Improving Probabilistic Models in Text Classification via Active Learning[*]

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

First draft: September 10, 2020
This draft: July 18, 2023

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 to train automated classifiers. 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 the classification performance of our algorithm is comparable to state-of-the-art methods at a fraction of the computational cost. Moreover, we replicate two recently published articles and reach the same substantive conclusions with only a small proportion of the original labeled data used in those studies. We provide *activeText*, an 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.

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

[‡] Ph.D. Candidate, Department of Political Science, University of Michigan. Email: mcbosley@umich.edu.

[§] Ph.D. Candidate, Department of Political Science, University of Michigan. Email: skuzushi@umich.edu

[¶] 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.

[‖] 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.

Introduction

As the amount and diversity of available information have rapidly increased, social scientists are increasingly resorting to multiple forms of data to answer substantive questions. In particular, the use of text-as-data in social science research has exploded over the past decade.[1] Document classification has been the primary task in political science, with researchers classifying documents such as legislative speeches (Peterson and Spirling, 2018; Motolinia, 2021), correspondences to administrative agencies (Lowande, 2018, 2019), 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), treaties (Spirling, 2012), religious speeches (Nielsen, 2017), and human rights text (Cordell et al., 2021; Greene et al., 2019) into two or more categories, which are then used as the outcome or as a predictive variable to test substantive hypotheses.

While many researchers undertake the laborious task of manually labeling a substantial number of documents, the cost of having human coders categorize all documents is often prohibitively high. Gohdes (2020), for example, manually labels approximately 2000 documents to analyze the relationship between internet access and state repression in Syria. Similarly, Park et al. (2020) conduct an analysis of the association between information communication technologies (ICTs) and the U.S. Department of State's human rights reports by manually labeling approximately 4000 documents. While these numbers are much smaller than their entire data sets ($65,274$ and $2,473,874$, respectively), as recognized by these authors, having human coders label thousands of documents still requires substantial time and effort.

We propose a new algorithm called *activeText* to make it easier for researchers to reduce the number of documents they manually code. Our algorithm has two key features. First, it uses a semi-supervised probabilistic model for text classification (Nigam et al., 2000) that incorporates information from both labeled and unlabeled documents. In this model, we observe latent classes as labels for labeled documents and estimate them as latent variables for unlabeled documents using the Expectation-Maximization (EM) algorithm (Dempster et al., 1977). Second, instead of randomly selecting documents for labeling, we employ active learning, which identifies the most informative documents for labeling based on measures of label uncertainty. Additionally, our method is designed to be computationally efficient and user-friendly, making it highly accessible for implementation on personal computers. We believe this feature will greatly support various research projects that involve text classification tasks.

[1] See Grimmer et al. (2022) for an excellent overview of the application of these methods in political science research.

We show that through the combination of semi-supervision and active learning, researchers can substantially reduce the number of documents that need to be hand-coded without sacrificing classification accuracy, and that this is particularly the case when the classification categories are imbalanced.

To test the performance of our model, we compare it to alternatives such as state-of-the-art Bidirectional Encoder Representations from Transformers (BERT) classifiers. Using a diverse set of validation tasks focused on the classification of political text, we present compelling evidence that *activeText* consistently outperforms alternative methods when limited labeled documents are available. Moreover, our findings reveal that *activeText* offers significant computational advantages over BERT, thereby reducing computational costs. Additionally, as our model is generative in nature and its key parameters are interpretable, we provide insights into leveraging a researcher's domain expertise beyond manual document labeling. We demonstrate the effectiveness of augmenting text classification through the annotation of class-associated keywords, thus enhancing the overall classification process.

We also show how researchers could have used our method to reach the same substantive conclusions with fewer labeled documents. To do so, we use *activeText* to replicate the Gohdes (2020) and Park et al. (2020) studies, both of which use text classification to test substantive hypotheses regarding human rights, and conduct the same empirical analyses for each study using the estimated document labels. Our replication analysis recovers their original 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—using far fewer labeled documents. These replication exercises demonstrate that *activeText* performs well on complex documents commonly used in political science research, such as human rights reports.

This paper proceeds as follows. In Machine Learning for Classifying Political Texts, 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 and simulated data sets. Then, Reanalysis with Fewer Human Annotations presents the results of our replication studies. Finally, we discuss a couple of practical concerns, directions for future research, and possible improvements to the algorithm in Discussion, and describe an **R** package called *activeText* with the goal of providing researchers from all backgrounds with easily accessible tools to minimize the amount of hand-coding of documents and improve the performance of classification models for their own work.

Machine Learning for Classifying Political Texts

Suppose that a researcher has a collection of social media text data, called a corpus, and wishes to classify whether each text in the corpus is political (e.g., refers to political protest, human rights violations, unfavorable views of a given candidate, targeted political repression, etc.) or not solely based on the words used in a given observation.[2] Critically, the researcher does not yet know which of the texts are political or not at this point.

Before analyzing text as data, the researcher needs to make several decisions. In many instances, these decisions include choosing how to represent tokens, selecting which tokens to include, and determining which pre-processing techniques to apply, etc.[3] Once the corpus is encoded, the classification question then arises: given the set of features of a document, how can we effectively determine its political nature?

In this section, we provide an overview of the two main components of our *activeText* method that facilitates text classification tasks: 1) semi-supervised learning and 2) active learning.[4]

Semi-supervised Learning

Supervised and unsupervised learning are two fundamental approaches in machine learning for text-as-data (Grimmer et al., 2022). In the supervised approach, the researcher follows these steps: (1) acquiring accurate labels for a subset of the documents through human coding, where e.g., the researcher determines that a news headline such as "What a DeSantis presidential run means for the 2024 election" refers to politics due to its focus on a political figure (Gov. Ron DeSantis) and his potential role in the 2024 Election; (2) establishing a connection between the textual features of each document in the corpus as represented in the matrix $\mathbf{D}$ and the true labels represented in the vector $\mathbf{Z}$ for the labeled documents. This involves understanding how terms like "DeSantis," "presidential," "run," "2024," and "election" are important in determining the political nature of the headline; and (3) utilizing the acquired understanding of the relationship between the text data and the known labels to later predict whether the remaining unlabeled documents in the corpus are political or not (Hastie et al., 2009).

[2] For simplicity, the exposition here focuses on a binary classification task, however, our proposed method can be extended to multiple classes e.g., classifying a document as either a positive, negative, or neutral position about a candidate. See Sections The Method and Reanalysis with Fewer Human Annotations, and Supplementary Information (SI) D for more details.

[3] Tokenization refers to the division of textual data into discrete units known as tokens, which can be words, terms, sentences, symbols, or other meaningful components (Grimmer et al., 2022, p. 52).

[4] For an introduction of machine learning concepts applied to text data for classification tasks, including topics like feature representation, discriminative versus generative models, and model evaluation metrics, please refer to SI A.

On the other hand, an unsupervised approach does not require the use of labeled data. Instead, the researcher employing an unsupervised approach would select a model that groups documents in the corpus based on shared patterns in the features as represented by the matrix $\mathbf{D}$. After assigning documents to clusters, the researcher would determine which cluster corresponds to the desired outcome of interest, namely whether a document is political or not. However, it's important to note that there is no guarantee that there will be a direct connection between clusters and the outcomes of interest (Knox et al., 2022).

Semi-supervised learning combines supervised and unsupervised approaches (Miller and Uyar, 1996; Nigam et al., 2000), making it particularly useful when there is a large amount of unlabeled data and labeling is expensive. In a semi-supervised model, the relationship between the text data matrix $\mathbf{D}$ and the classification outcome $\mathbf{Z}$ is learned using both labeled and unlabeled data. Although $\mathbf{Z}$ is not known for unlabeled data, it still provides information about the joint distribution of the features $\mathbf{D}$. Thus, by incorporating patterns recovered from the unlabeled data and using the labeled data as a foundation for measurement, semi-supervised learning produces more accurate and robust predictions compared to purely supervised or unsupervised methods (Nigam et al., 2000).

Active Learning

If the researcher in our running example decides to use a supervised or semi-supervised approach for predicting whether documents in their corpus are political or not, the next step is to decide how many documents to label, and how to choose them. Since labeling is the bottleneck of any classification task of this kind, it is critical that she also selects an approach to label observations that minimizes the number of documents to be labeled in order to produce an accurate classifier.

There are two popular strategies for retrieving cases to be labeled: 1) passively and 2) actively. The difference between a passive and an active approach amounts to whether the researcher randomly chooses which documents to label (i.e., choose documents *passively*), or whether to use some selection scheme (i.e., choose documents *actively*). Ideally, an active approach should require fewer labels than the amount of randomly labeled data sufficient for a passive approach to achieve the same level of accuracy.

Authors such as Cohn et al. (1994) and Lewis and Gale (1994) established that a good active learning approach should reliably choose documents for labeling that provide more information to the model than a randomly chosen document, particularly in situations when the amount of labeled data is scarce.[5] One of the most studied active learning approaches is called *uncertainty sampling* (Lewis and Gale, 1994; Yang et al., 2015), a process where

[5] See also Dasgupta (2011); Settles (2011); Hanneke (2014); Hino (2021) and the references therein.

documents are chosen for labeling based on how uncertain the model is about the correct classification category for each document in the corpus.[6]

Consider the scenario where the researcher aims to classify each unlabeled (U) document as either political (P) or non-political (N). For simplicity, suppose each document can be represented in a two-dimensional Euclidean space, based on the frequency of "Spending" and "Gridlock". The former being a term that is used across political and non-political documents alike, the latter being a term that if used at a high frequency within a document, then makes it more likely that such a document refers to politics. After having to hand label many documents independently without any guidance, the researcher decides it is time to find a better way to prioritize subsequent labeling efforts.

To illustrate this situation, Figure 1 (Panel A) presents a corpus where all documents represented by P and N have already been labeled. However, all of those represented by U are unlabeled. A passive learning method (Panel B) would assign equal labeling priority to any unlabeled document, i.e., the next unlabeled document to be hand labeled will be selected at random – as represented by the dashed circles around each U. Under this paradigm, any input from the classifier is not used as selection of the next case to be labeled is random. However, an active learning approach (Panel C) will prioritize labeling efforts for any unlabeled document in the region where the the model used for classification is more uncertain about its true label – the U surrounded by a solid circle in the region of uncertainty (shaded region). For unlabeled documents outside the region of uncertainty, the classifier is highly confident that those to the left of the region are non-political, and those to the right are political. Thus, manually assigning labels to documents situated within the region of uncertainty would provide the researcher and classifier with a greater understanding of the "dividing line" between documents with political and non-political orientations in this two-dimensional space.

If the labeling process employs an active approach, then we can say that an active learning *algorithm* entails 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, as described above, actively selecting the documents that the model is most uncertain about and focusing manual labeling efforts among those documents.[7] Then, the class probabilities are re-estimated using the newly

[6] This is just one of many possible approaches. Other uncertainty-based approaches to active learning include query-by-committee, variance reduction, expected model change, etc. We refer the interested reader to Settles (2011) for an accessible review on active learning and Hanneke (2014) for a more technical exposition.

[7] While our presentation focuses on instances of labeling one observation per iteration, exactly how many observations to select and label at each active iterations is also an important practical consideration for any researcher. As noted by Hoi et al. (2006), to reduce the cost of retraining the model per instance of labeling,

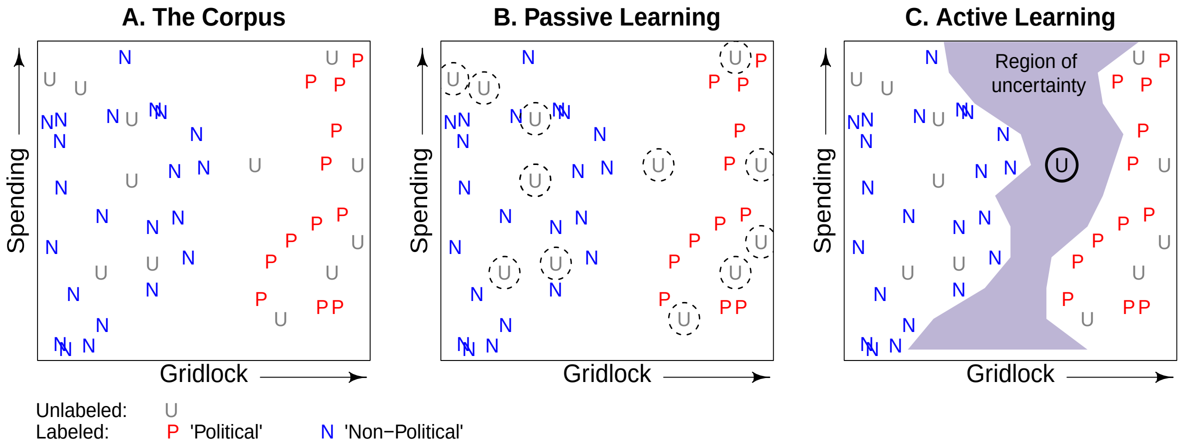

Figure 1: **Labeling: Passive vs Active Learning.** Panel A presents a corpus where a classifier based on term frequencies of ”Spending” and ”Gridlock” is utilized to categorize unlabeled (U) documents as political (P) and non-political (N). Panel B depicts a passive learning approach where the next document to be labeled is randomly selected i.e., any U inside a dashed circle is a target for labeling. However, some U's are surrounded by just P's or by just N's, and as a result, can be classified with high accuracy making their true labels less informative for the classifier. 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. This is because such a label will provide more informative insights into learning the dividing line between P's and N's.

labeled data. The algorithm cycles through these steps until a stopping criterion is met. For instance, in a fixed-budget approach, the active learning algorithm stops when the number of newly labeled data points reaches a predetermined value.[8] Another example is when the difference in measures of out-of-sample accuracy between two consecutive iterations is below a threshold set by the researcher e.g., the F1 score does not improve by more than 0.01 units from one iteration to the next (Altschuler and Bloodgood, 2019).[9]

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. In other words, our model makes the latent classes for the unlabeled data interpretable by connecting them to the hand-coded classes from the labeled data. It also takes advantage of the fact that the unlabeled data provides more information about the features used to predict the classes for each document. As we will discuss below, we insert our model into an active learning algorithm and use the Expectation-Maximization (EM) algorithm (Dempster et al., 1977) 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 size of features. We use $\mathbf{Z}$, a vector of length N, where each entry represents the latent classes assigned to each document. If a document i is assigned to the kth class, 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

labeling many documents per iteration (a batch) is the best approach. This is especially important when working with a large amount of data.

[8] The problem with such an approach is that it may lead to under- or over-sampling. This is due to the fact the fixed budget has not been set using an optimality criterion other than to stop human coding at some point. See Ishibashi and Hino (2020) for further discussion of this point.

[9] Note that if labeled data does not exist or cannot be set aside for testing due to its scarcity, we could use a stopping rule where the algorithm stops once in-sample predictions generated by the model (i.e., using the documents that have been labeled by the researcher during the active learning process) do not change from one iteration to the next. This is often referred to as a stability-based method (Ishibashi and Hino, 2020).

approach can be extended to accommodate 1) multiclass classification setting, where $k > 2$ and each document needs to be classified into one of the k classes (e.g., classifying news articles into 3 classes: politics, business, and sports); and 2) modeling more than two classes but keeping the final classification output binary.[10]

The following sets of equations summarize the model:

Labeled Data

$$\begin{aligned} Z_i = k &\sim \text{hand-coded}, \quad k \in \{0,1\} \\ \eta_{\cdot k} &\overset{i.i.d}{\sim} Dirichlet(\boldsymbol{\beta}_k) \\ \mathbf{D}_{i\cdot}|Z_i = k &\overset{i.i.d}{\sim} Multinomial(n_i, \boldsymbol{\eta}_{\cdot k}) \end{aligned}$$

$$+$$

$\lambda \cdot$ Unlabeled Data

$$\begin{aligned} \pi &\sim Beta(\alpha_0, \alpha_1) \\ Z_i = k &\overset{i.i.d}{\sim} Bernoulli(\pi), \quad k \in \{0,1\} \\ \eta_{\cdot k} &\overset{i.i.d}{\sim} Dirichlet(\boldsymbol{\beta}_k) \\ \mathbf{D}_{i\cdot}|Z_i = k &\overset{i.i.d}{\sim} Multinomial(n_i, \boldsymbol{\eta}_{\cdot k}) \end{aligned}$$

If document i is unlabeled, we first draw $\pi = p(Z_i = 1)$, 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. 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 latent cluster assignment indicator Z_i from a Bernoulli distribution. 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 $\boldsymbol{\eta}_{\cdot k}$, where $\eta_{vk} = p(D_{iv}|Z_i = k)$, whose prior is the Dirichlet distribution with hyperparameter vector $\boldsymbol{\beta}_k$. If document i is labeled, the main difference with the unlabeled data case is that Z_i has been hand-coded, and as a result, we do not draw it from a Bernoulli distribution, but the rest of the model's structure remains the same. A key feature of our method is that the key parameters are interpretable.

[10] 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 (the main category of interest). For more details, see SI B, C, and D.

For example, π is the probability that a document belongs to the first class, and $\boldsymbol{\eta}$ is the probability of observing a word given the class of the document. In Section Active Keyword Upweighting, we show how to take advantage of this interpretability to augment the model.

The scarcity of labeled data compared to the abundance of unlabeled data is a significant challenge in implementing semi-supervised learning approaches. 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 down-weight information from unlabeled documents by utilizing a decision factor, $\lambda \in [0, 1]$ (Nigam et al. (2000)). 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.

Finally, because the observed data log-likelihood of our model is difficult to maximize, we use the EM algorithm to estimate the parameters.[11]

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 (see section Active Learning). 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.

Estimation

In the first iteration, the model is initialized with a small number of labeled documents.[12] The information from these documents is used to estimate the parameters of the model: the probability of a document being of class 1, π, and the probability of generating each word given a class, the $V \times 2$ matrix $\boldsymbol{\eta}$. 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

[11] For a full derivation of the EM algorithm, see SI B.

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

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
```

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-likelihood of unlabeled documents being down-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 posterior probability that each unlabeled document belongs to class 1.

Selection

Using the predicted probability that each unlabeled document belongs to class 1, 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 of the class 1 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 ⟶ [Political]
>
> Argentina Wins the 2022 FIFA World Cup, Defeating France ⟶ [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. We use the out-of-sample validation stopping rule with a bound of 0.01 for the F1 score in our reanalyses 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.[13] 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,[14] 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.[15] 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.[16]

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

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

[15] 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.

[16] See SI I.1 for more discussion about the consequences of mislabeling keywords.

Using the set of keywords for each class, *activeText* creates a $V \times 2$ keyword matrix $\boldsymbol{\kappa}$ where each element $\kappa_{v,k}$ 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.[17] Finally, we show how keyword upweighting can improve classification accuracy.

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).[18] 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.[19]

Classification Performance

Figure 2 shows the results from three model specifications: *activeText* (denoted by the solid line), a version of *activeText* that uses random sampling instead of active sampling (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.[20] 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 iterations in the case

[17] We trained the BERT models using Nvidia V100 GPUs on an HPC platform.

[18] More information about preprocessing and descriptions about the dataset are in SI E

[19] See SI A.3 for a detailed description of the F1-score and other commonly used model evaluation metrics.

[20] See SI E for how we generate validated data with class-imbalance.

of the *activeText* models and the DistilBERT model, respectively. In the *activeText* models, 20 documents are labeled in each iteration.[21]

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.[22] The exception is the Human Rights corpus, where the benefit of active learning is marginal, and where at the 0.50 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 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 0.05 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 0.50 and Population levels.[23]

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 word-class parameter $\boldsymbol{\eta}$ 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

[21] While we simulate human coders who label all documents correctly at the labeling stage, this may not be the case because humans can make mistakes in practice. SI I.2 shows that honest (random) mistakes in the labeling of documents can hurt the classification performance.

[22] In SI G, we further examine how the class imbalance influences the benefit of active learning by varying the proportion of the positive class between 5% and 50%.

[23] Figure G.1 in SI G includes comparisons of our generative approach, *activeText*, in terms of predictive performance against Support Vector Machines (SVM), a popular discriminative method used for classification tasks. For a discussion of generative vs. discriminative models, see SI A.2.

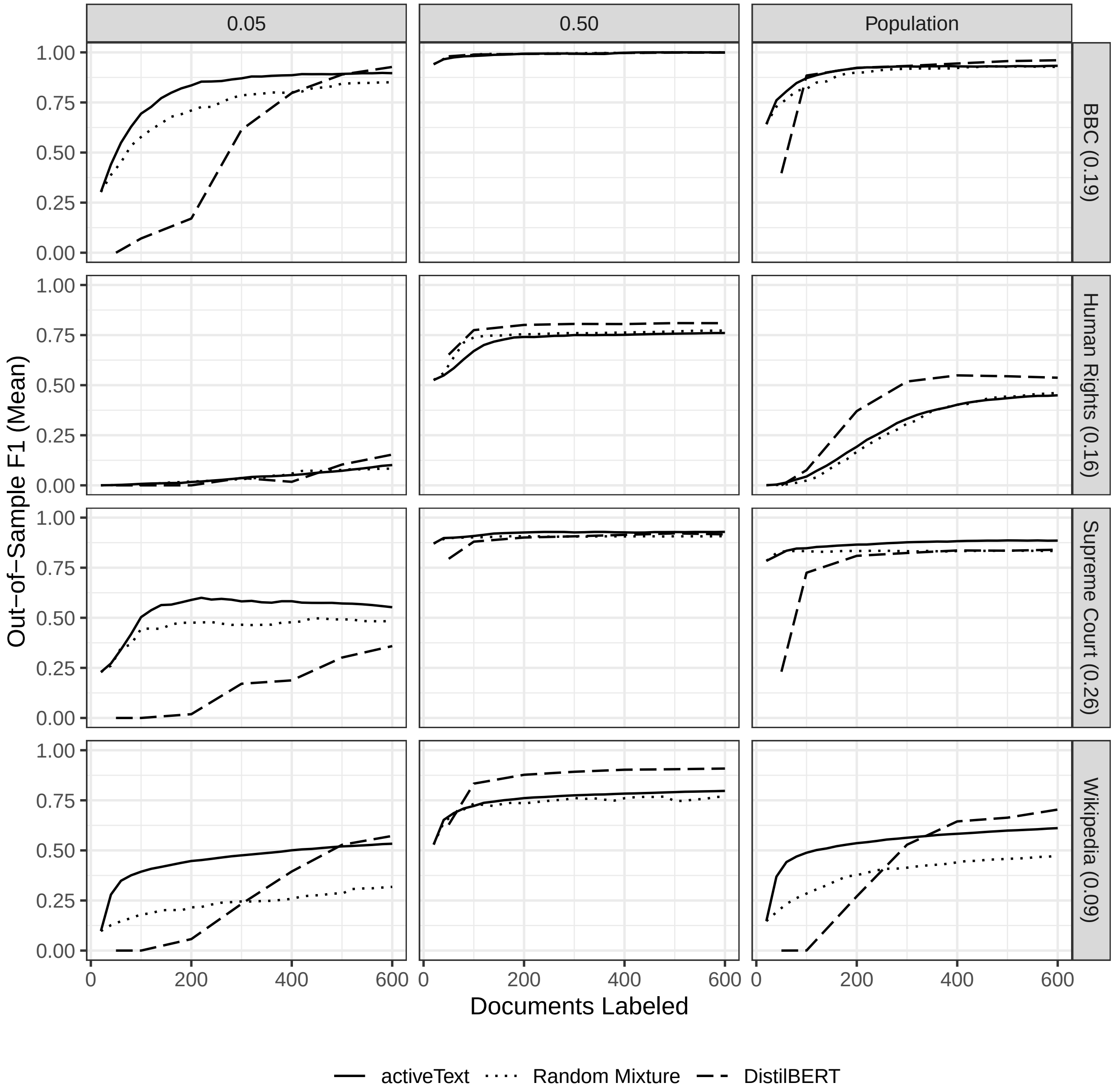

Figure 2: **Comparison of Classification Results with Random and Active Versions of *activeText* and DistilBERT**

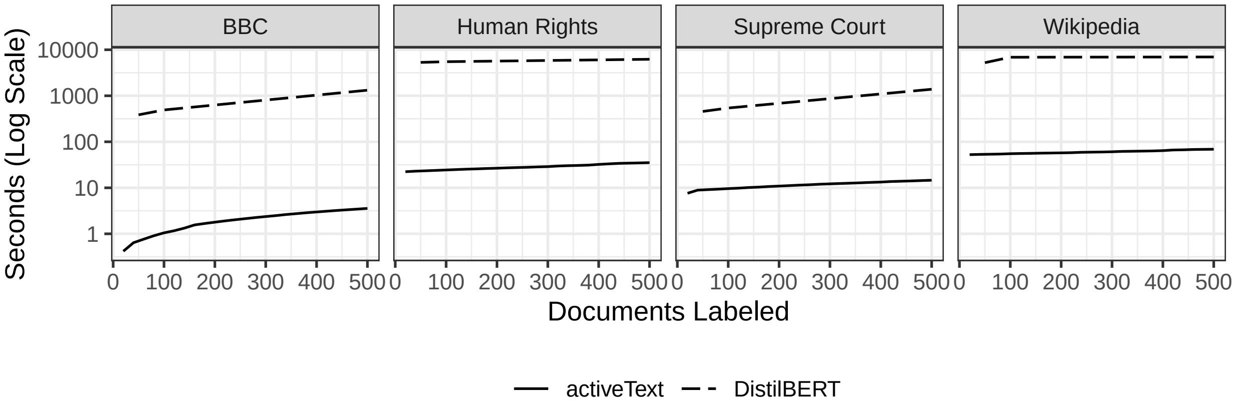

Figure 3: **Comparison of Classification and Time Results across *activeText*and DistilBERT**

Air with 8 GB of RAM and 7 GPU cores. While the *activeText* models were trained using a single CPU, we used the recent implementation of support for the GPU in M1 Macs in PyTorch[24] to parallelize the training of the BERT model using the M1 Mac's GPU cores.[25] 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. 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).

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

[25] 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.

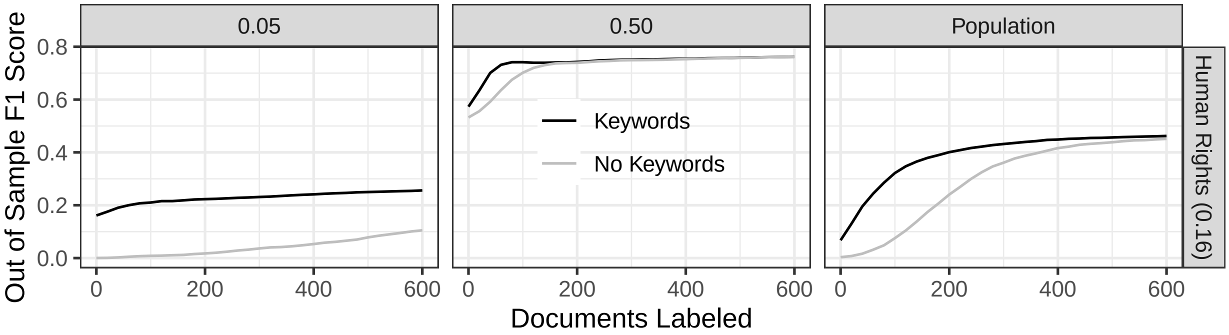

Figure 4: **Classification Results of *activeText* with and without Keywords**

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.[26] This means that the information the models can learn from labeled documents is less compared to the other corpora with longer documents.[27] In situations like this, providing keywords in addition to document labels can improve classification performance because it directly shifts the values of the word-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 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

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

[27] In our simulation studies described in Section Simulations, we confirmed that the classification performance is poor when the document length is short. Please refer to Supplementary Appendix for the Simulation Studies for the full set of results.

mixture model with the complete set of labeled documents.[28]

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.[29]

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 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, 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 I.1, 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.

Simulations

To complement the evidence from our validation using real-world labeled data, we conducted a series of simulation studies involving 162 different configurations. Recently, Farquhar et al. (2021) discuss unresolved issues, tradeoffs, and complexities related to the in-sample statistical bias arising from active learning, and its impact on out-of-sample classification performance. To examine whether the in-sample statistical bias has adverse effects on the out-of-sample classification performance, in our simulation studies, we manipulated various aspects of the simulated data, such as the size of the corpus (number of documents), the size of the vocabulary, 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.

Overall, our findings demonstrate that *activeText* is effective when dealing with imbal-

[28] 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.

[29] SI H demonstrates how active keyword works by visualizing the word-class matrix, $\boldsymbol{\eta}$, at each active iteration.

anced data and a limited number of manually labeled documents. Additionally, we observe that the proposed approach's predictive performance is not significantly affected by any potential bias resulting from labeling the most uncertain cases first. These findings validate our previous conclusions obtained through the validation studies above (using actual labeled documents), indicating that *activeText*, especially in the early stages of the labeling process, outperforms alternative methods. Moreover, our results show that as the number of hand-labeled documents increases, both in active and passive sampling schemes, the mean squared error (MSE) of $\boldsymbol{\eta}$ and π decreases. Although active learning at earlier stages results in a slightly larger MSE for π due to the possible bias induced by labeling the most uncertain cases first, it is this process of effective annotation that ultimately appears to improve predictive performance.[30]

Reanalysis with Fewer Human Annotations

To further illustrate the benefits of our proposed approach for text classification, we conduct reanalyses of 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

[30] For more detailed information and a comprehensive summary of the simulation results, we direct interested readers to the SI M and the Supplementary Appendix for the Simulation Studies, respectively.

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 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 in Active Learning, 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

[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.

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 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 in SI K 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. Gohdes (2020) finds that in the Alawi region (known to be loyal to the regime) when Internet access is at its highest, the expected proportion of targeted killings is significantly smaller compared to other regions of 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). As noted above, these gains come from our active sampling scheme as it can select the most informative documents to be labeled.

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.

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

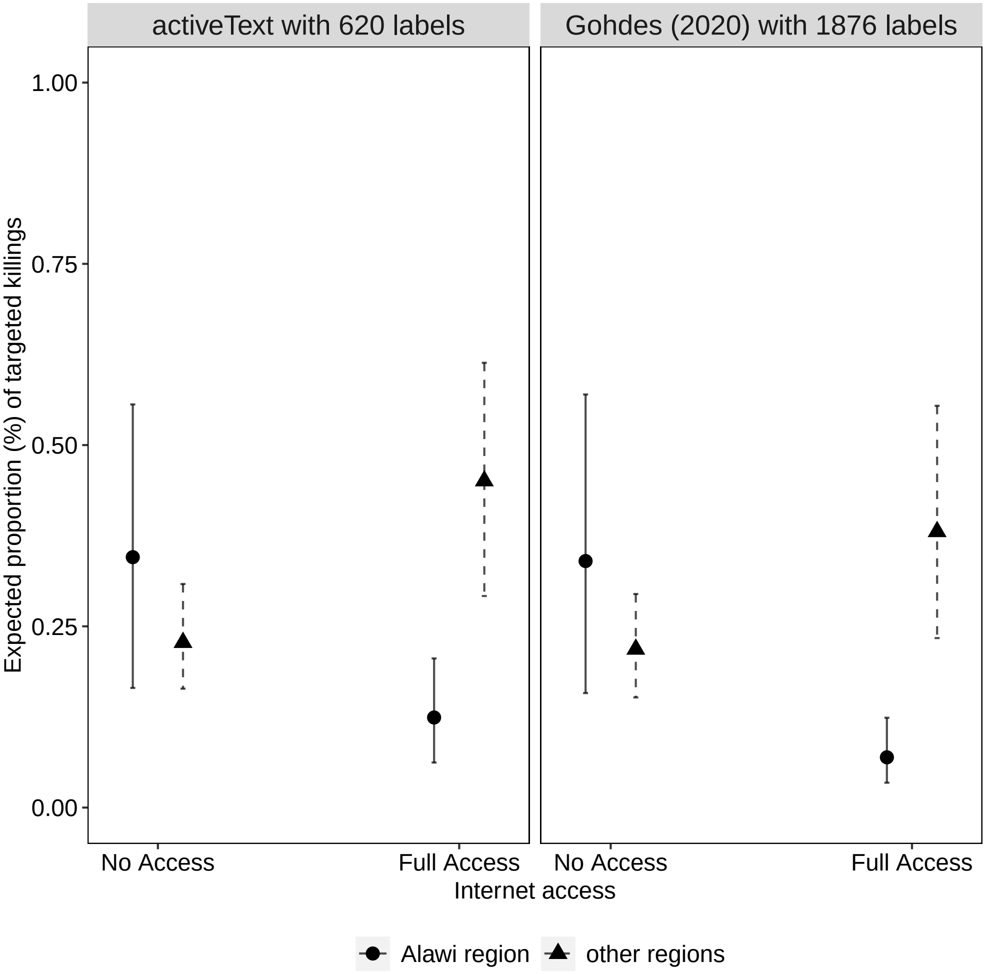

Figure 5: **Replication of Figure 3 in Gohdes (2020): Expected Proportion of Target Killings, Given Internet Accessibility and Whether a Region is Inhabitated by the Alawi Minority.** The results from *activeText* are presented in the left panel and those of Gohdes (2020) are on the right.

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 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.

[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.

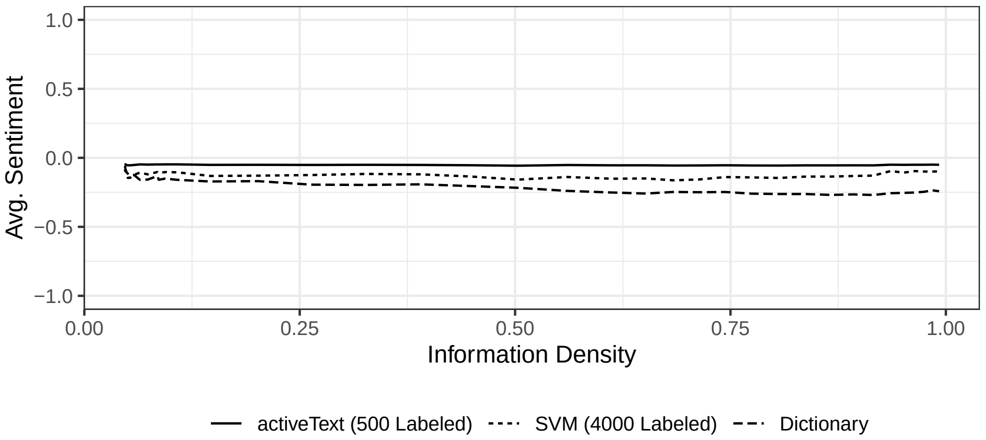

Figure 6: **Replication of Figure 1 in Park et al. (2020): The Relationship Between Information Density and Average Sentiment Score.**

As in the replication 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 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. 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 (see Figure L.1 in SI L).[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 L.2 of SI L, 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

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.[37] However, cross-validation may not be practical when the labeled data is scarce (or absent at the beginning of the process). We have observed across a variety of applications that very small values (e.g., 0.001 or 0.01) work the best on the corpora we used (see SI F). However, more work is needed to clearly understand the optimality criteria needed to select λ. We leave this question for future research.

Labeling Error

While our empirical applications assume that labelers are correct, human labelers do make mistakes. In SI I, we examine how mislabeling keywords and documents affect classification performance. Our results show that, if compared to the no-keyword approach, a small amount of random noise (classical measurement error) on keyword labeling does not hurt the classification performance. In contrast, random perturbations from true document labels do hurt the classification performance. A promising avenue for future research should center on developing new active learning algorithms that assign labelers based on their labeling ability and/or are robust to more pervasive forms of labeling error (differential and non-differential measurement error). For instance, assigning the most competent labelers with the most uncertain or difficult documents and assigning the least competent labelers with easier documents can optimize the workload of the labelers. At the same time, we note that users may be able to improve the quality of human labeling by other means, such as clarifying category concepts and better training of coders, in practical settings.

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

[37] Indeed, it may be beneficial to tune the lambda value *across* active learning iterations.

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 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. Towards this, we continue to work towards publishing the R package *activeText* on CRAN. We believe that our model 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.

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.

Cohn, D., Atlas, L., and Ladner, R. (1994), "Improving generalization with active learning," *Machine Learning*, 15(2), 201–221.

Cordell, R., Clay, K. 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.

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.

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*

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.

Hanneke, S. (2014), “Theory of Disagreement-Based Active Learning,” *Foundations and Trends in Machine Learning*, 7(2-3), 131–309.

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.

Hino, H. (2021), “Active Learning: Problem Settings and Recent Developments,” *Journal of the Japan Statistical Society, Japanese Issue*, 50(2), 317–342.

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.

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*

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.

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.

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

Lowande, K. (2019), "Politicization and Responsiveness in Executive Agencies," *The Journal of Politics*, 81(1), 33–48.

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.

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.

Spirling, A. (2012), "US treaty making with American Indians: Institutional change and relative power, 1784–1911," *American Journal of Political Science*, 56(1), 84–97.

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.

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