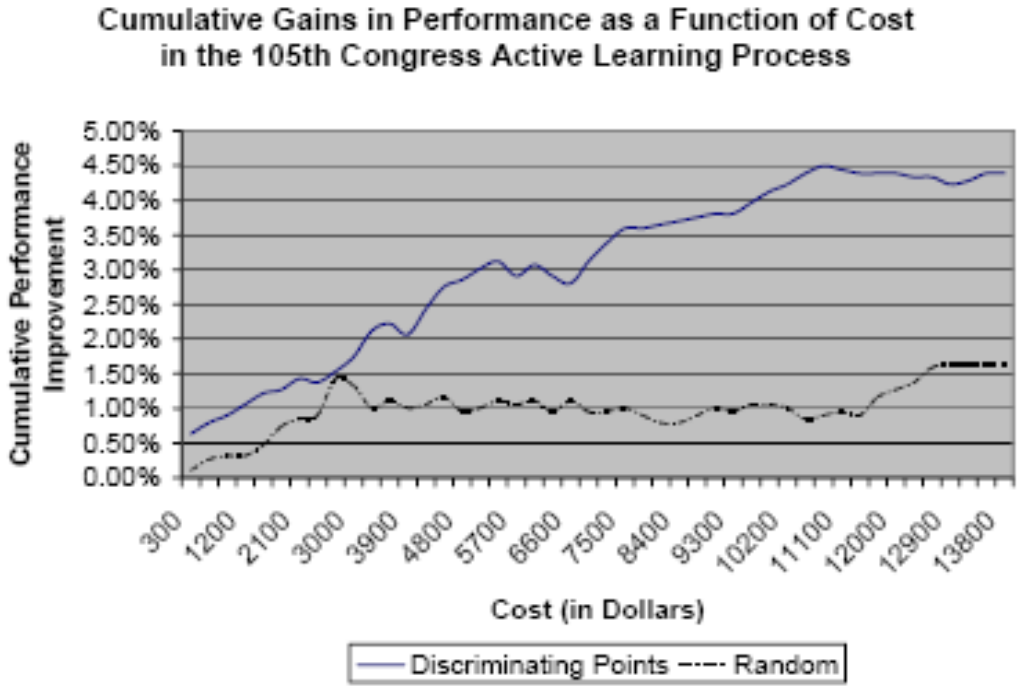
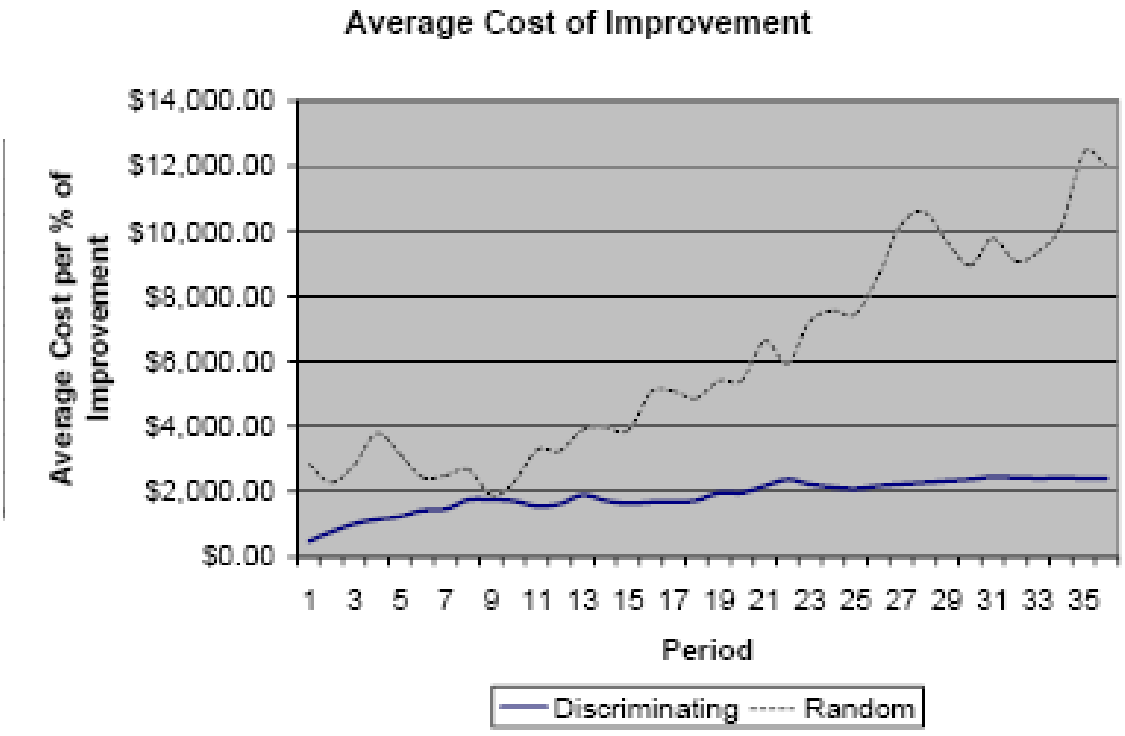


Figure 2: This graph shows the average cost of increasing quality in the data set. Its first derivative is the marginal cost of quality increase, which is sometimes negative in our process due to fluctuations in probabilistic outcomes. When the marginal cost of quality increase exceeds the marginal benefit, the user should stop labeling bills using our Active Learning approach and switch to a different approach. In our example, this occurs at around 40 iterations of Active Learning (20 of discriminating and 20 of random sampling), or labeling approximately 600 bills. The net gain in performance at that point was about 4.5% over the threshold of using the 104th Congress as training data.



What Happens If We Start from Scratch?

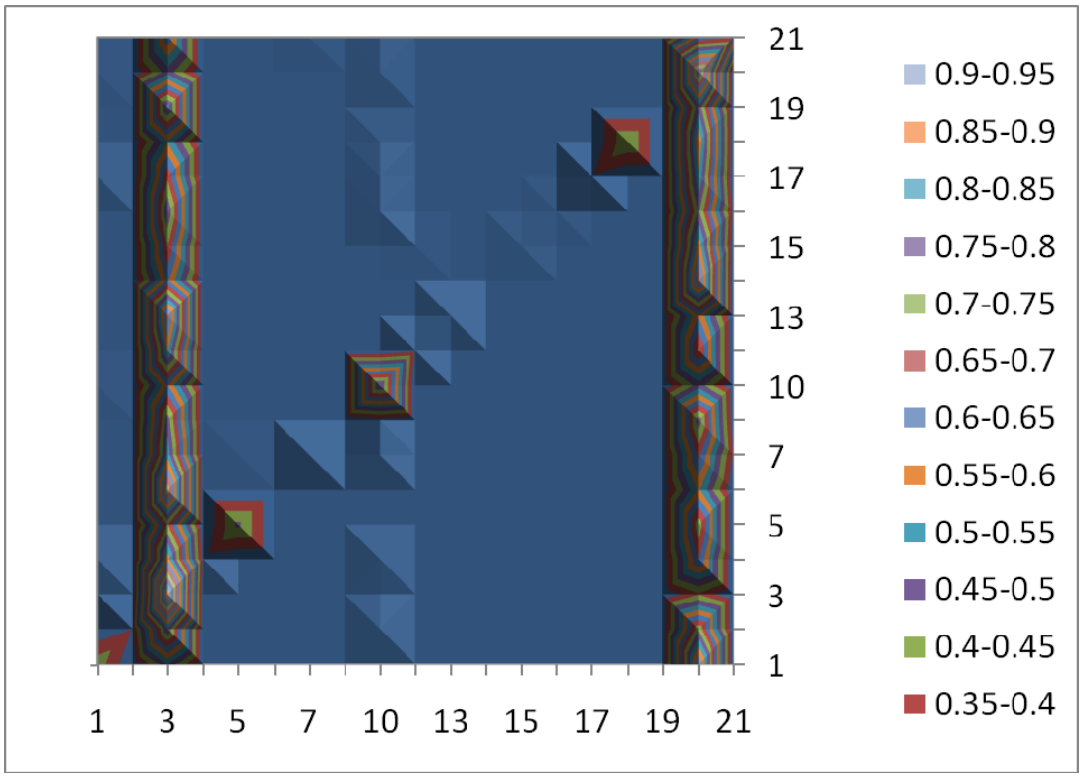
Confusion tables represent the estimated convergence of the categories after N iterations of Active Learning. In the example below, the 4 confusion tables presented estimate convergence for the 105th Congress after 8, 25, 50, and 100 iterations of Active Learning *if we begin from scratch*. In other words, in the first iteration, the total size of the training sample is 8 x 15 (120) bills. In the fourth, it increases to 100 x 15 (1500) bills.

Each row and column of a confusion table is labeled with a category number and each row sums to 100% (and each column should sum to 100% if the system were 100% accurate). If the system is predicting at 100%, the diagonal from the lower left corner to the upper right corner will be highlighted while all other areas of the table will be solid blue. "Hot spots" are areas where the first prediction of the system does not agree with the human assigned code. The sequence indicates that at 8 iterations, the system is (incorrectly) categorizing most bills as falling within two topic areas (3 and 20). However,

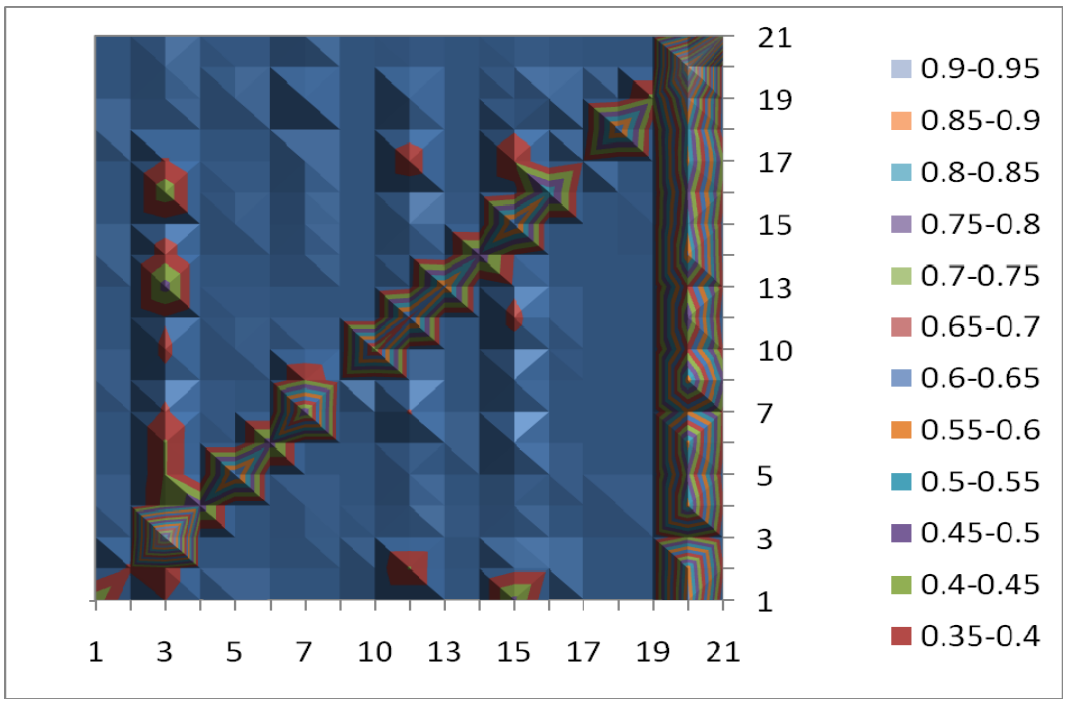
after 100 iterations (for a total training sample size of 1500), the vast majority of bills are being correctly classified, and in most cases, the errors can be attributed to legitimate disagreements about where a bill belongs (e.g. health (3) and social welfare (13) are much less distinct in substance than public lands (21) and civil rights and liberties (2).

At the same time, the confusion tables point to topic 20 (government operations) as central to explaining many of the remaining prediction errors. Despite inter-coder agreement of 0.9 for the bills that humans have assigned to this category (20x20), the system is erroneously labeling many other bills as category 20 bills. The tables appear to highlight the effects of a lopsided training set. Category 3 (health care bills) and category 20 (government operation bills) constitute significantly greater portions of the training sets. The confusion tables taunt us for not using a stratified sample (by distribution of topic cases) to more evenly train the system. An obvious response would be to employ the optimum (disproportionate) stratified sample for training, and, in fact, a test on the 105th Congress demonstrates that it speeds convergence by 4 rounds of Active Learning. In practice, we would also be able to counter these effects by training on hundreds of thousands of additional cases in the form of coded bills from previous Congresses.

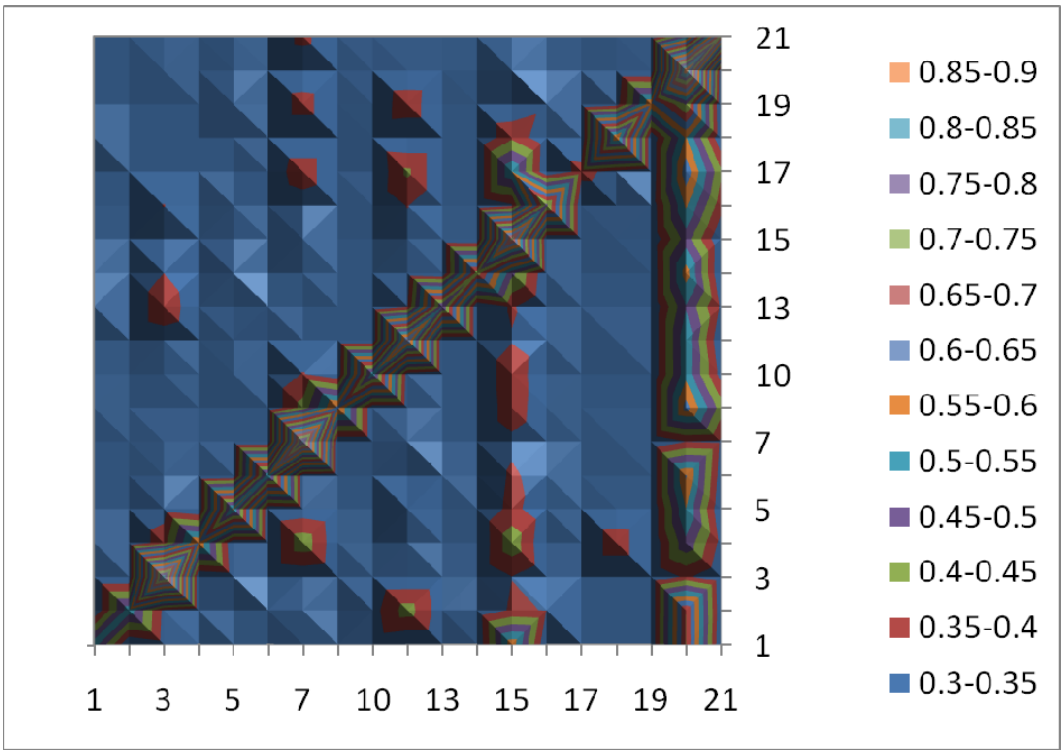
As research continues on the system, the confusion tables will play a key role in helping social scientists visualize mistakes and suggesting specific resolutions. We've even developed a script to export confusion tables from a run into a YouTube video to visualize convergence. See http://www.youtube.com/watch?v=sqMMlY84go4 for an example.



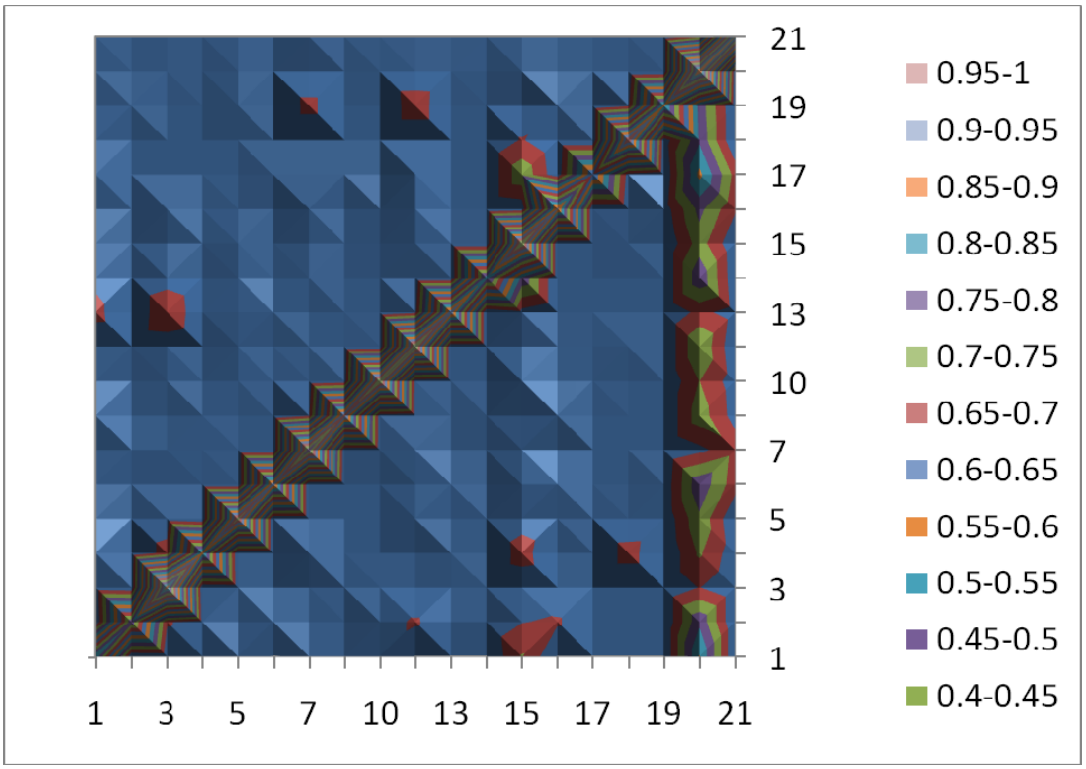
Confusion Table #1: Predictions for 105th Congress after 8 iterations of Active learning



Confusion Table #2: Predictions for 105th Congress after 25 iterations of Active Learning



Confusion Table #3: Predictions for 105th Congress after 50 iterations of Active Learning



Confusion Table #4: Predictions for 105th Congress after 100 iterations of Active Learning

Discussion

Locating individual documents in a subject area is a requirement of many, if not most of the questions that interest social scientists. "Topic classification," where documents are pre-coded according to some organizing framework, is widely used to facilitate search and summarization in these situations. We have found that machine learning methods can substantially lower the costs (both time and financial) of topic labeling individual events in large digitized databases. Starting from scratch, a sample of 1500 human labeled bills generates high levels of accuracy, even before additional valuable information (such as the labeled bills from prior congresses or stratified sampling) have been introduced. Compared to existing human labeling approaches, this virgin approach reduces labor requirements by about 80%.

The requirement to label 20% of the bills is the conservative estimate. When using training data from previous Congresses, labeling only 600 bills (8% of the total for the session) would achieve AC1 inter-coder agreements of 0.75 for Major Topics while labeling enough randomly sampled bills to have relative confidence in the finished data set. And this is without considering any benefits from humans recognizing human mistakes. An AC1 score of 0.75 is the best use of research funds, unless the group values quality more than the baseline required to achieve the goals we've enumerated. To move from an AC1 score of 0.75 to 0.80 using our Active Learning approach, they need to label an additional 900 bills (12%). Alternatively, they can take the feedback data the system provides on its current state and attempt to find and fix problematic categories on their own through systemic review. This may be more productive in some cases where a trained researcher can realize that a systemic fix is necessary.

Machine learners do not constitute a simple solution to the challenge of topic classification. They are not a silver bullet. They require high quality training samples and, as we have shown, regular human intervention to mitigate considerations such as topic drift as they are applied to new domains. Other methods may be more appropriate (i.e. less costly) for less ambitious tasks, such as research studies geared towards estimating proportions with little concern for search or validation, or those that investigate changing issue frames with limited concern for inter-temporal reliability. But in our view (one that is apparently shared by many in the field of computational linguistics) there is currently no substitute for machine learners when the objective is accurate computer-generated tags of complex individual events. We believe that topic classification is, and will continue to be, a central objective of an increasing number of data collection projects in Political Science and beyond. Properly applied, machine learning systems hold considerable promise as cost effective approaches to accomplishing these objectives.

Some have argued that this approach is too "black box", too complex, or not formally demonstrated. With regard to the black box critique, we hope readers will appreciate that every assigned code can be scrutinized, and the results can be evaluated according to accepted standards of reliability. To the complexity argument, we appreciate the allure of simpler solutions, but in many respects we consider our approach to be more transparent and understandable than other available methods. To the formalism critique, we point to existing research that shows that machine learning

methods not based on formal modeling consistently outperform other approaches based on formal modeling (Hand, 2006; Breiman, 2001; Yang and Liu, 1999).

But perhaps the most relevant critique pertains to the cost issue that motivated this project in the first place. Our methods dramatically reduce the costs of labeling, but other methods exist that can apply labels to many more (even millions more) records in about the same amount of time. Why bother? This question reminds us of a longstanding disciplinary debate about parsimony versus accuracy. However, while that debate centered on trade-offs between simple theories of human behavior that explains a little and complicated theories that explain more, the current controversy centers on whether measurement precision on a per record basis is desirable. In *The Logic of Comparative Social Inquiry*, Przeworski and Teune (1970) argued that parsimony is preferred (even at the expense of accuracy):

> "The goal of social science is to explain social phenomena [… Moreover,] the generality and parsimony of theories should be given primacy over their accuracy. In other words social science theories, rather than explaining phenomena as accurately as possible in terms relative to specific historical circumstances, should attempt to explain phenomena wherever and whenever they occur" (17).

The counterpoint was made in 1994 by a prominent group of researchers who challenged this perspective by arguing that neither accuracy nor parsimony is always preferred. Referring to this approach as the "injunction of Przeworski and Teune (1982)," King, Keohane, and Verba suggest that it has spawned an entire research tradition that "tries to say something about a class of events or units without saying anything in particular about a specific event or unit." (35) The problem with this injunction, according to the authors, is that it sometimes ignores "the requirement that the facts...that go into the general analysis must be accurate" (36).

We couldn't agree more. Parsimony guards against "over-fitting" of models. But the attraction of parsimony as a scientific standard is lessened to the extent that theories of underlying data structure are falsifiable. And even concise theories can require knowledge of individual cases. For example, the hypothesis that "western legislators cosponsor more environmental bills than eastern legislators" is a simple prediction about proportions. But it cannot be studied without individual case labels.

References

Adler, E. Scott and John D. Wilkerson. (forthcoming) "Intended Consequences? Committee Reform and Jurisidctional Change in the House of Representatives." Legislative Studies Quarterly.

Baum, M. *Soft News Goes to War: Public Opinion and American Foreign Policy in the New Media Age*. 2003. Princeton N.J.: Princeton University Press.

Baumgartner, Frank, Bryan Jones, and John Wilkerson. 2002. "Studying Policy Dynamics." In Baumgarter and Jones, *Policy Dynamics*, Chapter 2.

Eric Brill and Jun Wu. 1998. "Classifier combination for improved lexical disambiguation". In *Proc. ACL*, pages 191–195.

Breiman, L. 2001. "Statistical Modeling: The Two Cultures". *Statistical Science*. 16(3):199-231.

Cristianini, N., Shawe-Taylor, J., and Lodhi, H. Latent semantic kernels. in Brodley, C. and Danyluk, A. Proceedings of ICML-01, *18th International Conference on Machine Learning*. (San Francisco, US, 2001), Morgan Kaufmann Publishers, pages 66–73.

Cohen, J. 1968. "Weighted Kappa: Nominal Scale Agreement with Provision for Scaled Disagreement or Partial Credit." *Psychological Bulletin*, 70(4):213–220.

Cohn, D. and L. Atlas, and R. Ladner. 1994. "Improving generalization with active learning". *Machine Learning*, 315:201–221.

Curran, J. 2002. "Ensemble methods for automatic thesaurus extraction". Proc. Empirical Methods in *Natural Language Processing*, pages 222–229.

Dagan, I. and S. Engelson. 1995. "Committee-based sampling for training probabilistic classifiers". Proc. *Int. Conf. on Machine Learning*, pages 150–157.

Dekel, O., J. Keshet, and Y. Singer. 2004. "Large margin hierarchical classification". ACM International Conference Proceeding Series.

Dietterich, T. 2000. "Ensemble methods in machine learning". Lecture Notes in Computer Science, 1857:1–15.

Fowler, J.ames H. 2006. "Legislative cosponsorship networks in the US House and Senate." *Social Networks* 28: 454-465.

Freund, Y., H.S. Seung, E. Shamir, and N. Tishby. 1997. "Selective Sampling Using the Query by Committee Algorithm". *Machine Learning*, 28(2):133–168.

Gwet, K. "Kappa Statistic is not Satisfactory for Assessing the Extent of Agreement Between Raters." in *Statistical Methods For Inter-Rater Reliability Assessment, No. 1,* April, 2002.

Hand, D., Mannila, H., Smyth, P. 2001. *Principles of Data Mining (Adaptive Computation and Machine Learning).* Cambridge: The MIT Press.

Hand, D. 2006. "Classifier Technology and the Illusion of Progress". *Statistical Science*. 21(1):1–14.

Joachims, T. 1998. "Text categorization with support vector machines: Learning with many relevant features". In Proc. *European Conference on Machine Learning*.

King, D. 1997. *Turf Wars*. Chicago: University of Chicago Press.

King, G. and Lowe, W. 2003. "An Automated Information Extraction Tool for International Conflict Data with Performance as Good as Human Coders: A Rare Events Evaluation Design." *International Organization* 57(3, July):617–642.

King, G. R. O. Keohane, S. Verba. 1994. *Designing Social Inquiry: Scientific Inference in Qualitative Research.* Princeton: Princeton University Press.

Koller. D. and M. Sahami. 1997. "Hierarchically classifying documents using very few words". Proc. Int. Conf. on Machine Learning, pages 170–178.

Lewis, D. and J. Catlett. 1994. "Heterogeneous uncertainty sampling for supervised learning". In Proc. Int. Conf. on Machine Learning.

Liere, R. and P. Tadepalli. 1997. "Active learning with committees for text categorization". Proceedings of the Fourteenth National Conference on Artificial Intelligence, pages 591–596.

Mann, G., Mimno, D. and McCallum, A. 2006. "Bibliometric impact measures leveraging topic analysis". JCDL '06: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries. Pp 65—74. Chapel Hill, NC, USA.

McCallum, A., R. Rosenfeld, T. Mitchell, and A. Ng. 1998. "Improving text classification by shrinkage in a hierarchy of classes". Proc. Int. Conf. on Machine Learning, 367.

McCallum, A. 1996. "Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering". http://www.cs.cmu.edu/mccallum/bow.

Muslea, I. 2006. "Active learning with multiple views". *Artificial Intelligence Research*, 27:203–233.

Papineni, K. 2001. "Why inverse document frequency?" In Proc. North American Chapter of the Association for Computational Linguistics.

Pollack, H. and R. Zeckhauser. 1996. "Budgets as dynamic gatekeepers". *Management Science*, 42:642–658.

Porter, M.F. 1980. "An algorithm for suffix stripping". *Program*, 16(3):130–137.

Purpura, S and Hillard, D. 2006. “Automated classification of congressional legislation”. In Proc. Digital Government Research, pages 219–225.

Przeworski, A and Teune, H. 1970. *The Logic of Social Inquiry.* New York: Krieger Pub Co (March 1982).

Quinn, K, Monroe, B, Colaresi, M., Crespin, M. and Radev, D. 2006. “An Automated Method of Topic-Coding Legislative Speech Over Time with Application to the 105th-108th US Senate.” Annual Meetings of the Society for Political Methodology.

Schrodt P., Davis, S., and Weddle, J. 1994. “Political Science: KEDS—A Program for the Machine Coding of Event Data”. Social Science Computer Review, Vol. 12, No. 4, 561-587 (1994).

Schapire, R.E. and Y. Singer. 2000. “Boostexter: A boosting based system for text categorization”. Machine Learning, 39(2/3):135–168.

Schrodt, P and D. Gerner. 1994. “Validity Assessment of a Machine-Coded Event Data Set for the Middle East, 1982-92” *American Journal of Political Science*. 38 (3): 825-44/

Sebastiani, F. “Machine learning in automated text categorization”. *ACM Computing Surveys*, 34(1).

Soroka, S., C. Wlezien, and I. McLean. 2006. “Public Expenditure in the UK: How Measures Matter”. Journal of the Royal Statistical Society, pages 255–271.

Sugiyama, M. and H. Ogawa. 1999. “Functional analytic approach to model selection–Subspace information criterion”. Proc. Workshop on Information-Based Induction Sciences, pages 93–98.

Sulkin, T. 2005. *Issue Politics in Congress*. Cambridge University Press.

Yang, Y. and X. Liu. 1999. “A re-examination of text categorization methods”. In Proc. of SIGIR-99.

Yang, H., Callan, J., and Shulman, S. 2006. “Next steps in near-duplicate detection for eRulemaking.” *Proceedings of the Sixth National Conference on Digital Government Research*. San Diego, CA.