Automated Classification of Congressional Legislation

Stephen Purpura
John F. Kennedy School of Government
Harvard University
+1-617-314-2027
stephen_purpura@ksg07.harvard.edu

Dustin Hillard
Electrical Engineering
University of Washington
+1-206-789-1029
hillard@ee.washington.edu

ABSTRACT

For social science researchers, content analysis and classification of United States Congressional legislative activities has been time consuming and costly. The Library of Congress THOMAS system provides detailed information about bills and laws, but its classification system, the Legislative Indexing Vocabulary (LIV), is geared toward information retrieval instead of the pattern or historical trend recognition that social scientists value. The same event (a bill) may be coded with many subjects at the same time, with little indication of its primary emphasis. In addition, because the LIV system has not been applied to other activities, it cannot be used to compare (for example) legislative issue attention to executive, media, or public issue attention.

This paper presents the Congressional Bills Project's (www.congressionalbills.org) automated classification system. This system applies a topic spotting classification algorithm to the task of coding legislative activities into one of 226 subtopic areas. The algorithm uses a traditional bag-of-words document representation, an extensive set of human coded examples, and an exhaustive topic coding system developed for use by the Congressional Bills Project and the Policy Agendas Project (www.policyagendas.org). Experimental results demonstrate that the automated system is about as effective as human assessors, but with significant time and cost savings. The paper concludes by discussing challenges to moving the system into operational use.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Clustering, Information Filtering, Retrieval Models

General Terms

Algorithms, Performance, Experimentation

Keywords

U.S. Congress, legislative activities, text analysis, SVMs, support vector machines, institutions.

The 7th Annual International Conference on Digital Government Research '06, May 21–24, 2006, San Diego, CA, USA.

1. INTRODUCTION

The Congressional Bills Project received NSF funding in 2000 (SES 0080061) to assemble a dataset[1] of all federal public bills introduced since 1947. The project's data set contains 390,000 records that include details about each bill's substance, progress and sponsors. Each bill is also assigned a single topic code drawn from the 226 subtopics of the Policy Agendas Project[2]. The resulting database is of high quality and used by researchers, instructors, students and citizens to study relative policy attention across time and venues. Researchers on other project teams are also classifying other government, media and public activities according to the same system, expanding the scope of comparison. A subset of published research, including articles and books, that consume the data may be found at the Policy Agendas web site[3].

At this time, a common classification scheme from the Policy Agendas Project makes possible comparisons of all Congressional bill activity with all Congressional hearings activity, Presidential State of the Union addresses, New York Times stories (sample), Solicitor General Briefs, and Gallup's Most Important Problem poll indices, among others for the period 1947-present. To date, these classification projects have depended on the efforts of trained human coders. However, the time and cost involved in expanding to new datasets and continually updating existing systems are substantial. A high quality, automated approach, especially one that allows lessons learned in one venue to be applied to another, would greatly speed the availability of the data to researchers.

Unfortunately, published attempts detailing the development of automated sorting and classification tools for projects of this scale and complexity are few. Recent research from Benoit, Laver, and Garry [7] has examined automated classification of issue appeals in party platforms using a word scoring technique. In addition, Shulman and others [6][12] have examined regulatory comment email duplicate detection using Kullback-Leibler (KL) distance and clustering techniques. Although Shulman's work is closer to our approach, we will instead propose a general purpose method borrowed form research in newswire topic spotting in computational linguistics.

---

[1] See www.congressionalbills.org

[2] See www.policyagendas.org and the codebook at:
http://www.policyagendas.org/codebooks/topicindex.html

[3] See http://www.policyagendas.org/publications/index.html

On first appearance, legislative bills have similar document characteristics to newswire data. Topic spotting in legislative bills has similar goals to topic spotting in newswire data because both involve scanning a text segment for the predominance of a theme. Numerous techniques for topic classification have been well documented. In this work, support vector machines (SVMs) are chosen due to their strong performance on a wide variety of tasks.

SVMs are a natural fit for topic classification because they deal well with sparse data and large dimensionality. But legislative text has different language patterns and characteristics from the typical news stories or broadcasts usually classified in newswire topic spotting. Unlike news stories or broadcasts, legislative text uses a standard template and the language may be very similar for specific types of bills. We propose the commonalities will overwhelm the difficulties and make the task of topic spotting in legislation quite successful.

The remainder of this paper documents our approach to building a prototype of a SVM system to classify the legislative text of the U.S. Congress using the Policy Agendas coding scheme and human coded samples. The approach was tested on roughly 108,000 of the 390,000 records in the Congressional Bills Project databases, as this was the largest sample available at the time of analysis. The approach to classifier design is developed in Section 2. The evaluation methodology is presented in Section 3. Experimental results are detailed in Section 4, and the main conclusions of this work are summarized in Section 5.

2. ALGORITHM OVERVIEW

Our goal is a software system that assists the Congressional Bills Project in classifying bills from the U.S. Congress according to the Policy Agendas coding scheme. Based on training examples (known as 'the truth') from expert coders, the system should scan each bill and determine which of 226 subtopic codes best fits each bill. The section below describes an algorithm that accomplishes the objective.

2.1 Support Vector Machines

SVMs were introduced in [14] and the technique attempts to find the best possible surface to separate positive and negative training samples. The best possible surface produces the greatest possible margin among the boundary points.

SVMs were developed for topic classification in [4]. Joachims motivates the use of SVMs using the characteristics of the topic classification problem: a high dimensional input space (the words), few irrelevant features, sparse document representation, and the knowledge that most text categorization problems are linearly separable. All of these factors are conducive to using SVMs because SVMs can train well under these conditions. That work performs feature selection with an information gain criterion and weights word features with a type of inverse document frequency. Various polynomial and RBF kernels are investigated, but most perform at a comparable level to (and sometimes worse than) the simple linear kernel. A software package for training and evaluating SVMs is available and described by [5]. That package is used for these experiments.

2.2 Word Feature Processing

Text input to topic classification systems is usually preprocessed and then word features are given weights depending on importance measures. Most text classification work begins with word stemming to remove variable word endings and reduce words to a canonical form so that different word forms are all mapped to the same token (which is assumed to have essentially equal meaning for all forms). Word features usually consist of stemmed word counts, adjusted by some weighting. Inverse document frequency is commonly used, and has some justification in [8]. More complex measures of word importance have shown to provide additional gains though. A weighted inverse document frequency is an extension of inverse document frequency to incorporate term frequency over texts, rather than just term presence [11]. Term selection can also help improve results and many past approaches have found information gain to be a good criterion ([13] and [10]).

During word feature processing, we remove non-word tokens, map text to lower case, and then apply the Porter Stemming Algorithm described in [9][4]. The text is then distilled into features. Features such as inverse document frequency have been generally effective but more detailed forms of word weighting have shown improvements. This work adopts a weighting related to mutual information. Each word is given a feature value w_i as shown in equation 1.

$$w_i = log(\frac{p(w,t)}{p(w)p(t)}) = log(\frac{p(w|t)p(t)}{p(w)p(t)}) \qquad (1)$$

In this equation, the top term, $p(w|t)$, is the probability of a word in a particular bill (the number of occurrences in this bill, divided by the number of total words in the bill). The denominator term $p(w)$ is the probability of a word across all bills (the number of occurrences of this word in all bills, divided by the total number of words in all bills). This also reduces to an intuitive form as in equation 2 where it can be thought of as a ratio of word frequency given a bill, divided by the overall frequency in all available bills.

$$w_i = log(\frac{p(w|t)}{p(w)}) \qquad (2)$$

Finally, only words with $w_i > 0$ are placed in the term by conversation matrix (this is all terms with a ratio greater than 1, or in other words those that occur more frequently than the corpus average).

2.3 Hierarchical Approach

Our approach is unique because our problem demands innovation on the typical use of SVMs. We have chosen a two-phase hierarchical approach to SVM training which mimics the method employed by human coders. Human coders first classify a bill as falling under one of 20 major topic codes (see Table 1) and then further classify it as falling under one of 226 subtopics. For example, a bill proposing to reform the health care insurance system is assigned to fall under subtopic 301, where the 3 indicates health, and the 01 indicates health insurance reform.

[4] Note that this step reduces performance in international environments. See discussions of stemming.

Table 1: Major Topic Codes

1 = Macroeconomics
2 = Civil Rights, Minority Issues, and 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, and Domestic Commerce
16 = Defense
17 = Space, Science, Technology, and Communications
18 = Foreign Trade
19 = International Affairs and Foreign Aid
20 = Government Operations
21 = Public Lands and Water Management
99 = Other

The advantages of the two phase approach were many, but two reasons stand out. First, training SVMs on 226 subtopic codes across large numbers of bills is computationally expensive. Using this hierarchical approach greatly reduces the computational expense of the sorting. The hierarchical approach can be implemented on a common laptop computer with a complete sorting of the full data set in much less than a day of processing. Second, human coders are more likely to disagree on subtopic coding than they are on major topic coding. Thus, correctly predicting the major topic of a bill has more value to the coding team than completely missing the mark.

The hierarchical approach's two-phase system begins with a first pass which trains a set of SVMs to assign one of 20 major topics to each bill. The second pass iterates once for each major topic code and trains SVMs to assign subtopics within a major class. For example, we take all bills that were first assigned the major topic of health (3) and then train a collection of SVMs on the health subtopics (300-398). Since there are 20 subtopics of the health major topic, this results in an additional 20 sets of SVMs being trained for the health subtopics.

Once the SVMs have been trained, the final step is subtopic selection. In this step, we assess the predictions from the hierarchical evaluation to make our best guess prediction for a bill. For each bill, we apply the subtopic SVM classifiers from each of the top 3 predicted major topic areas (in order to obtain a list of many alternatives). This gives us subtopic classification for each of the top 3 most likely major categories. The system can then output an ordered list of the most likely categories for the research team.

3. EVALUATION METHODOLOGY

Evaluation of success is straightforward because high quality information which describes "the ground truth" is available. This section describes the data sets used in our experiments and our methodology for assessing performance against human labelers.

3.1 Data Sets

This research was conducted using the Congressional Bills Project's public data set[5]. At the time (April 2004), 'only' 108,000 records were available for analysis. All statistics are generated from the 108,000 record set.

For the purposes of testing, the 108,000 records were divided into two groups and processed using the "train on 50%, test on 50%" methodology. We report results for the entire set using cross validation, which means we run the system twice (the second run swaps the train and test examples), allowing us to test on all available bills. To select the groups, random sampling without replacement was applied across all of the bills. The experiment was repeated many times, and the statistics were comparable. We report the last run.

3.2 Evaluation Metrics

We use metrics common in topic spotting and clustering analysis work in our evaluation of performance. The usefulness of our system was measured by its ability to predict the truth for every record. For analysis convenience, we also summarize consistency with the truth by major topic and subtopic classifications. Finally, we report Cohen's Kappa and AC1 to assess inter-coder agreement with the human team, as described in [3] and [12].

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 results between two human coders, but the computational linguistic field uses the metric as a standard mechanism to assess agreement between a human and machine coder.

Cohen's Kappa statistic is defined as:

$$\kappa = \frac{p(A) - p(E)}{1 - p(E)} \qquad (3)$$

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

$$p(A) = \frac{1}{N}\sum_{n=1}^{N} I(Human_n == Computer_n) \qquad (4)$$

Where N is the number of examples, and I() is an indicator function that is equal to one when the two annotations (human

[5] Data is available from www.congressionalbillsproject.org

and computer) agree on a particular example. P(E) is the probability of the agreement expected by chance:

$$p(E) = \frac{1}{N^2} \sum_{c=1}^{C} (HumanTotal_c \times ComputerTotal_c) \quad (5)$$

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.

For reasons of bias documented by [3], 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 problems 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.

4. EXPERIMENTAL RESULTS

The Congressional Bills Project assessed the system by its ability to reliably predict the major topic and subtopic about as well as a human. These results are reported in Tables 3 through 6, and they express that the system is about as accurate as a trained human coder at identifying the major topic of a bill, and sometimes as accurate at identifying the subtopic of a bill, with some exceptions.

The results in Table 2 illustrate that the system automatically determines the correct major category for over 80% of the bills. The single worst category is Category 99, which makes sense because this is an 'Other' category only used for bills that could not reasonably be assigned to any other category. Performance on other categories varies, but is mostly above 80% correct. The single best category was Category 18, 'Foreign Trade' at almost 90%. Excluding the 'Other' category, the most difficult category was Category 19, 'International Affairs and Foreign Aid' at only 68% correct.

Table 2: Major Category Precision; Number of Bills Predicted Correctly by Major Category, including totals.

Category	Correct	Possible	Percent
Macroeconomics (1)	4148	5481	75.68
Civil Rights … (2)	1682	2397	70.17
Health (3)	7246	8200	88.37
Agriculture (4)	3137	3703	84.72
Labor … (5)	5232	7323	71.45
Education (6)	3131	3613	86.66
Environment (7)	4108	4871	84.34
Energy (8)	4128	4660	88.58
Transportation (10)	4518	5378	84.01
Law, Crime … (12)	5417	6491	83.45
Social Welfare (13)	5249	6080	86.33
Community … (14)	1851	2447	75.64
Banking … (15)	5261	6876	76.51
Defense (16)	6255	7440	84.07
Space, Science (17)	1500	1845	81.30
Foreign Trade (18)	4127	4647	88.81
International (19)	1613	2372	68.00
Government Op (20)	13416	15607	85.96
Public Lands … (21)	6830	7894	86.52
Other (99)	145	943	15.38
Total	88994	108268	82.20

Table 3: Subcategory Precision; Number of Bills Predicted Correctly for Subtopic Categories (totals only).

Subtopic	Correct	Possible	Percent
Total	76800	108143	71.02

Table 3 presents the overall statistics for categorization at the subtopic category level. The number of possible bills is slightly lower (only by 0.1%) because our hierarchical approach only hypothesizes minor categories within the top three major categories for each bill. This provides for significant computational savings, while missing only a negligible number of bills. The overall percentage of correct bills is 71% and is lower than for the major categories, but this task is significantly more complex with over 200 possible categories instead of 20 for the major category case.

Tables 4 and 5 present the 15 best and worst individual minor category results. The single best category is 1807 'Tariff and Import Restrictions, Import Regulation.'

Table 4: Subcategory Precision; Number of Bills Predicted Correctly for Subtopic Categories (best 15 subtopic categories).

Category	Correct	Possible	Percent
Tariff and Export Restrictions (1807)	2754	2974	92.60
Federal Holidays (2030)	322	351	91.74
Relief Claims Against the U.S. Government (2015)	3071	3378	90.91
Airports, Airlines, Air Traffic Control, and Safety (1003)	1022	1155	88.48
Food Stamps, Food Assistance, and Nutrition Monitoring Programs (1301)	520	591	87.99
Regulation of Political Campaigns, Political Advertising, PAC Regulation, Voter Registration, Government Ethics (2012)	1257	1447	86.87
Worker Safety and Protection, Occupational and Safety Health Administration (OSHA) (501)	470	542	86.72
Government Subsidies to Farmers and Ranchers, Agricultural Disaster Insurance (402)	1379	1594	86.51
Highway Construction, Maintenance and Safety (1002)	623	721	86.41
Tobacco Abuse, Treatment, and Education (341)	258	299	86.29
Broadcast Industry Regulation (TV, Cable, and Radio) (1707)	538	624	86.22
Natural Gas and Oil (Including offshore Oil and Gas) (803)	1532	1783	85.92
Recycling (707)	176	205	85.85
Postal Service Issues (including Mail Fraud) (2003)	806	942	85.56
Native American Affairs (2102)	854	1009	84.64
Higher Education (601)	1397	1653	84.51

Many of the minor categories that had a large number of examples had better performance in the end, probably because the SVM was better able to learn the category characteristics when more examples were available. The 15 worst categories are primarily those categories with very few examples, and often were again those categories that were 'Other' categories within a major topic (those ending in 99).

Table 5: Subcategory Precision; Number of Bills Predicted Correctly for Subtopic Categories

(worst 15 subtopic categories)

Category	Correct	Possible	Percent
Unemployment Rate (103)	0	17	0.00
Social Welfare, Other (1399)	0	39	0.00
Banking, Finance, and Domestic Commerce, Other (1598)	0	6	0.00
Foreign Trade,Other (1899)	0	14	0.00
Anti-Government Activities (209)	0	17	0.00
Public Lands and Water Management, Other (2199)	0	6	0.00
Drugs and Alcohol or Substance Abuse Treatment (344)	0	42	0.00
Education Research and Development (698)	0	15	0.00
International Affairs and Foreign Aid, Other (1999)	1	23	4.35
Military Nuclear and Hazardous Waste Disposal, Military Environmental Compliance (1614)	2	41	4.88
Energy, Other (899)	1	17	5.88
Other, Other (9999)	65	863	7.53
Transportation,Other (1099)	2	26	7.69
Labor, Employment, and Immigration, Other (599)	3	29	10.34
Civil Rights, Minority Issues, and Civil Liberties, Other (299)	2	19	10.53

4.1 Systems-to-Human Inter-coder Agreement

The second set of calculations assessed inter-coder reliability, as calculated using Cohen's Kappa and AC1. We use a single coder to express the performance of the entire Congressional Bills team and note that in future research we will integrate the system as a coder within the team for testing. The calculations are summarized in Table 6, and demonstrate, using either Cohen's Kappa or AC1 as metrics, the system performs about as well as humans would be expected to perform.

TABLE 6: Cohen's Kappa and AC1, humans versus system

	p(A)	p(E)	Statistic
κ for all major topics	0.822	0.069	0.809
κ for all subtopics	0.710	0.013	0.706
AC1 for all major topics	0.822	0.049	0.813
AC1 for all subtopics	0.710	0.004	0.709

5. CONCLUSION AND NEXT STEPS

Researchers are now classifying government, media and public activities according to common coding systems to expand the scope of comparison across government institutions. The Congressional Bills Project and the Policy Agendas Project are just two examples. Their experience makes clear that the shift from paper documents to electronic documents should make their job easier, but without new tools and methods, progress will be slow and expensive.

This research focused on the process of sorting United States Congressional bills using an established classification system. Extensive work by the Congressional Bills team set the benchmark for measuring an automated system. And the techniques in this paper demonstrate that support vector machines are effective for efficiently classifying Congressional bills. On some types of bills, the system has difficulty compared to an expert coder. But, in the balance, the algorithm is quite compact and robust. Considering the complexity of coding legislative text into one of 226 subtopics, its effectiveness is about as good as can be expected when using techniques based solely on the "bag of words" principal. Future research should examine using other features which could improve the system as well as other algorithms.

The described algorithm also displays another highly desirable trait for the task – it is easily extensible with additional features. The SVM system is capable of considering out-of-band data to aid in reaching a conclusion in text classification. In concrete terms, the system could be told to consider a count of THOMAS LIV classifications, sponsor committee membership, and other relevant information when predicting the subtopic of a bill. With the correct tools, extending the system to improve its accuracy would then become an exercise for any political science student interested in taking up the task.

The next step for the team is to integrate the algorithm with the human coding team of the Congressional Bills project. Use of the system in their daily work would provide them with the ability to predict the major and subtopic codes for each new Congress' set of bills. Although the system cannot be trusted to generate a 100% accurate answer, it already generates meaningful information useful to understanding when it is making a systemic, likely true prediction versus a wild guess for each bill. This information is critical to the successful adoption of systems like this, and methods to expose this information will be the subject of future research. The team is applying for National Science Foundation funding to pursue these opportunities.

6. ACKNOWLEDGMENTS

Thanks to Dr. John Wilkerson for providing assistance with the Congressional Bills' data. Also, thanks to Dr. Stuart Shulman for encouraging us to submit this document.

7. REFERENCES

[1] 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.

[2] Deerwester, S. et al. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407.

[3] 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.

[4] Joachims, T. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. *Proceedings of the European Conference on Machine Learning (ECML)*. (Springer, 1998)

[5] Joachims, T. Making Large-Scale SVM Learning Practical. in: *Advances in Kernel Methods - Support Vector Learning*, B. Schölkopf, C. Burges, and A. Smola (ed.), MIT Press, 1999.

[6] Kwon, N., Shulman, S.W., and Hovy, E.H.. (Under review). "Collective text analysis for eRulemaking." *Proceedings of the Sixth National Conference on Digital Government* Research. San Diego, CA.

[7] Laver, M., Benoit, K., and Garry, J. Extracting policy positions from political texts using words as data. In *American Political Science Review* 97(2).

[8] Papineni, K. "Why inverse document frequency?" IN *Proceedings of the North American Association for Computational Linguistics, NAACL*, pp. 25–32. (2001)

[9] Porter, M. F. An algorithm for suffix stripping. Program, 16(3):130–137.

[10] Sebastiani, F. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1).

[11] Tokunaga, T. and Iwayama, M. Text categorization based on weighted inverse document frequency. *Technical Report 94*

TR0001, Department of Computer Science, (Tokyo Institute of Technology, 1994).

[12] Yang, H., Callan, J., and Shulman, S. (Under review) "Next steps in near-duplicate detection for eRulemaking." *Proceedings of the Sixth National Conference on Digital Government Research*. San Diego, CA.

[13] Yang, Y. and Liu, X. 1999. A re-examination of text categorization methods. In *Proceedings of SIGIR-99*, November.

[14] Vapnic, V. *The Nature of Statistical Learning Theory*. Springer, New York, NY. 1995.