Improving Probabilistic Models in Text Classification via Active Learning*

Mitchell Bosley[†‡] Saki Kuzushima[†§] Ted Enamorado[¶]
Yuki Shiraito[‖]

First draft: September 10, 2020
Final submission: April 22, 2024

Abstract

Social scientists often classify text documents to use the resulting labels as an outcome or a predictor in empirical research. Automated text classification has become a standard tool, since it requires less human coding. However, scholars still need many human-labeled documents for training. To reduce labeling costs, we propose a new algorithm for text classification that combines a probabilistic model with active learning. The probabilistic model uses both labeled and unlabeled data, and active learning concentrates labeling efforts on difficult documents to classify. Our validation study shows that with few labeled data the classification performance of our algorithm is comparable to state-of-the-art methods at a fraction of the computational cost. We replicate the results of two published articles with only a small fraction of the original labeled data used in those studies, and provide open-source software to implement our method.

*We thank Ken Benoit, Yaoyao Dai, Chris Fariss, Yusaku Horiuchi, Kosuke Imai, Walter Mebane, Daichi Mochihashi, Kevin Quinn, Luwei Ying, audiences at the 2020 Annual Meeting of the American Political Science Association, the 2021 Annual Meeting of the Midwest Political Science Association, the 11th Annual Conference on New Directions in Analyzing Text as Data, the 2022 Summer Meeting of the Japanese Society for Quantitative Political Science, and the 40th Annual Summer Meeting of the Society for Political Methodology, and seminar participants at the University of Michigan and members of the Junior Faculty Workshop at Washington University in St. Louis for useful comments and suggestions. We also appreciate detailed and constructive comments from four anonymous reviewers of the journal. Finally, we are extremely grateful to the editor, Michelle Dion, for guiding us through the rigorous review process of *APSR*.

[†]These authors have contributed equally to this work.

[‡]Ph.D. Candidate, Department of Political Science, University of Michigan. Email: mcbosley@umich.edu. ORCID: 0000-0002-9172-966X.

[§]Ph.D. Candidate, Department of Political Science, University of Michigan. Email: skuzushi@umich.edu. ORCID: 0000-0003-3014-5203.

[¶]Assistant Professor, Department of Political Science, Washington University in St. Louis. Siegle Hall, 244. One Brookings Dr. St Louis, MO 63130-4899. Phone: 314-935-5810, Email: ted@wustl.edu, URL: www.tedenamorado.com. ORCID: 0000-0002-2022-7646.

[‖]Assistant Professor, Department of Political Science, University of Michigan. Center for Political Studies, 4259 Institute for Social Research, 426 Thompson Street, Ann Arbor, MI 48104-2321. Phone: 734-615-5165, Email: shiraito@umich.edu, URL: shiraito.github.io. ORCID: 0000-0003-0264-1138.

Introduction

Text classification—the act of measuring underlying concepts by categorizing sequences of text into two or more categories—is a fundamental task in social science research. In political science, researchers have used this approach to classify a wide variety of textual data, including legislative speeches (Peterson and Spirling, 2018; Motolinia, 2021), correspondences to administrative agencies (Lowande, 2018), public statements of politicians (Airoldi et al., 2007; Stewart and Zhukov, 2009), news articles (Boydstun, 2013), election manifestos (Catalinac, 2016), social media posts (King et al., 2017), religious speeches (Nielsen, 2017), and human rights text (Cordell et al., 2021; Greene et al., 2019).[1]

Because manually labeling a large number of documents to classify text is too costly, researchers are increasingly turning to machine learning and Natural Language Processing (NLP) methodologies to automate this task. For example, to investigate the relationship between internet access and state repression in Syria, Gohdes (2020) manually labeled 2,000 out of 65,274 documents in order to train a machine learning model to predict the class of the documents in the rest of the corpus. Similarly, Park et al. (2020) train a classifier using 4,000 of the 2,473,874 documents in their corpus to analyze the association between Information Communication Technologies (ICTs) and the U.S. Department of State's human rights reports. Although these approaches are more efficient than manually labeling an entire corpus, labeling thousands of documents still demands considerable time and effort, as the authors of these studies acknowledge.

To help researchers reduce the amount of labeled data required to train an accurate classification model, we introduce *activeText*, a fast and easy-to-use algorithm for text classification that can be run on a standard laptop. Our method combines probabilistic modeling, semi-supervised learning (Nigam et al., 2000; Zhu and Goldberg, 2022) and active learning (McCallum et al., 1998; Settles, 2011; Miller et al., 2020) to help guide researchers to label documents more efficiently.

Recently, deep learning models like Bidirectional Encoder Representations from Transformers (BERT, Devlin et al. 2018) have become popular due to the impressive performance on many text classification tasks (Liu et al., 2019). However, they require substantial computational resources to train and can be prone to overfitting when labeled data is scarce (Sun et al., 2019). In addition, it is difficult to understand how their predictions are generated (Rudin, 2019). Consequently, deep learning models may not be the best choice when researchers have limited computational resources and labeled data. We present *activeText*

[1]See Grimmer et al. (2022) for a further overview of the application of text classification methods in political science research.

as an additional option for researchers who face these challenges.

The simple mixture model we use at the core of *activeText* runs an order of magnitude faster than deep learning models by employing the Expectation-Maximization (EM) algorithm (Dempster et al., 1977) for parameter estimation. Because of the computational efficiency of our model, we embed it in an active learning framework where the most informative documents are selected for human labeling. As a result, users of *activeText* can rapidly iterate between training the model to generate estimates of label uncertainty and selecting informative documents for human labeling to train a classification model with high accuracy using only a small fraction of the documents in their corpus.

Another key feature of our approach is that the parameters of our mixture model can be easily inspected to understand how the model is making its predictions. We leverage this interpretability to allow users with existing subject expertise to boost classification performance by upweighting keywords associated with each classification category. All these features make of *activeText* an attractive option for social science researchers for whom labeled data and computational resources are scarce.

We demonstrate the performance of *activeText* in three ways. First, we conduct a series of validation experiments to assess our performance on four common political science classification tasks using real text corpora: identifying news articles as political, identifying toxic hate speech, classifying the topic of supreme court rulings, and identifying mentions of physical integrity violations in human rights reports. Our validation experiments show that when there are few labeled documents, *activeText* generally outperforms alternatives, including DistilBERT (Sanh et al., 2019), a more computationally efficient variant of BERT, in terms of classification performance. We show that this benefit is most pronounced when the corpus is unbalanced, and that upweighting keywords can boost further model performance. We also show that in contrast to models such as BERT, which require substantial computational resources to train, *activeText* can easily be run on a standard laptop using the statistical programming language **R**, with the prediction of classification labels typically completing in seconds for a corpus with tens of thousands of documents.

Second, we replicate Gohdes (2020) and Park et al. (2020) using *activeText* to show how researchers could have used our method to reach the same substantive conclusions—a higher level of internet access is associated with a larger proportion of targeted killings, and ICTs are not associated with the sentiment of the State Department's human rights reports, respectively—with far fewer labeled documents.

Third, we use simulations to explore the general conditions under which *activeText* performs well, and to evaluate the impact of mislabeling documents and the potential biases introduced by active learning on the classification performance of *activeText*. We show that

activeText is robust to minor instances of mislabeling and that the in-sample bias introduced by active learning does not affect out-of-sample classification performance.

This paper proceeds as follows. In "Machine Learning Approaches to Text Classification," we introduce readers to the concepts of semi-supervised and active learning approaches to text classification. In "The Method," we describe both the semi-supervised and the active learning components of *activeText*, and how we combine the two. In "Validation Performance," we show the results from comparing our model to popular alternatives on validation data sets. Then, "Reanalysis with Fewer Human Annotations" presents the results of our replication studies. Finally, we discuss several practical concerns, directions for future research, and possible improvements to the algorithm in "Discussion." Code and data to reproduce the analysis of this paper is available at the American Political Science Review Dataverse (Bosley et al., 2024).

Machine Learning Approaches to Text Classification

In social science research, it is common for researchers to want to classify a large collection of text documents into two or more categories based on the content of the text in order to test a substantive hypothesis. The biggest impediment to this process is the cost of manually labeling a large collection of text documents. Rather than exhaustively labeling all documents, researchers often use machine learning techniques to automate the process, with supervised learning being the most common paradigm. In supervised learning, a model is trained on a labeled dataset to learn the relationship between text features and class labels (Kotsiantis et al., 2007), and a variety of supervised learning algorithms, such as Naive Bayes, Support Vector Machine (SVM), and Logistic Regression, have been applied to text classification tasks in political science research (Hillard et al., 2008; Colleoni et al., 2014).

Even though supervised learning reduces the amount of labeling required relative to hand-coding all documents, it still requires a substantial amount of labeled data to train a model that generalizes well to the entire corpus, typically in the order of several thousand labeled documents. In most applied political science applications, however, researchers start with little to no labeled data, making it laborious to label sufficient data to train accurate classifiers. In this section, we discuss two machine learning approaches that we leverage in our method to address the challenge of labeled data scarcity: semi-supervised learning and active learning.[2]

[2]For an introduction of basic concepts in machine learning applied to text data for classification tasks, including topics like feature representation, supervised and unsupervised learning, discriminative versus generative models, and model evaluation metrics, please refer to Supplemental Information (SI) A.

Solutions to Labeled Data Scarcity

To address the challenge of labeled data scarcity, several approaches have been proposed in the machine learning literature, including semi-supervised and active learning. Semi-supervised learning aims to leverage the structure of large amounts of unlabeled data to improve classification performance (Miller and Uyar, 1996; Nigam et al., 2000). In a semi-supervised setting, the model learns from both labeled and unlabeled data, using the labeled data as a foundation for measurement and incorporating patterns recovered from the unlabeled data to produce more accurate and robust predictions. This approach is particularly useful when labeled data is scarce, but unlabeled data is abundant.

Active learning, on the other hand, focuses on strategically selecting the most informative instances for labeling, minimizing the labeling effort while maximizing the model's performance (Settles, 2011). One of the most studied active learning approaches is uncertainty sampling (Lewis and Gale, 1994; Yang et al., 2015), where documents are chosen for labeling based on how uncertain the model is about their correct classification. By focusing labeling efforts on these informative documents, active learning can learn the decision boundary more efficiently than randomly selecting documents for labeling. In addition, active learning approaches have been shown to be particularly effective when the classification categories are imbalanced, which is a common occurrence in social science classification exercises (Miller et al., 2020).

An active learning algorithm typically involves a sequence of iterative steps applicable to any classification methodology. The first step is to estimate the probability that each document belongs to a specific classification outcome. The second step involves actively selecting the documents that the model is most uncertain about and focusing manual labeling efforts among those documents (Hoi et al., 2006). Then, the class probabilities are re-estimated using the newly labeled data. The algorithm cycles through these steps until a stopping criterion is met, such as a fixed budget for labeling (Ishibashi and Hino, 2020) or a threshold for improvement in accuracy metrics such as precision, recall, or F1 score (Altschuler and Bloodgood, 2019).

To illustrate the difference between passive and active learning for labeling a document, consider the scenario where a researcher aims to classify each unlabeled (U) document as either political (P) or non-political (N) based on the frequency of terms like Spending" and Gridlock" (Figure 1, Panel A presents the corpus). In passive learning (Panel B), the next document to be labeled is randomly selected, regardless of its position in the feature space. In contrast, active learning (Panel C) prioritizes labeling documents in the region of uncertainty (shaded region), where the model is less confident about their true labels.

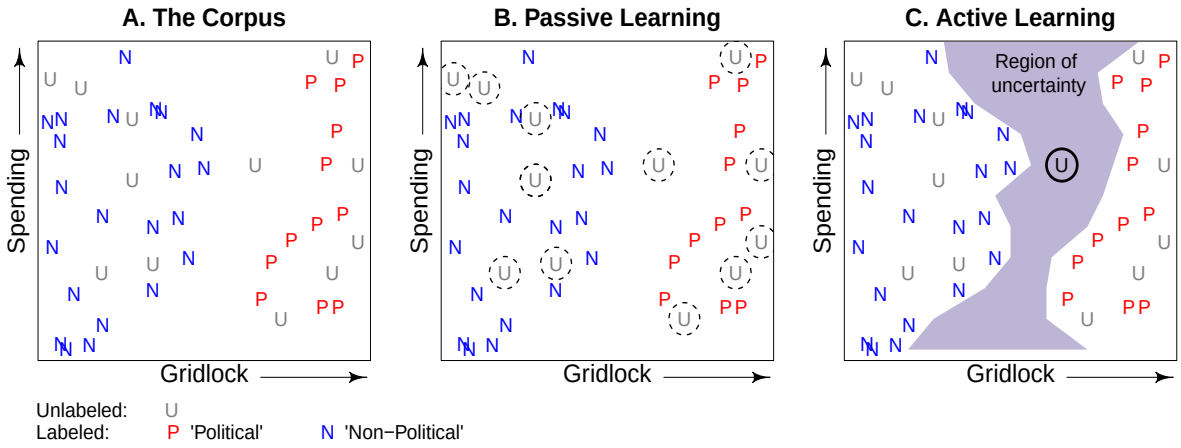

Figure 1: **Labeling: Passive vs Active Learning.** Panel A presents a corpus where a classifier based on term frequencies of Spending" and Gridlock" is utilized to categorize unlabeled (U) documents as political (P) and non-political (N). Panel B depicts a passive learning approach where the next document to be labeled is randomly selected. In contrast, Panel C demonstrates active learning, where obtaining the true label of the U located in the region of uncertainty for the classifier (shaded region) is prioritized, as it provides more informative insights into learning the decision boundary between P and N.

By focusing labeling efforts on these informative documents, active learning can learn the decision boundary more efficiently than passive learning

Deep Learning and *activeText*

In recent years, transfer learning, especially using deep learning approaches and pre-trained embeddings, has become popular. Transfer learning is a machine learning technique where knowledge gained from solving one problem is applied to a different but related problem, often leading to improved performance and reduced training time compared to training from scratch (Ruder et al., 2019). This approach relies on leveraging large pre-trained models, such as BERT (Devlin et al., 2018), which have been trained on vast amounts of unlabeled text data using the Transformer architecture (Vaswani et al., 2017) to learn rich, contextual representations of words and sentences. Rather than encode text as a sparse vector of word frequencies and learn the relationship between text features and class labels, as in the bag-of-words representation, these models learn embeddings—dense, vector representations of words that capture semantic and syntactic similarities between words and documents. Once the embedding representations of text data are learned, they can be fine-tuned towards a specific classification task using a collection of labeled data[3], allowing the model to adapt

[3]Fine-tuning involves adding a classification layer on top of the pre-trained model and training it on the target task while keeping the pre-trained model weights mostly fixed (Howard and Ruder, 2018).

its learned representations to the specific domain and task at hand.

While transfer learning with deep learning models has been shown to excel at many text classification tasks (Devlin et al., 2018; Liu et al., 2019), simpler models still have a place in the text classification toolkit, especially when labeled data and computing resources are scarce. Deep learning models require significant computational resources and can be time-consuming to train, even when fine-tuning pre-trained models (Strubell et al., 2019), and require substantial technical expertise in machine learning and natural language processing to implement relative to simpler models. They also have an extremely large number of parameters, making them more prone to overfitting when labeled data is limited (Sun et al., 2019).[4] In addition, their complex architectures and high-dimensional representations can make them difficult to interpret (Guidotti et al., 2018). For a model to be interpretable, we mean both that the model's predictions can be explained in terms of the input features and that the model's parameters can be used to gain insights into the underlying phenomena and test substantive theories, both of which are essential in political science research.[5]

Because of these limitations, we argue that when labeled data is scarce, computational resources are limited, and model interpretability is crucial—i.e., the conditions under which the typical political scientist operates—combining semi-supervised and active learning techniques with a simple mixture model[6] is a viable alternative to deep learning approaches at a fraction of the computational cost.[7]

In the following sections, we propose *activeText*, a novel method that combines semi-supervised learning and active learning with a generative mixture model based on bag-of-words representations of text data. Our approach leverages the EM algorithm to learn from both labeled and unlabeled data and incorporates uncertainty-based active learning to strategically select examples for labeling. We demonstrate the effectiveness of our approach through experiments and case studies on real-world political science datasets, highlighting its performance, interpretability, and computational efficiency compared to alternative methods.

[4]Overfitting occurs when a model learns to fit the noise or random fluctuations in the training data, rather than the underlying patterns, leading to poor performance on new, unseen data, and is a common problem in machine learning, particularly when the amount of labeled data is small and/or the model is complex (Hastie et al., 2009).

[5]See Rudin (2019) for a discussion of the importance of interpretability in machine learning.

[6] Mixture models are probabilistic models that can effectively capture the underlying structure of the data while remaining computationally efficient and interpretable (McLachlan et al., 2019).

[7] This is not to say that social scientists should not use deep learning models. To the contrary, we expect that in many cases, deep learning models will outperform simpler models, especially when labeled data is abundant and computational resources are not a constraint.

The Method

In this section, we present our modeling strategy and describe our active learning algorithm. For the probabilistic model (a mixture model for discrete data) at the heart of the algorithm, we build on the work of Nigam et al. (2000), who show that probabilistic classifiers can be augmented by combining the information coming from labeled and unlabeled data. As we will discuss below, we insert our model into an active learning algorithm and use the EM algorithm to maximize the observed-data log-likelihood function and estimate the model parameters.

Model

Consider the task of classifying N documents as one of two classes (e.g., political vs. non-political). Let $\mathbf{D}$ be a $N \times V$ document feature matrix, where V is the number of features.[8] In most applications, features are words, but they can also be bi-grams, tri-grams, or other tokens such word embeddings, etc. We use $\mathbf{Z}$, a vector of length N, where each entry represents the class assigned to each document. If a document i is assigned to the kth class (out of K classes), then $Z_i = k$, where $k \in \{0, 1\}$ (e.g., $k = 1$ represents the class of documents about politics, and $k = 0$ those that are non-political). Because we use a semi-supervised approach, some documents are already hand-labeled. This means that the value of Z_i is known for the labeled documents and is unknown for unlabeled documents.

To facilitate exposition, we assume that the classification goal is binary, however, our approach can be extended to accommodate 1) multiclass classification, where $K > 2$ and each document is classified into one of the K classes e.g., classifying news articles into 3 classes: politics, business, and sports (see SI D); and 2) modeling more than two classes but keeping the final classification output binary (see SI E).[9]

Equations (1) to (7) summarize the model:

Labeled Data

$$Z_i = k \quad \rightarrow \quad \text{hand-coded}, \quad k \in \{0, 1\} \tag{1}$$

$$\boldsymbol{\eta}_{\cdot k} \overset{\text{i.i.d.}}{\sim} \text{Dirichlet}(\boldsymbol{\beta}_{\cdot k}) \tag{2}$$

$$\mathbf{D}_{i\cdot} | Z_i = k \overset{\text{i.i.d.}}{\sim} \text{Multinomial}(n_i, \boldsymbol{\eta}_{\cdot k}) \tag{3}$$

[8] Throughout the paper, we denote a row or a column of a matrix by using $\cdot$ in the subscript, where the subscript $\cdot a$ represents the ath column and the subscript $b\cdot$ represents the bth row.

[9] In this second approach, we hierarchically map multiple sub-classes into one class e.g., collapsing the classification of documents that are about business and sports into a larger class (non-politics), and letting the remaining documents be about politics.

$\lambda \times$ Unlabeled Data

$$\pi \sim \text{Beta}(\alpha_0, \alpha_1) \tag{4}$$

$$Z_i = k \overset{\text{i.i.d.}}{\sim} \text{Bernoulli}(\pi), \quad k \in \{0, 1\} \tag{5}$$

$$\boldsymbol{\eta}_{\cdot k} \overset{\text{i.i.d.}}{\sim} \text{Dirichlet}(\boldsymbol{\beta}_{\cdot k}) \tag{6}$$

$$\mathbf{D}_{i\cdot} | Z_i = k \overset{\text{i.i.d.}}{\sim} \text{Multinomial}(n_i, \boldsymbol{\eta}_{\cdot k}) \tag{7}$$

If document i is unlabeled, we first draw the parameter $\pi = p(Z_i = 1)$, the overall probability that any given document belongs to the first class (e.g., political documents), from a Beta distribution with hyperparameters α_0 and α_1.[10] Similarly, for the other class (e.g., non-political documents), we have that $1 - \pi = p(Z_i = 0)$. Given π, for each document indexed by i, we draw the class assignment indicator Z_i from a Bernoulli distribution.[11] Then, we draw features for document i from a multinomial distribution governed by the total number of words in document i (n_i) and the vector $\boldsymbol{\eta}_{\cdot k}$ represents the kth column of the $V \times K$ matrix $\boldsymbol{\eta}$, where each entry of $\boldsymbol{\eta}_{\cdot k}$ is represented by the scalar $\eta_{vk} = p(D_{iv} | Z_i = k)$. The prior of $\boldsymbol{\eta}_{\cdot k}$ is the Dirichlet distribution with hyperparameter vector $\boldsymbol{\beta}_{\cdot k}$ (the kth column of the $V \times K$ matrix $\boldsymbol{\beta}$). Finally, $\mathbf{D}_{i\cdot}$ is a row vector of length V that represents the word counts of document i. Conditional on $Z_i = k$, $\mathbf{D}_{i\cdot}$ is drawn from a multinomial distribution with parameters n_i and $\boldsymbol{\eta}_{\cdot k}$. If document i has a label, the key distinction from the scenario with unlabeled data is that each Z_i is not drawn from a Bernoulli distribution. Instead, its value is manually determined through hand-coding.[12] Other than this point, the structure

[10] An anonymous reviewer asked us to further justify our choice of the beta prior over other prior distributions such as the uniform distribution. We opted for a Beta distribution with hyperparameters α_0 and α_1 for a couple of reasons. First, it is conditionally conjugate in our model, allowing for efficient computation of posterior updates for π, as demonstrated in SI C. Conjugate prior distributions often provide good approximations and simplify computations, similar to standard likelihood models (Gelman et al., 2014, p. 36). This principle also applies to our model, since the model for Z_i given π is the Bernoulli distribution for which the Beta distribution is conjugate. We note that the uniform prior is a special case of the Beta prior with $\alpha_0 = \alpha_1 = 1$ (e.g., Blitzstein and Hwang, 2019, p. 380). Second, unless α_0 and α_1 are significantly large compared to the number of documents in each class, their selection has minimal impact on estimating π, as discussed in SI C. We set $\alpha_0 = \alpha_1 = 2$ in our study to avoid prior density on extreme values of π such as $\pi = 0$ and $\pi = 1$ while ensuring computational feasibility, but our package provides the option for setting the prior parameter values of the user's choice. We thank the anonymous reviewer for raising this point.

[11] An alternative approach would be to allow groups of documents to have distinct values of π. In such a setting, for each observation i in group g, we could have $\pi_g = p(Z_i = k | G_i = g)$, where G_i is a variable indicating the group assignment of document i and the total number of groups is smaller than N. This modeling strategy can be beneficial for datasets with inherent group structures like longitudinal data, especially when the group hierarchy is observed. Yet, the datasets utilized in this paper lack a clear pre-established group structure. Therefore, instead of specifying π_g, we opted for specifying π. While incorporating a hierarchical structure to π could be an interesting future extension of our model, we leave it for future research. We thank an anonymous reviewer for highlighting this point.

[12] In equation 1, we use $\rightarrow$ to represent a deterministic assignment of the classes to documents.

of the model remains unchanged for the labeled data.

Altogether, if we denote $\mathcal{L}_{\text{obs}}(\pi, \boldsymbol{\eta}|\mathbf{D}, \mathbf{Z}, \lambda)$ as the observed data log-likelihood for all the data, based on equations (1) to (7), then we can express it as:

$$\mathcal{L}_{\text{obs}}(\pi, \boldsymbol{\eta}|\mathbf{D}, \mathbf{Z}, \lambda) = \mathcal{L}_{\text{labeled}}(\pi, \boldsymbol{\eta}|\mathbf{D}_{\text{labeled}}, \mathbf{Z}_{\text{labeled}}) + \lambda \times \mathcal{L}_{\text{unlabeled}}(\pi, \boldsymbol{\eta}|\mathbf{D}_{\text{unlabeled}}, \mathbf{Z}_{\text{unlabeled}}) \tag{8}$$

In other words, the result of adding the information from the observed log-likelihoods for the labeled and unlabeled data, respectively, and where, $\lambda \in [0, 1]$ is a parameter that adjusts the influence originating from the unlabeled data on the observed data log-likelihood. The inclusion of such a parameter follows from the scarcity of labeled data compared to the abundance of unlabeled data, which is a significant challenge in implementing semi-supervised learning approaches, as the likelihood function of text in unlabeled documents is likely to overwhelm that of the labels. This is a common problem with combining information from text and other sources through likelihood-based methods, as text data usually contain an order of magnitude more observed variables—features—than other types of data. To ensure that a classifier effectively extracts information from labeled data and is not solely influenced by unlabeled data, it is crucial to enhance the relative importance of labeled data; otherwise, the signal from labeled data will be overshadowed by the overwhelming presence of unlabeled data. To address this, we weight information from unlabeled documents by utilizing a decision factor, λ (Nigam et al., 2000).[13] When λ equals 1, the model equally considers each document, irrespective of whether it is labeled by human supervision or labeled probabilistically by the algorithm. As λ moves from 1 to 0, the model reduces the importance of information contributed by probabilistically labeled documents in the estimation of $\boldsymbol{\eta}$ and π. When λ reaches 0, the model disregards the information from all probabilistically labeled documents, turning it into a supervised algorithm.

Additionally, note that an important advantage of the interpretability of the key model parameters facilitates augmentation of the model using additional information such as domain knowledge. For example, each element of $\boldsymbol{\eta}_{\cdot k}$ represents the probability of observing a unique feature given the class of the document. In the Section "Active Keyword Upweighting," we show how to augment the model to allow some features to be highly associated with an specific class and in that way improve performance.

Finally, because the observed data log-likelihood of our model is difficult to maximize, we use the EM algorithm to estimate the parameters.[14]

[13] Kim et al. (2018, pp. 217–8) use a similar strategy to balance the information from both a smaller dataset (roll calls) and a larger dataset (textual data) within a model designed for estimating ideal points.

[14] For a full derivation of the EM algorithm for our binary classification model and its graphical represen-

Algorithm 1: Active learning with EM algorithm to classify text

Result: Obtain predicted classes of all documents.
Randomly select a small subset of documents, and ask humans to label them;
[**Active Keyword**]: Ask humans to provide initial keywords;
while *Stopping conditions are not met yet* **do**
  (1) [**Active Keyword**]: Up-weight the important of keywords associated with a class;
  (2) Predict labels for unlabeled documents using EM algorithm;
  (3) Select documents with the highest uncertainty among unlabeled documents, and ask humans to label them;
  (4) [**Active Keyword**]: Select words most strongly associated with each class, and ask humans to label them;
  (5) Update sets of labeled and unlabeled documents for the next iteration;
end

An Active Learning Algorithm

Our active learning algorithm (see Algorithm 1) can be split into the following steps: *estimation* of the probability that each unlabeled document belongs to the positive class, *selection* of the unlabeled documents whose predicted class is most uncertain, and *labeling* of the selected documents by human coders. The algorithm iterates until a stopping criterion is met. In this section, we also describe an optional keyword upweighting feature, where a set of user-provided keywords provide prior information about the likelihood that a word is generated by a given class to the model. These keywords can either be provided at the outset of the model or identified during the active learning process.

We now proceed to describe in detail each step of our algorithm:

Estimation

In the first iteration, the model is initialized with a small number of labeled documents.[15] The information from these documents is used to estimate the parameters of the model: the probability of a document being e.g., about politics, π, and $V \times 2$ matrix $\boldsymbol{\eta}$, represents the feature-class probabilities. If there is no labeled data, the model can be initialized by manually assigning initial values to the model parameters. These values can be set randomly or to a fixed value. From the second iteration on, we use information from both labeled and unlabeled documents to estimate the parameters using the EM algorithm, with the log-

tation, see SI C. Furthermore, refer to SI D for the corresponding details (model description, estimation, and graphical representation) for our model extension to multiclass classification and to SI E for the details regarding our second extension i.e., binary classification with multiple classes.

[15] While we assume that these documents are selected randomly, the researcher may choose any subset of labeled documents with which to initialize the model.

likelihood of unlabeled documents being weighted by λ, and with the $\boldsymbol{\eta}$ and π values from the previous iteration as the initial values. Using the estimated parameters, we compute the probability that each unlabeled document belongs to the politics class.

Selection

Using the predicted probability that each unlabeled document belongs to the politics class, we use Shannon Entropy (that is, the level of uncertainty) to determine which of the probabilistically labeled documents it was least certain about. In the binary classification case, this is the equivalent of calculating the absolute value of the distance between the politics class probability and 0.50 for each document. Using this criterion, the model ranks all probabilistically labeled documents in descending order of uncertainty. The n most uncertain documents are then selected for human labeling, where n is the number of documents to be labeled by humans at each iteration.

Labeling

A human coder reads each document selected by the algorithm and imputes the "correct" label. For example, the researcher may be asked to label as political or non-political each of the following sentences:

> The 2020 Presidential Election had the highest turnout in US history $\longrightarrow$ [Political]
>
> Argentina wins the 2022 FIFA World Cup, defeating France $\longrightarrow$ [Non-political]

These newly-labeled documents are then added to the set of human-labeled documents, and the process is repeated from the estimation stage.

Stopping Rule

Our method is highly modular and supports a variety of stopping rules. This includes an internal stability criterion, where stoppage is based on small amounts of change of the internal model parameters, as well as the use of a small held-out validation set to assess the marginal benefit of labeling additional documents on measures of model evaluation such as accuracy or F1. With either rule, the researcher specifies some bound such that if the change in model parameters or out-of-sample performance is less than the pre-specified bound, then the labeling process ends. For example, we use the out-of-sample validation stopping rule with a bound of 0.01 for the F1 score in Section "Reanalysis with Fewer Human Annotations."

Active Keyword Upweighting

The researcher also has the option to use an active keyword upweighting scheme, where a set of keywords is used to provide additional information. This is done by incrementing

elements of the $\boldsymbol{\beta}$ (the prior parameter of $\boldsymbol{\eta}$) by γ, a scalar value chosen by the researcher. In other words, we impose a tight prior on the probability that a given keyword is associated with each class.[16] To build the set of keywords for each class, 1) *activeText* proposes a set of candidate words, 2) the researcher decides whether they are indeed keywords or not,[17] and 3) *activeText* updates the parameters based on the set of keywords.

To select a set of candidate keywords, *activeText* calculates the ratio that each word was generated by a particular class using the $\boldsymbol{\eta}$ parameter. Specifically, it computes $\eta_{vk}/\eta_{vk'}$ for $k = \{0, 1\}$ with k' the opposite class of k, and chooses top m words whose $\eta_{vk}/\eta_{vk'}$ are the highest as candidate keywords to be queried for class k.[18] Intuitively, words closely associated with the classification classes are proposed as candidate keywords. For example, words such as "vote," "election," and "president," are likely to be proposed as the keywords for the political class of documents in the classification between political vs. non-political documents.

After *activeText* proposes candidate keywords, the researcher decides whether they are indeed keywords or not. This is where the researcher can use her expertise to provide additional information. For example, she can decide names of legislators and acronyms of bills as keywords for the political class.

Using the set of keywords for each class, *activeText* creates a $V \times 2$ keyword-class matrix $\boldsymbol{\kappa}$ where each element κ_{vk} takes the value of γ if word v is a keyword for class k, otherwise 0. Before we estimate parameters in each active iteration, we perform a matrix sum $\boldsymbol{\beta} \leftarrow \boldsymbol{\kappa} + \boldsymbol{\beta}$ to incorporate information from keywords. The keyword approach therefore effectively upweights our model with prior information about words that the researcher thinks are likely to be associated with one class rather than another.

Validation Performance

This section shows the performance comparisons between *activeText* and other classification methods. First, we show comparisons between active and passive learning. Then, we compare classification and time performance between *activeText* and a version of BERT called DistilBERT, a state-of-the-art text classification model using word embeddings as vector representations of the data.[19] Finally, we show how keyword upweighting can improve classification accuracy.

[16] See Eshima et al. (2020) for a similar approach for topic models.

[17] The researcher may also provide an initial set of keywords, and then iteratively adds new keywords.

[18] Words are excluded from candidate keywords if they are already in the set of keywords, or if they are already decided as non-keywords. Thus, no words are proposed twice as candidate keywords.

[19] We trained the BERT models using Nvidia V100 Graphics Processing Units (GPUs) on an High-Performance Computing (HPC) platform.

We compare the classification performance on each of the following sets of documents: internal forum conversations of Wikipedia editors (class of interest: toxic comment), BBC News articles (political topic), the United States Supreme Court decisions (criminal procedure), and Human Rights allegations (physical integrity rights allegation).[20] We use 80% of each dataset for the training data and hold out the remaining 20% for evaluation. Documents to be labeled are sampled only from the training set, and documents in the test set are not included to train the classifier, even in our semi-supervised approach. The out-of-sample F1 score is calculated using the held-out testing data.[21]

Classification Performance

Figure 2 shows the results from three model specifications: *activeText* (denoted by the solid line); *Random Mixture*, a version of *activeText* that uses passive instead of active learning (denoted by the dotted line); and DistilBERT (denoted by the dashed line).

Each panel corresponds to a unique combination of a dataset and the proportion of documents associated with the class of interest, with the rows corresponding to the datasets and the columns corresponding to the proportions. The parentheses beside the name of each corpus represent the proportion of positive labels in the population configuration i.e., the proportion of documents in the corpus that are labeled as the class of interest.[22] Within each panel, the x-axis represents the number of documents labeled, and the y-axis represents the average out-of-sample F1 score averaged across 50 and 10 Monte Carlo simulations in the case of the *activeText* models and the DistilBERT model, respectively. In the *activeText* models, 20 documents are labeled in each iteration.[23]

There are two key takeaways from Figure 2. First, we show that *activeText* is either equivalent to or outperforms its random sampling counterpart in nearly all cases, and the benefit from active learning is larger when the proportion of documents in the class of interest is smaller. The exception is the Human Rights corpus, where the benefit of active learning is marginal, and where at the 50% proportion, random sampling slightly outperforms active learning with less than 200 labeled documents.

Second, we show that in nearly all cases, *activeText* either outperforms or performs comparably to the DistilBERT model. As in the comparison between the active and random

[20] SI B presents a comprehensive description of the validation data and the preprocessing required for analyses.

[21] See SI A.4 for a detailed description of the F1-score and other commonly used model evaluation metrics.

[22] See SI B for more details on how we generate validated data with class-imbalance.

[23] Table H.1 in SI (Dataverse-only) H.1 presents similar evidence, based on other evaluation metrics (precision and recall). In addition, Figure H.4 in SI (Dataverse-only) H.3 includes comparisons of our generative approach, *activeText*, in terms of predictive performance against SVM, a popular discriminative method used for classification tasks.

versions of *activeText*, the advantage of *activeText* is larger when the proportion of documents in the class of interest is small. This is particularly true in the case of the BBC, Supreme Court, and Wikipedia corpora for the 5% and Population specifications. This advantage is not permanent, however: as the number of labeled documents increases, DistilBERT (as expected) performs well and even exceeds the F1 score of *activeText* in the case of Wikipedia. As before, the exception is the Human Rights corpus, where DistilBERT outperforms *activeText* at the 50% and Population levels.[24]

The early poor performance of *activeText* on the Human Rights corpus may be due to the fact that documents are short. Short labeled documents provide less information, making it more difficult for the model to distinguish between classes. We discuss how the information can be augmented using keywords to improve our method's classification performance in Section "Benefits of Keyword Upweighting." The keyword upweighting we propose takes advantage of the substantive interpretability of the feature-class matrix $\boldsymbol{\eta}$ in our generative model.

Runtime

In Figure 3, we compare computational runtime for *activeText* and DistilBERT. For this analysis, our goal was to compare how long it would take a researcher without access to a High-Performance Computing (HPC) platform or an expensive GPU to train these models. To this end, we trained the *activeText* and DistilBERT models on a base model M1 Macbook Air with 8 GB of RAM and 7 GPU cores. While the *activeText* models were trained using a single central processing unit (CPU), we used the recent implementation of support for the GPU in M1 Macs in PyTorch[25] to parallelize the training of the BERT model using the M1 Mac's GPU cores.[26] We also computed the time values *cumulatively* for *activeText* since it is expected that model will be fit over and over again as part of the active learning process, whereas for a model like BERT we expect that the model would only be run once, and as such do not calculate its run-time cumulatively. For the Human Rights and Wikipedia corpora, which each have several hundred thousand entries, we used a random subsample of 50,000 documents. For the Supreme Court and BBC corpora, we used the full samples.

[24]Figure D.2 of SI D illustrates that the multiclass version of *activeText* performs better than other alternative models for the BBC and Supreme Court datasets. Additionally, our findings in Figures E.1 and E.2 in SI E indicate that even in binary classification tasks, *activeText* excels when considering the presence of multiple classes. Again, this is especially noticeable in datasets such as the BBC and Supreme Court corpora, where the number of underlying classes exceeds 2.

[25]See https://pytorch.org/blog/introducing-accelerated-pytorch-training-on-mac/.

[26]Specifically, we trained a DistilBERT model (see Sanh et al., 2019) for three epochs (the number of passes of the entire training dataset BERT has completed) using the default configuration from the Transformers and PyTorch libraries for the Python programming language and used the trained model to predict the labels for the remaining documents for each corpus.

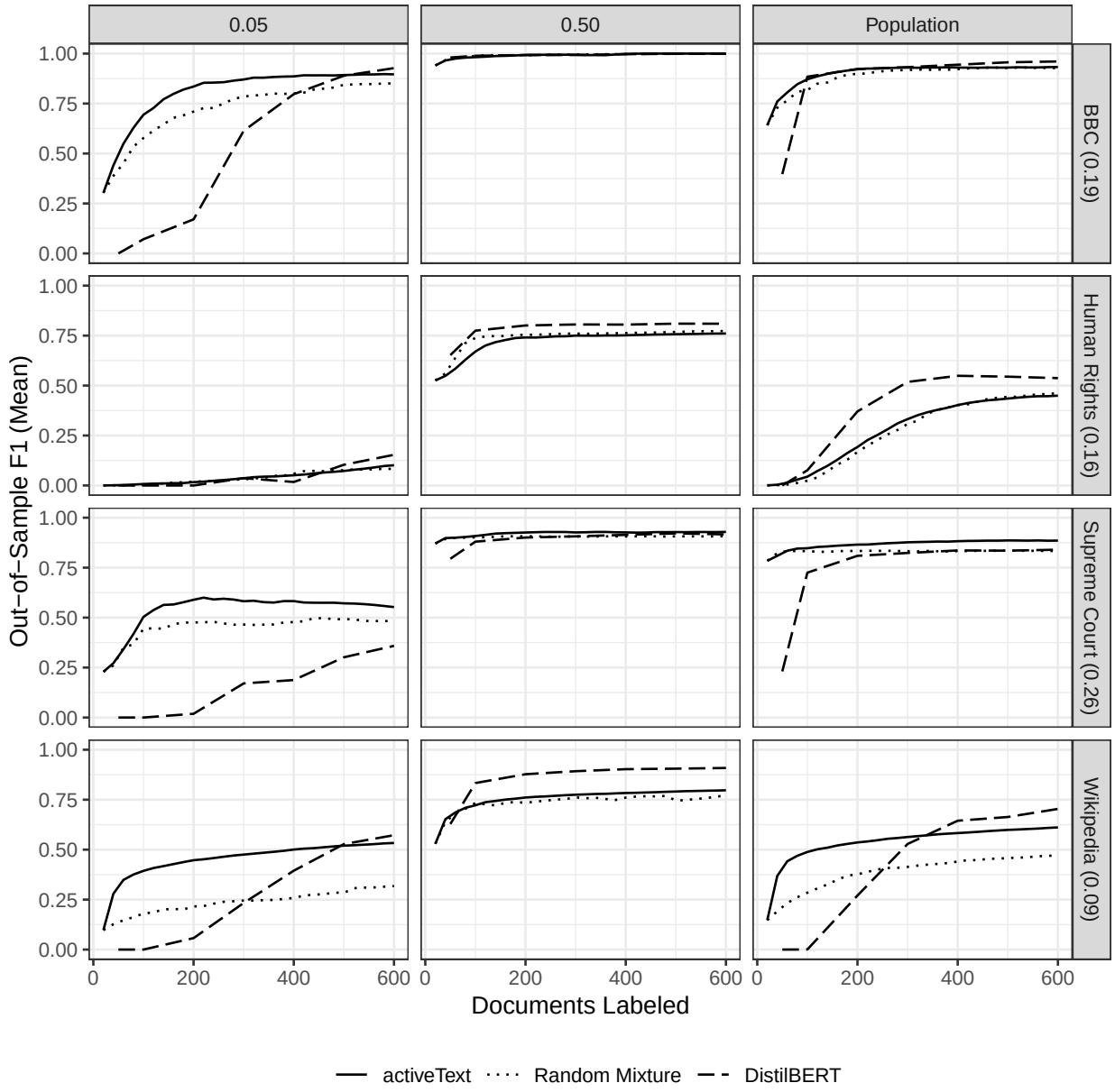

Figure 2: **Comparison of Classification Results with Random and Active Versions of *activeText* and DistilBERT**

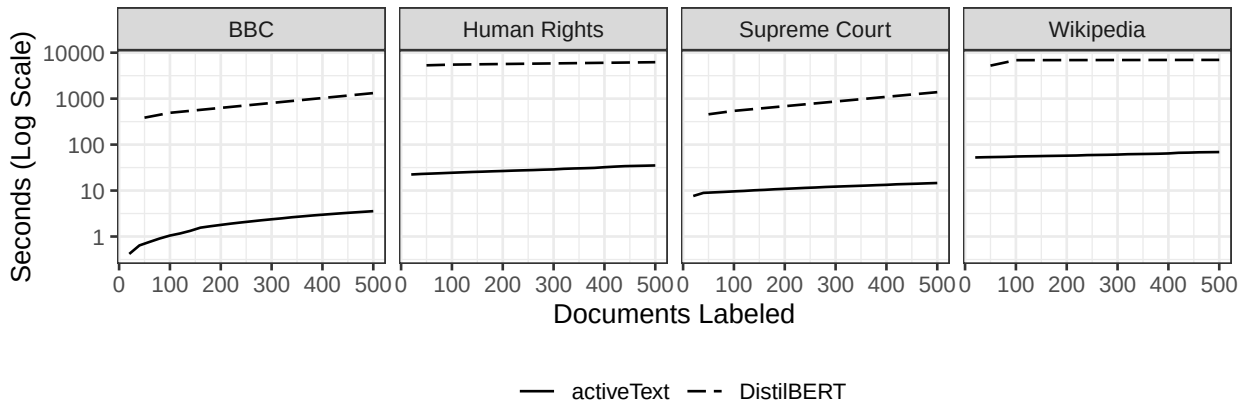

Figure 3: **Comparison of Classification and Time Results across *activeText* and DistilBERT**

Finally, we present the time results in logarithmic scale to improve visual interpretation.

Figure 3 shows that using DistilBERT comes at a cost of several orders of magnitude of computation time relative to *activeText*. Using the Wikipedia corpus as an example, at 500 documents labeled the baseline *activeText* would have run to convergence 25 times, and the sum total of that computation time would have amounted to just under 100 seconds. With DistilBERT, however, training a model with 500 documents and labeling the remaining 45,500 on an average personal computer would take approximately 10,000 seconds (2.78 hours).

Benefits of Keyword Upweighting

In Figure 2, active learning did not improve the performance on the human rights corpus, and the F1 score was lower than other corpora in general. One reason for the early poor performance of *activeText* may be the length of the documents. Because each document of the human rights corpus consists of one sentence only, the average length is shorter than other corpora.[27] This means that the information the models can learn from labeled documents is less compared to the other corpora with longer documents.[28] In situations like this, providing keywords in addition to document labels can improve classification performance because it directly shifts the values of the feature-class probability matrix, $\boldsymbol{\eta}$, even when the provided keywords is not in the already labeled documents.

Figure 4 compares the performance with and without providing keywords. The darker

[27]With the population data, the average length of each document is 121 (BBC), 17 (Wikipedia), 1620 (Supreme Court), and 9 (Human Rights)

[28]In our simulation studies described in Section "The Bias of Active Learning" and SI F, we confirmed that the classification performance is poor when the document length is short. Please refer to SI (Dataverse-only) K for the full set of results.

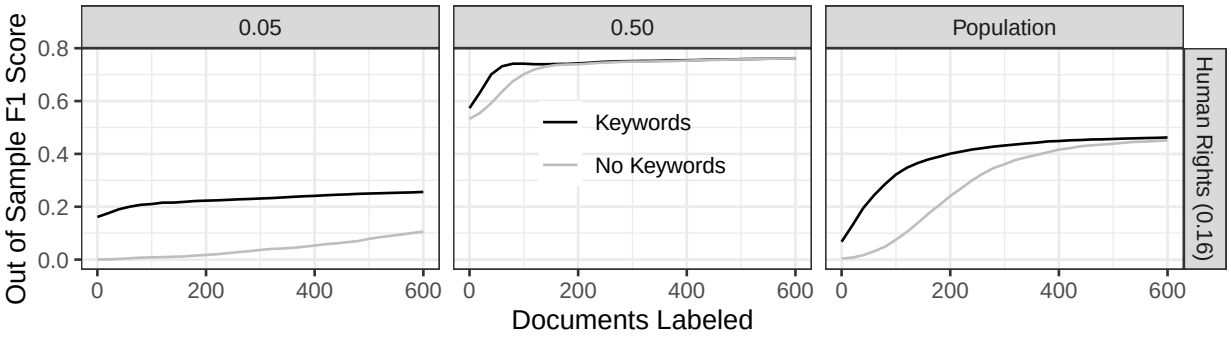

Figure 4: **Classification Results of *activeText* with and without Keywords**

lines show the results with keywords and the lighter lines without. The columns specify the proportion of documents associated with the class of interests: 5%, 50%, and the population proportion (16%). As in the previous exercises, 20 documents are labeled at each sampling step, and 100 Monte Carlo simulations are performed to stabilize the randomness due to the initial set of documents to be labeled. We simulated the process of a user starting with no keywords for either class and then being queried with extreme words indexed by v whose $\eta_{vk}/\eta_{vk'}$ is the highest for each class k, with up to 10 keywords for each class being chosen based on the estimated $\boldsymbol{\eta}$ at a given iteration of the active process. To determine whether a candidate keyword should be added to the list of keywords or not, our simulated user checked whether the word under consideration was among the set of most extreme words in the distribution of the 'true' $\boldsymbol{\eta}$ parameter, which we previously estimated by fitting our mixture model with the complete set of labeled documents.[29]

The results suggest that providing keywords improves performance when the proportion of documents is markedly imbalanced across classes. The keywords scheme improved the performance when the number of labeled documents is smaller on the corpus with 5% or 16% (population) labels associated with the class of interest. By contrast, it did not on the corpus where both classes were evenly balanced. These results highlight that our active keyword approach benefits the most when the dataset suffers from serious class imbalance problems.[30]

One caveat is that we provided 'true' keywords, in the sense that we used the estimated $\boldsymbol{\eta}$ from a fully labeled dataset. In practice, researchers have to decide if candidate keywords

[29]Specifically, the simulated user checked whether the word in question was in the top 10% of most extreme words for each class using the 'true' $\boldsymbol{\eta}$ parameter. If the candidate word was in the set of 'true' extreme words, it was added to the list of keywords and upweighted accordingly in the next active iteration.

[30]Figure H.3 in SI (Dataverse-only) H.2 demonstrates how active keyword works by visualizing the feature-class matrix, $\boldsymbol{\eta}$, at each active iteration. In particular, we show how the keyword scheme accelerates the learning process of the feature-class matrix $\boldsymbol{\eta}$.

are indeed keywords using their substantive knowledge. In this exercise, we believe that the keywords supplied to our simulation are what researchers with substantive knowledge about physical integrity rights can confidently adjudicate. For example, the keywords, such as "torture," "beat," and "murder," match our substantive understanding of physical integrity right violation. Nevertheless, as we explain in Section "Labeling Error" and SI G, humans can make mistakes, and some words may be difficult to judge. Thus, we examined the classification performance with varying degrees in the amount of error at the keyword labeling step. In SI G.2, we show that the active keyword approach still improves the classification performance compared to the no-keyword approach – even in the presence of small amounts (less than 20%) of "honest" (random) measurement error in keyword labeling.

Reanalysis with Fewer Human Annotations

To further illustrate the benefits of our proposed approach for text classification, we conduct reanalyses of two recently published aritcles: Gohdes (2020) and Park et al. (2020). We show that with *activeText*, we can arrive at the same substantive conclusions advanced by these authors but using only a small fraction of the labeled data they originally used.

Internet Accessibility and State Violence (Gohdes, 2020)

In the article "Repression Technology: Internet Accessibility and State Violence," Gohdes (2020) argues that higher levels of Internet accessibility are associated with increases in targeted repression by the state. The rationale behind this hypothesis is that through the rapid expansion of the Internet, governments have been able to improve their digital surveillance tools and target more accurately those in the opposition. Thus, even when digital censorship is commonly used to diminish the opposition's capabilities, Gohdes (2020) claims that digital surveillance remains a powerful tool, especially in areas where the regime is not fully in control.

To measure the extent to which killings result from government targeting operations, Gohdes (2020) collects 65,274 reports related to lethal violence in Syria. These reports contain detailed information about the person killed, date, location, and cause of death. The period under study goes from June 2013 to April 2015. Among all the reports, 2,346 were hand-coded by Gohdes, and each hand-coded report can fall under one of three classes: 1) government-targeted killing, 2) government-untargeted killing, and 3) non-government killing. Using a document-feature matrix (based on the text of the reports) and the labels of the hand-coded reports, Gohdes (2020) trained and tested a state-of-the-art supervised decision tree algorithm (extreme gradient boosting, `XGboost`). Using the parameters learned at the training stage, Gohdes (2020) predicts the labels for the remaining reports for which

the hand-coded labels are not available. For each one of the 14 Syrian governorates (the second largest administrative unit in Syria), Gohdes (2020) calculates the proportion of biweekly government targeted killings. In other words, Ghodes collapses the predictions from the classification stage at the governorate-biweekly level.

We replicate Gohdes (2020) classification tasks using *activeText*. In terms of data preparation, we adhere to the very same decisions made by Gohdes (2020). To do so, we use the same 2,346 hand-labeled reports (1,028 referred to untargeted killing, 705 to a targeted killing, and 613 a non-government killing) of which 80% were reserved for training and 20% to assess classification performance. In addition, we use the same document-feature matrices.[31] As noted in Section "An Active Learning Algorithm," because *activeText* selects (at random) a small number of documents to be hand-labeled to initialize the process, we conduct 100 Monte Carlo simulations and present the average performance across initializations. As in "Validation Performance," we set $\lambda = 0.001$. The performance of *activeText* and `XGboost` is evaluated in terms of out-of-sample F1 score. Following the discussion above, we stopped the active labeling process at the 30th iteration when the out-of-sample F1 score stopped increasing by more than 0.01 units (our pre-specified threshold). Table 1 presents the results.[32] Overall, we find that as the number of active learning steps increases, the classification performance of *activeText* is similar to the one in Gohdes (2020). However, the number of hand-labeled documents that are required by *activeText* is significantly smaller (around one-third) if compared to the ones used by Gohdes (2020).

Table 1: Classification Performance: Comparison with Gohdes (2020) results

			Ouf-of-sample F1 Score per class		
Model	Step	Labels	Untargeted	Targeted	Non-Government
activeText	0	20	0.715	0.521	0.800
	10	220	0.846	0.794	0.938
	20	420	0.867	0.828	0.963
	30	**620**	**0.876**	**0.842**	**0.963**
	40	820	0.879	0.845	0.961
Gohdes (2020)		1876	0.910	0.890	0.940

In social science research, text classification is often not the end goal but a means to quantify a concept that is difficult to measure and make inferences about the relationship between this concept and other constructs of interest. In that sense, to empirically test

[31] Gohdes (2020) removed stopwords, punctuation, and words that appear in at most two reports, resulting in 1,342 features and a document-feature matrix that is 99% sparse. The median number of words across documents is 13.

[32] The values in the bottom row are based on Gohdes (2020), Table A9.

her claims, Gohdes (2020) conducts regression analyses where the proportion of biweekly government targeted killings is the dependent variable and Internet accessibility is the main independent variable – both covariates are measured at the governorate-biweekly level. Gohdes (2020) finds that there is a positive and statistically significant relationship between Internet access and the proportion of targeted killings by the Syrian government. Using the predictions from *activeText*, we construct the main dependent variable and replicate the main regression analyses in Gohdes (2020).[33]

Tables I.2 and I.3 in SI (Dataverse-only) I.2 report the estimated coefficients, across the same model specifications in Gohdes (2020). The point estimates and the standard errors are almost identical whether we use `XGboost` or *activeText*. Moreover, Figure 5 presents the expected proportion of targeted killings by region and Internet accessibility, using the preferred regression specification by Gohdes. This model, as detailed in column V of Tables I.2 and I.3, incorporates the interaction between region and Internet accessibility. Gohdes finds that in the Alawi region, which is recognized for its loyalty to the regime, higher levels of Internet access correspond to a significantly lower expected proportion of targeted killings compared to other regions in Syria. In the absence of the Internet, however, there is no discernible difference across regions (see Figure 5, right panel). Our reanalysis does not change the substantive conclusions by Gohdes (2020) (Figure 5, left panel), however, it comes just at a fraction of the labeling efforts (labeling 620 instead of 1876 reports).

Human Rights are Increasingly Plural (Park et al., 2020)

Park et al. (2020) investigate how the rapid growth (in the last four decades) of information communication technologies (ICTs) has changed the composition of texts referring to human rights, and show that the average sentiment with which human rights reports are written has not drastically changed over time. They claim that if one wants to really understand the effect of changes in the access to information on the composition of human rights reports, it is necessary to internalize the fact that human rights are plural (i.e., bundles of related concepts). In other words, the authors argue that having access to new information has indeed changed the taxonomy of human rights over time, even when there has not been a change in tone.

To empirically test such a proposition, Park et al. (2020) conduct a two-step approach. First, by training an SVM for text classification with three classes (negative, neutral, and positive sentiment), the authors show that the average sentiment of human rights reports has

[33]The results presented in SI (Dataverse-only) I.1 demonstrate two main findings. First, the classification results of *activeText*, as shown in Table I.1, are almost identical to that of Gohdes (2020). Second, the proportion of biweekly government targeted killings from *activeText*, depicted in Figure I.1, is also highly consistent with the same measure by Gohdes (2020).

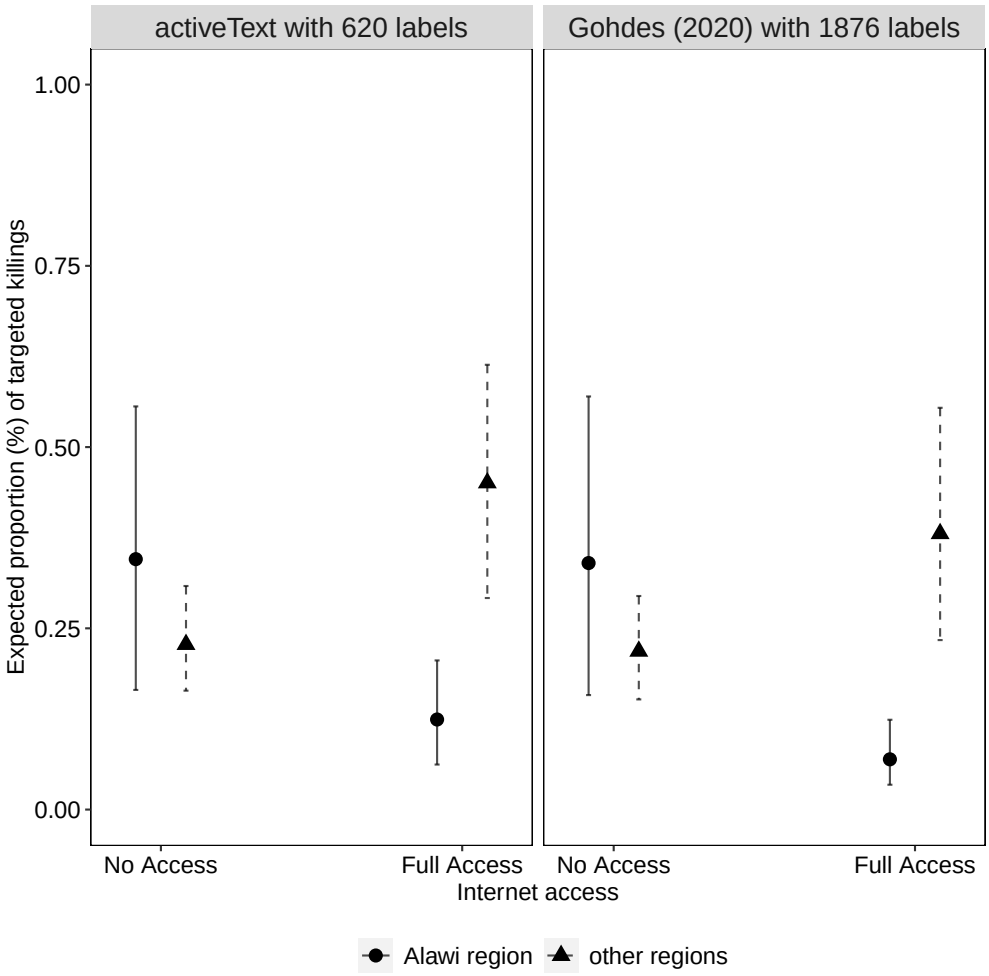

Figure 5: **Replication of Figure 3 in Gohdes (2020): Expected Proportion of Target Killings, Given Internet Accessibility and Whether a Region is Inhabitated by the Alawi Minority.** The results from *activeText* are presented in the left panel and those of Gohdes (2020) are on the right.

indeed remained stable even in periods where the amount of information available has become larger.[34] Second, they use a network modeling approach to show that while the average sentiment of these reports has remained constant over time, the taxonomy has drastically changed. In this section, using *activeText*, we focus on replicating the text classification task of Park et al. (2020), which is key to motivating their puzzle.

As in the reanalyses of Gohdes (2020), we adhere to the same pre-processing decisions made by Park et al. (2020) when working with their corpus of Country Reports on Human

[34]As explained in Appendix A1 of Park et al. (2020), negative sentiment refers to text about a clear ineffectiveness in protecting or to violations of human rights; positive sentiment refers to text about clear support (or no restrictions) of human rights; and neutral sentiment refers to stating a simple fact about human rights.

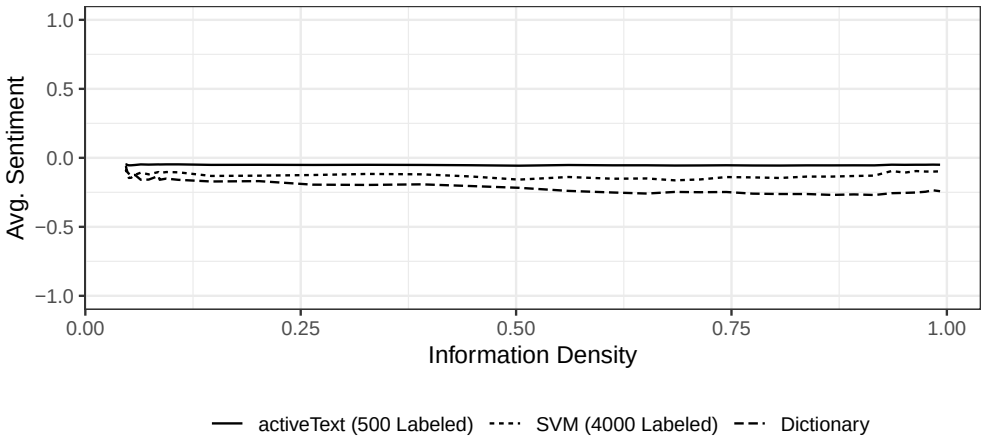

Figure 6: **Replication of Figure 1 in Park et al. (2020): The Relationship Between Information Density and Average Sentiment Score.**

Rights Practices from 1977 to 2016 by the US Department of State. In particular, we use the same 4000 hand-labeled human rights reports (1182 are positive, 1743 are negative, and 1075 are neutral) and use the same document-feature matrices (which contain 30,000 features, a combination of unigrams and bigrams). Again, we conduct 100 Monte Carlo simulations and present the average performance across initializations. As shown in Figure J.1 in SI (Dataverse-only) J, we stopped the active labeling process at the 25th iteration of our algorithm as the out-of-sample F1 score (from an 80/20 training/test split) does not increase by more than 0.01 units.[35] Using the results from the classification task via *activeText*, the sentiment scores of 2,473,874 documents are predicted. With those predictions, we explore the evolution of the average sentiment of human rights reports per average information density score.[36]

Figure 6 shows that by labeling only 500 documents with *activeText*, instead of 4000 labeled documents used by Park et al. (2020) to fit their SVM classifier, we arrive at the same substantive conclusion: the average sentiment of human rights reports has remained stable and almost neutral over time. In Figure J.2 of SI (Dataverse-only) J, we also show that this result is not an artifact of our stopping rule and it is robust to the inclusion of additional label documents (e.g, labeling 1000, 1500, and 2000 documents instead of just 500).

[35]The only point where we depart from Park et al. (2020) is that we use an 80/20 split for training/testing, while they use k-fold cross-validation. Conducting k-fold cross-validation for an active learning algorithm would require over-labeling because the labeling process should be repeated k times as well. Because of this difference, we refrain from comparing our model performance metrics to theirs.

[36]Information density is a proxy for ICTs based on a variety of indicators related to the expansion of communications and access to information, see Appendix B in Park et al. (2020).

Discussion

In this section, we address three key issues concerning our proposed approach: 1) the (in-sample) bias that occurs when actively selecting observations to train a model; 2) the impact of mislabeling documents and keywords; and 3) the practical considerations about down-weighting unlabeled data.

The Bias of Active Learning

As highlighted by Dasgupta (2011); Dasgupta and Hsu (2008); Farquhar et al. (2021), active learning introduces in-sample statistical bias due to training non-i.i.d data. As described above, active learning involves selecting the most informative data points for training. This selection can introduce bias because the newly labeled documents may not represent the entire of the target population.

To illustrate the nature of this bias, let's denote the population risk as $r = \mathbb{E}_{\mathbf{Z},\mathbf{D}}[L(Z_i, \widehat{Z}_i)]$. Here, L represents a loss function, such as the L1 or L2 norms. For each observation indexed by i, Z_i denotes its true label, and $\widehat{Z}_i$ denotes the predictions made based on the model parameters and the data's features ($\mathbf{D}$). It is important to note that $\mathbb{E}_{\mathbf{Z},\mathbf{D}}$ indicates the expectation calculated over the joint distribution of $\mathbf{Z}$ and $\mathbf{D}$. In the training stage, we want to find the model parameters that minimize the population risk. However, the population risk is not observed, and we can only estimate it using the labeled data. Hence, the empirical risk, derived from an i.i.d sample of the population with a size denoted as N^l, can be computed by the formula: $\widehat{R} = \frac{1}{N^l}\sum_{j=1}^{N^l}[L(Z_j, \widehat{Z}_j)]$. This calculation provides an unbiased estimate of r because the labeled data is randomly selected from the population. With active learning, the empirical risk is $\widetilde{R} = \frac{1}{M}\sum_{m=1}^{M}[L(Z_m, \widehat{Z}_m)]$, where M is the number of labeled data selected actively. Because these M data points are not a random sample of the population, $\widetilde{R}$ is no longer an unbiased estimator of r.

Farquhar et al. (2021) propose an unbiased estimator of the population risk with active learning, called Leveled Unbiased Risk Estimator (LURE), $\widetilde{R}_{\text{lure}} = \frac{1}{M}\sum_{m=1}^{M}[v_m L(Z_m, \widehat{Z}_m)]$, where v_m is a function of the sampling probabilities of the actively selected documents (refer to SI F for more details). They show that $\widetilde{R}_{\text{lure}}$ is an unbiased estimate with minimum variance in its class of weighted estimators for the empirical risk. In other words, the in-sample bias in the active learning process can be corrected by $\widetilde{R}_{\text{lure}}$ (refer to Theorems 3 and 4 in Farquhar et al. 2021).

Importantly, according to Farquhar et al. (2021), one factor that determines whether the in-sample bias correction improves out-of-sample predictive performance is the complexity of the model. For example, they empirically show that the in-sample bias correction with

LURE improves the out-of-sample classification performance for a simple linear regression, but not for a neural network. Farquhar et al. (2021) argue that for overparameterized models, correcting the in-sample bias from active learning might not be advantageous to improve out-of-sample classification because active learning can serve as a regularization mechanism against overfitting bias. While *activeText* does not possess as many parameters as a neural network, it has many more parameters than linear regressions depending on the number of features of the data. This implies that it is an open question whether the in-sample bias correction improves the out-of-sample predictive performance of *activeText*.

To examine whether the in-sample statistical bias has adverse effects on the out-of-sample classification performance, we conducted a series of simulation studies involving 108 different configurations. In our simulation studies, we manipulated various aspects of the simulated data, such as the number of unique words, the average length of the documents (measured in number of words), the difficulty of classification, and the proportion of positive class documents in the corpus.

Figure 7 presents the results of implementing the LURE bias correction to *activeText* under a simulation setup. This setup involves generating simulation data with 1000 documents, 500 unique features, an average of 50 features per document, and the reference class accounting for 10% of the corpus. We perform 100 Monte Carlo simulations in this context. The left panel presents the in-sample bias of the $\widetilde{R}$ and $\widetilde{R}_{\text{lure}}$, and the right panel presents the corresponding out-of-sample F1 scores. The bias is calculated as the difference between the population risk and the empirical risk. For active learning, this is represented as $r - \mathbb{E}[\widetilde{R}]$, while for its bias-corrected version, it is $r - \mathbb{E}[\widetilde{R}_{\text{lure}}]$.

Figure 7 shows that regarding the in-sample bias of the empirical risk, $\widetilde{R}$ exhibits an upward bias in the early stages of the labeling process, which gradually decreases as more documents are labeled. This bias arises because the most uncertain documents are labeled first in active learning, causing the empirical risk to initially surpass the population risk. In contrast, we find that LURE weights effectively eliminate the in-sample bias of *activeText*. However, as the right panel of Figure 7 shows, the unadjusted version of *activeText* demonstrates better out-of-sample classification performance compared to its bias-corrected counterpart. Consequently, our findings indicate that addressing in-sample bias does not necessarily improve the out-of-sample classification performance of *activeText*. These results hold across different simulation settings and in our validation data. For a more detailed overview of the simulation results, interested readers can refer to SI F.

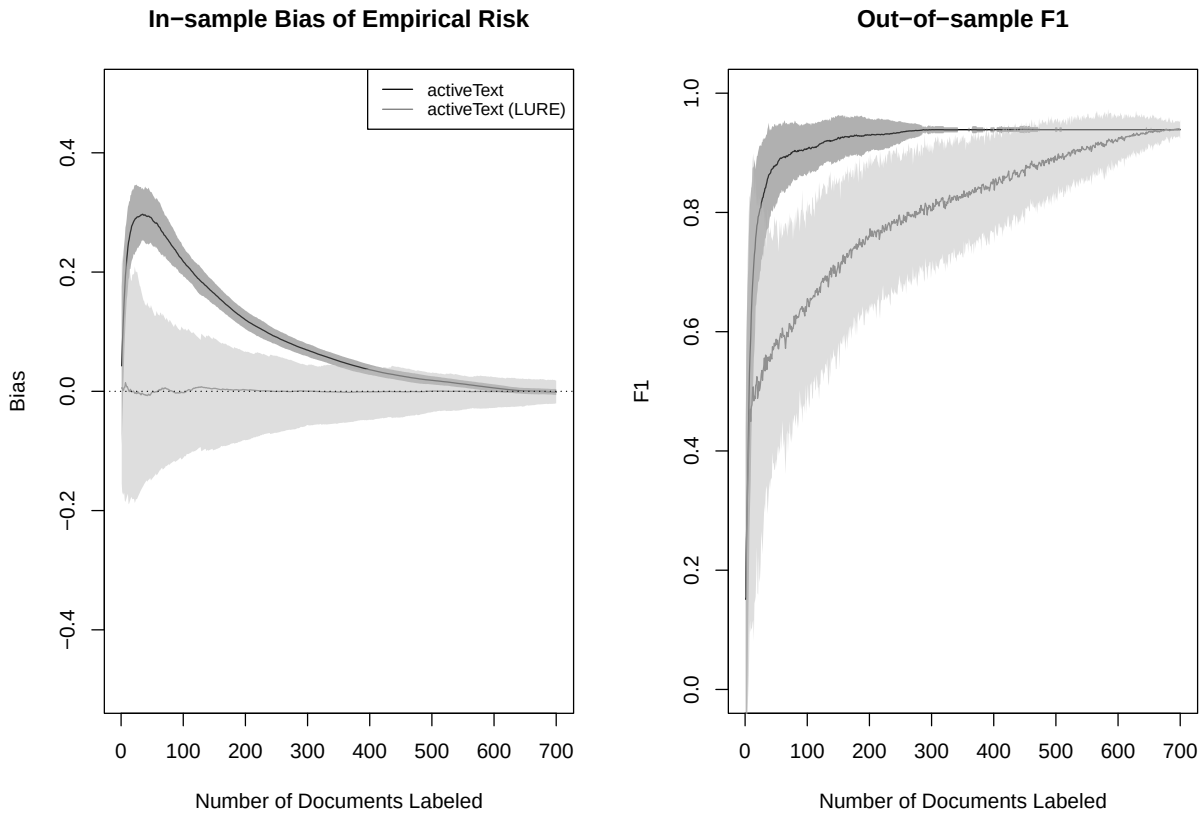

Figure 7: **Bias of the Empirical Risk for Labeled Data (left panel) and Out-of-sample Classification Performance (right panel) of *activeText*, *activeText*+LURE. For each panel, the x-axis represents the number of documents labeled, and the y-axis represents the average bias and average out-of-sample F1 score across 100 Monte Carlo simulations. Shaded areas represent the 95% confidence intervals across Monte Carlo simulations.**

Labeling Error

Although our main findings operate under the assumption that labelers are accurate, it is important to acknowledge that human labelers can still make errors. We will now investigate how mislabeling of documents and keywords that are actively selected impacts the classification performance of *activeText*. Specifically, we aim to demonstrate the potential impact of measurement errors in labeling, particularly focusing on the effect of "honest" mistakes (classical measurement error) on classification performance and basic downstream analysis.

In the case of mislabeling actively selected documents, our results show that random perturbations from true document labels do hurt the classification performance.[37] For ex-

[37]As outlined in SI G.1, in binary classification, honest mistakes in labeling documents entails choosing (at random) the opposite label from the true one.

ample, in SI G.1, we find that in the case of the BBC News articles dataset described above, when about 20 out of 200 documents are labeled with the incorrect label (10% mislabeling), the out-of-sample F1 score remains high at around 0.87. However, when the mislabeling of documents exceeds 1 in 5 documents ($\geq$ 20% mislabeling), there is a significant decrease in the out-of-sample F1 score (the F1 is less than 0.75). This pattern holds across all validation datasets (refer to Figure G.1 in SI G.1).

To illustrate how mislabeling affects downstream analyses, we consider a simple example. Suppose we are trying to predict the number of documents related to a specific category, like politics in the BBC data. In our validation studies, we already know the actual proportions of documents in the categories we are interested in. For instance, in the BBC dataset, 19% of the articles cover politics, while in the Wikipedia corpus, 9% of the documents are deemed toxic. Similarly, 26% of Supreme Court cases involve criminal procedure, and 16% of Human Rights reports include allegations of physical integrity rights violations. To gauge the impact of mislabeling, we assess the bias in the predictions for the proportion of documents in the target class made by our model. As illustrated in Figure G.2 in SI G.1, the bias increases as the rate of mislabeling rises. For example, in the Wikipedia comments dataset, if we accurately label around 200 documents, the bias is minimal. However, introducing 30% mislabeling results in a bias increase of 0.25 units. This trend is consistent across all datasets, mirroring what we observed with the F1 score. Essentially, even minor unintentional errors (honest mistakes) diminish accuracy when computing simple summary statistics such as a sample mean.

In contrast to our results for the mislabeling of documents, if compared to the no-keyword approach, a small amount of classical measurement error on keyword labeling does not hurt the classification performance.[38] For example, the results presented in Figure G.3 in SI G.2 shows that when random noise is introduced, the classification performance of *activeText* for binary classification decreases slightly as the proportion of mislabeled keywords increases to 30% or more. This trend remains consistent across different validation datasets and values of γ (the upweight assigned to each keyword).

In light of these results, we believe that future research focus on developing new active learning algorithms that prioritize assigning labelers based on their labeling expertise and adapt to various types of labeling errors. One approach could involve allocating the most skilled labelers to annotate the most uncertain or challenging documents, while assigning simpler tasks to less proficient labelers. This strategy could optimize the efficiency of the labeling process. Additionally, as we discussed above, inaccurate predictions can introduce

[38] As described in more detail SI G.2 mislabeling keywords implies randomly labeling non-keyword as keywords, and vice versa.

bias, particularly in downstream tasks. This bias can be exacerbated by departures from classical measurement error, making it difficult to determine its direction. Therefore, as recently recognized by authors such as Knox et al. (2022) and Fong and Tyler (2021) further investigation is necessary to directly tackle these potential biases, especially in settings such as a poplar inference methods such as a regression framework.[39]

Tuning the value of λ

As noted above, we downweight the information from unlabeled documents as we typically have more unlabeled than labeled documents. Moreover, since the labeled documents have been classified by an expert, we want to rely more on the information they bring for prediction.

An important practical consideration is how to select the appropriate value of λ. One possible approach would be to adopt popular model selection methods (e.g. cross-validation) to choose the appropriate λ value during the model initialization process. However, cross-validation may not be practical when the labeled data is scarce (or absent at the beginning of the process). We have consistently observed across a variety of applications that very small values (e.g., 0, 0.001 or 0.01) work the best on the corpora we used (see Figures H.1 and H.2 in SI (Dataverse-only) H.1). However, more work is needed to clearly understand the optimality criteria needed to select λ. We leave this question for future research.

Conclusion

Human labeling of documents is the most labor-intensive part of social science research that uses text data. For automated text classification to work, a machine classifier needs to be trained on the relationship between text features and class labels, and the labels in training data are given manually. In this paper, we have described a new active learning algorithm that combines a mixture model and active learning to incorporate information from labeled and unlabeled documents and better select which documents to be labeled by a human coder. Our validation and simulation studies showed that a moderate number of documents are labeled, and the proposed algorithm performed at least as well as state-of-the-art methods such as BERT at a fraction of the cost. We replicated two published political science studies to show that our algorithm lead to the same conclusions as the original papers but needed much fewer labeled documents. In sum, our algorithm enables researchers to save their manual labeling efforts without sacrificing quality.

Machine learning techniques are becoming increasingly popular in political science, but

[39]Refer to Egami et al. (2023) for a recent approach to recover regression estimates when noisy predictions from a model are employed as outcomes in regression analysis.

the barrier to entry remains too high for researchers without a technical background to make use of advances in the field. As a result, there is an opportunity to democratize access to these methods. We believe that our model and **R**-package will provide applied researchers with a tool that they can use to efficiently categorize documents in corpora of varying sizes and topics.

References

Airoldi, E., Fienberg, S., and Skinner, K. (2007), "Whose ideas? Whose words? Authorship of Ronald Reagan's radio addresses," *PS: Political Science & Politics*, 40(3), 501–506.

Altschuler, M., and Bloodgood, M. (2019), Stopping Active Learning Based on Predicted Change of F Measure for Text Classification,, in *2019 IEEE 13th International Conference on Semantic Computing (ICSC)*, pp. 47–54.

Blitzstein, J., and Hwang, J. (2019), *Introduction to Probability*, 2nd edn Chapman and Hall/CRC.

Bosley, M., Kuzushima, S., Enamorado, T., and Shiraito, Y. (2024), "Replication Data for: Improving Probabilistic Models in Text Classification via Active Learning,".
URL: *https://doi.org/10.7910/DVN/7DOXQY*

Boydstun, A. (2013), *Making the news: Politics, the media, and agenda setting* University of Chicago Press.

Catalinac, A. (2016), *Electoral reform and national security in Japan: From pork to foreign policy* Cambridge University Press.

Colleoni, E., Rozza, A., and Arvidsson, A. (2014), "Echo chamber or public sphere? Predicting political orientation and measuring political homophily in Twitter using big data," *Journal of communication*, 64(2), 317–332.

Cordell, R., Clay, C., Fariss, C., Wood, R., and Wright, T. (2021), "Recording repression: Identifying physical integrity rights allegations in annual country human rights reports," *International Studies Quarterly*, .

Dasgupta, S. (2011), "Two Faces of Active Learning," *Theoretical Computer Science*, 412(19), 1767–1781.

Dasgupta, S., and Hsu, D. (2008), Hierarchical sampling for active learning,, in *Proceedings of the 25th international conference on Machine learning*, pp. 208–215.

Dempster, A., Laird, N., and Rubin, D. (1977), "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society, Series B*, 39(1), 1–38.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018), "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, .

Egami, N., Hinck, M., Stewart, B., and Wei, H. (2023), Using Imperfect Surrogates for Downstream Inference: Design-based Supervised Learning for Social Science Applications of Large Language Models,, in *37th Conference on Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1–13.

Eshima, S., Imai, K., and Sasaki, T. (2020), "Keyword assisted topic models," *arXiv preprint arXiv:2004.05964*, .

Farquhar, S., Gal, Y., and Rainforth, T. (2021), On Statistical Bias In Active Learning: How and When to Fix It,, in *International Conference on Learning Representations*.
URL: *https://openreview.net/forum?id=JiYq3eqTKY*

Fong, C., and Tyler, M. (2021), "Machine Learning Predictions as Regression Covariates," *Political Analysis*, 29(4), 467–484.

Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A., and Rubin, D. (2014), *Bayesian Data Analysis*, 3rd edn Chapman and Hall/CRC.

Gohdes, A. (2020), "Repression technology: Internet accessibility and state violence," *American Journal of Political Science*, 64(3), 488–503.

Greene, K., Park, B., and Colaresi, M. (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(2), 223–230.

Grimmer, J., Roberts, M., and Stewart, B. (2022), *Text as data: A New Framework for Machine Learning and the Social Sciences* Princeton University Press.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. (2018), "A survey of methods for explaining black box models," *ACM computing surveys (CSUR)*, 51(5), 1–42.

Hastie, T., Tibshirani, R., and Friedman, J. (2009), *The Elements of Statistical Learning*, Springer Series in Statistics, New York, NY, USA: Springer New York Inc.

Hillard, D., Purpura, S., and Wilkerson, J. (2008), "Computer-assisted topic classification for mixed-methods social science research," *Journal of Information Technology & Politics*, 4(4), 31–46.

Hoi, S., Jin, R., and Lyu, M. (2006), Large-Scale Text Categorization by Batch Mode Active Learning,, in *WWW 06: Proceedings of the 15th International Conference on World Wide Web, Edinburgh, Scotland, May 23*, Vol. 26, pp. 633–642.

Howard, J., and Ruder, S. (2018), "Universal language model fine-tuning for text classification," *arXiv preprint arXiv:1801.06146*, .

Ishibashi, H., and Hino, H. (2020), Stopping criterion for active learning based on deterministic generalization bounds,, in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, eds. S. Chiappa, and R. Calandra, Vol. 108 of *Proceedings of Machine Learning Research*, PMLR, pp. 386–397.
URL: *https://proceedings.mlr.press/v108/ishibashi20a.html*

Kim, I. S., Londregan, J., and Ratkovic, M. (2018), "Estimating Spatial Preferences from Votes and Text," *Political Analysis*, 26(2), 210–229.

King, G., Pan, J., and Roberts, M. (2017), "How the Chinese government fabricates social media posts for strategic distraction, not engaged argument," *American political science review*, 111(3), 484–501.

Knox, D., Lucas, C., and Cho, W. K. T. (2022), "Testing Causal Theories with Learned Proxies," *Annual Review of Political Science*, 25(1), 419–441.

Kotsiantis, S., Zaharakis, I., Pintelas, P. et al. (2007), "Supervised machine learning: A review of classification techniques," *Emerging artificial intelligence applications in computer engineering*, 160(1), 3–24.

Lewis, D., and Gale, W. (1994), A Sequential Algorithm for Training Text Classifiers,, in *SIGIR '94*, eds. B. W. Croft, and C. J. van Rijsbergen, Springer London, London, pp. 3–12.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019), "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, .

Lowande, K. (2018), "Who Polices the Administrative State?," *American Political Science Review*, 112(4), 874–890.

McCallum, A., Nigam, K. et al. (1998), Employing EM and Pool-Based Active Learning for Text Classification.,, in *ICML*, Vol. 98, Citeseer, pp. 350–358.

McLachlan, G. J., Lee, S. X., and Rathnayake, S. I. (2019), "Finite mixture models," *Annual review of statistics and its application*, 6, 355–378.

Miller, B., Linder, F., and Mebane, W. R. (2020), "Active Learning Approaches for Labeling Text: Review and Assessment of the Performance of Active Learning Approaches," *Political Analysis*, pp. 1–20.

Miller, D., and Uyar, H. (1996), A Mixture of Experts Classifier with Learning Based on Both Labelled and Unlabelled Data,, in *Advances in Neural Information Processing Systems*, eds. M. Mozer, M. Jordan, and T. Petsche, Vol. 9, MIT Press.

Motolinia, L. (2021), "Electoral Accountability and Particularistic Legislation: Evidence from an Electoral Reform in Mexico," *American Political Science Review*, 115(1), 97–113.

Nielsen, R. (2017), *Deadly clerics: Blocked ambition and the paths to jihad* Cambridge University Press.

Nigam, K., McCallum, A. K., Thrun, S., and Mitchell, T. (2000), "Text classification from labeled and unlabeled documents using EM," *Machine learning*, 39(2-3), 103–134.

Park, B., Greene, K., and Colaresi, M. (2020), "Human Rights are (Increasingly) Plural: Learning the Changing Taxonomy of Human Rights from Large-scale Text Reveals Information Effects," *American Political Science Review*, 114(3), 888–910.

Peterson, A., and Spirling, A. (2018), "Classification accuracy as a substantive quantity of interest: Measuring polarization in westminster systems," *Political Analysis*, 26(1), 120–128.

Ruder, S., Peters, M., Swayamdipta, S., and Wolf, T. (2019), Transfer learning in natural language processing,, in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Tutorials*, pp. 15–18.

Rudin, C. (2019), "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature machine intelligence*, 1(5), 206–215.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019), "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, .
URL: *https://arxiv.org/abs/1910.01108*

Settles, B. (2011), *Synthesis Lectures on Artificial Intelligence and Machine Learning : Active Learning* Morgan & Claypool Publishers.

Stewart, B., and Zhukov, Y. (2009), "Use of force and civil–military relations in Russia: an automated content analysis," *Small Wars & Insurgencies*, 20(2), 319–343.

Strubell, E., Ganesh, A., and McCallum, A. (2019), "Energy and policy considerations for deep learning in NLP," *arXiv preprint arXiv:1906.02243*, .

Sun, S., Yu, D., Yu, D., and Cardie, C. (2019), "Fine-tuning pre-trained language model with weak supervision for document classification," *arXiv preprint arXiv:1909.05053*, .

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, Ł., and Polosukhin, I. (2017), "Attention is all you need," *Advances in neural information processing systems*, 30.

Yang, Y., Ma, Z., Nie, F., Chang, X., and Hauptmann, A. (2015), "Multi-Class Active Learning by Uncertainty Sampling with Diversity Maximization," *International Journal of Computer Vision*, 113, 113–127.

Zhu, X., and Goldberg, A. (2022), *Introduction to semi-supervised learning* Springer Nature.