Supplemental Appendix For "Multi-label Prediction for Political Text-as-Data"

Aaron Erlich[1,2], Stefano G. Dantas[3], Benjamin E. Bagozzi[4], Daniel Berliner[5], and Brian Palmer-Rubin[*6]

[1]Department of Political Science, McGill University

[2]Centre for the Study of Democratic Citizenship

[3]Department of Electrical and Computer Engineering, McGill University

[4]Department of Political Science and International Relations, University of Delaware

[5]Department of Government, London School of Economics and Political Science

[6]Department of Political Science, Marquette University

April 6, 2021

v1.11

[*]Erlich's and Dantas' research was made possible through Social Science and Humanities Research Council (SSHRC) Grant #430-2018-1069, Fonds de recherche du Québec - Société et culture (FQRSC) Grant #253243 and Compute Canada. Bagozzi's research was supported by the University of Delaware General University Research Fund and by the National Science Foundation under Grant No. DMS-1737865.

Contents

A Overview

In this supplementary material, we first summarize 12 examples of past political science and social science research that endeavor to classify multiple labels in text-as-data contexts. This is followed by elaborations on (i) the mathematical notation used in our main paper and (ii) the computational costs associated with each multi-label approach outlined in our main paper. We then provide additional information on our Mexico "Access-to-Information" (ATI) application, including additional background on the Mexico case, on the text sample employed, on our human coding of requests, on our classification results & hyperparameter selection choices, and on a set of supplemental classification tasks involving (i) evaluations of deep neural networks and (ii) comparisons of predicted proportions. We next present the full set of CIRI variables that we employ in our second application. Finally, we provide and discuss a series of Monte Carlo experiments.

B Extant Research Employing Multiple Labels

Extant political science—and related social science—research has widely implemented supervised machine learning for the task of classifying multiple non-exclusive textual labels, which are then later employed as independent or dependent variables, or for descriptive purposes. This section summarizes 12 examples. In every instance, these projects (i) have human coders label multiple document traits that can be thought of as distinct predictors or outcome variables and (ii) currently use independent—and sequential—supervised machine learning methods with these human-labels to code a larger set of documents.

1. Barisione and Ceron (2017) analyze a collection of 54,061 UK-, France-, and Italy-based tweets mentioning "austerity." As one component to this analysis, the authors use a supervised coding approach to separately code each tweet for (1) positive-vs-negative sentiment, (2) source (political actor vs. citizen/civil-society actor), and (3) the type of framing (national vs. European). The authors then evaluate how the relative shares of each of these three dichotomies change across their sample over both (i) time and (ii) country-context.

2. Burscher et al. (2014) compare the performance of two supervised classifiers for the task of accurately classifying four nonexclusive, generic news frames: (1) a conflict frame, (2) an economic consequences frame, (3) a human-interest frame, and (4) a morality frame. The authors undertake these classifications using a sample of Dutch daily newspaper articles that they pre-identify as each having some reference to politics. The authors evaluate the classified outcomes largely evaluated in a summary manner, in reference to each classifier's levels of classification accuracy.

3. Casas and Wilkerson (2017) use supervised machine learning to separately classify 11,505 Republican Congress-members' tweets according to four separate dichotomous indicators, corresponding to whether (= 1) or not (= 0) a tweet (1) was about the 2013 government shutdown, (2) mentioned policy, (3) mentioned party competence, or (4) blamed Democrats for the 2013 shutdown. The authors then analyze these indicators as separate outcomes via a variety of modeling approaches.

4. Courtney et al. (2020) apply a supervised machine classification approach to a set of randomly selected newspaper article paragraphs (drawn from the *Financial Times*, *El País*, and *Die Welt*) in order to code four separate binary outcomes pertaining to the presence vs. absence of discussions of (1) macroeconomic policy, (2) microeconomic policy, (3) political competition, and (4) other policy. The authors then primarily analyze these classifications in a summary manner, with respect to their overall prominence in the corpus and the performance of different classifiers (and supervised classification decisions) in accurately recovering each quantity.

5. Hemsley et al. (2018) study 35,639 tweets pertaining to US gubernatorial candidates. They use separate supervised machine learning algorithms to classify these tweets according to the following nine nonexclusive binary categories: (1) attack, (2) narrative, (3) call to action, (4) informing, (5) positioning, (6) alignment, (7) association, (8) targeting, and (9) acknowledge support. The authors then examine the the full-sample distributions of these classified categories, including examinations of some categories' breakdowns in relation to the use or non-use of "@-mentions" within the same underlying tweets.

6. Kostyuk and Zhukov (2019) collect and code event data on physical violence by rebel and government actors in the Ukraine context. To do so, they collect 72,010 Russian, Ukrainian, rebel, and international news reports and blog posts. They then separately human code a sample of these reports for (1) event type, (2) initiator, (3) target, (4) tactic, and (5) casualties, and apply a supervised classification approach to each of these human labeled report qualities in an effort to code all remaining unlabeled news stories along each dimension. Several dimensions of this coding scheme are then used to create distinct event data measures of kinetic operations, which are subsequently related to separately coded data on cyber operations in Ukraine within a vector autoregression context.

7. Larsen and Fazekas (2020) analyze Danish newspaper articles via a supervised clas-

sification and post-classification regression approach. For their overarching sample of 4,147 news articles related to political polling, they separately human code (and then apply supervised machine learning classifiers to) binary measures of (1) whether or not a newspaper article reports changes in political competition in its title, (2) whether or not a newspaper article's body mentioned statistical uncertainty or margin of error for the poll being discussed, and (3) whether or not any relevant persons were quoted. These binary classifications are then subsequently analyzed as distinct outcome variables in a set of regression models.

8. Lörcher and Taddicken (2017) analyze comments posted to a variety of German blog and media sources with an eye towards evaluating climate change communication across multiple platforms. They separately human code a sample of these comments for twenty dimensions pertaining to the (top three) climate change topics discussed, expressions of uncertainty, expressions of skepticism, and other facets of climate change causes, consequences, or responses. These codings are then separately classified for the full sample of media and blog comments via a variety of supervised classification algorithms. The author then evaluate the resultant supervised-coded measures using descriptive and statistics.

9. Minhas et al. (2015) leverage country-year texts from Freedom House and the US State Department alongside a set of discrete codings of political regime types to develop a supervised classification approach for the coding of political regimes. In their application, they use this framework to generate four separate binary measures for (1) Democracy, (2) Monarchy, (3) Military rule, and (4) One-party rule under the logic that these regime categories are not mutually exclusive for real world regimes. They primarily assess their resultant classifications in a descriptive sense via global heatmaps and comparisons to existing regime measures.

10. Mitts (2019) uses a supervised machine learning approach to separately classify 175,015 tweets for whether (= 1) or not (= 0) they topically intersected with (1) sympathy for ISIS, (2) Life in ISIS territories, travel to Syria, or foreign fighters,

(3) the Syrian War, or (4) anti-West rhetoric. The binary classifications in each of these four cases are then separately considered as dependent variables in further analysis.

11. Scharkow (2013) human-labels a corpus of German news stories—drawn from 12 distinct German websites—for a set of nine binary or ordinal variables pertaining to: (1) National politics, (2) International politics, (3) Political economics, (4) Sports, (5) Disasters/accidents, (6) Crime, (7) Controversy, and (8) Prominence. These variables are then separately classified under a supervised machine learning framework and assessed—primarily in the interest of assessing how different text preprocessing decisions affects classification accuracy.

12. Zhukov (2016) applies a supervised classification approach to a collection of 53,754 news reports, press releases, and blog posts to code each news report for the degree that it captured violent political events and related attributes. The latter attributes are separately coded via supervised classification for the following qualities: (1) event type, (2) initiator, (3) target, (4) tactic, and (5) casualties. These classifications are then used to generate a variable corresponding to rebel attacks, which is subsequently analyzed as a dependent variable.

C Mathematical Notation and Formal Presentation

Table C.1: Summary of Mathematical Notation

Notation	Mathematical Meaning
$\mathcal{X}$	d-dimensional input space $\mathbb{R}^d$
x	input feature vector $\in \mathcal{X}, x = (x_1, x_2, ..., x_d)^T$
$\mathcal{Y}$	q-dimensional output space $\{0,1\}^q = \{y_1, y_2, ..., y_q\}$
Y	label set $Y \subseteq \mathcal{Y}$
$\mathcal{D}$	training set $= \{x_i, Y_i \mid 1 \leq i \leq m\}$
$h(\cdot)$	mapping function (classifier) $h : \mathcal{X} \to \mathcal{Y}$, returns the set of q-labels
$f(\cdot,\cdot)$	confidence function, $f(x,y)$ returns the probability of y being the label of x
$P(A\|B)$	Probability of event A given B
$\mathcal{D}^{\dagger}_{\mathcal{Y}}$	Training data transformed to multi-class scheme
$\mathcal{D}_{\tau(j)}$	Available training data for the j-th classifier of the chain classifier
$g^{\dagger}_{\mathcal{Y}}$	multi-class classifier that maps the input space to the new defined class $\mathcal{D}^{\dagger}_{\mathcal{Y}}$
$\sigma_{\mathcal{Y}}$	injective function that maps the labels from multi-label space to multi-class space

C.1 ML-kNN

We formally define ML-kNN as follows. For a test instance (i.e., datapoint) x_d with $y_1 \ldots y_j \ldots y_q$ unknown labels, let C_j denote a membership counting vector of the number of x_d's neighbors in the training set that have label $y_j = 1$, and let H_j be the event that x_d has label $y_j = 1$. Under this algorithm adaptation framework, the posterior probability $P(H_j|C_j)$ represents the probability that H_j holds given that x_d has exactly C_j neighbors with label $y_j = 1$ and $P(\neg H_j|C_j)$ the probability that H_j does not hold. According to the maximum a posteriori (MAP) rule, we can predict each of the q labels of that datapoint

based on the ratio

$$Y = \{y_j \mid \frac{P(H_j|C_j)}{P(\neg H_j|C_j)} > 1, 1 \leq j \leq q\}, \tag{1}$$

where, based on Bayes' theorem, we can thereby write this ratio as

$$\frac{P(H_j|C_j)}{P(\neg H_j|C_j)} = \frac{P(H_j)P(C_j|H_j)}{P(\neg H_j)P(C_j|\neg H_j)} \tag{2}$$

C.2 CC and ECC

Formally, for CC, assume a problem with q possible labels $\{y_1, \ldots, y_q\}$ and a permutation function τ that specifies the ordering of labels, i.e., $y_{\tau(1)}$ comes before $y_{\tau(2)}$. For the j-th label in the order list $y_{\tau(j)}$, the available training data $\mathcal{D}_{\tau(j)}$ is defined as

$$\mathcal{D}_{\tau(j)} = \{([x_i, preceding^i_{\tau(j)}], \phi(Y_i, y_{\tau(j)}) \mid 1 \leq i \leq m\} \tag{3}$$

$$\text{where } \phi(Y_i, y_j) = \begin{cases} 1 & \text{if } y_j \in Y_i, \\ 0 & \text{otherwise} \end{cases}$$

The binary assignment of those labels preceding $y_{\tau(j)}$ is represented by $preceding^i_{\tau(j)}$. These outputs are then concatenated with the input vector x_i. Following this, a binary base classifier $g_{\tau(j)}$ is used to classify $y_{\tau(j)}$. Let $\lambda^x_{\tau(j)} \in \{1, 0\}$ be the output of this classifier for the input x. The outputs of each classifier in the chain are related as follows:

$$\begin{aligned} \lambda^x_{\tau(1)} &= g_{\tau(1)}(x) \\ \lambda^x_{\tau(j)} &= g_{\tau(j)}(x, \lambda^x_{\tau(1)}, \lambda^x_{\tau(1)}, ..., \lambda^x_{\tau(j-1)}) \ (2 \leq j \leq q) \end{aligned} \tag{4}$$

where the predicted label set is given by $Y = \{y_{\tau(j)} \mid \lambda^x_{\tau(j)} = 1, 1 \leq j \leq q\}$

C.3 Label Powerset

The Label Powerset (LP) method approaches the multi-label problem as an ensemble of multi-class classifiers can be achieved by using an injective function $\sigma_{\mathcal{Y}} : 2^{\mathcal{Y}} \to \mathcal{N}$ that maps each occurrence of $\mathcal{Y}$ to natural numbers, and $\sigma^{-1}_{\mathcal{Y}}$ the inverse function that

maps it back to the corresponding label set. Therefore, the new training set is $\mathcal{D}^{\dagger}_{\mathcal{Y}} = \{(x_i, \sigma_{\mathcal{Y}}(Y_i) \mid 1 \leq i \leq m\}$.

Let $g^{\dagger}_{\mathcal{Y}}$ be a multi-class learning algorithm that maps the input space to the new defined class $\mathcal{D}^{\dagger}_{\mathcal{Y}}$. For new examples of x, the LP first maps them to the multi-class classifier and then maps back to the label set of $\mathcal{Y}$:

$$Y = \sigma^{-1}_{\mathcal{Y}}(g^{\dagger}_{\mathcal{Y}}(x)). \tag{5}$$

D Computational Costs

The computational costs of the multi-label approaches presented in our main paper primarily depends on three main factors: the number of training examples m, the number of labels q, and the dimensionality of the input d (number of independent variables). We denote by $\mathcal{C}_B(m, d)$ ($\mathcal{C}_M(m, d, q)$) the complexity of the base binary (multi-class) classifier. The worst-case computational costs for training each multi-label model are described in Table D.1.

Algorithm	**Computation Cost in Big O Notation**
ML-kNN with k neighbors	$O(m^2 * d + q * m * k)$
Binary Relevance	$O(q * \mathcal{C}_B(m, d))$
Classifier Chain	$O(q * \mathcal{C}_B(m, d + q))$
Ensemble Classifier Chain with p permutations	$O(q * \mathcal{C}_B(m, d) * p)$
Label Powerset	$O(\mathcal{C}_M(m, d, 2^q))$
RAKEL with n partitions of k labels	$O(n * \mathcal{C}_M(m, d, 2^k))$

Table D.1: Summary of Computational Costs

E Mexico ATI Application

E.1 Additional Background

With its passage in 2002, Mexico's *Ley Federal de Transparencia y Acceso a la Información Pública Gubernamental* (LFTAIPG) established a groundbreaking online information platform to facilitate access to, and the administration of, Mexican federal government information and its provision. This platform has been in operation since mid-2003, and was primarily named INFOMEX during our period of analysis.[1] INFOMEX manages all citizen (and related actor) information requests, responses, and appeals for Mexico's federal government.[2] While the vast majority of requesters file their requests through INFOMEX's online interface, verbal or hand-written requests (and associated information) are also entered into the INFOMEX system by agency officials. This ensures that the INFOMEX system manages the totality of federal-level information requests in Mexico, starting in mid-2003 onward.[3]

Another important feature of LFTAIPG was its creation of an independent information commission, tasked with promoting awareness and use of the new law, monitoring compliance and sanctioning non-compliance, and resolving appeals (Bogado et al. 2007; Bookman and Guerrero Amparán 2009). The 2002 LFTAIPG law itself, and its associated INFOMEX system and independent information commission, have been frequently characterized as one of strongest ATI systems in place worldwide (Pinto 2009; Michener 2011; Berliner and Erlich 2015; Berliner et al. 2020). Moreover, LFTAIPG's subsequent

[1] It was originally known as Sistema Infomex (SISI).

[2] Additional systems now exist to varying degrees for Mexico's subnational units of government.

[3] The INFOMEX system also administers confidential requests for personal information, though these requests are governed by different legal and disclosure requirements. All confidential requests for personal information are omitted from the analysis and summaries provided above and below, as the request texts associated with confidential requests for personal information or data are not made publicly available.

implementation and usage rates have been likewise hailed as a model among developing countries (Bookman and Guerrero Amparán 2009; Michener 2011; Berliner et al. 2020).

E.2 Request Text Sample

For our period of analysis, there were 1,025,953 ATI requests made within Mexico's federal ATI system (i.e., INFOMEX). Each of these requests reflects an individual query made to a specific Mexican federal government agency or ministry. These requests were commonly made by individual citizens, legal representatives, businesses, and NGOs in relation to information on (e.g.) public security, government procurement contracts, environmental zoning queries, or investigations into government malfeasance.

As briefly noted in the main paper, requesters primarily describe the information that they are requesting by typing their request into a corresponding text-entry box on Mexico's INFOMEX system. Requesters can also enter contextual or supporting information into a separate "otros datos" (other data) textual entry box. Attachments can then be included for the request itself, and are included in roughly 11-15% of all entries—most commonly in the form of PDFs, Word/Excel documents, or image-files. Requesters then choose a target federal government agency for their intended request, and enter a set of personal information (including state/municipality/pots-code information entered via dropdowns) into Mexico's INFOMEX system.

Mexico's INFOMEX system makes all of the information described above—save for requesters' personal identifying information—publicly available online. This information is largely provided via annual CSV files wherein each row corresponds to a unique request made within a particular year. The exception is requester attachments, which need to be separately downloaded individually from Mexico's online information system via links provided within the annual CSV files mentioned above. We downloaded each relevant CSV file, 2003-2015, for the current project, and then individually downloaded[4] each corresponding attachment.

[4]For the human coding step, human coders manually downloaded and viewed each attachment. For the full sample, attachments were downloaded in an automated fashion, and were then converted to plain text via optical character recognition software.

E.3 Human Coding of Requests

The human coding process took place over a 10-week period in Summer 2019, with a team of six Mexico City-based research assistants. The research assistants were all Mexican nationals, alumni of CIDE, a prestigious social science university in Mexico City. They thus all had a strong basis of understanding of Mexican politics and federal government agencies. The coders were supervised by a head RA with prior experience working in Mexico's independent information commission (Mexico's Access to Information Institute). Each RA coded roughly 1,000 requests and corresponding responses. (Ten percent of each coder's requests were double assigned to permit tests of inter-coder reliability. For the variables in this paper, the average intraclass coefficient for multiple coders is .69) RAs were provided with spreadsheets with randomly selected public information requests from the full sample of requests and coded several traits of these using an online Qualtrics coding protocol. The spreadsheet included the file number (*folio*), date, agency, official response, and a link to access online the official government response.

The coding protocol was developed using a multi-stage process, incorporating both the research team's analytical goals as well as concerns about feasibility of coding variables in a reliable way. The research team first brainstormed a series of variables that were of analytical interest, including traits of the request and response. We then subjected this questionnaire to multiple rounds of piloting and revision. The questionnaire took coders roughly 20 minutes on average to complete for a single information request, with wide variation based on the complexity of the request and response.

In the main body of the paper, we analyze four sets of features (for a total of 21 dichotomous features) and one nominal choice feature with five categories. The dichotomous features include 1) traits of the language of the request: formal, legalistic, technical, accusatory; 2) topic(s) of information requested: activities, budget, evaluation, external contracts, institutional structure, other 3) traits of information requested: whether it exists, is the purview of the requested agency, and is not classified. 4) specific terms mentioned, including documents, person, date, place, institution, or NGO. The nominal choice question asked who the coder thought was the likely requester of the information:

academic/scholarly, commercial, monitoring, personal, impossible to say.[5]

E.4 Classification Results and Hyperparameters

Table E.1 reports the full classification results for each of the metrics displayed in the main text. We obtained the execution times by running all models on the Google Colab cloud service. Table E.2 displays the hyperparameters used in each model.

Algorithm	**Subset Accuracy**	**Hamming Loss**	**Ranking Loss**	**F1-micro**	**F1-macro**	**Execution Time (m:ss)**
Classifier Chain (CC)	8.95 %± 0.78	12.21% ± 0.26	0.088 ± 0.002	76.66 ± 0.43	41.84 ± 0.70	0:10
Ensemble CC (ECC)	8.22 % ± 0.75	11.89%± 0.17	0.076 ± 0.002	77.59 ± 0.27	45.20 ± 0.76	1:40
RAkEL, k = q/4	5.09% ± 0.62	12.19%± 0.23	0.084± 0.003	75.17 ± 0.39	30.51 ± 0.43	0:56
RAkEL, k = q/2	6.48% %± 0.85	12.27%± 0.24	0.090± 0.003	75.75± 0.40	35.34± 0.56	1:36
Label Powerset (LP)	8.61 %± 0.71	12.99 ± 0.28	0.114 ± 0.003	74.77± 0.50	33.10 ± 0.75	2:37
Binary Relevance (BR)	4.60% ± 0.66	11.42% ± 0.19	0.071± 0.002	76.84± 0.31	39.84± 0.64	0:06
BR Optimized Models	4.64% ± 0.49	11.60% ± 0.23	0.076 ± 0.001	76.66 ± 0.39	41.75± 0.55	5:49
BR Optimized Thresholds	3.44% ± 0.39	13.70% ± 0.14	0.071 ± 0.002	77.09 ± 0.18	53.31± 0.33	0:37
BR Optimized SMOTE	5.54% ± 0.75	12.42%± 0.24	0.078± 0.002	77.39 ± 0.40	51.83 ± 0.56	10:13
MLkNN	1.58 % ± 0.44	20.37 ± 2.94	0.182 ± 0.040	61.84 ± 8.45	35.51 ± 3.39	0:04

Table E.1: Results

Algorithm	**Hyperparameters**
Classifier Chain (CC)	Classifier chain using Logistic Regression with regularization coefficient $\lambda = 1$ as base classifier
Ensemble CC (ECC)	Classifier chain using Logistic Regression with regularization coefficient $\lambda = 1$ as base classifier
RAkEL, k = q/4	RAkEL using Logistic Regression with regularization coefficient $\lambda = 0.25$ as base classifier
RAkEL, k = q/2	RAkEL using Logistic Regression with regularization coefficient $\lambda = 1$ as base classifier
Label Powerset	LP using Logistic Regression with regularization coefficient $\lambda = 1$ as base classifier
Binary Relevance (BR)	Logistic Regression with regularization coefficient $\lambda = 1$
BR Optimized	Gradient Boosting Classifier (n_estimators = 100), Random Forest Classifier (n_estimators $\in [100, 500, 1000]$)
BR Optimized SMOTE	Gradient Boosting Classifier (n_estimators = 100), Random Forest Classifier (n_estimators $\in [100, 500, 1000]$)
BR Optimized Thresholds	Logistic regression with regularization coefficient $\lambda = 1$ and different decision thresholds for each label
MLkNN	number of neighbours of each input instance = 2, smoothing parameter = 0.5

Table E.2: Hyperparameters Used in the Mexico ATI Experiments

[5]Our key conclusions below also hold when we instead compare our preferred multi-label classifier to a sequential classification approach that treats this nominal variable as categorical via a multinomial logit model.

E.5 Evaluation of Deep Neural Networks

In the present subsection, we extend our multi-label evaluations of the Mexico ATI corpus via a direct comparison of Ensemble CC (ECC) and BR multi-label approaches to equivalent multi-label classifications from deep neural networks-based models. As noted in our main paper—and despite the impressive performance of deep neural networks (DNNs) in several areas—DNNs have not shown dominant performance across different multi-label problem domains when compared to classic models such as CC and BR.[6] Moreover, while similar approaches employing word embeddings[7] have proven to be quite effective in a wide variety of non-multi-label natural language tasks[8]—these extensions also have a number of limitations. Most notably, the corpus used to generate embeddings has to be quite large in order to be effective, which is not often the case in political science applications. One solution to this shortcoming is to use pre-trained embeddings. However, the availability of non-English embeddings—as as is the case in the current

[6]That being said, Xu et al. (2019) do present a selection of high performing DNNs for at least some specific multi-output learning models, such as hierarchical multi-label classification (Baker and Korhonen 2017) and time-series prediction (Smith and Jin 2014).

[7]Word embeddings capture both semantic and syntactic information of words. In this approach, each word is represented by a multi-dimensional vector, where each entry represents information about that word meaning and context. Some of the most popular examples of word embeddings methods are the *word2vec* (Mikolov et al. 2013), *GloVe* (Pennington et al. 2014) and *fastText* (Joulin et al. 2016). Political scientists have also used these techniques to ideologically scale political candidates based on their speech (**rheault_word_2020**), measure differences in campaign speech across parties (**arnold2018covering**), track the meaning of political concepts over time (**rodman_timely_2019**), and to build customized sentiment dictionaries (**rice_corpus-based_2019**).

[8]More specifically, the current state-of-the-art performance on NLP tasks is achieved by deep neural network models that incorporate context into word embeddings, such as *ELMo* (Peters et al. 2018), *BERT* (Devlin et al. 2018) and *GPT-2* (Radford et al. 2019).

Spanish language application—is also often limited. Moreover, if the embeddings are trained (generated) on a corpus that is significantly different than the corpus of the task, the embeddings might not be very informative as some words might have different meanings depending on the topic.

Despite the potential limitations for the approaches outlined above within the context of our current, middle-sized ATI dataset, we compare our previous results to a selection of deep neural networks approaches below so as to provide a benchmark for future innovations in this area. To this end, we employed two different deep neural networks to act as baseline for our problem: a long-short term memory network (LSTM; Hochreiter and Schmidhuber 1997) and a convolutional neural network (Kalchbrenner et al. 2014). The embeddings were generated using the available text from the ATI dataset. The details of the architecture are described in Table E.3 below.

Algorithm	**Hyperparameters**
Embedding Information	Vocabulary size = 10,000, embedding dimension = 32, maximum sequence length = 400
CNN	Embedding Layer -> 1D convolutional layer with 32 filters of size 3 -> average pooling layer -> fully connected layer with 32 hidden units -> dropout (p = 0.2) -> output layer
LSTM	Embedding Layer -> Bi-directional LSTM layer with 32 neurons -> fully connected layer with 32 hidden units -> dropout (p = 0.2) -> output layer

Table E.3: Details on Deep Neural Network Architectures Used

For the same Mexico ATI application detailed above and in the main paper, we present our comparison results for our ECC, BR, convolutional neural network, and LSTM network approaches in Table E.4. In this application, we find that ECC, followed by BR, each exhibit superior subset accuracy, hamming loss, and ranking loss to either of our deep neural networks approaches. On these three metrics, the LSTM network marginally outperforms the convolutional neural network. Our results are slightly more mixed for F1-micro and F1-macro. For these latter two metrics, ECC performs best in each case, followed BR in the case of F1-micro and the LSTM network in the case of F1-macro. In sum, ECC consistently outperforms our deep neural network approaches within the current Mexico application and furthermore—as the left-most column of Table

E.1 indicates—does so with an execution time that is 5-to-10 times lower than that of the two deep neural network approaches considered here. While these findings and conclusions are likely contingent upon the modest size of the current (ATI) dataset under consideration, they nevertheless suggest that ANNs may not always be preferable to our paper's proposed multi-label approaches within real world political science applications.

Algorithm	Subset Accuracy	Hamming Loss	Ranking Loss	F1-micro	F1-macro	Execution Time (m:ss)
Ensemble CC (ECC)	8.22 % ± 0.75	11.89%± 0.17	0.076 ± 0.002	77.59 ± 0.27	45.20 ± 0.76	1:40
Binary Relevance (BR)	4.60% ± 0.66	11.42% ± 0.19	0.071± 0.002	76.84± 0.31	39.84± 0.64	0:06
Convolutional Neural Network	4.45 % ± 0.44	13.98 ± 0.23	0.122 ± 0.003	73.79 ± 0.42	40.47 ± 2.01	5:52
LSTM Network	5.55 % ± 0.57	13.62 ± 0.32	0.112 ± 0.005	74.63 ± 0.47	42.15 ± 1.00	16:46

Table E.4: Results for BR, ECC and Two Deep Neural Network Variants

E.6 Comparison of Predicted Proportions

Despite the ECC approach's commensurate performance along all of the metrics considered within our ATI application, it would perhaps be more informative if we were able to evaluate ECC's performance in classifying all available requests. Since there are no labels for this set of all ATI requests (i.e., outside of the random sample that we human-labeled), we cannot evaluate this in a straightforward matter.

With this constraint in mind, we decided to compare three models in a different fashion. As the available labeled data was drawn from a random sample, we assume that that the proportions of each label in this subset of requests is similar to the proportion of each label in all requests. Hence, we want a model whose predictions have a similar proportion to the "true proportions."[9]

We evaluated three models: ECC, our standard Binary Relevance (BR) approach, and the BR with optimized threshold. The metric used for this evaluation was the average deviation from the hand-coded proportions. The results are BR: 8.75%; BR with optimized thresholds: 8.08%; Esemble Classifier Chain (ECC): 5.79%.

As can be seen in this Table, ECC obtained, on average, the lowest deviation from the hand-coded proportions. By comparison, BR and BR Optimized each fair decidedly

[9]I.e., the proportions found in the hand-coded data.

worse, but comparably, with regards to average deviation from hand-coded proportions. These findings do not guarantee that ECC's true performance is better than the other classification approaches considered here. However, since ECC also outperformed these other two models for the standard metrics within our human labeled data analysis in the main paper, this proportion deviation-based finding is strongly suggestive of ECC's superior performance in classifying all unlabeled requests.

F Human Rights Application

In Table F.1, we list the full set of CIRI (Cingranelli and Richards 2010) variables that we consider in our human rights application. Note that for our application, we omit CIRI's "women's social issues" variable as it was phased out in 2005. All results presented in the made paper are comparable when we instead include this measure. As noted in our main paper, we dichotomize each originally ordinal CIRI variable for our application. As indicated in Table F.1 and its corresponding note, most of CIRI's ordinal measures are comprised of three categories, which we dichotomize to compare the absence of a given human rights violation to the remaining two categories (which typically correspond to 'some' and 'widespread' violations). As is also noted below, we then comparably dichotomize CIRI's four-category (women's rights) variables at their midpoints.

Table F.1: CIRI Variables Used in Human Rights Application

CIRI Variable Name	No. Categories
Disappearance	Three
Extrajudicial Killing	Three
Political Imprisonment	Three
Torture	Three
Freedom of Assembly and Association	Three
Freedom of Foreign Movement	Three
Freedom of Domestic Movement	Three
Freedom of Speech	Three
Electoral Self-Determination	Three
Freedom of Religion	Three
Worker's Rights	Three
Women's Economic Rights	Four
Women's Political Rights	Four
Independence of the Judiciary	Three

Note: We dichotomize each three category measure listed above to compare the absence of a given human rights violation (= 0) to some/partial/widespread violations of that type (= 1). Each four category variable is dichotomized at its midpoint.

G Monte Carlo Experiments

This section presents the results from four main Monte Carlo experiments. These experiments collectively assess the performance of the ECC and BR approaches within the task of multi-label classification under four different scenarios of distinct training-test sample splits. Altogether, these experimental assessments thereby allow us to evaluate the relative performance of our preferred multi-label classification approach—ECC—under conditions where a researcher's available training and test data for multi-label classification varies.

In each experiment, we hold the total number of observations fixed at $N = 1,000$. Our first experimental assessment (Experiment 1) then considers a scenario where one's available training (testing) data comprises 85% (15%) of all observations. Experiment 2 then reassesses ECC's and BR's performance for a scenario of 70% training data and 30% test data. Experiment 3 further reduces our Monte Carlo assessments' available training to 55% of all observations, thus entailing 45% test data. Finally, Experiment 4 considers a scenario where one's available training data comprises only 40% of one's total $N = 1,000$ sample (thus implying 60% test data). Given the low N considered across all experiments, these differences in training sample shares are non-trivial, and help to directly inform political science researchers as to the viability of multi-label methods in instances where coding or collecting additional training labels is costly or infeasible.

For each experiment, we assign the number of *Sims* to be $1,000$, and then set about generating five correlated binary labels $\mathbf{y} = (\mathbf{y_1}, \mathbf{y_2}, \mathbf{y_3}, \mathbf{y_4}, \mathbf{y_5})$. To generate these five binary labels, we first defined a series of predictors, $\mathbf{x} = (\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, \mathbf{x_4}, \mathbf{x_5})'$. These five predictors ($\mathbf{x_{1:5}}$) were randomly drawn from Poisson distributions with $\lambda = (1, 0.25, 0.1, 2, 0.5)$, respectively, so as to ensure that our predictors mirror the types of document-level traits that researchers often encounter in text-as-data classification (e.g., document term matrices). Parameter values (ϕ's) were then randomly drawn from a $\mathcal{N}(0, 5)$. Using a unique vector of ϕ's for each label, a continuous $\mathbf{y}^*$ label was then

initially generated via an additive combination of five $\mathbf{x}$'s and/or $\mathbf{y}$'s[10] plus a constant term and an error term of $\mathcal{N}(0,1)$. Each resultant $\mathbf{y}^*$ was then discretized via a cutpoint which ensured that each final binary $\mathbf{y} = (\mathbf{y_1}, \mathbf{y_2}, \mathbf{y_3}, \mathbf{y_4}, \mathbf{y_5})$ were slightly imbalanced but not markedly so, with an average level of overall absolute correlation of 0.54.

The above steps created a series of five correlated binary labels—and a set of candidate predictors—within each of our individual Monte Carlo simulations. For each Monte Carlo experiment, we then assess the out-of-sample multi-label performance of BR and ECC in terms of each of the multi-label classification statistics reported in the main paper: subset accuracy, hamming loss, ranking loss, F1-micro, and F-macro.

Above and beyond accurate recovery of our $\mathbf{y}$'s according to these multi-label performance metrics, we are also interested in leveraging our Monte Carlo experiments to evaluate how well BR's and ECC's recovered $\mathbf{y}$'s perform within a set of auxiliary (post-classification) regressions. These post-classification assessments are of particular relevance to political science researchers, who often undertake multi-label classification with an intention of using their classified labels within subsequent explanatory regression analyses. To evaluate this scenario, and after generating our full $\mathbf{y}$ according to the steps described above (but *before* performing multi-label classification), each Monte Carlo simulation also generated a continuous, auxiliary outcome variable ($\mathbf{z}$) as an additive function of $\mathbf{y_2}$, $\mathbf{y_3}$, and $\mathbf{y_4}$ (now as predictors), a new set of parameter values (β),[11] and a univariate normal error term of $\mathcal{N}(0,1)$. This auxiliary simulation step, in combination with our earlier simulation steps, accordingly allowed us to evaluate the performance of our BR-derived and ECC-derived $\mathbf{y}$'s within three distinct post-classification scenarios:

1. A scenario where a researcher is interested in evaluating the individual effect of a single multi-label classified $\mathbf{y}$ upon an external variable: $\mathbf{z} = \beta_0 + \mathbf{y_4}\beta_1 + e$. We refer to this a the "bivariate regression" model below.

[10]Whilst ensuring that among these five predictors, no more than two other $\mathbf{y}$'s, and at times zero additional $\mathbf{y}$'s, were included as predictors.

[11]In order to facilitate the analysis, we set the β values to 1 for the bivariate and multiple regression scenarios, but the same results hold for randomly selected β's.

2. A scenario where a researcher is interested in evaluating the effects of multiple multi-label classified $\mathbf{y}$'s upon an external variable: $\mathbf{z} = \beta_0 + \mathbf{y_2}\beta_1 + \mathbf{y_3}\beta_2 + \mathbf{y_4}\beta_3 + e$. We refer to this as the "multiple regression" model below.

3. A scenario where a researcher is interested in evaluating the individual effect of a multi-label classified $\mathbf{y}$ upon another multi-label classified $\mathbf{y}$: $\mathbf{y_4} = \beta_0 + \mathbf{y_3}\beta_1 + e$. We refer to this as the generalized linear model (GLM) below.

After recovering our $\mathbf{y}$'s via BR and ECC, we implemented each of the above regressions—separately for our ECC-classified $\mathbf{y}$'s and for our BR-classified $\mathbf{y}$'s—within each simulation run. Auxiliary regressions 1-2 were estimated with ordinary least squares (OLS) regression, whereas auxiliary regression 3 was estimated via a generalized linear model (GLM) with a logit link given that the outcome variable in that case ($\mathbf{y_4}$) is binary. We then retained the parameter estimates and standard errors from each of these models. Using these estimates alongside the true values for each parameter, we next calculated each parameter's root mean square error (RSME) and empirical 95% coverage probability (CP)[12], which we averaged across all simulations, to compare the parameter estimates obtained from regressions 1-3 when using our ECC- versus BR-classified $\mathbf{y}$'s. We report these quantities of interest within a series of tables and figures further below. Additionally, we present a summary of the simulation's steps in Algorithm 1.

[12]I.e., the proportion of times—out of all $1,000$ $Sims$—that a true parameter fell within the 95% confidence intervals of a model's corresponding parameter estimate.

Algorithm 1 Monte Carlo Simulations

Inputs: $Sims$ (number of simulations), N (number of samples), $test_{split}$ (percentage of test data)

Initialize predictors: $\mathbf{x_1} \sim Poisson(1, N), \mathbf{x_2} \sim Poisson(0.25, N)$, $\mathbf{x_3} \sim Poisson(0.1, N), \mathbf{x_4} \sim Poisson(2, N), \mathbf{x_5} \sim Poisson(0.5, N)$

Initialize parameters: $\phi_1 \sim Normal(0, 1, N), \phi_2 \sim Normal(0, 1, N)$, $\phi_3 \sim Normal(0, 1, N), \phi_4 \sim Normal(0, 1, N), \phi_5 \sim Normal(0, 1, N), \epsilon_c \sim Normal(0, 1, 5 \times N)$, $\epsilon_r \sim Normal(0, 1, N)$

for $k = 1, ..., Sims$ **do**

  aggregate predictors into $\mathbf{x} = (\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, \mathbf{x_4}, \mathbf{x_5})'$

  generate each continuous label $\mathbf{y_n^*} = \phi_n X_n^* + \epsilon_c$, {where X_n^* is a combination of $\mathbf{x}$ and up to two previous labels $\in (\mathbf{y_{n-1}^*}, \mathbf{y_{n-2}^*}, ..., \mathbf{y_1^*})$}

  discretize y^* to $\mathbf{y}$ according to a cutpoint

  split data into $(1 - test_{split})\%$ training and $test_{split}\%$ testing

  fit BR and ECC on training data

  get predictions $\hat{\mathbf{y}}$ using testing data

  Initialize outcome variable:

  if Regression **then**

    $\mathbf{z} = \beta \mathbf{y} + \epsilon_r$

  else

    $\mathbf{y_4} = \beta_0 + \mathbf{y_3}\beta_1 + e_r$

  end if

  predict the outcome variable accordingly

  store results

end for

return classification and regression metrics

Before turning to our auxiliary regression Monte Carlo results, we first consider the overall multi-label classification performance of BR and ECC with respect to the primary multi-label task at hand: recovery of each relevant binary label $\mathbf{y} = (\mathbf{y_1}, \mathbf{y_2}, \mathbf{y_3}, \mathbf{y_4}, \mathbf{y_5})$.

These results can be found in Table G.1. Here we can first observe that ECC outperforms BR on every metric, and across each Experimental condition. Looking across Experiments 1-4, we can further note that the abilities of ECC and BR to accurately recover our $\mathbf{y}$'s also declines in performance as one's available training data decrease. However, in most cases, these declines in performance as one moves from Experiments 1 to Experiment 4 are negligible for both BR and ECC, especially in relation to the divergence in performance across the two approaches for any one metric. Altogether, these results hence strongly favor ECC over BR in the context of our simulations—thereby offering further support for this multi-label approach in contexts where one's overall (and training) sample size is far smaller than either of the applications presented in our main paper.

To better summarize and visualize the classification performance of ECC and BR in classifying $\mathbf{y_1}, \mathbf{y_2}, \mathbf{y_3}, \mathbf{y_4}$, and $\mathbf{y_5}$, we also plot the combined (i.e., pooled) results for each multi-label classification metric—Subset Accuracy, Hamming Loss, F1-Macro, F1-Micro, and Ranking Loss—across all of our different training and testing data splits. The results in this case are hence averaged over all Experiments, and are shown in Figure G.1. For the top-right and top-left corners plots, the higher the metric, the better. For the bottom-right and bottom-left corners, the lower, the better. As one can see in Figure G.1, ECC outperforms BR across all multiple classification metrics[13]—no matter the training-test split considered.

We now turn to the results obtained under each of our auxiliary regression assessments that employ the (BR and ECC) multi-label-classified $\mathbf{y}$ measures described above. The first set of results are related to the bivariate regression and we present in Table G.2 and in Figure G.2. For the sake of simplicity, recall that the auxiliary data generating process in this case had set all the true β's to 1. Turning to Table G.2, we first note that the estimated β's from ECC are closer to the corresponding true β value of 1 in all Experimental conditions, when compared to the β's estimated by BR. Based upon these averaged β's—and the reported RMSEs and 95% empirical CPs—the two approaches appear most comparable in terms of accuracy and coverage under our Experiment 1

[13]And especially so for subset accuracy and hamming loss.

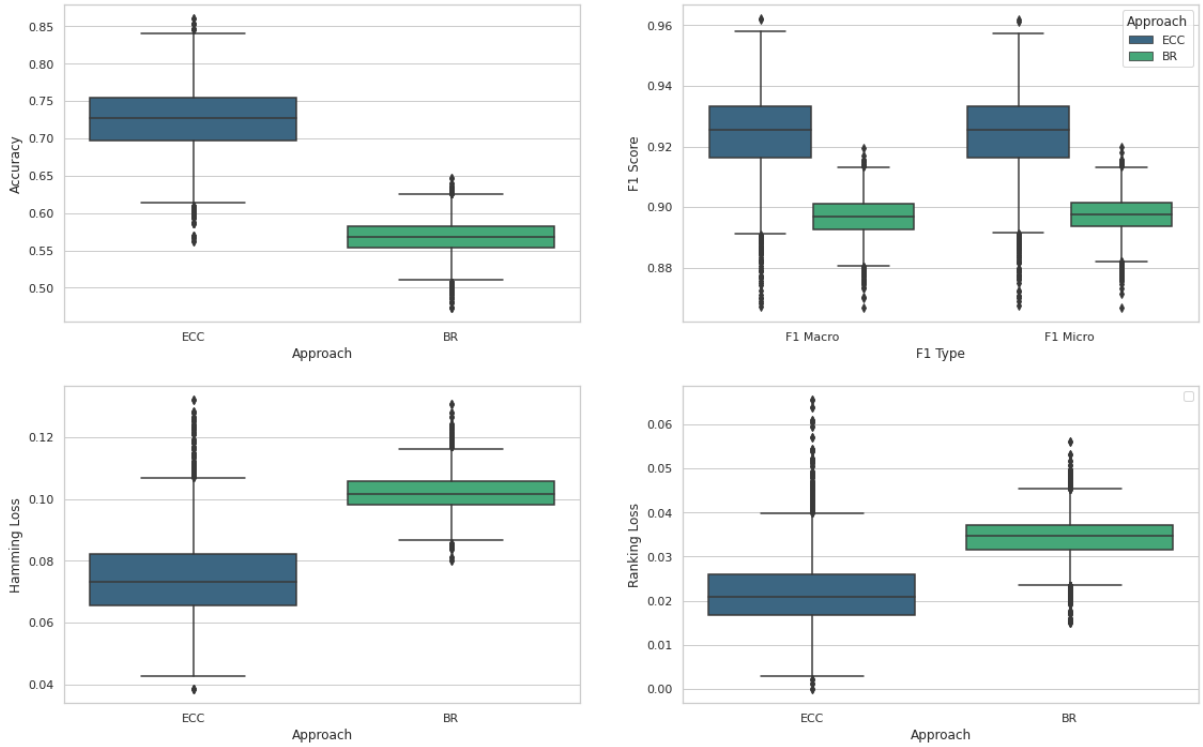

Figure G.1: Box-plots Comparisons of (ECC versus BR) Multi-label Classification Metrics, Across all Experiments

condition (85% training data). As one reduces each approach's available training data over our subsequent three Experimental conditions, we then observe relative declines in accuracy and coverage, though these performance declines are more substantial for BR. As a consequence, for our lowest training data condition—Experiment 4, where available training data corresponded to only 40% of all data, and the remaining 60% correspond to our test predictions—we find that the relative gains in accuracy and coverage for ECC's estimated β's (i.e., relative to those of BR) are more substantial than was the case for the earlier Experimental instances where more training data was available.

Why, in this case, does ECC perform increasingly better than BR as one's available training (test) data decreases (grows)? This may at first seem counterintuitive given the results discussed for in Table G.1, where ECC *did not* increasingly outperform BR in directly recovering our $\mathbf{y}$'s as one's training (test) data decreased (grew). However, in the current auxiliary regression context, recall that each regression input is now a combination of training $\mathbf{y_4}$ cases *and* classified (i.e., test) $\mathbf{y_4}$ cases. As such, for the auxiliary bivariate regressions conducted under Experiment 1, a majority of both approach's re-

Classification Metrics	Subset Accuracy		Hamming Loss		F1-macro		F1-micro		Ranking Loss	
	ECC	BR	ECC	BR	ECC	BR	ECC	BR	ECC	BR
Experiment 1 Training 85% Testing 15%	0.729	0.560	0.073	0.103	0.926	0.895	0.926	0.896	0.020	0.033
Experiment 2 Training 70% Testing 30%	0.722	0.561	0.076	0.103	0.922	0.894	0.923	0.895	0.022	0.036
Experiment 3 Training 55% Testing 45%	0.727	0.571	0.073	0.101	0.925	0.899	0.925	0.899	0.021	0.034
Experiment 4 Training 40% Testing 60%	0.724	0.577	0.075	0.101	0.924	0.899	0.924	0.899	0.024	0.035

Table G.1: Table summarizing the Classification results.

gression inputs ($\mathbf{y_4}$'s) were training cases, thus reducing ECC's potential "value added" in terms of more accurately recovered test $\mathbf{y_4}$'s. By contrast, under Experiment 4, the majority of values for the $\mathbf{y_4}$ regressor in our auxiliary bivariate regression correspond to classified (test) $\mathbf{y_4}$ cases. Since these regression inputs now encompass a larger share of classified (as opposed to training-labeled) $\mathbf{y_4}$ cases, ECC's more accurate classifications of $\mathbf{y_4}$ naturally ensure better auxiliary regression performance.

For scenarios where researchers are interested in using classified labels individually within one or more regression models, this implies that multi-label classification methods such as ECC are especially preferable to BR in cases where available training (testing) data for classification is limited (expansive). This is very likely to be the case for many political science contexts, where researchers' human coded labels for any given text-as-data application often comprise only a small fraction of all cases of interest. Lastly, and reaffirming the broader conclusions drawn above, we can also observe that the distribution of ECC-recovered β's—now combined for all Experiments—are closer to the true values of each β than are the BR-recovered β's in the box-plots presented in Figure G.2. This provides further indication of ECC's consistent advantages over BR for auxiliary regressions employing a single classified $\mathbf{y}$ as a regressor.

Bivariate	**Intercept**			**Y4**		
Regression	True β_0	ECC	BR	True β_1	ECC	BR
Experiment 1		1.01	1.02		0.98	0.95
Training 85%	1	(0.04)	(0.04)	1	(0.06)	(0.07)
Testing 15%		[0.940]	[0.937]		[0.926]	[0.880]
Experiment 2		1.03	1.05		0.94	0.89
Training 70%	1	(0.05)	(0.06)	1	(0.07)	(0.11)
Testing 30%		[0.884]	[0.803]		[0.836]	[0.617]
Experiment 3		1.05	1.08		0.91	0.84
Training 55%	1	(0.05)	(0.08)	1	(0.10)	(0.16)
Testing 45%		[0.827]	[0.635]		[0.677]	[0.314]
Experiment 4		1.06	1.10		0.87	0.79
Training 40%	1	(0.07)	(0.10)	1	(0.13)	(0.21)
Testing 60%		[0.732]	[0.482]		[0.514]	[0.114]

Note: cell values are $\hat{\beta}$'s; values in parentheses are RMSE's; values in brackets are 95% CP's.

Table G.2: Bivariate Regression Results

The above results illustrate that ECC's recovered $\mathbf{y}$'s outperform those of BR when said $\mathbf{y}$'s are individually considered as explanatory variables within subsequent regressions. Do these same findings hold when a researcher wishes to jointly include multiple classified $\mathbf{y}$'s as explanatory variables within such a regression? To evaluate this, we next turn to our multiple regression comparisons, as outlined under auxiliary regression scenario two above. The results from our multiple regression model assessments are presented in Table G.3 and in Figure G.3.

Turning first to Table G.3 we can begin by noting that our overall findings are fairly similar to those obtained for the auxiliary bivariate regression assessment above. The ECC-estimated β's are generally closer to the true β values than are those of BR for each parameter estimate of interest and across each Experimental condition. As such, ECC's averaged RMSEs and 95% empirical CPs generally outperform those of BR across our

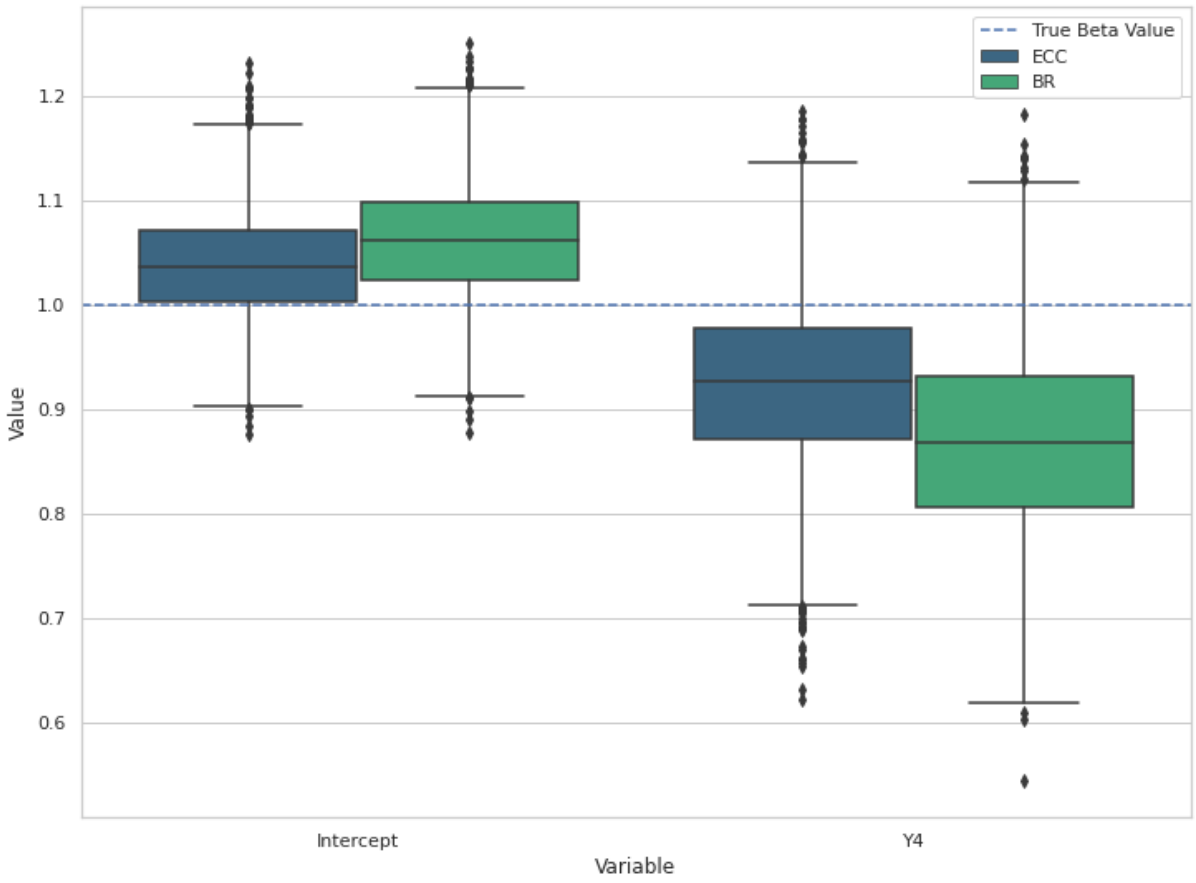

Figure G.2: Box-plot of Estimated β's from the BR and ECC Approaches for the Bivariate Regression setting.

various parameter estimates and Experimental conditions, and increasingly so as the level of training (test) data decreases (increases). As was the case for the previous auxiliary regression assessment, this again suggests that the advantages of ECC over BR will be especially relevant for auxiliary regression scenarios where a researcher is faced with a small (large) share of training (test) data within a prior multi-label classification task.

That being said, we do find two exceptions to these broader trends. First, BR's averaged estimates for β_1 remain very close to those of ECC across all Experimental conditions, and are in fact tied with that of ECC for Experiment 4. As a consequence, we find that our ECC- and BR-specific RMSEs and CPs are highly similar for this particular explanatory variable no matter the Experimental condition. Second, our findings for β_3 in Table G.3 much more dramatically favor ECC over BR as one decreases available training data than was the case for our bivariate regression assessment. For example, whereas ECC's and BR's 95% empirical CPs were 0.942 and 0.815 for Experiment 1 in the

current multiple regression context, BR's 95% empirical CP effectively falls to 0 (0.001) once one's training (testing) data has been reduced (increased) to $N = 400$ ($N = 600$), whereas ECC's CPs in this case maintained a decent degree of empirical coverage (0.423). Hence—and although using multiple ECC-classified $\mathbf{y}$'s as independent variables within an auxiliary regression may at times yield comparable findings to those of BR (as is the case for $\mathbf{y_2}$)—ECC remains far superior to BR for this task on the whole, and especially so if a researcher is faced with a low ratio of training-to-test data. These conclusions are underscored by Figure G.3, which combines our $\hat{\beta}$'s across all Experimental conditions. In this case, we again find that the distributions of the ECC-recovered β's are noticeably closer to each true β value than are comparable distributions for the BR-recovered β's.

Multiple	Intercept			**Y2**			**Y3**			**Y4**		
Regression	True β_0	ECC	BR	True β_1	ECC	BR	True β_2	ECC	BR	True β_3	ECC	BR
Experiment 1		1.03	1.08		0.99	0.98		0.99	0.94		0.97	0.91
Training 85%	1	(0.07)	(0.09)	1	(0.10)	(0.10)	1	(0.09)	(0.11)	1	(0.07)	(0.10)
Testing 15%		[0.931]	[0.849]		[0.962]	[0.964]		[0.940]	[0.904]		[0.942]	[0.815]
Experiment 2		1.07	1.19		0.95	0.92		0.97	0.89		0.92	0.80
Training 70%	1	(0.10)	(0.19)	1	(0.12)	(0.12)	1	(0.10)	(0.13)	1	(0.10)	(0.20)
Testing 30%		[0.823]	[0.366]		[0.926]	[0.911]		[0.928]	[0.832]		[0.800]	[0.327]
Experiment 3		1.14	1.29		0.90	0.87		0.98	0.86		0.85	0.70
Training 55%	1	(0.15)	(0.29)	1	(0.15)	(0.15)	1	(0.10)	(0.16)	1	(0.15)	(0.30)
Testing 45%		[0.602]	[0.046]		[0.840]	[0.824]		[0.930]	[0.735]		[0.574]	[0.035]
Experiment 4		1.19	1.36		0.88	0.88		0.94	0.78		0.80	0.61
Training 40%	1	(0.20)	(0.36)	1	(0.17)	(0.15)	1	(0.12)	(0.22)	1	(0.21)	(0.39)
Testing 60%		[0.442]	[0.001]		[0.804]	[0.808]		[0.885]	[0.510]		[0.423]	[0.001]

Note: cell values are $\hat{\beta}$'s; values in parentheses are RMSE's; values in brackets are 95% CP's.

Table G.3: Multiple regression results

Finally, it may also be the case that researchers are interested in relating two classified variables (i.e., $\mathbf{y}$'s) to one another in an auxiliary regression. One concern in this context is that multi-label approaches such as ECC may risk overstating (i.e., biasing upwards) a relationship in this context, relative to BR, due to potential overfitting of one's label associations. To evaluate this, we present the results for the third scenario, where evaluate

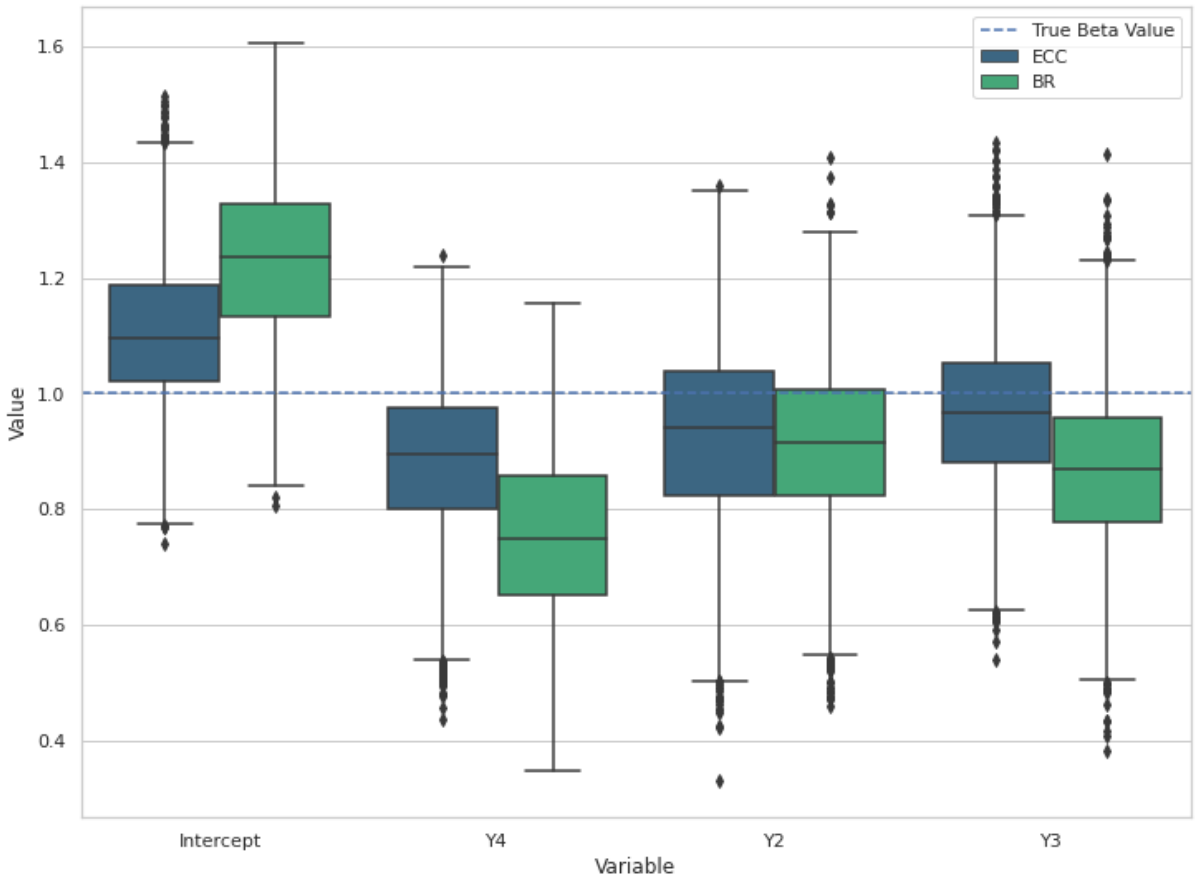

Figure G.3: Box-plot of estimated betas from the BR and ECC approaches for the multiple regression setting

the individual effect of a label upon another label using a GLM. The results were evaluated using the same metrics presented above. However, one main difference in relation to our prior two auxiliary regression evaluations is that the true value of each β in this cases change according to the labels' relation for each individual simulation run. Thus, we report the average value of β_0 and β_1 across all simulations for this set of evaluations, whose summary quantities can be found in Table G.4 and Figure G.4.

Turning to this Table and Figure, we confirm in this case that even for instances where a researcher plans to relate a pair of classified labels to one another in an auxiliary regression, ECC still performs better than BR in terms of accuracy and coverage. This is illustrated in the aggregate for both parameter estimates of interest, and across all Experimental conditions, within the $\hat{\beta}$ box-plots presented in Figure G.4. Comparing our two approaches more extensively for each Experimental condition in Table G.4, we reach similar conclusions for our RMSEs and CPs on the whole. Therein, we can further

observe that the relative advantages of ECC over BR in this context again grow in size as one's available training (test) data declines (increases). For instance, looking at β_1, we find that both approaches exhibit similar RMSEs (of 0.05 and 0.07) and CPs (of 1 in each case) under scenarios with 85% training data (Experiment 1), but then markedly diverge in accuracy and coverage when one's available training (test) data is decreased (increased) to 40% (60%) as in Experiment 4. In the latter case, we specifically find that BR's RMSE is over twice the size of ECC's RMSE for β_1, and that BR's 95% CP is remarkably poor (0.023) in comparison to that of ECC (0.726). Hence, even in instances where one intends to subsequently use a set of classified labels as one's outcome *and* explanatory variables, ECC is preferable to BR. That being said, it is worth emphasizing that the Monte Carlo findings and insights discussed here and earlier are dependent upon our specific choices of sample size, label correlation, parameter values, and choices of multi-label classifiers. Future research should continue to extend and assess each evaluation scenario presented above under alternate conditions.

GLM	**Intercept**			**Y3**		
	Avg. True β_0	CC	BR	Avg. True β_1	CC	BR
Experiment 1		1.16	1.13		-2.30	-2.19
Training 85%	1.13	(0.04)	(0.02)	-2.26	(0.06)	(0.07)
Testing 15%		[1]	[1]		[1]	[1]
Experiment 2		1.18	1.06		-2.33	-2.06
Training 70%	1.13	(0.06)	(0.07)	-2.26	(0.13)	(0.21)
Testing 30%		[0.999]	[0.997]		[0.991]	[0.881]
Experiment 3		1.17	0.97		-2.32	-1.90
Training 55%	1.13	(0.09)	(0.16)	-2.26	(0.18)	(0.36)
Testing 45%		[0.96]	[0.757]		[0.872]	[0.149]
Experiment 4		1.19	0.94		-2.33	-1.81
Training 40%	1.13	(0.12)	(0.19)	-2.26	(0.22)	(0.45)
Testing 60%		[0.867]	[0.572]		[0.726]	[0.023]

Note: cell values are $\hat{\beta}$'s; values in parentheses are RMSE's; values in brackets are 95% CP's.

Table G.4: Individual effect experiment

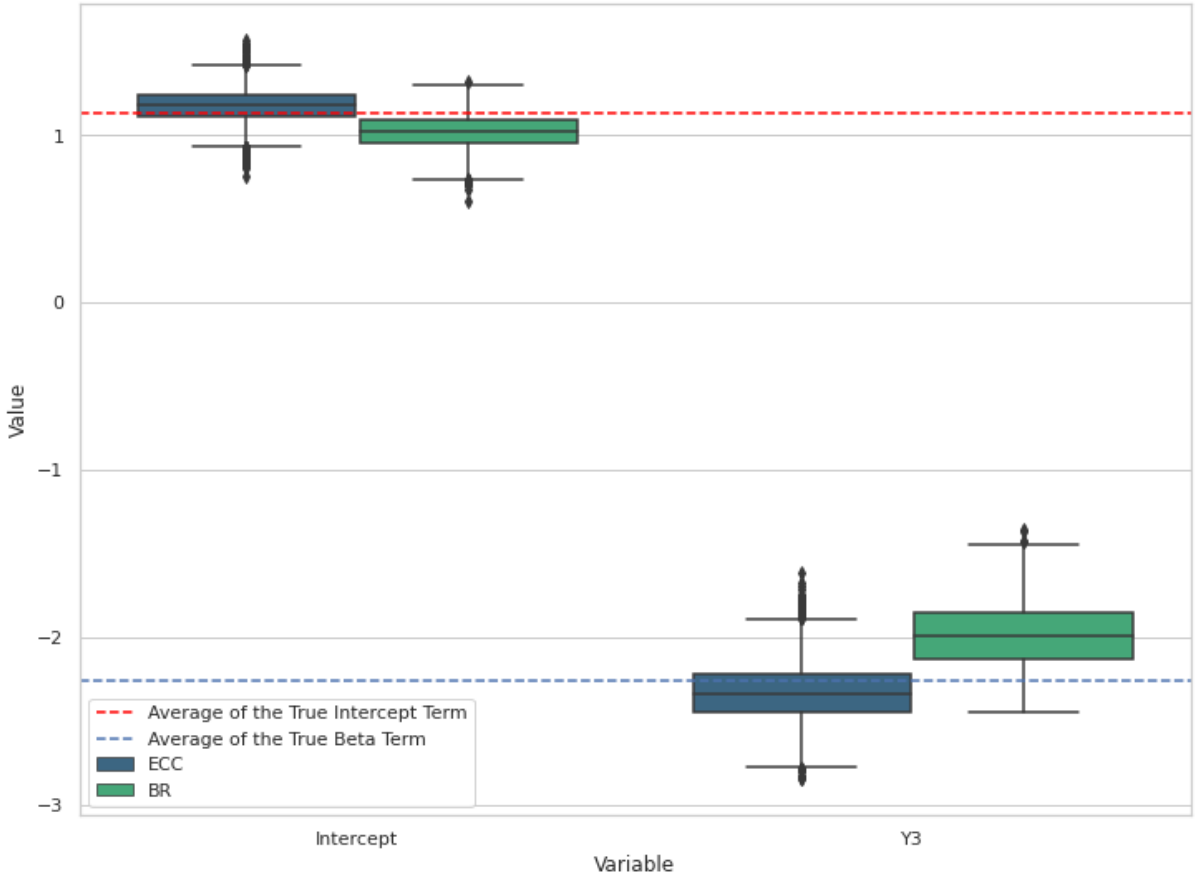

Figure G.4: Box-plot of estimated betas from the BR and ECC approaches for the individual effect setting

References

Baker, Simon, and Anna-Leena Korhonen. 2017. "Initializing neural networks for hierarchical multi-label text classification." Association for Computational Linguistics.

Barisione, Mauro, and Andrea Ceron. 2017. "A Digital Movement of Opinion? Contesting Austerity Through Social Media." In *Social Media and European Politics: Rethinking Power and Legitimacy in the Digital Era,* edited by M. Barisione and A. Michailidou. Palgrave Macmillan.

Berliner, Daniel, Benjamin E Bagozzi, Brian Palmer-Rubin, and Aaron Erlich. 2020. "The Political Logic of Government Disclosure: Evidence from Information Requests in Mexico." *Journal of Politics.*

Berliner, Daniel, and Aaron Erlich. 2015. "Competing for Transparency: Political Competition and Institutional Reform in Mexican States." *American Political Science Review* 109.

Bogado, Benjamin Fernandez, Emilene Martinez-Morales, Bethany Davis Noll, and Kyle Bell. 2007. *The Federal Institute for Access to Information and a Culture of Transparency: Follow Up Report.* Technical report. Annenberg School of Communications, University of Pennsylvania. `http://www2.gwu.edu/~nsarchiv/NSAEBB/NSAEBB247/Annenberg.pdf`.

Bookman, Zachary, and Juan-Pablo Guerrero Amparán. 2009. "Two Steps Forward, One Step Back: Assessing the Implementation of Mexico's Freedom of Information Act." *Mexican Law Review* 1 (2).

Burscher, Björn, Daan Odijk, Rens Vliegenthart, Maarten de Rijke, and Claes H. de Vreese. 2014. "Teaching the Computer to Code Frames in News: Comparing Two Supervised Machine Learning Approaches to Frame Analysis." *Communication Methods and Measures* 8 (3).

Casas, Andreu, and John Wilkerson. 2017. "A Delicate Balance: Party Branding During the 2013 Government Shutdown." *American Politics Research* 45 (5).

Cingranelli, David L., and David L. Richards. 2010. "The Cingranelli and Richards (CIRI) Human Rights Data Project." *Human Rights Quarterly* 32 (2).

Courtney, Michael, Michael Breen, Ian McMenamin, and Gemma McNulty. 2020. "Automatic Translation, Context, and Supervised Learning in Comparative Politics." *Journal of Information Technology & Politics.*

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805.*

Hemsley, Jeff, Jennifer Stromer-Galley, Bryan Semaan, and Sikana Tanupabrungsun. 2018. "Tweeting to the Target: Candidates' Use of Strategic Messages and @Mentions on Twitter." *Journal of Information Technology & Politics* 15 (1).

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. "Long short-term memory." *Neural Computation* 9 (8).

Joulin, Armand, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. "Fasttext. zip: Compressing text classification models." *arXiv preprint arXiv:1612.03651.*

Kalchbrenner, Nal, Edward Grefenstette, and Phil Blunsom. 2014. "A convolutional neural network for modelling sentences." *arXiv preprint arXiv:1404.2188.*

Kostyuk, Nadiya, and Yuri M. Zhukov. 2019. "Invisible Digital Front: Can Cyber Attacks Shape Battlefield Events?" *Journal of Conflict Resolution* 63 (2).

Larsen, Erik Gahner, and Zoltán Fazekas. 2020. "Transforming Stability into Change: How the Media Select and Report Opinion Polls." *The International Journal of Press/Politics* 25 (1).

Lörcher, Ines, and Monika Taddicken. 2017. "Discussing climate change online. Topics and perceptions in online climate change communication in different online public arenas." *Journal of Science Communication* 16 (2).

Michener, Greg. 2011. "FOI Laws Around the World." *Journal of Democracy* 22 (2).

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. "Distributed representations of words and phrases and their compositionality." In *Advances in neural information processing systems.*

Minhas, Shahryar, Jay Ulfelder, and Michael D. Ward. 2015. "Mining Texts to Efficiently Generate Global Data on Political Regime Types." *Research and Politics.*

Mitts, Tamar. 2019. "From Isolation to Radicalization: Anti-Muslim Hostility and Support for ISIS in the West." *American Political Science Review* 113 (1).

Pennington, Jeffrey, Richard Socher, and Christopher D Manning. 2014. "Glove: Global vectors for word representation." In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP).*

Peters, Matthew E, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. "Deep contextualized word representations." *arXiv preprint arXiv:1802.05365.*

Pinto, Juliet G. 2009. "Transparency Policy Initiatives in Latin America: Understanding Policy Outcomes from an Institutional Perspective." *Communication Law and Policy* 14 (1).

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. "Language models are unsupervised multitask learners." *OpenAI blog* 1 (8).

Scharkow, Michael. 2013. "Thematic content analysis using supervised machine learning: An empirical evaluation using German online news." *Quality & Quantity* 47.

Smith, Christopher, and Yaochu Jin. 2014. "Evolutionary multi-objective generation of recurrent neural network ensembles for time series prediction." *Neurocomputing* 143.

Xu, Donna, Yaxin Shi, Ivor W Tsang, Yew-Soon Ong, Chen Gong, and Xiaobo Shen. 2019. "Survey on Multi-Output Learning." *IEEE transactions on neural networks and learning systems.*

Zhukov, Yuri M. 2016. "Trading Hard Hats for Combat Helmets: The Economics of Rebellion in Eastern Ukraine." *Journal of Comparative Economics* 44 (1).