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

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

First draft: September 10, 2020
This draft: December 11, 2021

Abstract

When using text data, social scientists often classify documents to use the resulting document labels as an outcome or predictor. Since it is costly to label documents manually, automated text classification has become a standard tool. However, current approaches for text classification do not take advantage of all the data at one's disposal. We propose a new model for text classification that combines information from both labeled and unlabeled data with an active learning component, where a human iteratively labels documents that the algorithm is least certain about. Using text data from Wikipedia discussion pages, BBC News articles, historical US Supreme Court opinions, and human rights abuse allegations, we show that our model improves performance relative to classifiers that (a) only use information from labeled data and (b) randomly decide which documents to label at the cost of manually labelling a small number of documents.

[*] We thank Ken Benoit, Yaoyao Dai, Chris Fariss, Walter Mebane, Kevin Quinn, and audiences at the 2020 Annual Meeting of the American Political Science Association, 2021 Annual Meeting of the Midwest Political Science Association, 11th Annual Conference on New Directions in Analyzing Text as Data, seminar participants at the University of Michigan, 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.

1 Introduction

As the amount and diversity of available information have rapidly increased, social scientists often resort to multiple forms of data to answer substantive questions.[1] In particular, the use of text-as-data in cutting-edge social science research has exploded over the past decade. Document classification has been the primary task in political science, with researchers classifying documents such as legislative speeches (Peterson and Spirling, 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), treaties (Spirling, 2012), religious speeches (Nielsen, 2017), and human rights text (Cordell et al., 2021; Greene et al., 2019).

Two types of classification methods are commonly used: supervised and unsupervised algorithms. Supervised approaches use associations between word frequencies and labels from a set of hand-coded documents to categorize unlabeled documents, whereas unsupervised schemes cluster documents without needing labeled documents. Both of these methods have downsides, however: in the former, hand-coding documents is labor-intensive and costly, requires expert knowledge and reconciliation of disagreements between coders to ensure label validity; in the latter, the substantive interpretation of the categories discovered by the clustering process can be difficult, and performance is severely threatened when the data lacks the necessary structure such that signal can be distinguished from noise.

Active learning is a technique that reduces the cost of hand-coding in the supervised approach. It uses measures of label uncertainty to iteratively flag highly informative documents, and has been shown to reduce the number of labeled documents needed to train an accurate classifier, particularly when the proportion of the document class in the data is very low (Miller et al., 2020). However, current implementations of active learning have only been used to augment supervised approaches. That is, in each iteration of an active learning algorithm, only labeled documents are used to train the classifier that indicates which documents should be labeled.

Our innovation is to augment active learning with unsupervised clustering, exploiting the benefits of both approaches to improve the performance of text classifiers. We extend the mixture model from Nigam et al. (2000) to combine the information from both labeled and unlabeled documents within an active learning framework. In the model, latent clusters are *observed* as labels for labeled documents and *estimated* as a latent variable for unlabeled documents, and active learning iteratively provides observed labels for the documents that the cluster estimates are most uncertain about. We show that our model outperforms Support Vector Machines (SVM), a popular supervised learning model, when both models are using active learning to choose which documents to label. Furthermore, because our model is generative, it is straightforward to use a researcher's domain expertise to improve classification. As an example, we show that classification performance is improved by iteratively upweighting keywords that the researcher identifies as being

[1] See e.g., Grimmer and Stewart (2013) for an excellent overview of these methods in Political Science. See also Appendix A.

highly associated with one of the possible document labels.

We provide a library for the statistical language *R* called *activeText* with the goal of providing researchers from all backgrounds easily accessible tools to minimize the amount of hand-coding of documents and improving the performance of classification models for their own work.

This research note proceeds as follows. In Section 2, we describe both the semi-supervised and active learning components of our model. In Section 3, we show the results from comparing our model to a popular alternative. In Section 5, we review the results, and discuss directions for future research. For an accessible primer on the use and interpretation of machine learning models for text classification, see the online Appendix B.

2 The Method

2.1 Model

Consider the task of classifying N documents as one of two discrete classification options. 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, to represent the latent cluster assigned to each document. If a document i is assigned to the k th cluster, $Z_i = k$. Without loss of generality, we assume that $K = 2$, however, the mixture model behind our approach is quite flexible and can be extended to encompass more than two clusters (see Appendix sections C and D).

The following equations summarize our generative model:

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

Where we first draw $\pi = p(Z_i = 1)$, the probability that any given document belongs to the positive class, from a Beta distribution with hyperparameters α_0 and α_1.Given π, for each document indexed by i, we draw from a Bernoulli distribution the latent cluster assignment indicator Z_i. Then, we draw features for document i from a multinomial distribution governed by the vector $\boldsymbol{\eta}_{\cdot k}$, where $\eta_{vk} = p(D_{iv} | Z_i = k)$, whose prior is the Dirichlet distribution.

We use the EM algorithm to estimate the parameters.[2] One important note on the estimation is that we down-weight information from unlabeled documents. The objective function to be maximized by the EM algorithm consists of the log prior, the log likelihood of labeled data, and the log likelihood of unlabeled data. Without weight, information from unlabeled data will dominate information from labeled data because the number of documents in the former is much larger than the latter typically. Therefore, we weight the log likelihood of unlabeled data by $\lambda \in [0, 1]$

[2] For a full derivation of the EM algorithm, see Appendix C.

following Nigam et al. (2000). When the λ is equal to 1, the model treats each document equally, regardless of whether the document is labeled deterministically by a human, or probabilistically by the algorithm. As λ moves from 1 towards 0, the model increasingly down-weights the information that the probabilistically labeled documents contribute to the estimation of $\boldsymbol{\eta}$ and π, such that when λ is 0, the model *ignores* all information from the probabilistically labeled documents and therefore becomes a supervised algorithm (see Appendix C).

2.2 Active Learning

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 then iterates until one of three stopping conditions are met: (1) the model runs out of unlabeled documents to label; (2) the remaining unlabeled documents do not meet a particular uncertainty threshold; or (3) the maximum number of allowed active steps is reached. 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.

2.2.1 Estimation

In the first iteration, the model is initialized with a small number of labeled documents.[3] The information from these documents is used to estimate the parameters of the model: the probability of a document having a positive label π, and the probability of generating each word given a class, the $V \times 2$ matrix $\boldsymbol{\eta}$. From the second iteration on, we use information from both labeled and unlabeled documents to estimate the parameters using 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 the positive class.

2.2.2 Selection

Using the predicted probability that each unlabeled document belongs to the positive class, we use Shannon Entropy to determine which of the probabilistically labeled documents that it was least certain about. In the binary classification case, this is the equivalent of calculating the absolute value of the distance of the positive class probability and 0.50 for each document. Using this criteria, 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.

[3] 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 the predicted classes of all documents at least with some certainty.
Initialize $\mathbf{D}^l_{old}$ by sampling some documents randomly, and have humans label them ;
Initialize $\mathbf{D}^u \leftarrow \mathbf{D} \setminus \mathbf{D}^l_{old}$;
[Active Keyword]: Initialize keyword matrix $\boldsymbol{\kappa}$, where each element $\kappa_{v,c}$ takes the value
  of $\gamma$ if the word $v$ is a keyword for class $c$, otherwise 0.
while Not all documents are classified with some certainty yet do
    (1) [Active Keyword]: Up-weight elements of the $\boldsymbol{\beta}$ prior using $\boldsymbol{\kappa}$;
        $\boldsymbol{\beta} \leftarrow \boldsymbol{\beta} + \boldsymbol{\kappa}$;
    (2) Predict labels for each document in $\mathbf{D}^u$ using EM algorithm;
    (3) Sample $n$ most uncertain documents in $\mathbf{D}^u$ and have humans labels them;
        $\mathbf{D}^l_{new} \leftarrow n$ most uncertain documents in $\mathbf{D}^u$;
    (4) [Active Keyword]: Sample $m$ non-keywords most associated with each class and
      have humans label to create $\boldsymbol{\kappa}_{new}$;
        $\boldsymbol{\kappa} \leftarrow \boldsymbol{\kappa}_{new}$;
    (5) Update labeled and unlabeled documents;
        $\mathbf{D}^l \leftarrow \mathbf{D}^l_{old} \cup \mathbf{D}^l_{new}$;
        $\mathbf{D}^u \leftarrow \mathbf{D} \setminus \mathbf{D}^l_{old}$
end
```

2.2.3 Labeling

A human coder reads each document selected by the algorithm and imputes the 'correct' label. The newly-labeled documents are then added to the set of human-labeled documents, and the process is repeated from the estimation stage.

2.2.4 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 to the mixture model by incrementing elements of the $\boldsymbol{\beta}$ matrix associated with a keyword for a given class by γ, a scalar value chosen by the researcher (see Eshima et al. 2020 for a similar approach for topic models). To build the list of keywords associated with each class, the researcher is queried after a set interval of active learning iterations to label a set of candidate words as keywords or not.[4] To select candidate keywords with the active keyword approach, we calculate the log ratio that each word was generated by a particular class using the $\boldsymbol{\eta}$ parameter from the mixture model, and choose m words (that are not already part of the set of keywords) with the most extreme ratio for each class. Using the set of keywords for each class, we create a $N \times C$ keyword matrix $\boldsymbol{\kappa}$ where each element $\kappa_{v,c}$ takes the value of γ if word v is a keyword for class c, otherwise 0. Before we fit the mixture model in each active iteration, we perform a matrix sum $\boldsymbol{\beta} \leftarrow \boldsymbol{\kappa} + \boldsymbol{\beta}$ where $\boldsymbol{\beta}$ is the $N \times C$ matrix that summarizes the prior information about the likelihood that a given word is associated with a given class. The keyword approach therefore effectively upweights our model with prior information about words

[4] The researcher may combine also provide an initial set of keywords, and then iteratively labeling candidate words to add to the set of keywords.

that we think are likely to be associated with one class rather than another.

3 Applications

This section first shows the performance comparisons between active vs passive learning (random sampling at each active step) as well as semi-supervised learning vs supervised learning, when classifying documents for the following datasets: internal forum conversations of Wikipedia editors (positive class: toxic comment), BBC News articles (political topic), the United States Supreme Court decisions (criminal procedure), and Human Rights allegation (physical integrity rights allegation). It also shows that different specifications of our methods can further improve the performance depending on various data structures.

3.1 Results

Figure 1 shows the results from four model specifications, each represents one of the combinations of active or passive learning, and semi-supervised or supervised learning. The first choice is between active learning (solid lines) vs passive learning (dashed lines). In the active sampling, we select the next set of documents to be labeled based on the entropy of the predicted probabilities of the classes when we use our mixture model, and they are selected based on the margin sampling when we use SVM as the underlying classification method. The second choice is between our semi-supervised learning (darker lines) vs an off-the-shelf supervised learning (lighter lines). For the supervised learning, we replicate the results from Miller et al. (2020) which uses SVM as the classifier. The rows correspond to different datasets and the columns correspond to various proportion with positive label documents in the corpus. The y-axis indicates the average out-of-sample F1 score (the harmonic mean of precision and recall) across 100 Monte Carlo iterations, and the x-axis shows the number of sampling steps. We label 20 documents at each sampling step.

Among the four models, the combination of active learning with the mixture model (*Active Mixture* in Figure 1) performs the best with most of the specifications. The results on the Wikipedia corpus with 5% and 9% (population) positive labels and on the Supreme Court corpus with 5% positive labels highlight this most clearly. With other specifications, *Active Mixture* performs slightly better or as well as other models. The gain from active learning tends to be higher when the proportion of positive labels are small. When the proportion of the positive labels are 5%, active learning outperforms passive learning consistently. An exception is on the human rights corpus where active learning did not improve the performance even when the proportion of positive labels is 5%, and this is the case for both SVM and our mixture model. By contrast, *Active Mixture* tends to perform better than SVM at the initialization, though the difference shrinks as the number of labeled documents increase. The results on BBC corpus with 5% and 19% (population), Wikipedia, and Supreme Court highlight this point. The *Random Mixture* (dashed dark line) is above the *Random SVM* (dashed light line) at the beginning but they converge later. This makes sense because the relative contribution from unlabeled documents in the mixture model decreases as the size of labeled documents increases.

We also test the effect of implementing the keyword upweighting scheme described above. In Figure 1, active learning did not improve the performance on the human rights corpus even when the proportion of positive labels is 5%, and the F1 score was also lower than other corpora. One reason for the early poor performance of *Active Mixture* may be length of each document. Because each document of the human rights corpus consists of one sentence only, the average length of each document is shorter than other corpora.[5] This means that the information the models can learn from adding one labeled document is less in the human rights corpus compared to the other corpora. In situations like this, providing keywords in addition to document labels will be effective in improving the classification performance because it directly shifts the values of the word-class probability matrix, η, even when the provided keywords is not included in the labeled documents.

Figure 2 compares the performance with and without providing keywords. The darker lines show the results with keywords and the lighter lines without. 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 η_{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.[6] The results suggest that providing keywords improves the performance when the proportion of positive labels is small. The keywords scheme improved the performance on the corpus with 5% or 16% (population) positive labels while it did not on the corpus with 50% positive labels.

Figure 3 illustrates how the word-class matrix $\boldsymbol{\eta}$ is updated with and without keywords across iterations. A subset of the keywords supplied are labeled and highlighted by black dots. The x-axis shows the log of η_{v1}/η_{v0}, where η_{v1} corresponds the probability of observing the word v in a document with a positive label and η_{v0} for a document with a negative label. The high value in x-axis means that a word is more strongly associated with positive labels. The y-axis is the log of word frequency. A word with high word frequency has more influence in shifting the label probability. In our mixture model, words that appear often and whose ratio of η_{vk^*} vs η_{vk} is high play a central role in the label prediction. By shifting the value of $\boldsymbol{\eta}$ of those keywords, we can accelerate the estimation of $\boldsymbol{\eta}$ and improve the classification performance.

One caveat is that we provided "true" keywords, in the sense that we used the estimated $\boldsymbol{\eta}$ from fully labeled dataset. In practice, researchers have to come up with the keywords using their prior substantive knowledge about the corpus. However, we believe that the keywords supplied to our

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

[6] 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.

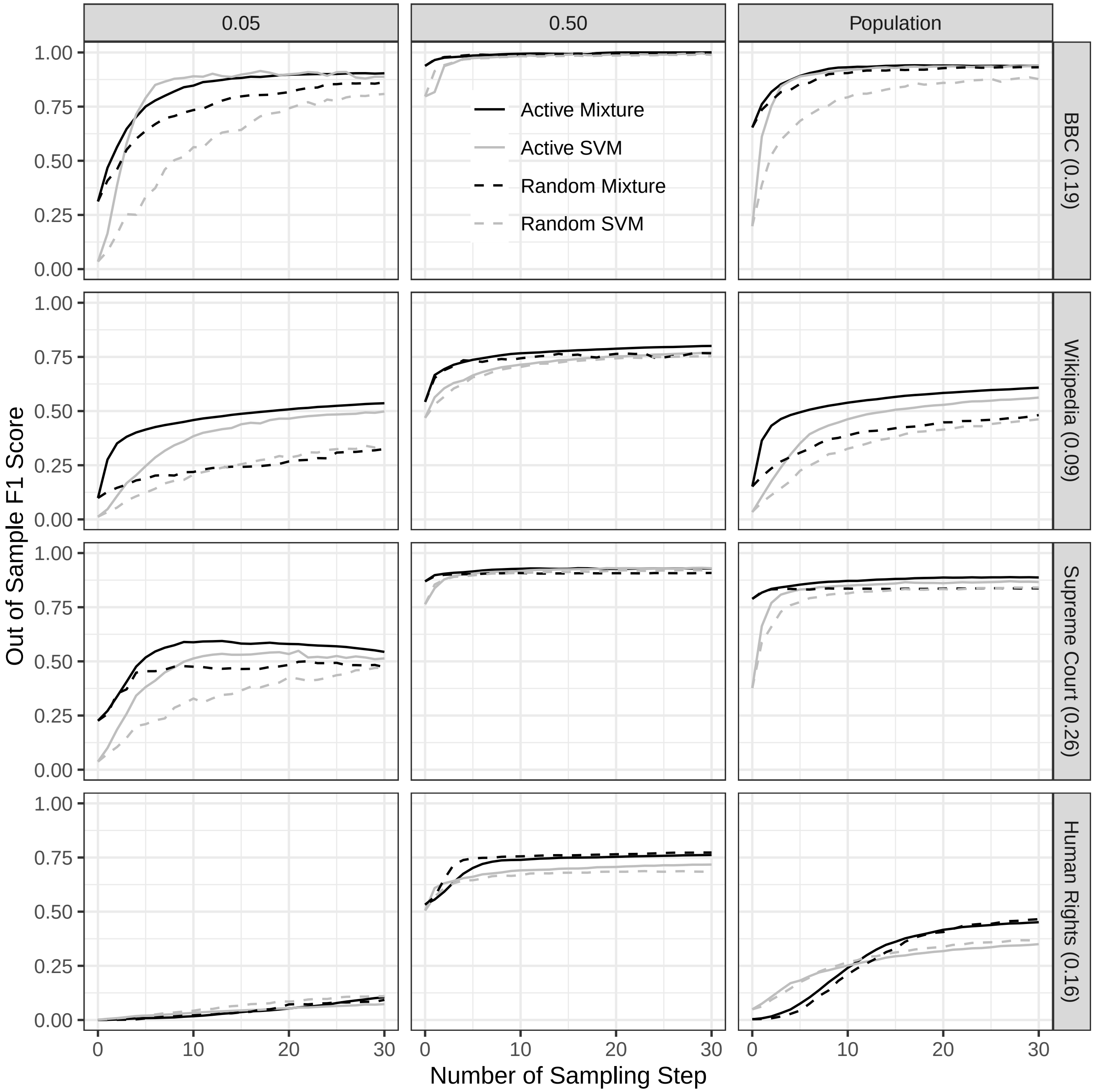

Figure 1: **Comparison of Classification Results across Active Mixture, Active SVM, Random Mixture, and Random SVM**

simulation are what researchers with reasonable substantive knowledge about physical integrity right can come up easily. Indeed, the "true" keywords used in this simulation, such as "torture," "beat," and "murder," match our substantive understanding of physical integrity right violation.

4 Discussion

4.1 Tuning the value of λ

The first practical question is how to choose the value of λ, which is the weight on the unlabeled documents relative to the labeled documents. Recall that we downweighted the information from unlabeled documents since we typically have much more unlabeled documents than labeled documents, and we want to rely more on the information from the labeled documents for prediction.

An important practical questions is how to select the value of λ that maximizes the performance. It is possible that we can adopt popular model selection methods (e.g. cross-validation) to choose the appropriate λ value during the model initialization process.[7] However, cross-validation may not be practical when the labeled data is scarce (or absent at the beginning of the process). Using our active learning approach is particularly, we have observed across a variety of applications that very small values (e.g., 0.001 or 0.01) seem to work the best on the corpora we used (see Appendix E).

4.2 Labelling Error

While our empirical applications assume that labellers are always correct, human labellers do make mistakes in labelling in reality. Future research can address this point by comparing the robustness to labelling error between supervised vs semi-supervised classifiers. Another direction is to develop a new active learning algorithm that assign labellers based on their labelling ability. For instance, assigning the most competent labellers with the most uncertain or difficult documents will increase the marginal benefit of adding one label.

5 Conclusion

In this paper we have described a new active learning algorithm that combines information from labeled and unlabeled documents in order to better select which documents to be labeled by a human coder. We have shown that across three diverse datasets, our model almost always outperforms the active SVM algorithm , and that when we use the λ to appropriately downweight the information the model learns from unlabeled data, we frequently outperform the active Naive Bayes baseline as well.

Machine learning techniques are becoming increasingly popular in Political Science, but frequently 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*

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

on CRAN. We believe that our model will provide applied researchers a tool that they can use to efficiently categorize documents in corpuses of varying sizes and topics.

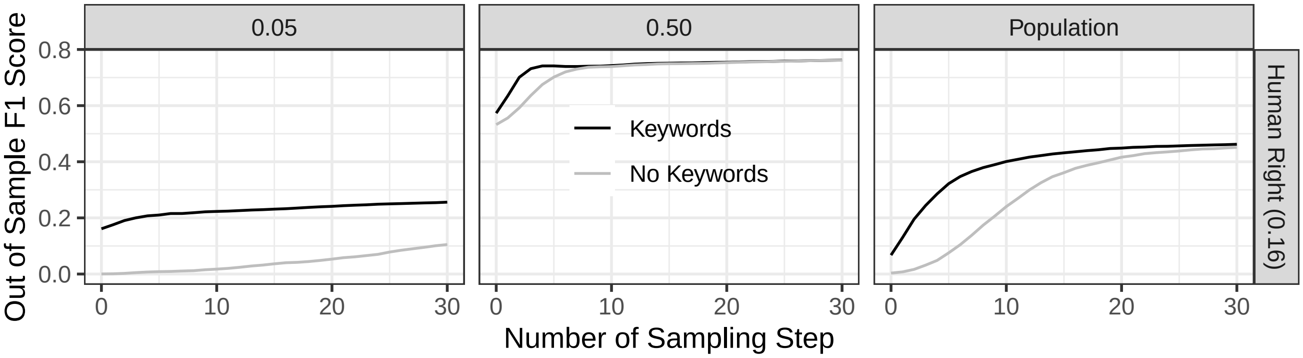

Figure 2: **Classification Results with and without Keywords**

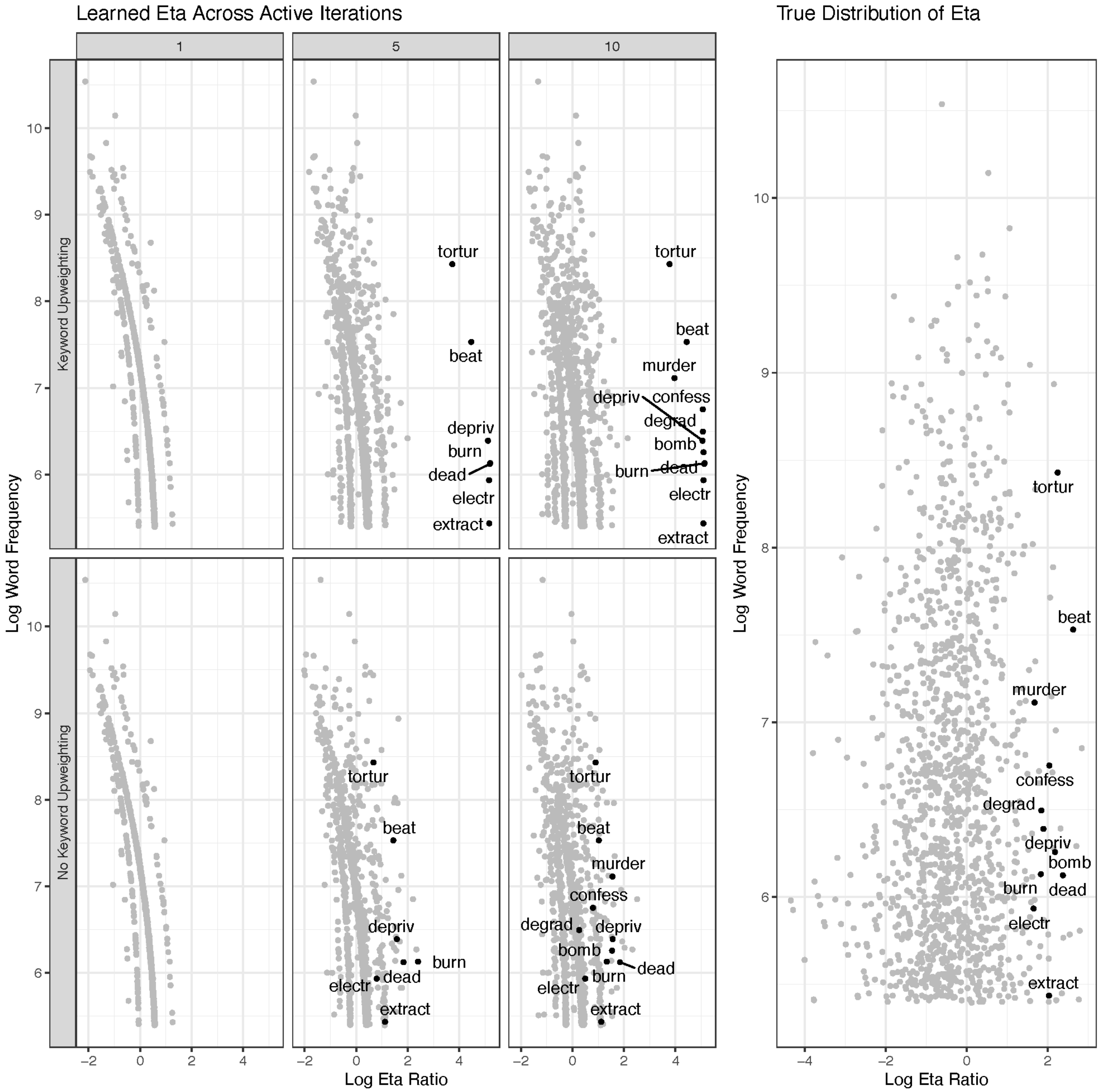

Figure 3: **Update of the Word-class Matrix (η) with and without Keywords**

References

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

Boydstun, A. E. (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.

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

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

Greene, K. T., 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., and Stewart, B. (2013), "Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts.," *Political Analysis*, 21(3), 267–297.

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

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.

Nielsen, R. A. (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.

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.

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