PRE-PUBLICATION VERSION. Cite as JITP 4:4, Forthcoming. There are a few known changes to the colors of the graphs and there may be other editorial changes as suggested by the editors.

Computer Assisted Topic Classification for Mixed Methods Social Science Research

Dustin Hillard

University of Washington

Stephen Purpura

Cornell University

John Wilkerson

University of Washington

Abstract

Social scientists interested in mixed methods research have traditionally turned to human annotators to classify the documents or events used in their analyses. The rapid growth of digitized government documents in recent years presents new opportunities for research but also new challenges. With more and more data coming online, relying on human annotators becomes prohibitively expensive for many tasks. For researchers interested in saving time and money while maintaining confidence in their results, we show how a particular supervised learning system can provide estimates of the class of each document (or event). This system maintains high classification accuracy and provides accurate estimates of document proportions, while achieving reliability levels associated with human efforts. We estimate that it lowers the costs of classifying large numbers of complex documents by 80% or more.

Keywords: Topic classification, data mining, machine learning, content analysis, information retrieval, text annotation, Congress, legislation.

Technological advances are making vast amounts of data on government activity newly available, but often in formats that are of limited value to researchers as well as citizens. In this paper, we investigate one approach to transforming these data into useful information. "Topic classification" refers to the process of assigning individual documents (or parts of documents) to a limited set of categories. It is widely used to facilitate search as well as the study of patterns and trends. To pick an example of interest to political scientists, a user of the Library of Congress' THOMAS website (http://thomas.loc.gov) can use its Legislative Indexing Vocabulary (LIV) to search for congressional legislation on a given topic. Similarly, a user of a commercial Internet service turns to a topic classification system when searching, for example, Yahoo! Flikr for photos of cars or Yahoo! Personals for postings by men seeking women.

Topic classification is valued for its ability to limit search results to documents that closely match the user's interests, when compared to less selective keyword-based approaches. However, a central drawback of these systems is their high costs. Humans—who must be trained and supervised—traditionally do the labeling. Although human annotators become somewhat more efficient with time and experience, the marginal cost of coding each document does not really decline as the scope of the project expands. This has led many researchers to question the value of such labor-intensive approaches, especially given the availability of computational approaches that require much less human intervention.

Yet there are also good reasons to cling to a proven approach. For the task of topic classification, computational approaches are useful only to the extent that they "see" the patterns that interest humans. A computer can quickly detect patterns in data, such as the number of *E*s in a record. It can then very quickly organize a dataset according to those

patterns. But computers do not necessarily detect the patterns that interest researchers. If those patterns are easy to objectify (e.g., any document that mentions *George W. Bush*), then machines will work well. The problem, of course, is that many of the phenomena that interest people defy simple definitions. *Bad* can mean good—or bad—depending on the context in which it is used. Humans are simply better at recognizing such distinctions, although computerized methods are closing the gap.

Technology becomes increasingly attractive as the size and complexity of a classification task increase. But what do we give up in terms of accuracy and reliability when we adopt a particular automated approach? In this paper, we begin to investigate this accuracy/efficiency tradeoff in a particular context. We begin by describing the ideal topic classification system where the needs of social science researchers are concerned. We then review existing applications of computer-assisted methods in political science before turning our attention to a method that has generated limited attention within political science to date: supervised learning systems.

The Congressional Bills Project (www.congressionalbills.org) currently includes approximately 379,000 congressional bill titles that trained human researchers have assigned to one of 20 major topic and 226 subtopic categories, with high levels of inter-annotator reliability.[1] We draw on this corpus to test several supervised learning algorithms that use case-based[2] or *learning by example* methods to replicate the work of human annotators. We find that some algorithms perform our particular task better than others. However, combining results from individual machine learning methods increases accuracy beyond that of any single method, and provides key signals of confidence regarding the assigned topic for each document. We then show how this simple confidence estimate can be employed to achieve additional classification accuracy more efficiently than would otherwise be possible.

Topic Classification for Social Science Document Retrieval

Social scientists are interested in topic classification for two related reasons: retrieving individual documents and tracing patterns and trends in issue-related activity. *Mixed method* studies that combine pattern analyses with case level investigations are becoming standard, and linked examples are often critical to persuading readers to accept statistical findings (King, Keohane, & Verba, 1994). In *Soft News Goes to War*, for example, Baum (2003) draws on diverse corpora to analyze media coverage of war (e.g., transcripts of Entertainment Tonight, the jokes of The John Stewart Show, and network news programs).

Keyword searches are fast and may be effective for the right applications, but effective keyword searches can also be difficult to construct without knowing what is actually in the data. A search that is too narrow in scope (e.g., "renewable energy") will omit relevant documents, while one that is too broad (e.g., "solar") will generate unwanted false positives. In fact, most modern search engines, such as Google, consciously reject producing a reasonably comprehensive list of results related to a topic as a design criterion.[3] The justification is that the systems cannot easily succeed at producing a comprehensive list without (expensive to obtain) domain-specific relevance feedback. When dealing with billions of documents, arbitrarily building topic classification systems for document sets is so expensive as to make it an unattainable goal for the same reasons that have previously motivated political scientists to shy away from conducting topic classification on large digital document sets. Despite this trade-off in design requirements, search systems are still useful because most users never examine results past the first few pages of result summaries.

Of course, in many cases topic classification systems have been applied to digital databases, but when applied they must also be designed to reflect the needs of researchers. Many political scientists rely on existing databases where humans have classified events (decisions, votes, media attention, legislation) according to a pre-determined topic system (e.g., Jones & Baumgartner, 2005; Poole & Rosenthal, 2003; Rohde, 2005; Segal & Spaeth, 2002).

In addition to enabling scholars to study trends and compare patterns of activity, reliable topic classification can save considerable research time. For example, Adler and Wilkerson (in press) wanted to use the Congressional Bills Project database to study the impact of congressional reforms. To do this, they needed to trace how alterations in congressional committee jurisdictions affected bill referrals. The fact that every bill during the years of interest had already been annotated for topic allowed them to reduce the number of bills that had to be individually inspected from about 100,000 to "just" 8,000.

Topic classification systems are also widely used in the private sector and in government. However, a topic classification system created for one purpose is not necessarily suitable for another. Well-known document retrieval systems such as the Legislative Indexing Vocabulary of the Library of Congress' THOMAS website allow researchers to search for documents using pre-constructed topics (http://thomas.loc.gov/liv/livtoc.html), but the THOMAS Legislative Indexing Vocabulary is primarily designed to help users (congressional staff, lobbyists, lawyers) track down contemporary legislation. This contemporary focus creates the potential for *topic drift*, whereby similar documents are classified differently over time as users' conceptions of what they are looking for change.[4]

For example, *Women's Rights* did not exist as a category in the THOMAS system until sometime after 1994. The new category likely was created to serve current users

better, but earlier legislation related to women's rights was not re-classified to ensure inter-temporal comparability. Topic drift may be of little concern where contemporary search is concerned, but it is a problem for researchers hoping to compare legislative activity or attention across time. If the topic categories are changing, researchers risk confusing shifts in the substance of legislative attention with shifts in coding protocol (Baumgartner, Jones, & Wilkerson, 2002).

So, what type of topic classification system best serves the needs of social scientists? If the goals are to facilitate trend tracing and document search, an ideal system possesses the following characteristics. First, it should be *discriminating*. By this we mean that the topic categories are mutually exclusive and span the entire agenda of topics. Search requires that the system indicate what each document is primarily about, while trend tracing is made more difficult if the same document is assigned to multiple categories. Second, it should be *accurate*. The assigned topic should reflect the document's content, and there should be a systematic way of assessing accuracy. Third, it should be *reliable*. Pattern and trend tracing require that similar documents be classified similarly from one period to the next, even if the terminology used to describe those documents is changing. For example, civil rights issues have been framed very differently from one decade to the next. If the goal is to compare civil rights attention over time, then the classification system must accurately capture attention despite these changing frames. Fourth, it should be *probabilistic*. In addition to discriminating a document's primary topic, a valuable topic system for search should also identify those documents that address the topic even though they are not primarily about that topic. Finally, it should be *efficient*. The less costly the system is to implement, the greater its value.

Human-centered approaches are attractive because they meet most of these standards. Humans can be trained to discriminate the main purpose of a document, and

their performance can be monitored until acceptable levels of accuracy and reliability are achieved. However, human annotation is also costly. In this paper, we ask whether supervised machine learning methods can achieve similar levels of accuracy and reliability while improving efficiency.

We begin by contrasting our approach to several computer-assisted categorization methods currently used in Political Science research. Only supervised learning systems have the potential to address the five goals of topic classification described above.

Computer Assisted Content Analysis in Political Science

Content analysis methods center on extracting meaning from documents. Applications of computer-assisted content analysis methods have developed slowly in political science over the past four decades, with each innovation adding a layer of complexity to the information gleaned from the method. Here we focus on a selected set of noteworthy projects that serve as examples of some of these important developments (Table 1).

[Table 1 here]

Data comparison, or keyword matching, was the first search method ever employed on digital data. Keyword searches identify documents that contain specific words or word sequences. Within political science, one of the most sophisticated is KEDS/TABARI (Schrodt, Davis, & Weddle, 1994; Schrodt & Gerner, 1994). TABARI turns to humans to create a set of computational rules for isolating text and associating it with a particular event category. The resulting system is used by researchers to analyze changing attention in the international media or other venues.

Systems based on keyword searching can meet the requirements for a solid topic classification system. Keyword search systems such as TABARI can be highly accurate and reliable because the system simply replicates coding decisions originally made by humans.[5] If the system encounters text for which it has not been trained, it does not classify that text. They can also be discriminating, because only documents that include the search terms are labeled. The system can also be probabilistic, by using rules to establish which documents are related to a topic area.

However, for non-binary classification tasks, achieving the ability to be both discriminating and probabilistic can be expensive because the system requires explicit rules of discrimination for the many situations where the text touches on more than one topic. For example, the topic *elderly health care issue*" includes subjects that are of concern to non-seniors (e.g. *health insurance*, *prevention*). Effective keyword searches must account for these contextual factors, and at some point other methods may prove to be more efficient for the same level of accuracy.

Unsupervised approaches, such as factor analysis or agglomerative clustering, have been used for decades as an alternative to keyword searching. They are often used as a first step to uncovering patterns in data including document content (Hand, Mannila, & Smyth, 2001). In a recent political science application, Quinn et al. (2006) have used this approach to cluster rhetorical arguments in congressional floor speeches.

Unsupervised approaches are efficient because typically they do not require human guidance, in contrast to data comparison or keyword methods. They also can be discriminating and/or probabilistic, because they can produce mutually exclusive and/or ranked observations. Consider the simplest case of unsupervised learning using agglomerative clustering—near-exact duplicate detection. As a researcher, if you know that 30% of the documents in a data set are near-exact duplicates (99.8% of text content is

equivalent) and each has the same topic assigned to it, it would be inefficient to ask humans to label all of these documents. Instead, the research would use an unsupervised approach to find all of the clusters of duplicates, label just one document in the cluster, and then trust the labeling approach to assign labels to the near-exact duplicate documents.[6]

But, to assess the accuracy and reliability of unsupervised methods on more complex content analysis questions, humans must examine the data and decide *relevance*. And once researchers begin to leverage information from human experts to improve accuracy and reliability (achieve a higher match with human scored relevance) in the data generation process, the method essentially evolves into a hybrid of the supervised learning method we focus on here.

Another semi-automated method, similar to the method proposed in this paper, is the supervised use of word frequencies in Wordscores. With Wordscores, researchers select model training *cases* that are deemed to be representative of opposite ends of a spectrum (Laver, Benoit, & Garry, 2003). The software then orders other documents along this spectrum based on their similarities and differences to the word frequencies of the end-point model documents. Wordscores has been used to locate party manifestos and other political documents of multiple nations along the same ideological continuum.

This method can be efficient because it requires only the human intervention required to select training documents and conduct validation. Wordscores is also probabilistic, because it can produce ranked observations. And the method has been shown to be accurate and reliable. However, its accuracy and reliability are *application dependent*, in that the ranks Wordscores assigns to documents will make sense only if the training documents closely approximate the user's information retrieval goal. Its small number of training documents limits the expression of the user's information needs. That is, Wordscores was not designed to place events in discrete categories.

Application-independent methods for conducting algorithmic content analysis do not exist. The goal of such a system would be to generate discriminative and reliable results efficiently and accurately for any content analysis question that might be posed by a researcher. There is a very active community of computer scientists interested in this problem, but, to date, humans must still select the proper method for their application. Many NLP researchers believe that an application independent method will never be developed (Kleinberg, 2002).[7]

As a part of this search for a more general method, Hopkins and King recently have developed a supervised learning method that gives, as output, "approximately unbiased and statistically consistent estimates of the proportion of all documents in each category" (Hopkins & King, 2007, p. 2). They note that "accurate estimates of these *document category proportions* have not been a goal of most work in the classification literature, which has focused instead on increasing the accuracy of *individual document classification*" (ibid). For example, a classifier who correctly estimates 90% of the documents belonging to a class must estimate incorrectly that 10% of those documents belong to other classes. These errors can bias estimates of class proportions (e.g., the proportion of all media coverage devoted to different candidates), depending on how they are distributed.

Like previous work (Purpura & Hillard, 2006), the method developed by Hopkins and King begins with documents labeled by humans, and then statistically analyzes word features to generate an efficient, discriminative, multi-class classification. However, their approach of estimating proportions is not appropriate for the researchers interested in mixed methods research requiring the ability to analyze documents within a class. Despite this limitation, mixed methods researchers may still want to use the Hopkins and King's method to validate estimates from alternative supervised learning systems. Because it is

the only other method (in the political science literature) of those mentioned that relies on human-labeled training samples, it does offer a unique opportunity to compare the prediction accuracy of our supervised learning approach in our problem domain to another approach (though the comparison must be restricted to proportions).

Supervised Learning Approaches

Supervised learning (or machine learning classification) systems, have been the focus of more than 1,000 scholarly publications in the computational linguistics literature in recent years (Mann, Mimno, & McCallum, 2006; Mitchell, 1997). These systems have been used for many different text annotation purposes but have been rarely used for this purpose in political science.

In this discussion, we focus on supervised learning systems that statistically analyze terms within documents of a corpus to create rules for classifying those documents into classes. To uncover the relevant statistical patterns in a corpus, annotators mark a subset of the documents in the corpus as being members of a class. The researcher then develops a *document representation* that draws on this *training set* to accurately machine annotate previously unseen documents in the corpus referred to as the *test set*.

Practically, a document representation can be any numerical summary of a document in the corpus. Examples might include a binary indicator variable, which specifies whether the document contains a picture, a vector containing *term weights* for each word in the document, or a real number in the interval (*0*, *infinity*) which represents the cost of producing the document. Typically, a critical selection criterion is empirical system performance. If a human can separate all of the documents in a corpus perfectly by asking whether a key email address appears, then a useful document representation would be a binary indicator variable specifying whether the email address appears in each

document. For classification tasks that are more complex, simplicity, calculation cost, and theoretical justifications are also relevant selection criteria.

Our document representation consists of a vector of term weights, also known as feature representation, as documented in Joachims (2002). For the term weights, we use both tf*idf (term frequency multiplied by inverse document frequency) and a mutual information weight (Purpura & Hillard, 2006). 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).[8] A list of common words (stop words) also may be omitted from each text sample.

Feature representation is an important research topic in itself, because different approaches yield different results depending on the task at hand. Stemming can be replaced by a myriad of methods that perform a similar task—capturing the signal in the transformation of raw text to numeric representation—but with differing results. In future research, we hope to demonstrate how alternative methods of pre-processing and feature generation can improve the performance of our system.

For topic classification, a relatively comprehensive analysis (Yang & Liu, 1999) finds that support vector machines (SVMs) are usually the best performing model. Purpura and Hillard (2006) applied a support vector machine (SVM) model to the corpus studied here with high fidelity results. We are particularly interested in whether combining the decisions of multiple supervised learning systems can improve results. This combined approach is known as *ensemble learning* (Brill & Wu, 1998; Curran, 2002; Dietterich, 2000). Research indicates that ensemble approaches yield the greatest improvements over a single classifier when the individual classifiers perform with similar accuracy, but make different types of mistakes.

Algorithms

We will test the performance of four alternatives: Naïve Bayes, SVM, Boostexter and MaxEnt.

Naïve Bayes. Our Naïve Bayes Classifier uses a decision rule and a Bayes probability model with strong assumptions of independence of the features (tf*idf). Our decision rule is based on MAP (maximum a posteriori) and it attempts to select a label for each document by selecting the class which is most probable. Our implementation of the Naïve Bayes comes from the rainbow toolkit (McCallum, 1996).

The SVM Model. The SVM system builds on binary pairwise classifiers between each pair of categories, and chooses the one that is selected most often as the final category (Joachims, 1998). Other approaches are also common (such as a committee of classifiers that test one vs. the rest), but we have found that the initial approach is more time efficient with equal or greater performance. We use a linear kernel, Porter stemming, and a feature value (mutual information) 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 results of the system are described in Purpura and Hillard (2006).

Boostexter Model. 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 (Schapire & Singer, 2000). 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 Boostexter model can accommodate multi-category tasks easily, so only one model need be learned.

The MaxEnt Model. The MaxEnt classifier assigns a document to a class by converging toward a model which is as uniform as possible around the feature set. In our case, the model is most uniform when it has maximal entropy. We use the rainbow toolkit (McCallum, 1996). This toolkit provides a cross validation feature that allows us to select the optimal number of iterations. We provide just the raw words to rainbow, and let it run word stemming and compute the feature values.

Figure 1 summarizes how we apply this system to the task of classifying congressional bills based on the word features of their titles. The task consists of two stages. In the first, we employ the ensemble approach developed here to predict each bill's major topic class. Elsewhere, we have demonstrated that the probability of correctly predicting the subtopic of a bill, given a correct prediction of its major topic, exceeds 0.90 (Hillard, Purpura, & Wilkerson, 2007; Purpura & Hillard, 2006). We leverage this valuable information about likely subtopic class in the second stage by developing unique subtopic document representations (using the three algorithms) for each major topic.[9]

[Figure 1 here]

Performance Assessment

We assess the performance of our automated methods against trained human annotators. Although we report raw agreement between human and machine for simplicity, we also discount this agreement for *confusion*, or the probability of that that the human and machine might agree by chance. Chance agreement is of little concern when the number of topics is large. However, in other contexts, chance agreement may be a more relevant concern.

Cohen's Kappa statistic is a standard metric used to assess inter-annotator reliability between two sets of results while controlling for chance agreement (Cohen, 1968). Usually, this technique assesses agreement between two human annotators, but the computational linguistics field also uses it to assess agreement between human and machine annotators. The Cohen's Kappa statistic is defined as:

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

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

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

where *N* is the number of examples, and *I()* is an indicator function that is equal to 1 when the two annotations (human and computer) agree on a particular example. *P(E)* is the probability of the agreement expected by chance:

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

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

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

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

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

where C is the number of categories, and πc is the approximate chance that a bill is classified as category c.

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

In this study we report just AC1 because there is no meaningful difference between Kappa and AC1.[10] For annotation 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 annotators (Carletta, 1996).

Corpus: The Congressional Bills Project

The Congressional Bills Project (www.congressionalbills.org) archives information about federal public and private bills introduced since 1947. Currently the database includes approximately 379,000 bills. Researchers use this database to study legislative trends over time as well as to explore finer questions such as the substance of environmental bills introduced in 1968, or the characteristics of the sponsors of environmental legislation.

Human annotators have labeled each bill's title (1973-98) or short description (1947-72) as primarily about one of 226 subtopics originally developed for the Policy Agendas Project (www.policyagendas.org). These subtopics are further aggregated into 20

major topics (Table 2). For example, the major topic of environment includes 12 subtopics corresponding to longstanding environmental issues, including species and forest protection, recycling, and drinking water safety, among others. Additional details can be found online at http://www.policyagendas.org/codebooks/topicindex.html.

[Table 2 here]

The students (graduate and undergraduate) who do the annotation train for approximately three months as part of a year-long commitment. Typically, each student annotates 200 bills per week during the training process. To maintain quality, inter-annotator agreement statistics are regularly calculated. Annotators do not begin annotation in earnest until inter-annotator reliability (including a master annotator) approach 90% at the major topic level and 80% at the subtopic level.[11] Most bills are annotated by just one person, so the dataset undoubtedly includes annotation errors.

However, it is important to recognize that inter-annotator disagreements are usually legitimate differences of opinion about what a bill is primarily about. For example, one annotator might place a bill to prohibit the use of live rabbits in dog racing in the sports and gambling regulation category (1526), while another might legitimately conclude that it is primarily about species and forest protection (709). The fact that inter-annotator reliability is generally high, despite the large number of topic categories, suggests that the annotators typically agree on where a bill should be assigned. In a review of a small sample, we found that the distribution between legitimate disagreements and actual annotation errors was about 50/50.

Experiments and Findings

The main purpose of automated text classification is to replicate the performance of human labelers. In this case, the classification task consists of either 20 or 226 topic 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 that major class, as advocated by Koller and Sahami (1997).

We begin by performing a simple random split on the entire corpus: We split the corpus into halves and use the first subset for training and the second for testing. Thus, one set of about 190,000 labeled samples is used to predict labels on about 190,000 separate cases.

Table 3 shows the results produced when using our text pre-processing methods and four off-the-shelf computer algorithms. With 20 major topics and 226 subtopics, a random assignment of bills to topics and subtopics can be expected to yield very low levels of accuracy. It is therefore very encouraging to find high levels of prediction accuracy across the different algorithms. This is indicative of a feature representation—the mapping of text to numbers for analysis by the machine—which reasonably matches the application.

[Table 3 here]

The ensemble learning voting algorithm combining the best of the four (SVM, MaxEnt, and Boostexter) marginally improves inter-annotator agreement (compared to SVM alone) by 0.3% (508 bills). However, combining information from three algorithms yields important additional information that can be exploited to lower the costs of improving accuracy. When the three algorithms predict the same major topic for a bill, the prediction of the machine matches the human assigned category 94% of the time (Table 4).

When the three algorithms disagree by predicting different major topics, collectively the machine predictions match the human annotation team only 61% of the time. The AC1 measure closely tracks the simple accuracy measure, so for brevity we present only accuracy results in the remaining experiments.

[Table 4 here]

Predicting to the Future: When and Where Should Humans Intervene?

A central goal of the Congressional Bills Project (as well as many other projects) is to turn to automated systems to lower the costs of labeling new bills (or other events), as opposed to labeling events of the distant past. The previous experiments shed limited light on the value of the method for this task. How different are our results if we train on annotated bills from previous Congresses to predict the topics of bills of future Congresses?

From past research we know that topic drift across years can be a significant problem. Although we want to minimize the amount of time that the annotation team devotes to examining bills, we also need a system that approaches 90% accuracy. To address these concerns, we adopt two key design modifications. First, we implement a partial memory learning system. When the system uses data from the past to predict the future, it forgets everything it learned prior to the most recent congressional session. For example, to predict class labels for the bills of the 100th Congress (1987—1988), we only use information from the 99th Congress, plus whatever data the human annotation team has generated for the 100th as part of an active learning process. We find that this approach yields results equal to, or better than, what can be achieved using all available previous training data.

The second key design decision is that we only want to accept machine-generated class labels for bills when the system has high confidence in the prediction. In other cases, we wish to have humans annotate the bills, because we've found that this catches cases of topic drift and it minimizes mistakes. One implication of Table 4 is that the annotation team may be able to trust the algorithms' prediction when all three methods agree and limit its attention to the cases of disagreement where they disagree. But we need to confirm that the results are comparable when we use a partial memory learning system.

For the purposes of these experiments, we will focus on predicting the topics of bills from the 100th Congress to the 105th Congress using only the bill topics from the previous Congress as training data. This is the best approximation of the "real world" case that we are able to construct, because (1) these congressional sessions have the lowest computer/human agreement of all of the sessions in the data set; (2) the 105th Congress is the last human annotated session; and (3) the first production experiment with live data will use the 105th Congress' data to predict the class labels for the bills of the 106th Congress. The results reported in Table 5 are at the major topic only. As mentioned, the probability of correctly predicting the subtopic of a bill, given a correct prediction of major topic class, exceeds 0.90 (Purpura & Hillard, 2006; Hillard, Purpura, & Wilkerson, 2007).

Several results in Table 5 stand out. Overall, we find that when we train on a previous Congress to predict the class labels of the next Congress, the system correctly predicts the major topic about 78.5% of the time without any sort of human intervention. This is approximately 12% below what we would like to see, but we haven't spent any money on human annotation yet.

How might we strategically invest human annotation efforts to best improve the performance of the system? To investigate this question we will begin by using the major topic class labels of bills in the 99th Congress to predict the major topic class labels of the

bills in the 100th Congress. Table 6 reports the percentage of cases that agree between the machine and the human team in three situations: when the three algorithms agree, when two of them agree, and when none of them agree. When all three agree, only 10.3% of their predictions differ from those assigned by the human annotators. But when only two agree, 39.8% of the predictions are wrong by this standard, and most (58.5%) are wrong when the three algorithms disagree.

Of particular note is how this ensemble approach can guide future efforts to improve overall accuracy. Suppose that only a small amount of resources were available to pay human annotators to review the automated system's prediction for the purpose of improving overall accuracy. (Remember that in an applied situation, we would not know which assignments where correct and which were wrong). With an expected overall accuracy rate of about 78%, 78% of the annotator's efforts would be wasted on inspecting correctly labeled cases. But if the annotator were to instead focus only on the cases of algorithm disagreement, the percentage of wasted effort declines to 60%. And if resources are extremely limited, the annotator might inspect only the cases where all three algorithms disagree. In this situation, 41% of her time would be wasted on inspecting correctly classified bills.

A review of just 655 bills where the three methods disagree (i.e., less than 8% of the sample) can be expected to reduce overall annotation errors by 20%. In contrast, inspecting the same number of cases when the three methods agree would reduce overall annotation errors by just 3.5%. If there are resources to classify twice as many bills (just 1,310 bills, or about 15% of the cases), overall error can be reduced by 32%, bumping overall accuracy from 78% to 85%. Coding 20% of all bills according to this strategy increases overall accuracy to 87%.

[Tables 5 and 6 here]

In the political science literature, the most appropriate alternative approach for validating the methods presented here is the one recently advocated by Hopkins and King (2007). While their method, discussed earlier, does not predict to the case level and is therefore inadequate for the goals we've established in this work, it can be compared against a subset of our objectives. We can compare estimates of proportions by applying our software and the *ReadMe* software made available by Hopkins and King (http://gking.harvard.edu/readme/) to the same dataset. We trained the ReadMe algorithm and the best performing algorithm of our ensemble (SVM) on the human-assigned topics of bills of the 104th Congress (1995-96), and then predicted the proportion of bills falling into each of 20 major topics of the 105th Congress.

In Figure 2, an estimate that lies along the diagonal is perfectly predicting the actual proportion of bills falling into that topic category. The further the estimate strays from the diagonal, the less accurate the prediction. Thus, Figure 2 indicates that the SVM algorithm—which labels cases in addition to predicting proportions—is performing as well and sometimes much better than the *ReadMe* algorithm. These findings buttress our belief that estimation bias is just one of the considerations that should affect methods selections. In this case, the greater document level classification accuracy of the SVM discriminative approach translated into greater accuracy of the estimated proportions. In practice, a mixed method researcher using our approach gains the ability to inspect individual documents in a class (because they know the classification of each document) while still having confidence that the estimates of the proportions of the documents assigned to each class are reasonable. The technique for bias reduction proposed by

Hopkins and King could then be used as a post processing step—potentially further improving proportion predictions.

[Figure 2 here]

Conclusions

Topic classification is a central component of many social science data collection projects. Scholars rely on such systems to isolate relevant events, to study patterns and trends, and to validate statistically derived conclusions. Advances in information technology are creating new research databases that require more efficient methods for topic classifying large numbers of documents. We investigated one of these databases to find that a supervised learning system can accurately estimate a document's class as well as document proportions, while achieving the high inter-annotator reliability levels associated with human efforts. Moreover, compared to human annotation alone, this system lowers costs by 80% or more.

We have also found that combining information from multiple algorithms (ensemble learning) increases accuracy beyond that of any single algorithm, and also provides key signals of confidence regarding the assigned topic for each document. We then showed how this simple confidence estimate could be employed to achieve additional classification accuracy.

Although the ensemble learning method alone offers a viable strategy for improving classification accuracy, we anticipate that additional gains can be achieved through other active learning interventions. In Hillard, Purpura, and Wilkerson (2007) we show how confusion tables that report classification errors by topic can be used to target follow-up interventions more efficiently. One of the conclusions of this research was that

it stratified sampling approaches can be more efficient than random sampling, especially where smaller training samples are concerned. In addition, much of the computational linguistics literature focuses on feature representations and demonstrates that experimentation in this area is also likely to lead to improvements. The Congressional Bills Project is in the public domain (www.congressionalbills.org). We hope that this work inspires others to improve upon it.

We appreciate the attraction of less costly approaches such as keyword searches, clustering methodologies, and reliance on existing indexing systems designed for other purposes. Supervised learning systems require high quality training samples and active human intervention to mitigate concerns such as topic drift as they are applied to new domains (e.g., new time periods), but it is also important to appreciate where other methods fall short as far as the goals of social science research are concerned. For accurately, reliably, and efficiently classifying large numbers of complex individual events, supervised learning systems are currently the best option (Yang & Liu, 1999; Breiman, 2001; Hand, 2006).

References

Adler, E. S., & Wilkerson, J. (in press) Intended consequences? Committee reform and jurisdictional change in the House of Representatives. *Legislative Studies Quarterly*.

Baum, M. (2003). *Soft news goes to war: Public opinion and American foreign policy in the new media age*. Princeton NJ: Princeton University Press.

Baumgartner, F., Jones, B. D., & Wilkerson, J. (2002). Studying policy dynamics. In F. Baumgarter & B. D. Jones (Eds.), *Policy dynamics (pp. 37-56).* Chicago: University of Chicago Press.

Brill, E. & Wu, J. (1998). Classifier combination for improved lexical disambiguation. In *Proceedings of the COLING-ACL '98, 191-95*. Philadelphia, PA: Association for Computational Linguistics.

Carletta, J. (1996). Assessing agreement on classification tasks: The Kappa statistic. *Computational Linguistics, 22(2), 249-54.*

Cohen, J. (1968). Weighted Kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin, 70(4), 213–220.*

Curran, J. (2002). Ensemble methods for automatic thesaurus extraction. *Proceedings of the conference on empirical methods in natural language processing, 222-29.* Morristown, N.J.: Association for Computational Linguistics.

Dietterich, T. (2000). Ensemble methods in machine learning. *Lecture Notes in Computer Science, 1857, 1–15.*

Gwet, K. (2002). Kappa statistic is not satisfactory for assessing the extent of agreement between raters. *Statistical Methods for Inter-rater Reliability Assessment, 1(April), 1-6.*

Hand, D., Mannila, H., & Smyth, P. (2001). *Principles of data mining (adaptive computation and machine learning)*. Cambridge, MA: MIT Press.

Hand, D. (2006). Classifier technology and the illusion of progress. *Statistical Science, 21(1), 1-14.*

Hillard, D., Purpura, S., & Wilkerson, J. (2007, April). An active learning framework for classifying political text. Presented at the annual meetings of the Midwest Political Science Association, Chicago.

Hopkins, D., & King, G. (2007). *Extracting systematic social science meaning from text.* Unpublished manuscript (Sept. 15, 2007), Center for Basic Research, Harvard University.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the European conference on machine learning.* Chemnitz, Germany: Springer.

Jones, B. D., & Baumgartner, F. B. (2005). *The politics of attention: How government prioritizes problems*. Chicago: University of Chicago Press.

King, G., Keohane, R. O., & Verba, S. (1994). *Designing social inquiry: Scientific inference in qualitative research*. Princeton, NJ: Princeton University Press.

King, G., & 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), 617–642.*

Koller, D., & Sahami, M. (1997). Hierarchically classifying documents using very few words. In *Proceedings of the Fourteenth international Conference on Machine Learning,* 170-78. San Francisco, CA: D. H. Fisher, & E. Morgan, Publishers.

Laver, M., Benoit, K., & Garry, J. (2003). Estimating the policy positions of political actors using words as data. *American Political Science Review, 97(2), 311-31.*

Mann, G., Mimno, D., & McCallum, A. (2006). Bibliometric impact measures and leveraging topic analysis. *Proceedings of the ACM/IEEE-CS joint conference on digital libraries, 65-74*. New York: Association for Computer Machinery.

Manning, C., & Shütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge, MA: MIT Press.

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

Mitchell, T. (1997). *Machine learning*. New York: McGraw-Hill.

Papineni, K. (2001). Why inverse document frequency? In *Proceedings of the North American chapter of the Association for Computational Linguistics, 1-8.* Morristown, NJ: Association for Computing Machinery.

Poole, K., & Rosenthal, H. (1997). *Congress: A political-economic history of roll call voting.* New York: Oxford University Press.

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

Purpura, S., & Hillard, D. (2006). Automated classification of congressional legislation. In *Proceedings of the international conference on digital government research, 219-225*. New York: Association for Computer Machines.

Quinn, K., Monroe, B., Colaresi, M., Crespin, M., & Radev, D. (2006). An automated method of topic-coding legislative speech over time with application to the 105th-

108th U.S. Senate. Presented at the Annual Meetings of the Society for Political Methodology, Seattle, WA.

Rohde, D. W. (2005). *Roll call voting data for the United States House of Representatives, 1953-2004.* Compiled by the Political Institutions and Public Choice Program, Michigan State University, East Lansing, MI, 2004. http://crespin.myweb.uga.edu/pipcdata.htm

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

Schrodt P., Davis, S., & Weddle, J. (1994). Political Science: KEDS—A program for the machine coding of event data. *Social Science Computer Review, 12(4), 561-587.*

Schrodt, P., & Gerner, D. (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.*

Segal, J., & H. Spaeth. (2002). *The Supreme Court and the attitudinal model revisited*. Cambridge, England: Cambridge University Press.

Yang, Y., & Liu, X. (1999). A re-examination of text categorization methods. *Proceedings of the ACM-SIGIR conference on research and development in information retrieval*. San Francisco, USA.

Author Notes

Dustin Hillard is a Ph.D. candidate in the Department of Electrical Engineering, University of Washington.

Steven Purpura is a Ph.D. student in Information Science, Cornell University.

John Wilkerson is Associate Professor of Political Science at the University of Washington.

This project was made possible with support of NSF grants SES-0429452, SES-00880066, SES-0111443 and SES-00880061). An earlier version of the paper was presented at the Coding Across the Disciplines Workshop (NSF grant SES-0620673). The views expressed are those of the authors and not the National Science Foundation. We thank Micah Altman, Frank Baumgartner, Matthew Baum, Jamie Callan, Claire Cardie, Kevin Esterling, Eduard Hovy, Aleks Jakulin, Thorsten Joachims, Bryan Jones, David King, David Lazer, Lillian Lee, Michael Neblo, James Purpura, Julianna Rigg, Jesse Shapiro, and Richard Zeckhauser for their helpful comments.

Correspondence concerning this article should be addressed to John Wilkerson, Box 353530, University of Washington, Seattle WA 98195.

Table 1

Criteria for Topic Classification and the Appropriateness of Different Computer Assisted Content Analysis Methods

	Method				
Criteria	Unsupervised Learning (without human intervention)	Keds/Tabari	Wordscores	Hopkins and King 2007	Supervised Learning
System for Topic Classification?	Partial	Yes	No	Partial	Yes
Discriminates the primary subject of a document?	Yes	No	No	No	Yes
Document level accuracy is assessed?	No	No	Yes	No	Yes
Document level reliability is assessed?	No	No	Yes	No	Yes
Indicate secondary topics?	Yes	No	No	No	Yes
Efficient to implement?	Yes, integrating document level accuracy and reliability checks makes the process similar to supervised learning	Yes, but costs rise with scope of task	Yes, costs decline with scope of task	Yes, costs decline with scope of task	Yes, costs decline with scope of task

Table 2
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

Table 3
Bill Title Inter-annotator Agreement for Five Model Types

	SVM	MaxEnt	Boostexter	Naïve Bayes	Ensemble
Major topic N=20	88.7% (.881)	86.5% (.859)	85.6% (.849)	81.4% (.805)	89.0% (.884)
Subtopic N=226	81.0% (.800)	78.3% (.771)	73.6% (.722)	71.9% (.705)	81.0% (.800)

Note: Results are based on using approximately 187,000 human labeled cases to train the classifier to predict approximately 187,000 other cases (that were also labeled by humans but not used for training). Agreement is computed by comparing the machine's prediction to the human assigned labels. (AC1 measure presented in parentheses).

Table 4

Prediction Success for 20 Topic Categories when Machine Learning Ensemble Agrees and Disagrees

	Methods Agree	Methods Disagree
correct	94%	61%
incorrect	6%	39%
cases	85%	15%
(N of Bills)	(158,762)	(28,997)

Note: Based on using 50% of the sample to train the systems to predict to the other 50%.

Table 5.
Prediction Success when the Ensemble Agrees and Disagrees

Congress				Correctly Predicts Major Topic (%)			
Train	Test	Bills in Test Set (N)	Ensemble Methods Agree (%)	When 3 Methods Agree	When Methods Disagree	Combined Agree and Disagree	Best Individual Classifier
99th	100th	8508	61.5	89.7	59.3	78.0	78.3
100th	101st	9248	62.1	93.0	61.5	81.1	80.8
101st	102nd	9602	62.4	90.3	61.1	79.3	79.3
102nd	103rd	7879	64.8	90.1	60.2	79.6	79.5
103rd	104th	6543	62.4	89.0	57.5	77.1	76.6
104th	105th	7529	60.0	87.4	58.9	76.0	75.6
	Mean	8218	62.2	89.9	59.7	78.5	78.4

Note: The “best individual classifier” is usually the SVM system.

Table 6.
Prediction Success when Ensemble Agrees and Disagrees

	3 Methods Agree	2 Methods Agree	No Agreement	Overall
Correct	89.7%	64.2%	41.5%	78.0%
Incorrect	10.3%	36.8%	58.5%	22.0%
Share of incorrect cases	28.8%	50.8%	20.2%	-----
All cases	61.5%	30.8%	7.7%	100.0%
(N of Bills)	(5233)	(2617)	(655)	(8508)

Note: Training on bills of the 99th Congress to predict bills of the 100th Congress.

Figure 1. Topic Classifying Congressional Bill Titles

Figure 2. Predicting Document Proportions via Two Methods

Figure 1

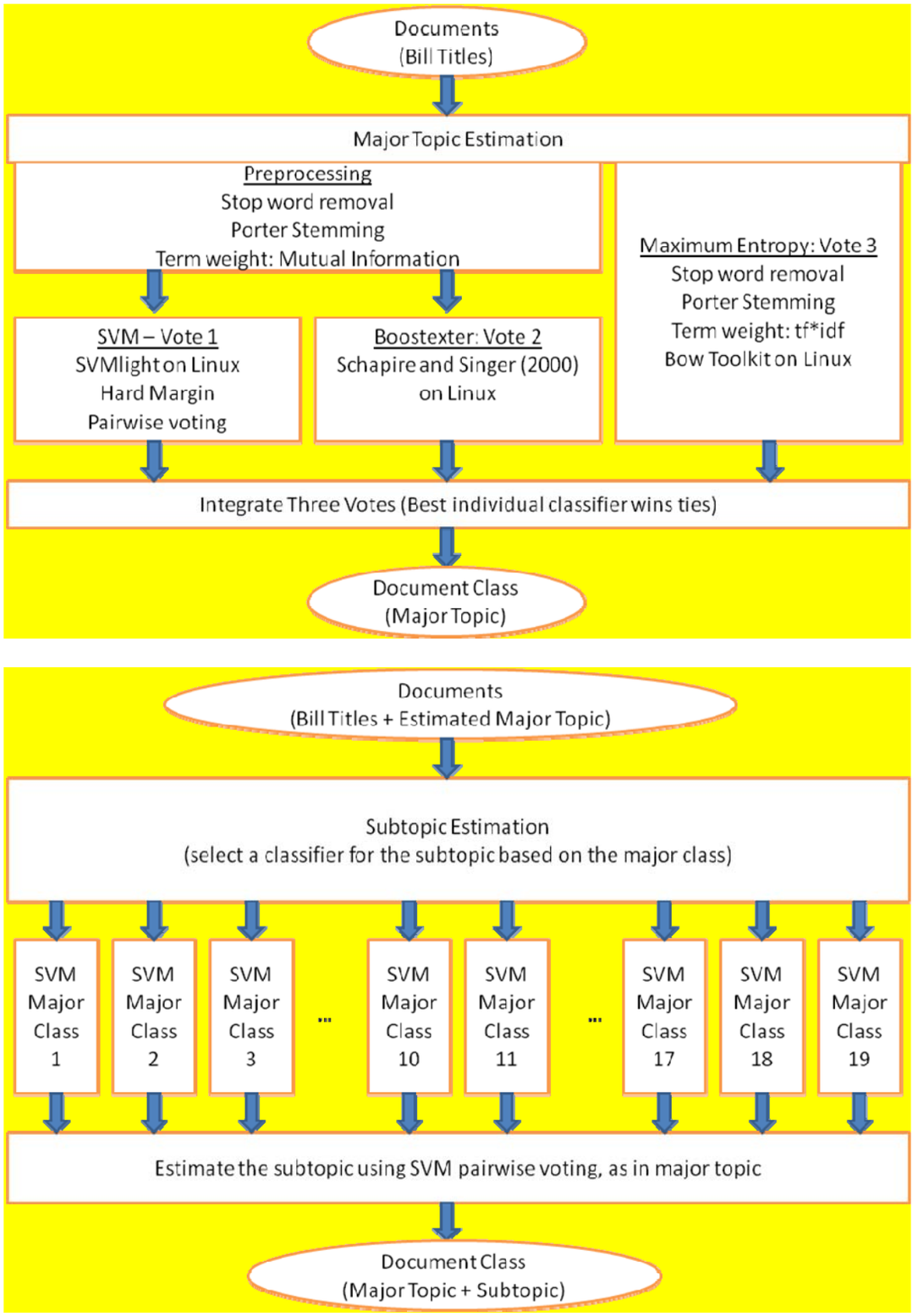
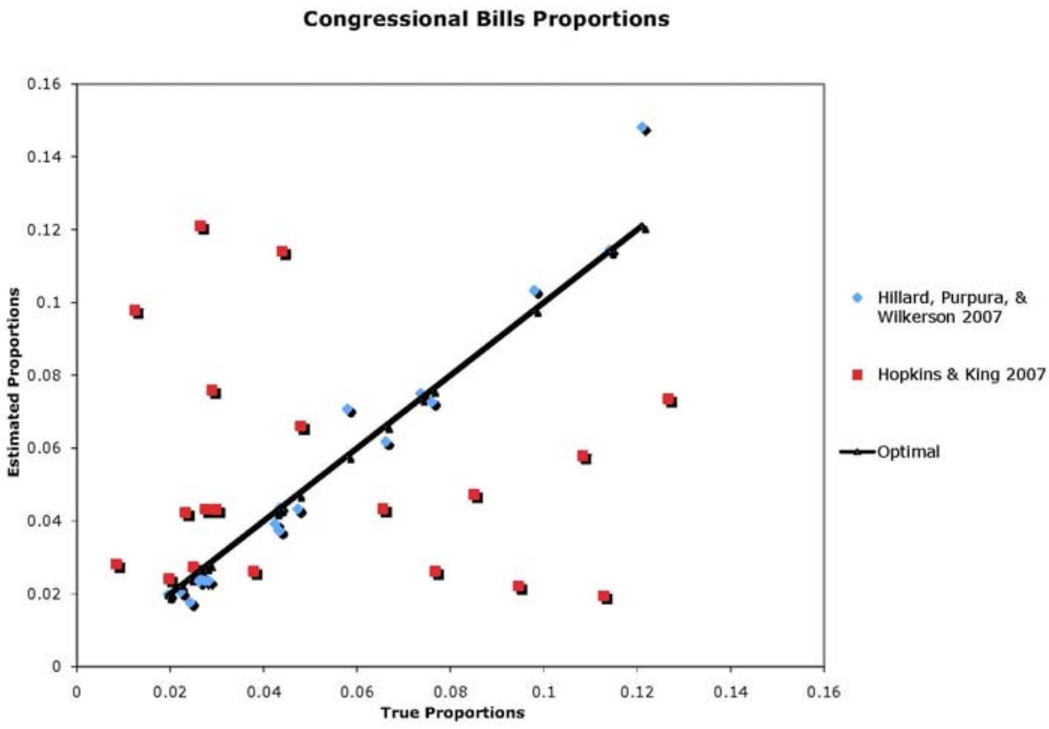

Documents
(Bill Titles)
Major Topic Estimation
Preprocessing
Stop word removal
Porter Stemming
Term weight: Mutual Information
Maximum Entropy: Vote 3
Stop word removal
Porter Stemming
Term weight: tf*idf
Bow Toolkit on Linux
SVM – Vote 1
SVMlight on Linux
Hard Margin
Pairwise voting
Boostexter: Vote 2
Schapire and Singer (2000)
on Linux
Integrate Three Votes (Best individual classifier wins ties)
Document Class
(Major Topic)
Documents
(Bill Titles + Estimated Major Topic)
Subtopic Estimation
(select a classifier for the subtopic based on the major class)
SVM Major Class 1
SVM Major Class 2
SVM Major Class 3
...
SVM Major Class 10
SVM Major Class 11
...
SVM Major Class 17
SVM Major Class 18
SVM Major Class 19
Estimate the subtopic using SVM pairwise voting, as in major topic
Document Class
(Major Topic + Subtopic)

Figure 2


Congressional Bills Proportions
Estimated Proportions
True Proportions
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
Hillard, Purpura, & Wilkerson 2007
Hopkins & King 2007
Optimal

Notes

[1] http://www.congressionalbills.org/Bills_Reliability.pdf

[2] We use *case-based* to mean that *cases* (examples marked with a class) are used to train the system. This is conceptually similar to the way that reference cases are used to train law school students.

[3]Google specifically rejects use of *recall* as a design criterion in their design documents, available at http://www-db.stanford.edu/~backrub/google.html

[4] Researchers at the Congressional Research Service are very aware of this limitation of their system, which now includes more than 5,000 subject terms. However, we have been reminded on more than one occasion that THOMAS's primary customer (and the entity that pays the bills) is the U.S. Congress.

[5] TABARI does more than classify but at its heart it is an event classification system just as at Google (at its heart) is an information retrieval system.

[6] If we were starting from scratch, we would employ this method. Instead, we use it to check whether near-exact duplicates are labeled identically by both humans and software. Unfortunately, humans make the mistake of mislabeling near-exact duplicates more times than we care to dwell upon and we are glad that we now have computerized methods to check them.

[7] Many modern popular algorithmic NLP text classification approaches convert a document into a mathematical representation using the *bag of words* method. This method reduces the contextual information available to the machine. Different corpus domains and applications require more contextual information to increase effectiveness. Variation in the document pre-processing (including morphological transformation) is one of the key methods for increasing effectiveness. See Manning and Shütze (1999) for a helpful introduction to this subject.

[8] Although the use of features similar to tf*idf (term frequency multiplied by inverse document frequency) dates back to the 1970's, we cite Papineni's literature review of the area.

[9] Figure 1 depicts the final voting system used to predict the major and subtopics of each Congressional Bill. The SVM system, as the best performing classifier, is used alone for the subtopic prediction system. However, when results are reported for individual classifier types (SVM, Boostexter, MaxEnt, and Naïve Bayes), the same classifier system is used to predict both major and subtopics.

[10] Cohen's Kappa, AC1, Krippendorf's Alpha, and simple percentage comparisons of accuracy are all reasonable approximations for the performance of our system because the number of data points and the number of categories are large.

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