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

Abstract

Political scientists increasingly use supervised machine learning to code multiple relevant labels from a single set of texts. The current "best practice" of individually applying supervised machine learning to each label ignores information on inter-label association(s), and is likely to under-perform as a result. We introduce multi-label prediction as a solution to this problem. After reviewing the multi-label prediction framework, we apply it to code multiple features of (i) access to information requests made to the Mexican government and (ii) country-year human rights reports. We find that multi-label prediction outperforms standard supervised learning approaches, even in instances where the correlations among one's multiple labels are low.

1 Introduction

Supervised machine learning has dramatically expanded researchers' abilities to measure and classify important concepts from political texts (e.g., Laver et al. 2003; Mitts 2019; Greene et al. 2019). Recent methodological innovations have likewise served to further tailor these methods to the needs of political scientists (e.g., Cantú and Saiegh 2011; D'Orazio et al. 2014; Chang and Masterson 2020; Miller et al. 2020). Despite these advancements, political scientists continue to primarily apply supervised machine learning to political texts in an independent manner for each target variable considered. Doing so treats each target variable as an unrelated quantities of interest during supervised classification. This standard, independent classification approach is consistent with the supervised machine learning frameworks discussed in past political science reviews of automated text analysis (Grimmer and Stewart 2013; Barberá et al. 2020).

However, political scientists also commonly endeavor to code multiple separate target variables from a single corpus of text, often with a future intention of using said measures as explanatory and/or outcome variables. For instance, Mitts (2019) uses a supervised approach to independently classify 175,015 tweets across four non-mutually exclusive labels: (1) sympathy for ISIS, (2) life in ISIS territories, travel to Syria, or foreign fighters, (3) the Syrian War, or (4) anti-West rhetoric, and *then* analyzes each as a distinct dependent variable in four separate statistical models. Kostyuk and Zhukov (2019) likewise use supervised classification to separately code political event attributes pertaining to event type, initiator/target, tactics, and casualties from 72,010 news reports and blog posts. These attributes are then leveraged to create a measure of Ukrainian kinetic operations, whose effects on a (separately coded) cyber-warfare variable are considered via vector autoregression. Appendix B offers 10 similar published examples.

In these contexts, substantial gains in classification accuracy—and thus also in variable measurement and any subsequent regression-based inferences—can be obtained by treating one's target variables as interdependent, and leveraging each variable's supervised predictions as features during the supervised classification of all other variables. We introduce one such supervised machine learning framework here: multi-label prediction.

Multi-label prediction offers substantial benefits over even the most flexible of independent classification alternatives, in that the former uniquely leverages pertinent auxiliary information that is already freely available—via the additional label-to-feature relations across one's remaining labels—during classification. In doing so, multi-label prediction avoids the significant training costs associated with searching over a potentially infinite set of combinations and/or transformations of one's original document features to arrive at this same on-hand information for classification. To illustrate these points below, we first formally present the multi-label framework, review a number of pertinent caveats and extensions, and introduce metrics to judge multi-label classification performance. Our main contribution is to introduce these items alongside guidance for applied text-as-data research. However, we also find that—as in other domains (Madjarov et al. 2012)—a(n ensemble) classifier chain multi-label approach has the best predictive performance over the largest number of relevant metrics and across a variety of types of text-as-data. We then verify that the predictions from this ensemble classifier chain multi-label approach also provides preferable (outcome and/or explanatory) variable measures for post-classification regression analyses, when compared to independent classification.

To arrive at these findings, we first illustrate our proposed multi-label approach's broader benefits through an application to the coding of access to information (ATI) request texts—a form of "big data" from citizen-government interactions with which political science research has engaged (e.g., Chen et al. 2016; Berliner et al. 2018; Berliner et al. 2020). Our second application then evaluates multi-target prediction within the context of the growing literature on the automated coding of human rights abuses from country-year human rights reports (Greene et al. 2019; Murdie et al. 2020; Park et al. 2020a). In both applications, and an accompanying Monte Carlo simulation, multi-label prediction outperforms several alternative, independent supervised classification approaches—even in circumstances of low correlation among target variables. Our applications and simulations further demonstrate that multi-label prediction performs much better than independent classification in instances where (i) correlations among target variables are high and/or (ii) the number of available features or labeled cases for classification is relatively low.

2 Multi-label Framework

This section begins by reviewing a set of key components to single variable classification, before generalizing this to instances where researchers seek to classify multiple variables (i.e., multi-label contexts). For supervised machine codings of text-as-data, we can define the main objective of classification algorithms as separating the classes of a variable using only human-coded training data. Ideally in such contexts, a model learns the underlying structure of a variable using said training data, and this structure then generalizes well to unseen (i.e., out-of-sample) data. If the target variable has only two possible values—for example, if the goal is to predict if a person will vote or not—we refer to this task hereafter as "binary classification." On the other hand, if there are more than two possible classes for a single target variable, we term this variable as "nominal" and characterize this task as "multi-class classification." The latter would include, e.g., efforts to classify candidate vote choice in a multi-party system with more than two parties.

More formally, let $\mathcal{X}$ denote the input space and $\mathcal{Y}$ the output space (target). The goal is to learn a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ that maps an instance from the input space to the output space. This function is learned from the training set $\{x_i, y_i \mid 1 \leq i \leq m\}$, where $x_i \in \mathcal{X}$ represents the features of an instance that will be mapped to a corresponding class (or label) $y_i \in \mathcal{Y}$ representing its characteristics. One fundamental assumption adopted by traditional classification algorithms is that each class is mutually exclusive. These are valid assumptions for the individual binary and categorical vote choice variables mentioned above.

However, as noted in the introduction, there are many learning tasks for which these simplifying assumption might not be reasonable. These situations commonly arise in researchers' efforts to code multiple, non-mutually exclusive traits—which we define as labels[1]—from corpora of political texts, as the examples from Mitts (2019) and Kostyuk and Zhukov (2019) illustrate above. Alternatively, for the aforementioned vote choice

[1] Whereas, we reserve "class" for only mutually-exclusive traits.

examples, a similar situation would arise in cases where a researcher is interested in predicting how a voter will vote across (e.g.,) six distinct ballot initiatives during a given election. As a third example, we note that more recent innovations in semantic role labeling can often require classification of an even larger (but more incomplete) number of overlapping labels per text-unit than the examples described thus far. In any of the above cases, a set of labels must ideally be assigned to each observation in order to express its *non-exclusive* characteristics in a manner that accounts for the mapping of multiple labels to a single observation (e.g., an individual voter). This assignment paradigm is referred to hereafter as "multi-label learning," whereby the goal of one's classification task becomes learning a function that can predict the proper label *sets* for unseen examples (Zhang and Zhou 2013).

For multi-label problems, political scientists continue to use independent label prediction—as illustrated by the examples highlighted in our introduction and Appendix. We refer to this standard approach as "binary relevance" (BR) hereafter. BR effectively decomposes multi-label problems into *multiple* independent binary label prediction tasks. In breaking one's classification tasks into a set of wholly independent, binary classification tasks, BR directly invokes the simplifying assumptions mentioned above. However, because any and all relationships between labels are accordingly ignored, BR will often achieve suboptimal performance in instances where this simplifying assumption does not hold. This is a substantial limitation, given that (as elaborated upon below) the effective exploitation of label correlations (i) is essential for accurate multi-label classification and (ii) is the main way to cope with the challenge of large output spaces that multi-label problems typically entail (Zhang and Zhang 2010).

As such, multi-label learning tasks require a distinct classification strategy from that which is used for the assignment of a single label (per observation) within the BR or multi-class classification paradigms defined above. Ignoring these multi-label attributes and classifying all labels independently likely to yields worse classification performance, and hence poorer measurement, of said variables. Depending on how researchers use these classified variables in subsequent statistical models, measurement shortcomings from independent classification could lead to biased and/or unreliable inferences. Indeed, to the extent that the latter classification approach leads to higher measurement error in one's post-classification independent (dependent) variable(s), inconsistency (inefficiency) in one's regression estimates—and a higher corresponding risk of bias and incorrect inferences when said measurement error arises in *either* one's independent *or* variable(s)—can arise (Wansbeek and Meijer 2000; deHaan et al. 2019).

Multi-label learning algorithms were developed to address these types of concerns in the context of news story and web-page categorization (McCallum 1999; Ueda and Saito 2003). In both usage cases, texts are classified into multiple non-mutually exclusive categories. After being successfully applied to problems involving text, multi-label algorithms have been widely used on diverse other tasks, such as automatic annotation of images and videos (Qi et al. 2007; Boutell et al. 2004), bioinformatics (Clare and King 2001), and web mining (Tang et al. 2009).

2.1 Formal definition

We now turn to formally define multi-label classification. For reference, we provide a summary of the mathematical notation used within this section in Appendix Table C.1. The multi-label paradigm can be more formally defined as follows: let $\mathcal{X} \in \mathbb{R}^d$ be a d-dimensional input space and the label space with q possible classes be $\mathcal{Y} = \{0, 1\}^q$. Similar to the standard classification task defined above, the goal is to learn a mapping function—in this case, defined as $h : \mathcal{X} \to \mathcal{Y}$—from the training set $\mathcal{D} = \{x_i, Y_i \mid 1 \leq i \leq m\}$. For each instance (x_i, Y_i), the input $x_i = (x_{i1}, ..., x_{id})$ is a d-dimensional vector and the output is Y_i is a q-dimensional set of labels. Therefore, for each new unseen example of $x \in \mathcal{X}$, the multi-label classifier $h(\cdot)$ predicts a set of labels $h(x) \subseteq \mathcal{Y}$.

For supervised text-as-data classification problems, the output of a multi-label model can thereby be seen as a real-valued function $f : \mathcal{X}$ x $\mathcal{Y} \to \mathbb{R}$, where $f(x, y)$ can be interpreted as a compatibility or confidence function that evaluated how compatible or likely $y \in \mathcal{Y}$ is the correct label of x. The classifier function $h(\cdot)$ can be obtained by taking the output with largest compatibility score $h(x) = \underset{y \in \mathcal{Y}}{arg\ max} f(x, y)$ or by using a thresholding function $t : \mathcal{X} \to \mathbb{R}$ such that $h(x) = \{y | f(x, y) \geq t(x), y \in \mathcal{Y}\}$.

Based on the formal definition presented above, it is evident that traditional supervised classification problems can be viewed as a simplified version of multi-label learning, where each target has only one (binary or nominal) label. The generality of multi-label classification makes this task much more difficult to solve, given that each directional inter-label relationship must be accounted for as a predictive feature in a manner that accounts for permutations. That is, in order to accommodate inter-label relationships, one ideally needs to not only apply a classifier once to each label (as is presently done within political science), but rather to classify each label once for every possible ordering (i.e., subset) of all remaining labels—since these remaining labels, and the order by which they themselves are classified, now have bearing on each label's subsequent prediction. In this regard, the key challenge of multi-label classification is thus the output space, which grows exponentially as the number of labels increase. For example, a problem with 5 binary labels has 32 different label subsets. If we increase this number to 15 binary labels, the number of possible combinations grows to 32,768.

To address these shortcomings, researchers have identified several ways to efficiently leverage the relations among labels within multi-label classification tasks. Two longstanding strategies include accounting for (i) pairwise correlations between any two labels (Qi et al. 2007; Ueda and Saito 2003) or (ii) rankings between relevant and irrelevant labels (Brinker et al. 2006; Elisseeff and Weston 2002). In comparison to BR, these alternative approaches better manage the trade-off between performance and computational cost within multi-label classification tasks.

However, these approaches encounter problems when the relations among labels becomes more complex than simple pairwise associations—which is oftentimes the case for real-world social science data. This is especially the case for many efforts to code quantities of interest from political texts. For example, with regards to the country-year human rights application presented below, CIRI's separate, non-mutually exclusive human-rights labels for state torture, political imprisonment, extrajudicial killings, and disappearances (Cingranelli and Richards 2010) are likely to be highly interdependent facets of states' overarching strategies of repression, as opposed to simply being linked via pairwise associations. In such cases, multi-label alternatives that accommodate the influences of all labels when predicting each label (Yan et al. 2007; Cheng and Hüllermeier 2009; Ji et al. 2008), can achieve superior performance, albeit at the expense of higher computational costs and more constrained scalability.

2.2 Categorization of Multi-label learning algorithms

Methods for multi-label classification can be divided into two general categories: problem transformation methods and algorithm adaptation methods. The former tackles the multi-label classification problem by reformulating this classification problem into other tasks, such as binary classification (Read et al. 2009) or multi-class classification (Tsoumakas and Vlahavas 2007). On the other hand, algorithm adaptation methods tackle the multi-label problem by adapting learning algorithms to directly deal with multi-labeled data (Zhang and Zhou 2007), oftentimes in manners that better account for the label associations in one's data. We first discuss algorithm adaptation below, before returning to problem transformation, and then a broader summary of all approaches.

2.2.1 Algorithm Adaptation

By tailoring existing algorithms to multi-label contexts, algorithm adaptation methods possess an inherent appeal in that they (i) often employ algorithms that are already familiar to researchers from single label contexts and (ii) most closely match the underlying data generating process (d.g.p) of one's multi-label data. Yet, such methods typically must sacrifice at least some ability to explicitly, and flexibly, accommodate inter-label correlations to achieve these aims. In multi-label contexts, this often leaves algorithm adaptation methods open to the same critiques that were previously highlighted for BR.

A canonical algorithm adaption method is the Multi-label k-Nearest Neighbor (ML-kNN) algorithm (Zhang and Zhou 2007). As the name suggests, this algorithm adaptation model is itself built upon the more widely known k-Nearest Neighbor (kNN) algorithm (Dudani 1976). In all (ML-)kNN approaches—and for each datapoint in a test set—the model identifies its k-nearest neighbors in the training set. Whereas standard kNN then assigns a label for a single trait to that datapoint based upon the most common label shared by that datapoint's k-nearest neighbors, ML-kNN instead considers *the set* of trait labels for each datapoint based upon a membership counting vector of its k-nearest neighbors' corresponding label-sets. Using the statistical information gained from these neighbors' label sets, all labels are then assigned to that datapoint via Bayesian inference.

Under this Bayesian framework, prior and posterior probabilities can then be directly estimated from a human-labeled text-as-data training set based on frequency counting.[2] The process of estimating prior probabilities for each label can also help to mitigate problems commonly faced by text-as-data researchers such as class-imbalance. This algorithm has been used in several real-world multi-label learning problems—in each case outperforming other multi-label learning algorithms that were considered at the time (Zhang and Zhou 2007).

However, one of ML-kNN's main limitations is that it does not explicitly consider the correlation between labels. As such, ML-kNN discards relevant information, and and accordingly risks assigning labels with sub-optimal accuracy rates that are no better than the BR approach outlined above. Several extensions have been proposed to address this potential deficiency. Examples include extensions that (i) incorporate all of the components of the counting vector C_j in the (non)assignment of label j (Younes et al. 2011) or (ii) consider the labels of neighbored instances as "features" of a logistic regression whose output is the label is to be estimated (Cheng and Hüllermeier 2009). Nevertheless, the applicability of this algorithm adaptation framework in a manner that fully accommodates correlations between labels remains limited. This leads us to tentatively favor problem transformation methods, to which we now turn.

2.2.2 Problem Transformation

Rather than tailoring a multi-label algorithm to an existing multi-label data structure, problem transformation methods first "break up" one's multiple labels—including nominal labels—into a simpler set of (often binary) labels. The latter methods then leverage these restructured data in a manner that more explicitly accounts for inter-label correlations. As such, problem transformation methods sacrifice information on the underlying d.g.p, and original structure, of one's multi-label data so as to better accommodate label

[2] More details can be found in Zhang and Zhou (2007).

correlations during classification—and the added computation that this often entails.

One of the most well-known and accessible problem transformation algorithms is the Classifier Chain (CC; Read et al. 2009). In a similar manner to strategies of multiple imputation by chained equations (**vanbuurenFlexibleImputationMissing2012**), the primary goal of the CC model is to transform the multi-label problem into a chain of binary classifiers, where the prediction of subsequent classifiers is based on the prediction of preceding elements of the chain. See Appendix C.1. for a more formal treatment.

The CC method does well at balancing predictive power and computational efficiency. Accordingly, it has now been successfully applied to domains as varied as music, scene, yeast, genbase and medical classifications (Madjarov et al. 2012). It is important to highlight, however, that the the ordering of the labels considered affects the CC algorithm's performance, so it is often necessary to run the model with several random permutations over the label space with and without replacement. To address this, we propose CC extensions that utilize ensemble methods within our applications further below, which we label as "Ensemble CC" (ECC) hereafter. This proposed innovation draws upon earlier CC extensions that have previously sought to optimize label ordering via (e.g.) genetic algorithms (Goncalves et al. 2013) or Monte Carlo methods (Read et al. 2014).

The multi-label problem can also be modeled as an ensemble of multi-class classifiers, where each component in the ensemble targets a random subset of the label space $\mathcal{Y}$ upon which a multi-class classifier is induced by what is known as the Label Powerset (LP; Boutell et al. 2004).[3]

This LP algorithm has been successfully applied to image classification, where it achieved commensurate performance (Boutell et al. 2004). However, it is important to note that LP has two major limitations. First, the prediction of new labels is limited by label sets that appear in the training set. That is, the model is not able to generalize to unseen combinations of labels. Secondly, the label space grows exponentially (2^q), so when $\mathcal{Y}$ is large, training becomes complex and computationally expensive.

[3] In other words, each distinct set (combination) of labels is mapped to a new class. For a more formal treatment, see Appendix C.2.

To overcome these two drawbacks, the LP algorithm can be extended under a random k-labelsets (RAkEL) framework (Tsoumakas and Vlahavas 2007). This extension's main innovation lies in its use of N different LP classifiers on k different random subsets of one's label space to guarantee computational efficiency. The approach then ensembles these N LP classifiers for a final prediction. The degree of label correlations is accordingly controlled for by k. For unseen examples, each of the N different classifiers predict their corresponding labels. The final output is determined by the ensemble of all N classifiers.

As an illustration, imagine a four-label classification task, $y = [L_1, L_2, L_3, L_4]$, with two training examples $y_1 = [0, 0, 1, 1]$ and $y_2 = [1, 0, 0, 0]$. With LP, the output of the model would be restricted to predicting either $[0, 0, 1, 1]$ or $[1, 0, 0, 0]$ as these are examples already seen during training. On the other hand, RAkEL allows the model to generalize to combinations of labels that are not present in the training set. For instance, for a RAkEL classifier with $k = 2$ that divides the label space in $k_1 = [L_1L_2]$ and $k_2 = [L_3L_4]$, a classifier N_1 is trained on the subset $k_1 = [0, 0], [1, 0]$ and another classifier N_2 is trained on the subset $k_2 = [1, 1], [0, 0]$. The final output is assembled for all combinations seen by N classifiers. In this case, the model is able to predict [0,0,1,1],[1,0,0,0],[0,0,0,0] and [1,0,1,1], wherein the last two examples were not present in the training set.

RAkEL has achieved good performance in multi-label domains involving document, image and protein classification (Tawiah and Sheng 2013). There are two types of RAkEL models: one considers only disjoint (non-overlapping) subsets (RAkEL$_d$) and a second that considers overlapping intervals (RAkEL$_o$). In our applications, we consider RAkEL$_d$.

2.3 Summary of Approaches

We provide (i) an overview of the computational costs associated with each multi-label approach in Appendix D and (ii) a summary of our (tentatively favored) problem transformation approaches, alongside the earlier-described BR approach, in Figure 1. Since these latter methods are based on transforming the multi-label problem into binary or multi-class classification, the training procedure is identical to standard supervised learning algorithms.[4] Likewise, these methods' flexibility in incorporating different base

[4] Consequently, the models are trained using cross-entropy as a loss function below.

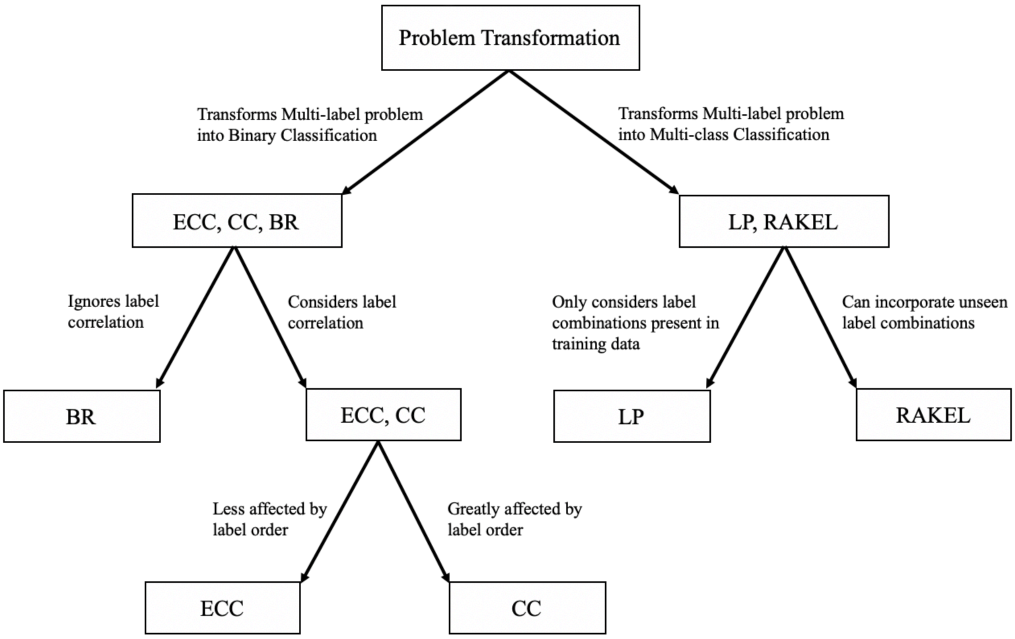
Problem Transformation
Transforms Multi-label problem into Binary Classification
Transforms Multi-label problem into Multi-class Classification
ECC, CC, BR
LP, RAKEL
Ignores label correlation
Considers label correlation
Only considers label combinations present in training data
Can incorporate unseen label combinations
BR
ECC, CC
LP
RAKEL
Less affected by label order
Greatly affected by label order
ECC
CC

Figure 1: Relationships Among Problem Transformation Approaches

classifiers of one's choosing helps to ensure comparable underlying interpretability relative to BR on this dimension. That being said, ECC, LP, and RAkEL's reliance on varying degrees of ensembling does raise practical challenges for these three approaches' interpretability, relative to simpler BR frameworks. Given that multi-label text-as-data problems are primarily oriented towards the measurement and accurate prediction of labels for future use—rather than explanation—this tradeoff in improved accuracy for some loss in interpretability is preferable for many researchers.

With the above caveats in mind, we note that many of the multi-label approaches reviewed above have exhibited good-to-excellent performance in text-as-data contexts in past comparisons of multi-label methods (Madjarov et al. 2012). Yet, in terms of relative performance, none of these multi-label methods has emerged as consistently superior to the others.[5] In light of this, our empirical applications compare the performance of a wide array of multi-label and standard supervised classification approaches. However, in order to do so, special consideration must first be given to choices of comparison metrics, given the multi-label context being considered. We hence now turn to discussing model comparison in multi-label contexts, before turning to our applications in full.

3 Model comparison

In single-label learning systems, performance is often evaluated by conventional metrics such as F-score, precision, recall, area under the curve (AUC) and accuracy. However, the evaluation of multi-label models is much more complex as each observation is associated with several labels simultaneously. Rather than simply denoting whether a prediction for a given observation is right or wrong, the latter quality implies that one needs to also evaluate (and hence aggregate over, in some manner) the share of correct labels predicted in multi-label contexts.

The multi-label evaluation metrics that have been designed for these purposes can be divided in three general categories: *example-based*, *label-based* and *ranking-based* metrics (Madjarov et al. 2012). Example-based metrics evaluate average differences between one's model prediction sets and the true label set of one's evaluation dataset. On the other hand, label-based metrics assess the predictive performance for each label separately and then average the performance over all labels. Finally, ranking-based metrics use the fraction of label pairs that are incorrectly ordered to evaluate the model. As multi-label metrics may be unfamiliar to some political scientists, we provide a summary table of each metric used in this paper in Table 1, and then formally present each metric below.

[5] Despite the impressive performance of deep neural networks (DNNs) in several areas, DNNs have not shown dominant cross-domain performance when compared to classic multi-label models. That being said, Xu et al. (2019) do present a selection of high performing DNNs for hierarchical (Baker and Korhonen 2017) and time-series (Smith and Jin 2014) multi-output learning contexts. Appendix Section E.5 accordingly provides a comparison of our baseline and primary multi-label approaches to two DNN approaches.

Metric	Type	Summary	Best Performance Value
Hamming Loss	Example-based	Computes the percentage of labels that were misclassified.	0
Subset Accuracy	Example-based	Computes the percentage of instances/examples that had all of their labels classified correctly.	1
F1-Macro	Label-based	Calculates the F1-score for each label independently, then averages them. All labels are treated equally.	1
F1-Micro	Label-based	All labels are aggregated before calculating the F1-score, making F-Micro more ideal for problems with class imbalance.	1
Ranking Loss	Ranking-based	Calculates the average number of incorrectly ordered label pairs.	0

Table 1: Multi-label Evaluation Metrics

3.1 Example-based metrics

Hamming loss is an example-based metric that computes the fraction of misclassified labels for each observation. It considers both prediction errors (when the prediction is incorrect) and omission errors (when the label is not predicted at all), where lower values are more optimal. For example, a hamming loss of 10% means that 90% of all labels were classified correctly. It can be formally defined as:

$$L_{Hamming}(Y,\hat{Y}) = \frac{1}{n_{\text{labels}}} \sum_{k=1}^{n_{\text{labels}}} \mathbf{1}(\hat{y}_k \neq y_k) \tag{1}$$

where n_{labels} is the total number of labels, $\mathbf{1}$ is the indicator function, $\hat{y}$ is the *k-th* predicted label and y is the actual label.

Subset accuracy is a stricter metric than hamming loss. Instead of evaluating the fraction of correctly classified labels, subset accuracy only considers a prediction as correct if **all** of an observation's predicted labels are identical to its true label set:

$$SubSet_{Acc}(Y,\hat{Y}) = \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}(\hat{Y}_i = Y_i) \tag{2}$$

where m is the total number of examples in the test set, $\mathbf{1}$ is the indicator function, $\hat{Y}_i$ is the *i-th* predicted label set, and Y_i is the ground-truth label. This metric can thus be interpreted as reporting the percentage of all observations that have all labels correctly classified, with higher values on this metric being more optimal.

3.2 Label-based metrics

For label-based metrics, we consider Macro- and Micro-F1 scores. Each is based on precision and recall.[6] Precision indicates the proportion of predicted positives that are truly positive while recall denotes the proportion of actual positives that are classified correctly. Macro-F1 is the harmonic mean of the precision and recall averaged across all labels. The precision-macro (p_{macro}) and recall-macro (r_{macro}) are defined as follows:

$$p_{macro_j} = \frac{TP_j}{TP_j + FP_j} \tag{3}$$

$$r_{macro_j} = \frac{TP_j}{TP_j + FN_j} \tag{4}$$

where TP_j, FP_j, FN_j denote the true positive, false positive and false negative rate for the label j respectively and q is the number of labels for each example. Macro-F1 can be defined in terms of these metrics as

$$Macro - F1 = \frac{1}{q}\sum_{j=1}^{q} \frac{2\ p_{macro_j} \cdot r_{macro_j}}{p_{macro_j} + r_{macro_j}} \tag{5}$$

On the other hand, for Micro-F1 both precision and recall are defined differently:

$$p_{micro_j} = \frac{\sum_{j=1}^{q} TP_j}{\sum_{j=1}^{q} TP_j + \sum_{j=1}^{q} FP_j} \tag{6}$$

$$r_{micro_j} = \frac{\sum_{j=1}^{q} TP_j}{\sum_{j=1}^{q} TP_j + \sum_{j=1}^{q} FN_j} \tag{7}$$

Finally, Micro-F1 is defined as follows:

$$Micro - F1 = \frac{2\ p_{micro_j} \cdot r_{micro_j}}{p_{micro_j} + r_{micro_j}} \tag{8}$$

Micro-F1 calculates the metrics by counting the total number of true positives, false

[6] As such, these particular metrics may not be feasible for applications where extremely imbalanced or incomplete labels preclude the calculation of precision and/or recall.

negatives, and false positives. That is, the scores of all labels are aggregated to compute the metric. On the other hand, Macro-F1 calculates the metrics for each label independently, and then averages them.

3.3 Ranking-based metrics

Ranking loss evaluates the fraction of label pairs that are incorrectly ordered. As such, it considers only the relative rankings (i.e., orderings) of one's label predictions—in terms of which labels exhibit higher versus lower predicted probabilities in that label set—and the correspondence between these rankings and a true label set. It can be defined as:

$$R_{loss} = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{|Y_i||\bar{Y_i}|}|\{(y', y''|f(x_i, y') \leq f(x_i, y''), (y', y'') \in Y_i \text{ x } \bar{Y_i}\}| \tag{9}$$

where $\bar{Y}$ is the complementary set of Y in $\mathcal{Y}$. In this case, ranking loss is interpreted such that the lower the ranking loss, the better the performance. In a similar manner to AUC in single-label prediction contexts, one strength of ranking loss is its reliance on relative orderings of label predictions rather than on arbitrary thresholds (e.g., 0.5). This ensures that predictive evaluations via ranking loss will be less sensitive to factors that lead to consistently (high or low) predictions relative to a chosen threshold.[7]

To illustrate this, imagine a set of true labels $Y1_{true} = [1, 0, 0], Y2_{true} = [1, 1, 0]$ and a corresponding set of label predictions given by a chosen classifier of $f1_{pred} = [0.4, 0.1, 0.2], f2_{pred} = [0.9, 0.8, 0.6]$. Using a threshold rule of 0.5 (1 if $f(\cdot) \geq 0.5$, 0 otherwise), we would obtain a relatively uninformative hamming loss of 33% and subset accuracy of 0%, as $Y1_{pred} = [0, 0, 0], Y2_{pred} = [1, 1, 1]$. On the other hand, the resulting ranking loss would be zero (i.e., perfect).

4 Applications

4.1 Mexican ATI Requests

Our first application examines access-to-information (ATI) requests made to the Mexican federal government during the period 2003-2015. ATI requests in this context have

[7] E.g., such as class-imbalance or one's choice of base classifier.

been previously analyzed in efforts to assess the degree to which Mexican citizens use this ATI system to hold their government publicly accountable (Berliner et al. 2018), or to understand government responsiveness (Almanzar et al. 2018; Berliner et al. 2020). We provide additional background on Mexico's ATI system in the Appendix. The textual content of information requests contains requesters' open-ended descriptions of their desired information, as entered into an online ATI request system's primary request field, supplemental information field, and attachments field. Attachments were webscraped, converted to machine readable text via optical character recognition, and then combined with other request text fields to form our primary textual entries of interest. Altogether, this process produced a sample of 1,025,953 requests for our consideration.

We are interested in a wide variety of traits associated with each of these request texts, pertaining to qualities such as the use of legalistic and technical language, the number of distinct pieces of information requested, and the appropriateness of the request for the targeted agency. Developing accurate and fine-grained measures of attributes like these will enhance understandings of the nature and dynamics of citizen demand for information, and of the request-specific determinants of government responsiveness—both in general and as it varies across agencies and time. We accordingly drew a random sample of 4,925 requests—stratified by year—from our full sample of request texts. Six Mexico City-based coders coded these request texts for distinct ATI-request traits. For this paper, we retained 26 total (ATI-request trait) labels for classification. Our original request traits—and overall human coding approach—are discussed in the Appendix.

With this sample of 4,925 human coded requests, we then trained all classifiers on this sample, classifying all 1,021,028 remaining (non-human coded) ATI requests for each of our 26 labels. All relevant text processing steps that were applied to the raw texts prior to classification—and the hyperparameters used for each model—are described in the Appendix. The overall level of correlation for our 26 labels is not high, suggesting that this application is a "hard test" for the potential benefits of multi-label classification. On average, the correlation between our pairs of hand-coded labels is 0.06 with the lowest and highest pairwise correlations being 0 and 0.40, respectively.

We next evaluate the value added of the multi-label framework (i.e., of considering inter-label relations for our document-level labels). In order to do so, we compare the results obtained from four plausible approaches to handling multi-label data. The first pair of general approaches that we consider are Binary Relevance (BR) and MLkNN. Recall that neither of these two approaches consider label relations. By contrast, the second pair of general multi-label approaches that we consider (CC and LP) *do* consider label correlations.

Within each of these general approaches, we then implement and consider several of the extensions described further above. Turning first to the CC approach, we consider its basic implementation and also a version where the label ordering was permuted. For the latter CC approach, the final output was composed of the average of each permuted chain, and is hereafter labeled ECC for "Ensemble CC."[8] Similarly, for the LP approach, we considered both its standard definition and the RAkEL extension. The base classifier for each of these methods was a standard logistic regression.

For the MLkNN, we only considered the standard version since its proposed extensions are not yet readily available. On the other hand, we evaluate four different varieties of BR classifiers. The first, presented as Standard BR, used a simple Random Forest with the same parameters applied to all labels. For the second BR variant, we perform a grid search to find the best performing classifier and corresponding hyperparameters for each label. This approach is labeled as BR Optimized below.

For the third BR variant, BR Optimized Threshold, we kept the same classifier for all labels. However, instead of using the standard 0.5 classification threshold, we optimized the threshold for each label by selecting 20 different splits of the training data and selecting the threshold value for each label that maximized the F1-score. Finally, to more directly address class-imbalance in some labels, we used an oversampling technique to generate synthetic samples from the minority class, known as SMOTE (Lusa et al. 2013). Using this technique, we trained the final BR classifier that we consider, hereafter referred

[8] As this algorithm was not present in Python's scikit-multilearn package (Szymański and Kajdanowicz 2017), we have made its implementation available in our repository.

to as BR Optimized SMOTE.

The multi-label results obtained from all classifiers considered are summarized in Table 2. The algorithms used were based on the implementations provided by Python's `scikit-multilearn` package (Szymański and Kajdanowicz 2017) and `sklearn` (Pedregosa et al. 2011)[9]. Each model was evaluated using 80% of the data for training and 20% for testing with 10 different splits. Further details on hyperparameter selection for each algorithm considered appear in Table E.2 of the Appendix.

Turning to Table 2, we find that ECC achieves the best performance among all tested classifiers. That is, we can determine from Table 2 that ECC routinely outperforms our alternate approaches, and can also observe that ECC is a top three performer across all metrics used. This finding supports our contentions regarding the importance of taking into account relationships between labels for multi-label classification of political text-as-data. The complete results for all metrics considered here can be found in Table E.1 of the Appendix and are consistent with this summary interpretation. These results suggest that ECC also exhibits the lowest average deviation when compared to other algorithms. Additional discussion of the predicted proportions for each label is likewise presented in the Appendix.

For the four BR classifiers considered in Table 2, the optimized version achieved the lowest hamming loss whereas the SMOTE version achieved the highest F1-macro. The latter finding underscores our earlier contentions regarding F1-macro being a poor metric for class imbalanced data. Regarding the former finding, we can note that each of the classifiers used in the BR optimized-component of this application was selected based upon the accuracy score for its corresponding label individually. Therefore, these BR classifiers were optimized by minimizing the hamming loss. Regarding the remaining classifiers in Tables 2 and E.1, we can further observe in this case that the results for MLkNN[10] are remarkably worse than all other models.

We can thus conclude that multi-label approaches that give careful consideration to

[9] See the `ml3` library (Probst et al. 2017) for multi-label methods in **R**.

[10] Recall that MLkNN, like our BR approaches, does not consider label relations.

Algorithm	**Subset Accuracy**	**Hamming Loss**	**Ranking Loss**	**F1-micro**	**F1-macro**
Classifier Chain (CC)	✓				
Ensemble CC (ECC)	✓	✓	✓	✓	✓
RAkEL$_d$, k = q/4					
RAkEL$_d$, k = q/2					
Label Powerset	✓				
Binary Relevance (BR)		✓	✓		
BR Optimized		✓			
BR Optimized SMOTE				✓	✓
BR Optimized Thresholds			✓	✓	✓
MLkNN					

Table 2: Results for 10 different data splits. The top three performers for each metric are highlighted. Full details appear in Table E.1 of the Appendix.

the relationships between one's labels tend to outperform less label-aware approaches that are more common in the political science literature. To this end, our CC approaches tended to exhibit higher subset accuracy than BR and comparable results for other metrics. This finding, and those outlined above, suggest that multi-label classification allows researchers to achieve superior labels within multi-label classification tasks, even in contexts where the correlation between these multiple labels is relatively low. This provides strong support for the multi-label approach in a real-world text-as-data context. Our next application offers researchers further guidance on *when*, specifically, multi-label methods are likely to be more or less effective.

4.2 Country-Year Human Rights Practices

Our second application allows us to evaluate the benefits of using multi-label algorithms in the context of a separate political science text-as-data domain. For this application, we specifically compare the best performing multi-label algorithm from our first application (ECC) to BR using a set of extant human rights texts and labels. Our human rights data include a combination of (i) country-year textual reports on human rights practices and (ii) overlapping labels of states' annual human rights practices, as human-labeled from these same texts by the CIRI data project (Cingranelli and Richards 2010). As noted in the introduction, automating the measurement of human rights abuses from the human rights texts is an area of growing interest in political science (Fariss et al. 2015; Greene et al. 2019; Murdie et al. 2020; Park et al. 2020a, 2020b). Our application of multi-label methods to these data thus stands to advance the state of the art for these automated endeavors.

The text-as-data component to this application is a corpus of US State Department Country Reports on Human Rights Practices for the years 1981 to 2011.[11] Each document corresponds to the State Department's assessment of a particular country's annual domestic human rights practices during the previous calendar year. We match these reports to 14 indicators, from the CIRI project, of specific types of human rights violations and practices at the country-year level. These indicators were human-coded by the CIRI project from the US State Department texts outlined above, and encompass categories of human rights abuse that range from targeted killings and torture to violations of women's political rights and freedom of speech. To simplify interpretation, we (i) dichotomize each (originally ordinal) CIRI variable and (ii) then reverse-code each dichotomized variable such that higher values imply worse (as opposed to better) human rights performance. We provide further details on these CIRI variables in the Appendix F.

In total, there are 4,756 reports, each with 14 binary indicators for (country-year) violations of different human rights wherein a 0 denotes "no violations" of a given violation type and 1 denotes a human rights violation of a certain type. As noted above—and unlike our first application—these labels were directly drawn from an existing human-coded country-year dataset (i.e., CIRI), rather than from human-labeling of our own. These human rights labels are furthermore much more highly correlated than the previous ATI Requests application. On average, the correlation between each current label is 0.33, with a maximum value of 0.60 and a minimum 0.06.

With respect to the text features considered, we begin with the full document term matrix (DTM) of raw term counts from our human rights reports of interest. This DTM format is consistent with formats used by past text-as-data research into these same State Department human rights reports (Fariss et al. 2015), and in our case follows the preprocessing steps applied by Bagozzi and Berliner (2018) to a similar corpus of human rights reports. After implementing these preprocessing steps, our final DTM contains

[11] While these reports are available for later years, CIRI is only available through 2011.

2,445 unique stemmed unigrams as features.

In addition to these DTM-features, the ECC algorithm also allows us to incorporate the label correlation information between our CIRI human rights violation target measures within our multi-label classification tasks. We anticipate that the latter information will be highly relevant, given that—as mentioned in Section 2 above—many of the specific CIRI violations considered here (e.g., disappearances, killings, and torture) are likely to arise (and be correlated with one another) under a common strategy of state repression. We are hence interested in quantifying the added improvement that is gained by accounting for these interdependencies. Within these evaluations, this second application also (uniquely) varies the set of retained features that we use in classification, so as to assess whether our findings with respect to the ECC's added improvements vary in relation to the number of features that a given researcher has available for use in classification.

We thus evaluated the performance of ECC and binary relevance (BR) with respect to the classification of our 14 binary human rights violation categories *when including different subsets of DTM features, which are chosen at random.* We posit that if a classifier has enough information from its original features to adequately classify all target labels, there is probably very little to gain in leveraging the relationships between target labels during classification.[12] On the other hand, if there are insufficient features to properly classify all target labels, the additional information available to the researcher via any empirical correlations between each multi-label label will likely improve prediction significantly. This suggests that our multi-label approach will improve in relative performance over BR as the number of available features for classification declines.

We utilize our human rights data within a series of experiments to evaluate the above contentions. These experiments corresponded to our evaluation of ECC and BR model performance across 10 different data splits (80% for training, 20% for testing) for each feature level considered. The number of features ranged from all features available to as low as 20% of the total number of features drawn at random. In evaluating the ECC and BR approaches, we consider three base classifiers in each case: support vector machine

[12] In fact, doing so could even potentially reduce classification performance.

(SVM) with linear kernel, random forest, and logistic regression.

The results from these experiments are presented in Figure 2, considering five of the model performance criteria outlined above: subset accuracy, hamming loss, F1-macro, F1-micro and ranking loss. For all results plotted in Figure 2, BR and ECC base classifiers were chosen according to best overall performance. The best base classifier for BR was a SVM with linear kernel, whereas for the ECC the best performing base classifier was a random forest. Our conclusions are comparable when we standardize the base classifier across BR and ECC.

Examining the performance criteria in Figure 2, we can first observe that ECC routinely outperforms BR no matter the model performance metric or percentage of retained features considered. As above, this result provides strong support for the use of ECC (and hence multi-label methods) in contexts where a researcher is faced with multi-label data. Further, when only a small percentage of features are available (20%), the gap in performance between ECC and BR is the widest. For instance, this gap is 11.09% for subset accuracy with 20% of features available. This implies that the desirability of ECC over the more commonly used BR is especially pronounced when researchers are faced with a multi-label problem but have a limited number of features for prediction. As the number of available features increases, we find that the difference in performance between ECC and BR becomes less and less notable.

Hence, when features are abundant, the benefits of ECC become less salient, suggesting that researchers whose prediction tasks already include an exceptionally large number of (relevant) features may find BR to be sufficient. By contrast, researchers faced with a more limited set of features—in terms of total number, relevance, or related traits (e.g., sparsity)—are likely to especially benefit from using ECC or other multi-label methods. These trade-offs notwithstanding, as both applications suggest, multi-label methods typically exhibit advantages over BR approaches even in cases where one's feature set is relatively large and even when one's multi-label targets exhibit a relatively low level of correlation.

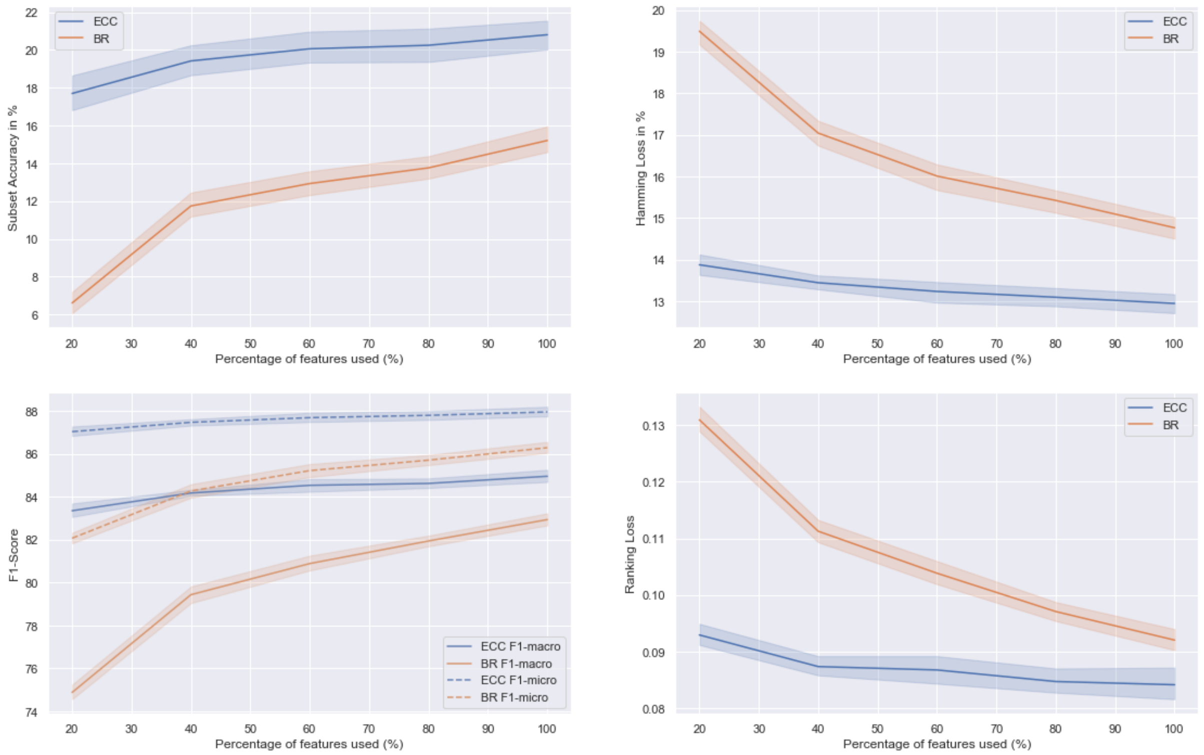

Figure 2: Performance of Binary Relevance (BR) and Ensemble Classifier Chain (ECC) for the proposed metrics. Shaded regions denote the standard deviation of the bootstrap splits. For the metrics reported in the left-hand panels, higher values imply better performance. For the metrics reported in the right-hand panels, lower values imply better performance.

4.3 Monte Carlo Simulations

The applications presented above assess the performance of a wide range of multi-label classifiers in two distinct empirical settings and across different levels of available features. However, these applications do not provide insight into the relative performance of these approaches under differing (i) scenarios of available training data and (ii) end-use cases for one's classified labels. Accordingly, our Appendix further compares our best performing multi-label approach (ECC) to BR across a series of Monte Carlo simulations.

We find that ECC outperforms BR for every classification metric considered—and especially for subset accuracy and hamming loss. Across three auxiliary regression setups, we then confirm in this context that ECC's superior classification performance also ensures that the parameter estimates obtained when subsequently using ECC's classified labels as regressors and/or regressands are superior in accuracy and coverage to those recovered by BR—and increasingly so as one's available training (test) data decrease (increase).

5 Conclusion

Supervised machine learning is now a commonly used means for coding measures from political and social texts. While such tools are typically applied independently to a single human-labeled variable of interest at a time, many political science projects now seek to code multiple, distinct human-labeled target variables from a single set of texts. We demonstrate that substantial gains can be made by recognizing this data structure, and by using multi-label prediction to leverage each target variable's predictions when predicting subsequent target variables. Given current trends in political text classification (Barberá et al. 2020; Chang and Masterson 2020; Miller et al. 2020)—and related trends in automated image and audio analyses (Dietrich et al. 2019; Williams et al. 2020; Torres and Cantú 2020)—the need for multi-label methods is only likely to grow in the future.

This paper has accordingly sought to introduce political scientists to multi-label prediction, and to highlight where and how it may benefit their own research. Our applications and simulations demonstrate that multi-label classification increasingly outperforms standard classification approaches (i) as the correlation across one's target variables increases and/or (ii) when one's share of training (test) data declines (increases). We also offer further insight into precisely *when* multi-label methods offer significant advantages in our second application's determination that the relative strengths of multi-label classification will decidedly increase as the number of available features declines. To facilitate these insights, we likewise provide a comprehensive overview of the requisite performance criteria for evaluations of multi-label predictions. Together, these insights will help to ensure that future multi-label classifications of political texts are as accurate as possible.

Funding

This work was supported by the Social Science and Humanities Research Council [430-2018-1069 to A.E., 430-2018-1069 to S.G.D.], the Fonds de recherche du Québec - Société et culture [253243 to A.E., 253243 to S.G.D.], Compute Canada [to A.E. and S.G.D.], the National Science Foundation [DMS-1737865 to B.E.B.] and the University of Delaware

General University Research Fund [to B.E.B.].

Acknowledgements

We thank Jeff Gill, four anonymous reviewers, Kevin Aslett, Vito D'Orazio, Ore Koren, and members of DemoTIP lab for their helpful comments and suggestions.

Data Availability Statement

Replication code for this article is available at Erlich et al. (2021) at `https://doi.org/10.7910/DVN/SOVPA4`.

References

Almanzar, T., M. Aspinwall, and D. Crow. 2018. "Freedom of information in times of crisis: The case of Mexico's war on drugs." *Governance* 31 (2).

Bagozzi, B. E., and D. Berliner. 2018. "The politics of scrutiny in human rights monitoring: Evidence from structural topic models of US State Department human rights reports." *Political Science Research and Methods* 6 (4).

Baker, S., and A.-L. Korhonen. 2017. "Initializing neural networks for hierarchical multi-label text classification." Association for Computational Linguistics.

Barberá, P., A. E.Boydstun, S. Linn, R. McMahon, and J. Nagler. 2020. "Automated Text Classification of News Articles: A Practical Guide." *Political Analysis.*

Berliner, D., B. E. Bagozzi, and B. Palmer-Rubin. 2018. "What Information Do Citizens Want?: Evidence from One Million Information Requests in Mexico." *World Development* 109.

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

Boutell, M. R., J. Luo, X. Shen, and C. M. Brown. 2004. "Learning multi-label scene classification." *Pattern recognition* 37 (9).

Brinker, K., J. Fürnkranz, and E. Hüllermeier. 2006. "A unified model for multilabel classification and ranking." In *Proceedings of the 2006 conference on ECAI 2006:*

17th European Conference on Artificial Intelligence August 29–September 1, 2006, Riva del Garda, Italy. IOS Press.

Cantú, F., and S. M. Saiegh. 2011. “Fraudulent Democracy? An Analysis of Argentina’s Infamous Decade Using Supervised Machine Learning.” *Political Analysis* 19.

Chang, C., and M. Masterson. 2020. “Using Word Order in Political Text Classification with Long Short-term Memory Models.” *Political Analysis* 28 (3).

Chen, J., J. Pan, and Y. Xu. 2016. “Sources of Authoritarian Responsiveness: A Field Experiment in China.” *American Journal of Political Science* 60.

Cheng, W., and E. Hüllermeier. 2009. “Combining instance-based learning and logistic regression for multilabel classification.” *Machine Learning* 76 (2-3).

Cingranelli, D. L., and D. L. Richards. 2010. “The Cingranelli and Richards (CIRI) Human Rights Data Project.” *Human Rights Quarterly* 32 (2).

Clare, A., and R. D. King. 2001. “Knowledge discovery in multi-label phenotype data.” In *european conference on principles of data mining and knowledge discovery.*

D’Orazio, V., S. T. Landis, G. Palmer, and P. Schrodt. 2014. “Separating the Wheat from the Chaff: Applications of Automated Document Classification Using Support Vector Machines.” *Political Analysis* 22 (2).

deHaan, E., A. Lawrence, and R. Litjens. 2019. “Measurement Error in Dependent Variables in Accounting: Illustrations Using Google Ticker Search and Simulations.” In *SSRN: https://ssrn.com/abstract=3398287.*

Dietrich, B. J., M. Hayes, and D. Z. O’Brien. 2019. “Pitch Perfect: Vocal Pitch and the Emotional Intensity of Congressional Speech.” *American Political Science Review* 113 (4).

Dudani, S. A. 1976. “The distance-weighted k-nearest-neighbor rule.” *IEEE Transactions on Systems, Man, and Cybernetics,* no. 4.

Elisseeff, A., and J. Weston. 2002. “A kernel method for multi-labelled classification.” In *Advances in neural information processing systems.*

Erlich, A., S. Dantas, B. Bagozzi, D. Berliner, and B. Palmer-Rubin. 2021. "Replication Data for: Multi-label Prediction for Political Text-as-Data." ***https://doi.org/10.7910/DVN/SOVPA4**, Harvard Dataverse V1.1.*

Fariss, C. J., F. J. Linder, Z. M. Jones, C. D. Crabtree, M. A. Biek, A.-S. M. Ross, T. Kaur, and M. Tsai. 2015. "Human Rights Texts: Converting Human Rights Primary Source Documents into Data." *PLOS One,* no. 0138935.

Goncalves, E. C., A. Plastino, and A. A. Freitas. 2013. "A genetic algorithm for optimizing the label ordering in multi-label classifier chains." In *2013 IEEE 25th International Conference on Tools with Artificial Intelligence.* IEEE.

Greene, K. T., B. Park, and M. Colaresi. 2019. "Machine Learning Human Rights and Wrongs: How the Successes and Failures of Supervised Learning Algorithms Can Inform the Debate About Information Effects." *Political Analysis* 27.

Grimmer, J., and B. M. Stewart. 2013. "Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts." *Political Analysis* 21.

Ji, S., L. Tang, S. Yu, and J. Ye. 2008. "Extracting shared subspace for multi-label classification." In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM.

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

Laver, M., K. Benoit, and J. Garry. 2003. "Extracting Policy Positions from Political Texts Using Words as Data." *American Political Science Review* 92 (2).

Lusa, L., et al. 2013. "SMOTE for high-dimensional class-imbalanced data." *BMC bioinformatics* 14 (1).

Madjarov, G., D. Kocev, D. Gjorgjevikj, and S. Džeroski. 2012. "An extensive experimental comparison of methods for multi-label learning." *Pattern recognition* 45 (9).

McCallum, A. K. 1999. "Multi-label text classification with a mixture model trained by EM." In *AAAI 99 workshop on text learning.* Citeseer.

Miller, B., F. Linder, and W. R. Mebane. 2020. “Active Learning Approaches for Labeling Text: Review and Assessment of the Performance of Active Learning Approaches.” *Political Analysis.*

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

Murdie, A., D. R. Davis, and B. Park. 2020. “Advocacy Output: Automated Coding Documents from Human Rights Organizations,” *Journal of Human Rights* 19 (1).

Park, B., K. Greene, and M. Colaresi. 2020a. “How to Teach Machines to Read Human Rights Reports and Identify Judgments at Scale.” *Journal of Human Rights* 19 (1).

Park, B., K. Greene, and M. Colaresi. 2020b. “Human Rights are (Increasingly) Plural: Learning the Changing Taxonomy of Human Rights from Large-scale Text Reveals Information Effects.” *American Political Science Review.*

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al. 2011. “Scikit-learn: Machine Learning in Python.” *Journal of Machine Learning Research* 12.

Probst, P., Q. Au, G. Casalicchio, C. Stachl, and B. Bischl. 2017. “Multilabel classification with R package mlr.” *arXiv preprint arXiv:1703.08991.*

Qi, G.-J., X.-S. Hua, Y. Rui, J. Tang, T. Mei, and H.-J. Zhang. 2007. “Correlative multi-label video annotation.” In *Proceedings of the 15th ACM international conference on Multimedia.* ACM.

Read, J., L. Martino, and D. Luengo. 2014. “Efficient monte carlo methods for multi-dimensional learning with classifier chains.” *Pattern Recognition* 47 (3).

Read, J., B. Pfahringer, G. Holmes, and E. Frank. 2009. “Classifier chains for multi-label classification.” In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases.* Springer.

Smith, C., and Y. Jin. 2014. “Evolutionary multi-objective generation of recurrent neural network ensembles for time series prediction.” *Neurocomputing* 143.

Szymański, P., and T. Kajdanowicz. 2017. “A scikit-based Python environment for performing multi-label classification.” *ArXiv e-prints.*

Tang, L., S. Rajan, and V. K. Narayanan. 2009. "Large scale multi-label classification via metalabeler." In *Proceedings of the 18th international conference on World wide web.*

Tawiah, C., and V. Sheng. 2013. "Empirical comparison of multi-label classification algorithms." In *Twenty-Seventh AAAI Conference on Artificial Intelligence.*

Torres, M., and F. Cantú. 2020. "Learning to See: Visual Analysis for Social Science Data."

Tsoumakas, G., and I. Vlahavas. 2007. "Random k-labelsets: An ensemble method for multilabel classification." In *European conference on machine learning.* Springer.

Ueda, N., and K. Saito. 2003. "Parametric mixture models for multi-labeled text." In *Advances in neural information processing systems.*

Wansbeek, T., and E. Meijer. 2000. *Measurement Error and Latent Variables in Econometrics.* North Holland.

Williams, N. W., A. Casas, and J. D. Wilkerson. 2020. *Images as Data for Social Science Research: An Introduction to Convolutional Neural Nets for Image Classification.* Cambridge University Press.

Xu, D., Y. Shi, I. W. Tsang, Y.-S. Ong, C. Gong, and X. Shen. 2019. "Survey on Multi-Output Learning." *IEEE transactions on neural networks and learning systems.*

Yan, R., J. Tesic, and J. R. Smith. 2007. "Model-shared subspace boosting for multi-label classification." In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM.

Younes, Z., F. Abdallah, T. Denoeux, and H. Snoussi. 2011. "A dependent multilabel classification method derived from the k-nearest neighbor rule." *EURASIP Journal on Advances in Signal Processing* 2011.

Zhang, M.-L., and K. Zhang. 2010. "Multi-label learning by exploiting label dependency." In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM.

Zhang, M.-L., and Z.-H. Zhou. 2007. "ML-KNN: A lazy learning approach to multi-label learning." *Pattern recognition* 40 (7).

Zhang, M.-L., and Z.-H. Zhou. 2013. “A review on multi-label learning algorithms.” *IEEE transactions on knowledge and data engineering* 26 (8).