Using supervised machine learning to code policy issues: Can classifiers generalize across contexts?

Burscher, B.; Vliegenthart, R.; de Vreese, C.H.

DOI
10.1177/0002716215569441

Publication date
2015

Document Version
Final published version

Published in
The Annals of the American Academy of Political and Social Science

Using Supervised Machine Learning to Code Policy Issues: Can Classifiers Generalize across Contexts?

By
BJORN BURSCHER,
RENS VLIEGENTHART,
and
CLAES H. DE VREESE

Content analysis of political communication usually covers large amounts of material and makes the study of dynamics in issue salience a costly enterprise. In this article, we present a supervised machine learning approach for the automatic coding of policy issues, which we apply to news articles and parliamentary questions. Comparing computer-based annotations with human annotations shows that our method approaches the performance of human coders. Furthermore, we investigate the capability of an automatic coding tool, which is based on supervised machine learning, to generalize across contexts. We conclude by highlighting implications for methodological advances and empirical theory testing.

Keywords: agenda setting; content analysis; machine learning; big data

Social scientists increasingly use supervised machine learning (SML) to automatically analyze media content (e.g., Hillard, Purpura, and Wilkerson 2008). SML is a technique in which a computer learns from a set of human-coded training documents to automatically predict variables (e.g., the topic of a news article) in texts. In this article, we apply SML to the coding of policy issues, which is central to the study of agenda setting—a major paradigm in various social sciences (e.g., Baumgartner and Jones 2010).

As agenda setting research is concerned with dynamics in issue salience among the media,

Bjorn Burscher is a PhD candidate in the Department of Communication Science at the Amsterdam School of Communication Research, University of Amsterdam. He works on the INFINITI project, which develops and enables the exploitation of open source and open standards tools to support semantic search.

Rens Vliegenthart is a full professor in communication science and chair in media and organizations in the Department of Communication Science at the Amsterdam School of Communication Research, University of Amsterdam.

DOI: 10.1177/0002716215569441

politicians, and citizens (Rogers, Dearing, and Bregman 1993), it requires large-scale over-time content analysis (CA) across different types of political texts. An automatic coding tool should be able to correctly predict policy issues in different sorts of political texts, from various sources and time periods. To investigate this, we conducted a series of validation experiments in which we employed SML to code policy issues in unknown datasets. Furthermore, we studied how a classifier's ability to predict the primary policy issue of a news article changes when using only words from its lead section in the training data. When it is necessary to code only a small fraction of each training document manually, SML becomes more cost efficient.

We found that SML is well suited to automatically code the primary policy issue of political texts. The ability of an SML model to generalize across contexts, however, is limited and depends on the characteristics of available training data. We conclude by discussing the strengths and limits of SML as compared to other approaches to automatic CA.

Computer-Aided Content Analysis

Scholars have followed different approaches to automatically code policy issues. In *dictionary-based CA*, previously defined character strings are used to code textual units into content categories (e.g., Schrodt, Davis, and Weddle 1994). This approach may compromise semantic validity, because manually compiled classification rules are at risk of being biased by the subjective conceptions and limited domain knowledge of the researcher(s). Furthermore, most people are not very good at determining how many different ways (e.g., senses, parts of speech) a word can be used when prompted with a specific category. This can lead to incomplete search strings and result in wrong predictions.

When applying *unsupervised machine learning*, issues are not defined a priori but are inductively extracted from the data by clustering documents that share the same words (e.g., Quinn et al. 2006). This is an efficient approach, because it requires very little guidance. However, for each identified cluster, a person needs to manually infer its meaning afterward. This can be a difficult task, because the found clusters might not necessarily represent the desired content categories. In the case of policy issues, some clusters might represent multiple issues, or might represent concepts other than policy issues (e.g., news coverage regarding a specific political actor or country). This poses a problem when one wants to code political texts according to a priori defined issues.

In SML, documents are automatically coded according to previously defined content categories by training a computer to replicate the coding decisions of humans (e.g., Hillard, Purpura, and Wilkerson 2008). A premise for the

Claes H. de Vreese is a professor and chair of political communication and director of the Program Group Political Communication & Journalism in The Amsterdam School of Communication Research (ASCoR) in the Department of Communication Science, University of Amsterdam.

application of SML is a set of documents that have been manually coded for the content categories of interest. This is called the training set. SML involves three steps: First, documents from the training set are converted in such a way that they are accessible for computational analysis. Each document is represented as a vector of quantifiable textual elements (e.g., word counts), which are called features. Second, feature vectors of all documents in the training set, together with the documents' content labels, are used to train a classifier to automatically code the content categories. In doing so, an SML algorithm statistically analyzes features of documents from each content category and generates a classifier to predict the content categories in future documents. Third, the classifier is used to code text documents outside the training set.

In SML, in contrast to dictionary-based CA, a computer automatically estimates a model that classifies texts according to content categories. This is likely to be more effective, because the rules used to identify the primary policy issue of a document are based on statistical analysis of human-coded training data. Compared to unsupervised machine learning, SML can apply a previously defined coding scheme. Being able to work with the same coding scheme in different studies facilitates the comparison as well as integration of findings across research contexts (John 2006).

Research Questions

In this study, we applied SML to the coding of policy issues in political texts. The aim of the study was twofold. First, we investigated the generalizability of policy issue classifiers across research contexts. To do so, we conducted a series of validation experiments, in which we applied classifiers to unknown datasets. As Grimmer and Stewart (2013) argue, the "performance of any one classifier can vary substantially across context, so validation of a classifier's accuracy is essential to establish the reliability of supervised learning methods" (p. 268). Information on the generalizability of classifiers helps scholars to decide on the suitability of an SML method. This is particularly relevant in comparative and longitudinal research, where documents from several outlets and time periods must be coded. In this article, we studied the generalizability of classifiers across two sorts of political texts (news articles and parliamentary questions [PQs]), across three different newspapers, and across a time frame of 15 years.

Second, we investigated how a classifier's ability to predict the primary policy issue of a news article changes when using only words from its lead section as features in the training set. Being able to reach similar classification accuracy with a training set in which only a small fraction of each article must be coded manually would significantly decrease the costs of applying SML to CA.

The chosen fraction must comply with two requirements. For an SML classifier, the fraction must be indicative of the primary policy issue. For human coders, the fraction must contain sufficient information to determine the primary policy issue when reading it. We chose to use the first 10 percent of words from

each article, because in news articles facts are generally presented in descending order of importance (Poettker 2003). Hence, this fraction of an article should inform human readers about the main policy issue discussed, and it should include words that are highly indicative of that policy issue.

Third, we studied the relationship between the amount of training data used to build a classifier and its performance to predict the primary policy issue. As manually coded training data are expensive and labor-intensive to obtain, it is important to know how much training data one must possess to build a well-performing issue classifier.

Data

To investigate our research questions, we used data that consist of front-page news articles of the three most-read Dutch newspapers (*Volkskrant*, *NRC Handelsblad*, and *Telegraaf*) and Dutch PQs for the period between 1995 and 2011. All news articles were collected digitally via the Dutch Lexis-Nexis database. PQs were downloaded from the official website of the Dutch government.[1] In the Netherlands, PQs are questions that members of parliament can direct to the government. Each question must be delivered in written form to the president of the House of Representatives, and must be orally answered by the addressed representative of the government during a weekly public session. For each year, a stratified sample of news articles (13 percent) and written PQs (N = 500) were manually coded for the main policy issue discussed. For each article/PQ, coders could choose one out of twenty different policy issues. The coding scheme that we used was developed by the Comparative Agendas Project (Baumgartner, Green-Pedersen, and Jones 2006). See Table 1 for an overview of all issue categories. The unit of coding was the distinct news article/PQ. Some PQs contained subquestions, which were grouped together. The resulting datasets consisted of 11,089 manually coded news articles and 4,759 manually coded PQs.

Manual coding was conducted by thirty trained coders. All coders were native Dutch speakers. To assess intercoder reliability, a random subset of articles (N = 198) and PQs (N = 200) was each coded by two coders. Krippendorff's alpha for issue category codings was equal to .69 for news articles and .60 for PQs. The coding was done as part of a large-scale research project about the influence of media coverage on parliamentarians.

Validation Experiments

First, we tested whether our classifiers could replicate the hand coding of documents from the original datasets of news articles (N = 11,089) and PQs (N = 4,759). In doing so, we used a stratified random sampling procedure to split each dataset into a training set (80 percent), on which we trained the classifier, and a test set (20 percent), on which we evaluated the classifier.

TABLE 1
F1 Scores for SML-Based Issue Coding in News Articles and PQs

Issue	News Articles			PQs	
		All Words	Lead Only		All Words
Features	*N*	F1	F1	*N*	F1
Macroeconomics	413	.54	.63	172	.46
Civil rights and minority issues	327	.34	.28	192	.53
Health	444	.70	.71	520	.81
Agriculture	114	.72	.76	159	.66
Labor and employment	217	.43	.49	174	.58
Education	188	.79	.71	229	.78
Environment	152	.34	.44	237	.59
Energy	81	.35	.59	67	.66
Immigration and integration	150	.50	.57	239	.78
Transportation	416	.58	.67	306	.81
Law and crime	1198	.70	.69	685	.77
Social welfare	115	.33	.34	214	.54
Community development and housing	113	.45	.44	136	.72
Banking, finance, and commerce	622	.62	.67	188	.58
Defense	393	.59	.55	196	.71
Science, technology, and communication	426	.64	.59	57	.53
International affairs and foreign aid	1,106	.70	.64	352	..65
Government operations	1,301	.71	.72	276	.48
Other issue	3,322	.84	.80	360	.51
Total	11,089	.71	.68	4,759	.69

NOTE: The F1 score is equal to the harmonic mean of recall and precision. Recall is the fraction of relevant documents that are retrieved, and precision is the fraction of retrieved documents that are relevant.

Second, to test a classifier's ability to correctly predict policy issues in another sort of political texts, we trained a classifier on a stratified random sample of four thousand news articles and tested the classifier on all PQs. Similarly, we trained a classifier on a stratified random sample of four thousand PQs and tested it on all news articles. Third, we tested whether a classifier could correctly predict the main policy issue in documents from unknown sources. We split the news dataset into two subsets, one included a stratified random sample of four thousand articles from two of the three newspapers, and the other included all articles from the third newspaper. Then, we used the former as the training set and the latter as the test set. We repeated this exercise for all possible combinations of newspapers. Finally, we tested whether a classifier could correctly predict the main policy issue in documents from unknown time frames. We split the news dataset in two subsets: a training set, which contained a stratified random sample of four

thousand articles from 1995 to 2003, and a test set that contained all articles from 2004 to 2011. We also did this in the reverse.

SML Implementation

We compared two different SML implementations: one in which we used all words from each document in the training set as features, and one in which we used only the first 10 percent of words from each document in the training set as features. We compared the performance of both implementations when classifying news articles. When classifying PQs, we always used all words of the document.

For both news articles and PQs, we applied the following processing steps. First, we tokenized all documents and applied stemming to each token using the Frog natural language processing modules (Van den Bosch et al. 2007). Then, contingent on the implementation, we used either all tokens of the document, or selected the first 10 percent of its tokens. From this selection of tokens, we removed punctuation, single-letter words, and common Dutch stop words. Then, we extracted all unique unigrams and bigrams from the remaining tokens and applied TF.IDF weighting (Russell and Norvig 2002)[2] to them. Therefore, each unigram or bigram was assigned the number of times it occurs in a document (TF), weighted by the inversed frequency of documents in the entire collection containing the unigram/bigram (IDF). The idea behind TF.IDF weighting is to evaluate the power of a word to discriminate between documents. In each classification task, we employed the Passive Aggressive learning algorithm,[3] which is known to perform well in various text classification tasks (Crammer et al. 2006).[4]

Our main evaluation measure is the F1 score, which is equal to the harmonic mean of recall and precision. Recall is the fraction of relevant documents that are retrieved, and precision is the fraction of retrieved documents that are relevant. The F1 score is a standard evaluation measure for SML applications and provides a good indication of classification performance.

To assess the relationship between the size of the training set and classification performance, we plotted learning curves for the classification of news articles and PQs. We used a stratified cross-validation generator to split the whole dataset five times into training (80 percent) and test data (20 percent). Subsets of the training set with varying sizes were used to train the classifier, and F1 scores for each training subset size and the test set were computed. Afterward, the scores were averaged over all runs for each training subset size. In all steps of the analysis, we used the scikit-learn machine learning library for the Python programming language (Pedregosa et al. 2011).

Results

In Table 1, we report F1 measures of coding performance per policy issue for news articles and PQs. In these analyses we split each of the datasets into a

TABLE 2
F1 Scores for Validation Experiments

	Baseline (N = 4,000)		Other Text Sort		Other Newspaper (News Dataset)			Other Time Frame (News Dataset)	
	News → News	PQs → PQs	News → PQs	PQs → News	VK/TEL → NRC	NRC/TEL → VK	VK/NRC → TEL	1995–2003 → 2004–2011	2004–2011 → 1995–2003
F1	.67	.68	.50	.49	.59	.63	.65	.59	.63

NOTE: VK = *Volkskrant*, NRC = *NRC/Handelsblad*, TEL = *Telegraaf*.

training set (80 percent) and a test set (20 percent), and then used the former for learning and the latter for evaluation. When using all words of each document in the training set as features, average coding performance was equal to F1 = .71 for news articles and F1 = .69 for PQs. When using only the first 10 percent of words from each document in the training set as features, classification performance was equal to F1 = .68 for news articles. This is only marginally lower as compared to using all words of each article as the training data.

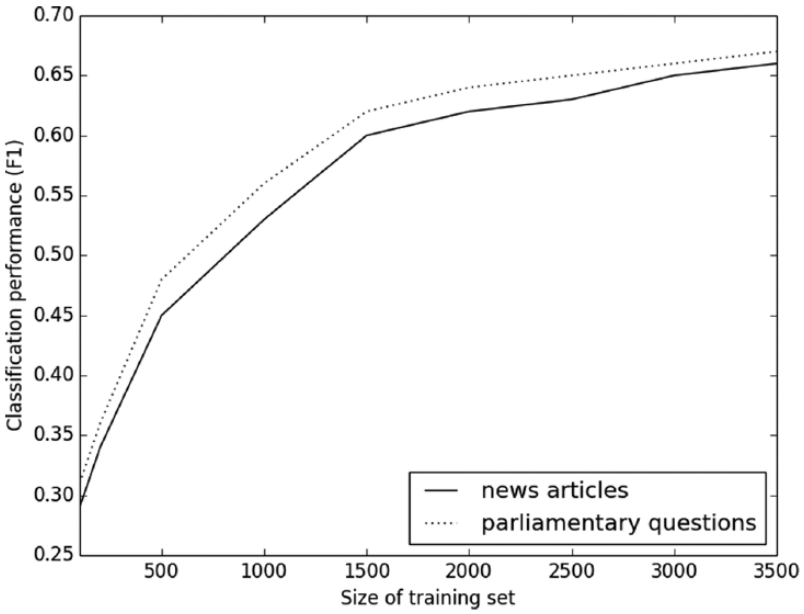
When looking at individual issue categories, we see that classification performance is higher for those issues that are more prevalent in the data. The correlation between F1 scores and the number of positive examples among the policy issues is equal to r = .40 for news articles and r = .50 for PQs.

Next, we turn to the validation experiments. To make results of all validation experiments comparable to one another, we set the training size in each validation experiment to four thousand documents. To make them comparable to the general analyses presented above, we reported general classifier performance when using only four thousand training documents as a baseline measure in Table 2. Results of all validation experiments are based on the implementation in which we used all words of each document in the training set as features.[5]

First, we present results of experiments, in which we used newspaper articles as training data and PQs as test data (and then PQs for training and news articles for testing). Table 2 reports F1 measures of such tests. Measures show that classification accuracy significantly decreases when applying a classifier to a different sort of political text on which it is not trained. When predicting PQs with a classifier that is trained on news articles, F1 is equal to .50. When predicting news articles with a classifier that is trained on PQs, F1 is equal to .49.

Second, we turn to results of experiments in which we predicted the policy issues of news articles from unknown papers and time periods. F1 measures for predicting news articles from another newspaper range from .59 to .65, which is clearly lower compared to measures for predicting papers that were included in the training data. Also, when predicting news articles from another time period, classification accuracy decreases. When training on the first half of the available time frame (1995–2003) and testing on the second half (2001–2011), F1 is equal

FIGURE 1
Learning Curves for the Classification of News Articles and PQs



to .59. When training on the second half of the available time frame and testing on the first half, F1 is equal to .63.

Finally, we turn to the relationship between the amount of training data and classification performance. The results are plotted in Figure 1. For news articles and PQs, classification performance increases as the amount of training data increases. This relationship, however, is not linear. After reaching a training size of around two thousand documents, coding performance increases only slowly when adding additional training documents. Moreover, the learning curve for PQs has a higher slope than the one for news articles. This indicates that PQs are easier to classify than news articles.

Discussion

Here we focused on two aspects of SML-based content analysis: the validation of SML classifiers across research contexts and the costs of training an SML classifier. To test the former, we applied policy issue classifiers to several unknown datasets. We found that classification accuracy decreases slightly when applying a classifier to an unknown newspaper, and strongly when applying it to articles from unknown time periods and content domains. From this, we conclude that training data must be representative of all outlets, time periods, and document types that one wants to study. When this is infeasible, a dictionary-based approach

might be preferred over an SML approach. An SML-based classification model is very specific to the word use within the training set. In a dictionary-based approach, in contrast, the classification model is more general. Therefore, it most likely performs more consistently across different contexts. Future research should focus on ways to improve the generalizability of policy issue classifiers by selecting less context-dependent features (e.g., names of persons and places).

To investigate the costs of training a policy issue classifier, we plotted the learning curves for both news articles and PQs. Based on the curves, we conclude that one does not need several thousand training documents to train a policy issue classifier. Actually, adding more hand-coded documents to the training set increases average coding performance only slowly after reaching a threshold of around two thousand training documents. Instead, it would be more effective to selectively sample positive examples for underrepresented categories. Several strategies for this are discussed in the literature (Hillard, Purpura, and Wilkerson 2008; Tong and Koller 2000).

Furthermore, we found that whether one uses all words of a news article or only words from its leading paragraph when presenting it in the training set has little effect on classification performance. This implies that, when creating training data, it might be sufficient to code only the leading paragraphs of each article. This makes supervised topic classification more cost-efficient, and facilitates the coding of more representative samples from several sources and time periods, which most likely will increase the robustness and generalizability of a policy issue classifier. This way, SML becomes more attractive compared to other approaches to automatic CA, which require no manually coded training data.

Finally, we are aware that the quality of our training data is not optimal. Disagreement between coders likely results from a combination of unsystematic coding errors and systematically different interpretation of policy issues across coders. The most relevant question is how this might influence our findings and conclusions. We expect classification performance to decrease as a result of inconsistencies in the training data. If texts with similar features are associated with different policy issues, it becomes more difficult for the SML algorithm to estimate a model that can clearly differentiate between content categories. Although classification performance is most likely influenced by the quality of the training data, we believe our conclusion to be largely unaffected.

Notes

1. See http://www.officielebekendmakingen.nl.

2. We also tried other bag-of-words implementations such as binary word presence and word counts. Findings showed that using TF.IDF weights was the most effective approach. When applying TF.IDF weighting, we normalized all data using the L2 norm.

3. We set the C-parameter to 100. This parameter trades off misclassification of training examples against simplicity of the decision surface.

4. We tried different state-of-the-art algorithms for text classification as well as an ensemble of classifiers. However, the Passive Aggressive algorithm outperformed all tested alternatives.

5. Results are nearly identical when using only the first 10 percent of words from each document in the training set as features.

References

Baumgartner, Frank R., and Bryan D. Jones. 2010. *Agendas and instability in American politics*. Chicago, IL: University of Chicago Press.

Baumgartner, Frank R., Christoffer Green-Pedersen, and Bryan D. Jones. 2006. Comparative studies of policy agendas. *Journal of European Public Policy* 13 (7): 959–74.

Crammer, Koby, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research* 7:551–85.

Grimmer, Justin, and Brandon M. Stewart. 2013. Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political Analysis* 21 (3): 267–97.

Hillard, Dustin, Stephen Purpura, and John Wilkerson. 2008. Computer-assisted topic classification for mixed-methods social science research. *Journal of Information Technology & Politics* 4 (4): 31–46.

John, Peter. 2006. The policy agendas project: A review. *Journal of European Public Policy* 13 (7): 975–86.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, and Mathieu Blondel. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–30.

Poettker, Horst. 2003. News and its communicative quality: The inverted pyramid—When and why did it appear? *Journalism Studies* 4 (4): 501–11.

Quinn, Kevin M., Burt L. Monroe, Michael Colaresi, Michael H. Crespin, and Dragomir R. Radev. 2006. An automated method of topic-coding legislative speech over time with application to the 105th–108th U.S. Senate. Paper presented at annual meeting of the Midwest Political Science Association, 20–23 April, Austin, TX.

Rogers, Everett M., James W. Dearing, and Dorine Bregman. 1993. The anatomy of agenda-setting research. *Journal of Communication* 43 (2): 68–84.

Russell, Stuart, and Peter Norvig. 2002. *Artificial intelligence: A modern approach*. Upper Saddle River, NJ: Prentice Hall.

Schrodt, Philip A., Shannon G. Davis, and Judith L. Weddle. 1994. Political science: KEDS—A program for the machine coding of event data. *Social Science Computer Review* 12 (4): 561–87.

Tong, Simon, and Daphne Koller. 2000. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research* 2:45–66.

Van den Bosch, Antal, Bertjan Busser, Sander Canisius, and Walter Daelemans. 2007. An efficient memory-based morphosyntactic tagger and parser for Dutch. In *Computational linguistics in the Netherlands: Selected papers from the Seventeenth CLIN Meeting*, 99–114. Leuven: CLIN.