Erschienen in: Research and Politics ; 2 (2015), 4
https://dx.doi.org/10.1177/2053168015615596

Research Article

Research and Politics
October-December 2015: 1–8

DOI: 10.1177/2053168015615596
rap.sagepub.com
SAGE

Improving the selection of news reports for event coding using ensemble classification

Mihai Croicu[1] and Nils B Weidmann[2]

Abstract

Manual coding of political events from news reports is extremely expensive and time-consuming, whereas completely automatic coding has limitations when it comes to the precision and granularity of the data collected. In this paper, we introduce an alternative strategy by establishing a semi-automatic pipeline, where an automatic classification system eliminates irrelevant source material before further coding is done by humans. Our pipeline relies on a high-performance supervised heterogeneous ensemble classifier working on extremely unbalanced training classes. Deployed to the *Mass Mobilization on Autocracies* database on protest, the system is able to reduce the number of source articles to be human-coded by more than half, while keeping over 90% of the relevant material.

Keywords

Protest data, text classification, machine learning

Introduction

Large-scale extraction of event data from unstructured news reports produced by global news agencies has been a topic in political science for almost three decades. Indeed, many leading data collections in political science follow this strategy. We can distinguish two different approaches: one involving manual reading and coding of events by trained human coders, used for example by the Uppsala Conflict Data Program's Geo-referenced Event Dataset (Sundberg and Melander, 2013) or the Armed Conflict Location and Events Dataset (Raleigh et al., 2010). The other approach, automated coding, uses various algorithms to extract events computationally from the source text, as for the Kansas Event Data System (Schrodt and Gerner, 1994) or the recent ICEWS event dataset (Boschee et al., 2013) for example.

Human coding, while producing high-quality content and being extremely flexible in terms of the type of information that can be extracted, is very costly and time consuming (Schrodt and Van~Brackle, 2013). Automated coding, while quite successful in categorizing news articles into various topics of interest or extracting actors from known actor lists (dictionaries), has been demonstrated to have shortcomings when it comes to extracting actual content from the text (e.g. the number of fatalities, the date of incident rather than the date of reporting, or the location), the actors involved, or the relations between them (Boschee et al., 2013; Hammond and Weidmann, 2014). In fact, there is relative agreement that the signal-to-noise ratio of automatically coded datasets make their usage at the most granular level (the individual incident) difficult, and that various aggregation techniques are required to eliminate unwanted noise (Schrodt and Van~Brackle, 2013).

In this paper, we present a coding pipeline that uses the best of both worlds by combining the two approaches described above—in other words, a semi-automatic procedure. In this pipeline, the source material is first screened for relevant news articles through machine learning techniques.

[1]Department of Peace and Conflict Research, Uppsala University, Sweden
[2]Department of Politics and Public Administration, University of Konstanz, Germany

Corresponding author:
Nils B Weidmann, Department of Politics and Public Administration, University of Konstanz, Universitätsstr. 10, 78457 Konstanz, Germany.
Email: nils.weidmann@uni-konstanz.de

The remaining, much smaller set of potentially relevant articles are assigned to human coders for further processing. This paper introduces the first part of this pipeline, the automatic pre-selection of source material. In a typical event coding project, the large amount of articles irrelevant for coding is a huge issue, and the prime reason why human coders are so slow in comparison to computers. Therefore, improving the pre-selection of relevant articles is a key issue we need to resolve. We proceed by describing our use case—a protest event data project focusing on autocracies. We then implement a pipeline for the pre-selection of articles using a versatile bagging ensemble classifier. In an out-of-sample validation, we show that this pipeline is able to achieve high predictive performance when applied to a number of different countries.

Use case and technical considerations

When coding political events from news reports, coders typically encounter large amounts of articles unrelated to the dataset under construction. The main objective of the first part of our semi-automatic coding process is therefore to eliminate as many of these "irrelevant" news articles, while retaining as many as possible of all "relevant" articles such that the latter can be given to a trained human for further coding. We demonstrate this approach in the context of an ongoing coding project, the *Mass Mobilization on Autocracies* database, a new event dataset on mass mobilization under autocratic regimes (Rød and Weidmann, 2013). However, the design is project-agnostic, such that it can be used in other data projects.

Sources. The source material for the coding are news articles furnished by global news bureaus (e.g. Reuters, Agence France Presse, Associated Press, or BBC Monitoring) through aggregators such as LexisNexis or Factiva. This type of source material is probably the most common basis for large-scale event datasets used in political science and international relations (Brandt et al., 2011).

We obtain the source material through a simple keyword search with different synonyms for political protest. No pre-filtering of articles is done at the time of retrieval. Given the fact that the keywords used are extremely common, a vast majority of the retrieved articles do not contain relevant information (i.e., they cover topics such as sports, finance, education, etc.). Thus, the number of irrelevant articles vastly exceeds those covering events of interest to the project. Alternative pre-filtering through proprietary tools such as the categories provided by LexisNexis was ruled out for lack of transparency and replicability. The extracted articles consist of a headline, a dateline, a body, and a unique ID.

Training set. During the first one year phase of the project, an initial set of roughly 250,000 articles was hand-coded according to the coding procedure described in Rød and Weidmann (2013).[1] This set of articles constitutes the training set of our machine learning-based procedure. While the coders extracted all the information relevant to the project such as the number of protesters, the issue, and the actors involved, we only use the information whether an article was actually considered relevant—in other words, whether at least one protest event was coded from it by the coder. This way, a large training set with human-annotated binary class labels (relevant/irrelevant) was generated. Descriptives for the training set are presented graphically in Figure 1.

A machine learning task. Our goal is to create a pre-filtering pipeline keeping as many of the relevant articles as possible, while discarding most of the irrelevant ones. Therefore, the overall requirements for this pipeline are somewhat different than those for most machine learning procedures. Traditionally, machine learning algorithms attempt to achieve the best trade-off between precision and recall. For our application, as the machine filtering is just one stage of the complete coding pipeline, recall clearly has a higher priority than precision. Ideally, we want all relevant articles to be labeled as true (recall as close to 100% as possible), whereas the number of correctly labeled irrelevant articles is of lesser importance (if some irrelevant articles are wrongly labeled as relevant, they can still be eliminated by coders). Therefore, the main performance indicator we aim to optimize is the recall of truly relevant articles, and we consider values of 0.85–0.95 as acceptable.

Implementation

We implement the solution as a pipeline, with the starting point being raw, unprocessed news articles extracted directly from LexisNexis. Following standard procedures, from each article we extract a set of features that is later used for classification.

Text processing and feature extraction

We apply standard natural language pre-processing such as sentence and word tokenization, lemmatization and stop-word removal, as well as removal of punctuation. We also experimented with part-of-speech tagging methods with the goal of eliminating low-information words such as adjectives and numerals. Most of these methods performed extremely well at their task, but were not included in the final pipeline due to slow performance and extremely modest improvements to the classification results. Similarly, named entity recognition was attempted in order to make the classifying pipeline as geography- and name-agnostic as possible; however,

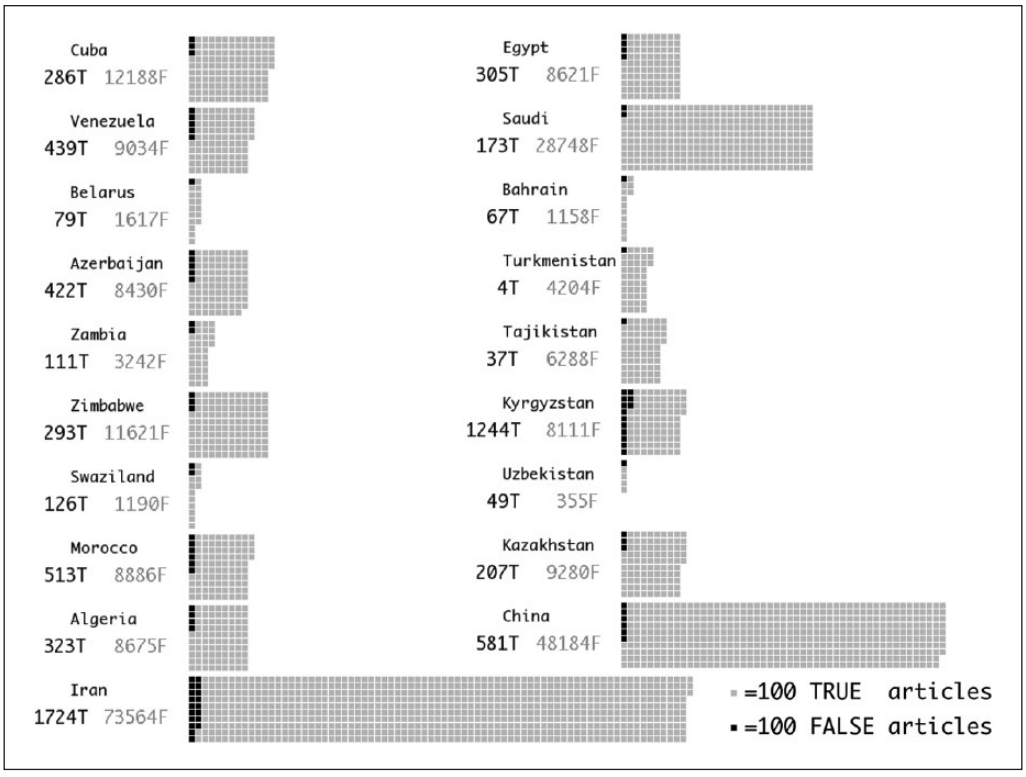

Figure 1. The composition and distribution of the dataset employed for training and testing.

again, the computational performance costs were massive and produced only small improvements in predictive performance.

Therefore, training and classification is done at the word-level. For each article in each iteration, a vector of classification features consisting of individual words in that article is extracted ("bag of words"). We extract the most used individual word stems (unigrams) as well as the most used two-word groupings/expressions (bigrams) from the training set used for each classifier, and test for their existence in each article. Some classifiers (such as Naive Bayes (NB) can only deal with a limited number of features. For that reason, we selected the most frequent 750 unigrams and 250 bigrams for each corpus of text used by each individual classifier.

This number was reached through multiple small-scale tests on blocks of 5000 articles, and offers an excellent speed (computer performance)–accuracy (model performance) compromise for classification. In small scale tests, increasing the number of unigrams in the features vector from 50 to 250 increased the recognition of relevant articles in out-of-sample tests from 86.4% to 97.2%. This increase tapered off at approximately 500 unigrams. Similarly, increasing the number of bigrams, as well as increasing the proportion of bigrams to unigrams provided increases in performance, but at the cost of increasing the number of false positives substantially (at 500 unigrams and 165 bigrams, almost 40% of all irrelevant articles were correctly labeled and thus eliminated; while at 500 unigrams and 500 bigrams, this dropped to under 20%; recognition of relevant articles was one percentage point better in the latter case). We considered this trade-off (20% more irrelevant articles kept for manual observation in exchange of 1% more relevant articles saved) unacceptable, and thus set a low number of bigram features.

The ensemble classifier

The next step in the pipeline is the classification stage. As described above, the key challenge we face is the extreme imbalance of the class distribution; only about 2% of all articles are truly relevant. Most classifiers tend to perform extremely poorly in such an environment, sometimes simply ignoring the smaller class by labeling all instances as belonging to the majority class (Tang et al., 2009; Chawla et al., 2004). Multiple solutions have been proposed, such as using weighting, various random sampling techniques (Chawla et al., 2004), and even multi-step classification with a large-class structured sampling methodology as a pre-selection phase (Tang et al., 2009).

Our approach to solving the problem of class imbalance is inspired by the random sampling methodology: employing an ensemble classifier consisting of multiple base classifiers, each of which is trained on a balanced

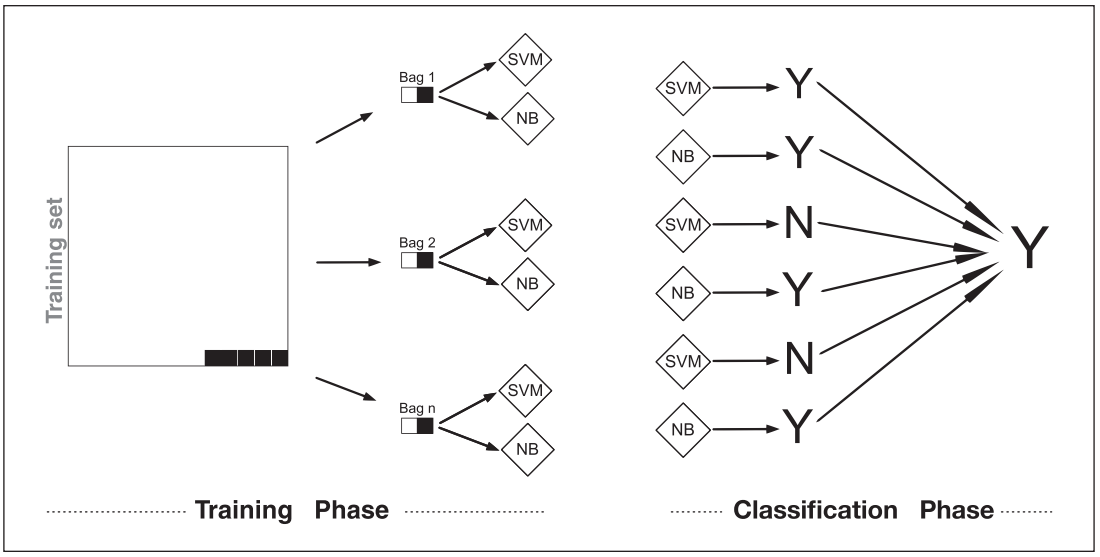

Figure 2. General architecture of the classifier ensemble. Dark squares represent the relevant articles in the training set.

subset of the training set. Ensemble classification is an approach where, rather than a single classifier, a *set* of so-called base classifiers is created during the training phase. When classifying a new instance, each of these classifiers then votes on the new instance, and the aggregated vote (for example, the majority) is then used as the ensemble's prediction. Ensemble prediction has long been used in machine learning, and is known to have a number of advantages (Aggarwal and Zhai, 2012: 209–211). The construction of ensemble classifiers can be done in a number of ways. Bagging (short for "bootstrap aggregation") is one of the simplest procedures (Breiman, 1996). Bagging involves the random creation of multiple training samples ("bags"), each of which is then used for the creation of one base classifier. These samples are created by randomly drawing (with replacement) from the set of all training instances. This, however, means that the bags exhibit the same class distribution as the entire training dataset. In our case, this would inevitably lead to the problem described above, i.e. "degenerated" base classifiers predicting the majority class only.

For that reason, we modify the standard bagging procedure to address the problem of class imbalance. As proposed by Weidmann (2008), we draw *balanced* random samples from the training instances that have an equal proportion of relevant and irrelevant articles. Since having partially overlapping data is generally considered as advantageous for classification (Dietterich, 2000), we deviate from Weidmann's approach of using the entire set of relevant articles along with randomly selected irrelevant ones. Instead, we split the relevant training articles into a number of "shards" (five shards, each 1200 articles long), and include a single shard with an equal-sized random sample from the irrelevant articles category in each bag. Therefore, each bag has a total of 2400 training articles (1200 relevant ones, 1200 irrelevant). This means that each base classifier can be trained on a balanced sample, which avoids the problem discussed above (see Figure 2, left). When given a new article for classification as relevant/irrelevant, each base classifier first generates an individual prediction. Then, the ensemble's aggregate prediction is generated through a voting procedure where an article is classified as relevant if a certain percentage of the base classifiers have classified the article as relevant (Figure 2, right).

Three questions remain before we can launch the ensemble classifier. First, we need to select the base classifiers to be used in the ensemble. We chose algorithms that are well suited for text classification purposes: support vector machines (SVM) and NB (Joachims, 1998; Manning et al., 2008). Due to their high computational costs, other commonly used algorithms used in text classification such as Decision Trees (Aggarwal and Zhai, 2012: 163–222) were not considered. The individual classifiers themselves are simple SVMs (with a radial basis function) provided by the Python *scikit-learn* module, and the NB classifier with a multinomial distribution (Manning et al., 2008: 234–251) from Python's *Natural Language Toolkit*. For the SVM, linear kernels were also considered, but prior experimental testing indicated worse results (a decrease of 1.5–5%) on the same dataset compared to RBFs. The combination of both classifiers was optimized such that every pair of SVM and NB classifiers share single a bag of data, which speeds up the process.

The second question we need to resolve is the number of bags. Theoretically, we expect that the number of bags required to maximize the quality of prediction is fairly limited. Both SVM and NB classifiers tend to be relatively

stable (Bousquet and Elisseeff, 2002; Ting and Zheng, 2003), which means that changes in the training data have a comparatively small impact on the structure of the classifier. This means that the number of required bags need not be larger than the amount of new information brought to the model. Given the relative homogeneity of the "relevant" category, we presume that the value of additional bags will be limited after a certain threshold has been reached. In limited experiments, this stability point was identified at approximately 10–15 bags. However, to be on the safe side, given some concerns with regards to data heterogeneity, and since performance did not alter beyond reason, all real-usage training was conducted with 50 bags. In total, 100 base classifiers are trained, i.e. 50 pairs of one NB and one SVM classifier. Each such pair is trained on a bag of negative data and one of the five shards (discussed above) of positive data.

The third question that remains is the voting threshold. For example, a threshold of 0.01 means that at least 1% of all classifiers must predict "relevant" for an article to be classified as "relevant." We implement and test different values of this parameter between 0.01 and 1. The evaluation below describes the performance of the ensemble at different threshold values.

Evaluation

In order to assess the applicability of our classification approach to the problem of pre-selecting news reports according to their relevance, we need to determine its predictive accuracy out-of-sample, i.e. on unseen news articles. The standard way to do this would have been a standard N-fold cross validation, where articles are split randomly into N folds, $N-1$ to use for teaching and the remaining one for evaluation (the whole procedure repeated N times). However, we expect strong regional or country patterns in our data, where relevant reports about protest may be characterized by certain region- or country-specific words or combinations. In essence, this would mean that a substantial number of the features determining the result of classification of any given article article are dependent on the country the article referred to. In a real application, however, users may want to filter articles from *previously unseen* countries.

Therefore, we perform a variation of the standard cross-validation approach by binning the articles by country ($N = 18$) to which they belong. In each loop, the classifier is trained on data from every country except one, and the excluded country constitutes the test set. Such a strategy reflects the actual usage of the classifier in practice and exposes the potential country bias discussed above. In particular, this evaluation methodology eliminates any overestimation of performance due to country-level information present in both the training and testing sets: a normal k-fold validation strategy would contain country data from any given country in both training and testing, allowing country-specific terms, names or locations to affect the labeling of articles in the test set. This is something we need to avoid.

Cross validation results are presented graphically in Figure 3. Overall, the ensemble exhibits good performance, identifying an unweighted average of 93% of recall across the cross-validation test sets at a 0.05 cutoff, while eliminating 56% of all irrelevant articles across the set.[2] In effect, the classifier is exceeding the goals set out at the beginning of the project, while conducting an evaluation similar to actual use in practice. This is equivalent to eliminating approximately 150,000 articles for an 18-country set similar to the sample used in this example, saving up to 200 work-days by trained humans. Results hold across sample sizes as well as various levels of balance (proportion of relevant articles) in the test sets. In effect, there is no observable difference between the classification of Belarus and that of Kyrgyzstan, even though in Belarus this proportion is more than three times higher (1:20 vs. 1:6.5).

Choosing a more demanding voting cutoff substantially increases the performance of the ensemble in eliminating irrelevant articles, with 80% being eliminated at a 0.75 cutoff. The cost is a converse loss of recall, dropping to only 80% of all positives. However, as recall-false increases much faster than recall-true decreases (i.e. the number of false articles eliminated grows much faster than the number of lost true articles), users should choose the cutoff level depending on the ambitions and human resources available to the project.

Further, the classifier performs very well in areas with very low numbers of positives, such as Turkmenistan or Belarus, where it identifies 100% of all positives in their respective test samples. This is essential in practice. In regions with low reporting rates we do not expect that other (potentially duplicated) articles describing the same incident would make up for a wrongly eliminated article; instead, losing a single article can mean omitting the entire protest event.

Moreover, predictive performance is more dependent on the choice of cutoff for geographic regions with less data (such as Latin America). For countries in the former Soviet Union, performance is nearly unaffected by an increase in cutoff (indicating that most classifiers actually vote the same). In Kyrgyzstan, 90% of relevant articles are identified at a cutoff of 0.75, whereas performance drops sharply as the cutoff is increased for countries such as Venezuela; here, only a little over 52% of all relevant articles are correctly identified. However, we consider such behavior acceptable from a practical point of view, as a vast majority of recall values are in the 0.9–1.0 range. Further, as no country remains to be coded in areas with the lowest

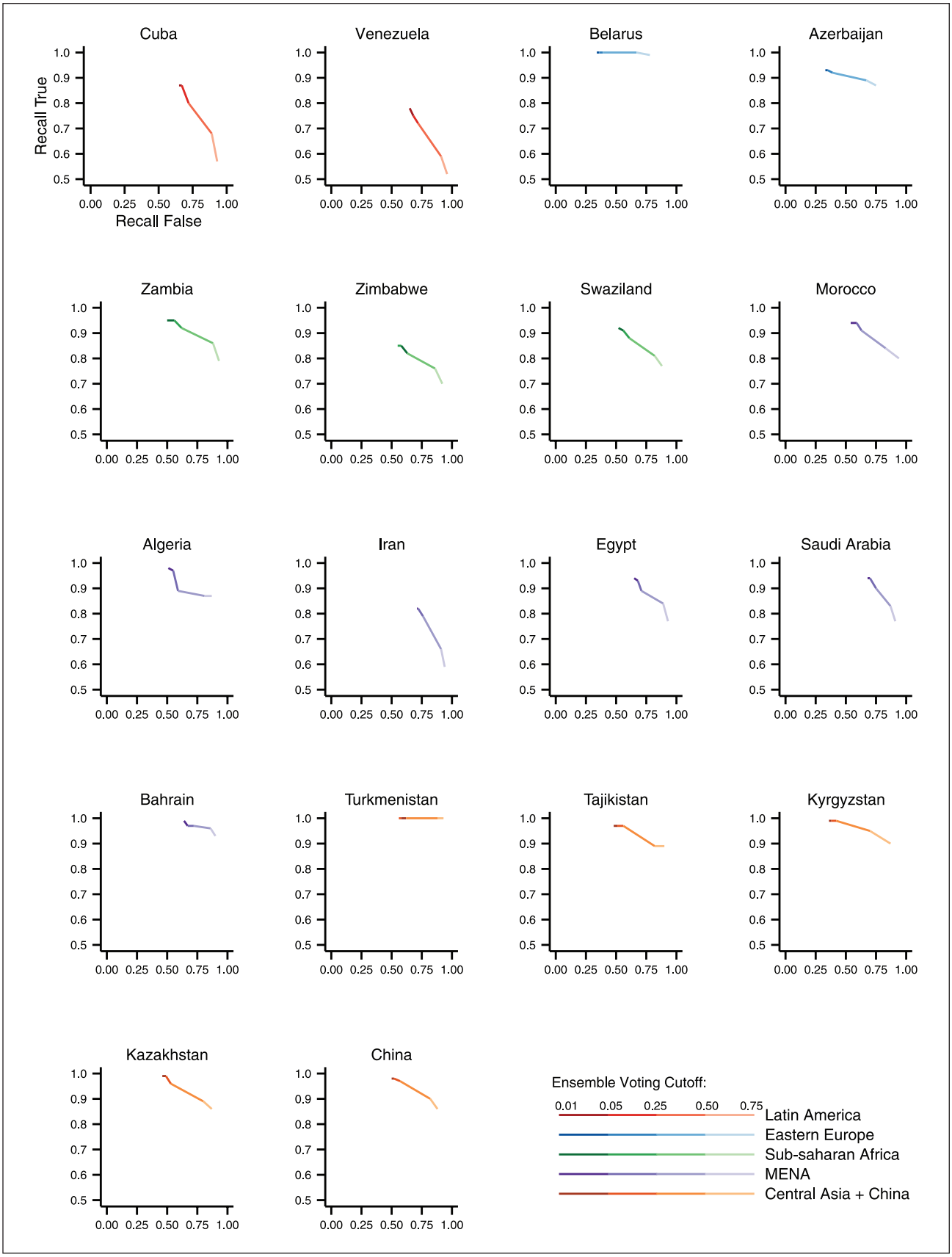

Figure 3. Cross-validation results for the proposed ensemble.

training data density (such as Latin America), the resulting bias will be further attenuated in practice. When deploying the coding pipeline in practice, we chose a low cutoff (0.05) that provides more homogeneous results.

Further, we assess the suitability of the classifier for other potential tasks it may be applied to. First, we analyze the classifier's performance when trained on an existing data set, in order to predict the relevance of new, incoming articles. We train the classifier on the first 80% of the articles (ordered by date, cutoff is 2008-06-20) and evaluate out-of-sample on the latest 20% of the data (combing all countries). The results are excellent. The classifier identifies 94% of all relevant articles while discarding 58% of all irrelevant articles at the 0.05 cutoff.[3]

Second, we test the classifier's performance when trained on a much smaller sample, since not all projects may have a training set as big as ours. We randomly select 15,000 total articles (382 positives) for training, and test on another random sample of 15,000 articles. The performance is again perfectly within an acceptable range, with the classifier identifying 97% of all positives in the sample at the standard 0.05 cutoff. However, trained with less data, the filtering performance suffers, and the classifier eliminates about one third of all irrelevant articles as opposed to over 50% when trained on the full dataset. One solution for better performance is to alter the value of the cutoff parameter, with values such as 0.25 or 0.5 being probably more appropriate for many users.[4]

Conclusion

The screening of large amounts of texts can quickly and efficiently be done by computers, whereas humans are better at extracting individual pieces of information from these texts. In order to combine the strengths of both automatic and human coding into a feasible coding pipeline, we devised a hybrid, semi-automatic approach for coding protest events from news reports. This paper describes the first stage of our pipeline. We presented a machine learning ensemble classifier for the pre-selection of news reports for event coding. In order to overcome the problem of a hugely imbalanced training set, this classifier relies on a large number of base classifiers, each built on a balanced random sample of the training data. We have shown that this approach is able to achieve good results in an out-of-sample validation, and can therefore be of great use also in other coding projects. While we have not explored the possible parameters and settings of our classifier exhaustively, we believe that our results provide sufficient reason to pursue this line of development in future research.

Acknowledgements

The authors would like to thank participants at the July 2015 workshop on "Automated Content Analysis in the Social Sciences" at the University of Zurich for comments.

Funding

Funding from the Research Council of Norway (project 204454/V10), the European Network for Conflict Research (COST Action IS1107) and the Alexander von Humboldt Foundation (Sofja Kovalevskaja Award) is gratefully acknowledged.

Notes

1. Active learning would have been an alternative approach that avoids the creation of a training set this large. In active learning, a classifier is trained on a small set of labeled entities, and then iteratively improves its performance by asking the user to label new instances (Tong and Kolle, 2002). The logistics of our coding project made such an approach infeasible, but the approach presented here could in principle be combined with such an approach.
2. Evaluating whether the selected articles are representative of the full set of relevant articles is difficult, since there are many dimensions along which we could compare the two samples. However, due to the fact that the loss is low (only 7% on average), we do not think that this is a major issue.
3. Other cutoff levels also exhibit good performance: the classifier eliminates between 56% of all irrelevant articles at the 0.01 cutoff to 90% of all irrelevant articles at the 0.75 cutoff while identifying between 75% of all positives at the 0.75 cutoff and 94% of all positives at the 0.75 cutoff.
4. At a cutoff value of 0.5, performance was within our goal, with 58% of all irrelevant articles identified and 90% of the relevant ones kept.

References

Aggarwal C and Zhai C (2012) *Mining Text Data*. New York: Springer Science & Business Media.

Boschee E, Natarajan P and Weischedel R (2013) Automatic extraction of events from open source text for predictive forecasting. In: Subrahmanian VS (ed.) *Handbook of Computational Approaches to Counterterrorism*. New York: Springer, pp.51–67.

Bousquet O and Elisseeff A (2002) Stability and generalization. *The Journal of Machine Learning Research* 2: 499–526.

Brandt PT, Freeman JR and Schrodt PA (2011) Real time, time series forecasting of inter- and intra-state political conflict. *Conflict Management and Peace Science* 28(1): 41–64.

Breiman L (1996) Bagging predictors. *Machine Learning* 24(2): 123–140.

Chawla NV, Japkowicz N and Kotcz A (2004) Editorial: Special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter* 6(1): 1–6.

Dietterich TG (2000) Ensemble methods in machine learning. In: Kittler J and Roli F (eds), *Multiple Classifier Systems*. Heidelberg: Springer, pp.1–15.

Hammond J and Weidmann NB (2014) Using machine-coded event data for the micro-level study of political violence. *Research & Politics* 1(2): 1–8.

Joachims T (1998) Text categorization with support vector machines: Learning with many relevant features. In: Nedellec C and Rouveirol C (eds), *Machine Learning: ECML-98*. Heidelberg: Springer, pp.137–142.

Manning CD, Raghavan P and Schütze H (2008) *Introduction to Information Retrieval*, vol 1. Cambridge, UK: Cambridge University Press.

Raleigh C, Linke A, Hegre H, et al. (2010) Introducing ACLED: An armed conflict location and event dataset. *Journal of peace research* 47(5): 651–660.

Rød EG and Weidmann NB (2013) Coding instructions for the mass mobilization in autocracies database. Codebook. Available at http://www.cnc.uni-konstanz.de/research/mmad/.

Schrodt PA and Gerner DJ (1994) Validity assessment of a machine-coded event data set for the Middle East, 1982–92. *American Journal of Political Science* 38(3): 825–854.

Schrodt PA and Van Brackle D (2013) Automated coding of political event data. In: Subrahmanian VS (ed.) *Handbook of Computational Approaches to Counterterrorism*. Heidelberg: Springer, pp. 23–49.

Sundberg R and Melander E (2013) Introducing the UCDP Georeferenced Event Dataset. *Journal of Peace Research* 50(4): 523–532.

Tang Y, Zhang YQ, Chawla NV, et al. (2009) SVMs modeling for highly imbalanced classification. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 39(1): 281–288.

Ting KM and Zheng Z (2003) A study of AdaBoost with Naive Bayesian classifiers: weakness and improvement. *Computational Intelligence* 19(2): 186–200.

Tong S and Koller D (2002) Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research* 2: 45–66.

Weidmann NB (2008) Conflict prediction via machine learning: Addressing the rare events problem with bagging. Poster presented at the 25th Annual Summer Conference of the Society for Political Methodology. Available at: http://nils.weidmann.ws/papers.html