Supplementary Information for 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

Contents

[*] 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.

A Table of Political Science papers that use text classification

	Class	Methods
Public Statements		
Airoldi et al. (2007)	Authorship	Poisson and Negative Binomial models
Stewart and Zhukov (2009)	Activist tone	Emsemble
Gillion (2016)	Race-related discourse	Emsemble
Legislative Speeches		
Peterson and Spirling (2018)	Party identification	SGD, Passive Agressive, Penalized Logsitic
Diermeier et al. (2012)	Ideology	SVM
News Articles		
Boydstun (2013)	Policy issue coverage	Manual coding
Election Manifestos		
Catalinac (2016)	Particularistic policy	Topic model
Social Media Posts		
Lopez et al. (2017)	Leave or Remain EU	SVM
King et al. (2017)	Issue category	ReadMe (Hopkins and King, 2010)
Treaties		
Spirling (2012)	Harssness	PCA
Religious Speeches		
Nielsen (2017)	Jihadist	Structural Topic Model (Roberts et al., 2013)
Human Rights Text		
Cordell et al. (2021)	Rights allegation	SVM, Naive Bayes, Logistic, Emsamble
Greene et al. (2019)	Political Terror Score	SVM, Naive Bayes, Logistic, Random Forest, Emsamble

B Using Machine Learning for Text Classification

In this section we introduce readers to several basic concepts in machine learning: the difference between supervised and unsupervised learning, between discriminative and generative models, and between active and passive learning.

Suppose that a researcher has a collection of social media text data, called a corpus, and wishes to classify whether each text in a corpus is political or not solely on the basis of the words used in a given observation. Critically, the researcher does not yet know whether any of the texts are political or not at this point. Suppose further that the researcher has chosen some scheme for translating the corpus into a matrix $\mathbf{X}$ with n rows and m columns, where n is the number of observations and m is the number of features,[1]and that there exists a vector of true labels Y, where each element of Y indicates whether a given document is political or not. Then, we can repose the classification question as follows: given the matrix of text data $\mathbf{X}$, how might we best learn Y, that is, whether each document is political or not?

B.1 Supervised vs. Unsupervised Learning

One of the first decisions that a researcher must make is whether to use a supervised or unsupervised approach to machine learning. The supervised approach to this problem would be to (1) obtain true labels of the some of the documents using human coding; (2) learn the relationship between the text features encoded in the matrix $\mathbf{X}$ and the true label encoded in the vector Y for the documents with known labels[2]; and (3) using the learned association between the text data and the known labels, predict whether the remaining documents in the corpus (that is, those that were not coded by a human) are political or not.

In contrast, an unsupervised approach would *not* obtain the true labels of some of the documents. Rather, a researcher using an unsupervised approach would choose a model that *clusters* documents from the corpus that have common patterns of word frequency.[3] Using this model, the researcher would choose the number of discrete clusters to divide the corpus into, and learn the relationship between the the matrix of text data $\mathbf{X}$ and each of the possible clusters in order to assign each document to a cluster. Using the assignment of documents to clusters, the researcher would then use some scheme to decide which of the clusters corresponds to the actual outcome of interest: whether a document is political or not.

The main advantage of a supervised approach over an unsupervised approach is the direct interpretability of results because a well-defined measure of the concept of interest exists. Consequently, it does not include the step of translating the clustering of documents to the classification

[1] Note that in the machine learning literature, the concept typically described by the term “variable” is communicated using the term “feature”.

[2] That is, learn $P(Y_{\text{labeled}}|\mathbf{X}_{\text{labeled}})$. This can be accomplished with a variety of models, including e.g. linear or logistic regression, support vector machines (SVM), Naive Bayes, k-nearest neighbor, and many more

[3] Examples of clustering algorithms include K-Nearest Neighbor (KNN) and Latent Dirichlet Allocation (LDA).

of documents as political or not.[4] On the other hand, the main disadvantage of a supervised approach is that obtaining labels for the documents in the corpus is often costly. Researchers using an unsupervised approach instead will avoid this cost, since they do not require a set of labels *a priori*.

Semi-supervised methods have been developed to combine the strengths of supervised and unsupervised approaches, and are particularly useful in situations where there is a large amount of unlabeled data, and acquiring labels is costly. A semi-supervised model proceeds similarly to the supervised approach, with the difference being that the model learns the relationship between the matrix of text data $\mathbf{X}$ and the classification outcome Y using information from both the labeled and unlabeled data. How exactly the information from the labeled and unlabeled data is balanced varies depending on the model used. In general, though, because a supervised approach learns the relationship between the labels and the data solely on the basis of the labeled documents, a classifier trained with a supervised approach may be less accurate than if it were provided information from both the labeled and unlabeled documents.

B.2 Discriminative vs. Generative Models

In addition to choosing a supervised, unsupervised, or semi-supervised approach, a researcher must also choose whether to use a discriminative or generative model. When using a discriminative model (e.g., logistic regression, SVM, etc.), the goal is to directly estimate the probability of the classification outcomes Y given the text data $\mathbf{X}$.[5] In contrast, when using a generative model (e.g., Naive Bayes), learning the relationship between the Y and $\mathbf{X}$ is a two-step process. In the first step, the likelihood of the matrix of text data $\mathbf{X}$ and outcome labels Y is estimated given the data and a set of parameters θ that indicate structural assumptions about how the data is generated.[6] In the second step, the researcher uses Bayes' rule to calculate the probability of the outcome vector given the features and the learned distribution of the parameters.[7]

The main benefit of a generative rather than discriminative model is that the researcher can include information they know about the data generating process by choosing appropriate functional forms.[8] This can help prevent overfitting when the amount of data in a corpus is small.[9] Conversely, because it is not necessary to model the data generating process directly, the main benefit of a discriminative rather than generative model is simplicity. Discriminative models are

[4] In most political science applications of unsupervised learning techniques, the author either is conducting an exploratory analysis and is therefore uninterested in classification, or performs an *ad hoc* interpretation of the clusters by reading top examples of a given cluster, and on that basis infers the classification from the clustering.

[5] That is, directly estimate $p(Y|\mathbf{X})$.

[6] That is, $p(\mathbf{X}, Y|\theta)$ is directly estimated.

[7] That is, $p(Y|\mathbf{X};\theta)$.

[8] This is particularly true when the researcher knows that the data has a complicated hierarchical structure since the hierarchy can be incorporated directly into the generative model.

[9] Overfitting occurs when a model learns to predict classification outcomes based on patterns in the training set that do not generalize to the broader universe of cases to be classified. A model that is overfitted may predict the correct class with an extremely high degree of accuracy for items in the training set, but will perform poorly when used to predict the class for items that the model has not seen before.

therefore appropriate in situations where the amount of data in a corpus is very large, and/or when the researcher is unsure about the data-generating process.[10]

B.3 Active vs. Passive Learning

If the researcher in our running example decides to use a supervised or semi-supervised approach to predicting whether documents in their corpus are political or not, she must also decide whether to use a passive or active approach. As described in Section B.1, a researcher using a supervised (or semi-supervised) approach must choose to label some documents, and on the basis of the learned relationship between those documents and the classification outcome, predict whether the rest of the documents in the corpus are political or not. The difference between a passive and active approach to this process 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*).

An active approach is superior to a passive approach when (1) the information that some documents contribute to the model results in more accurate predictions than the information contributed by other documents would and (2) there is some way of predicting which unlabeled documents will provide the best information. When both of these conditions hold, an active approach will be more efficient than a passive one. Conversely, when either of these conditions does not hold (as when randomly selected documents provide as good or better information to the model as one chosen by a particular scheme), a passive approach is superior to an active one. Alternatively, if the active approach performs slightly better than the passive approach, but is computationally intensive and/or very time-consuming to run, then one may be better off using the passive approach.

Therefore, a good active learning scheme should be fast, and should reliably choose documents for labeling that provide more information to the model than a randomly chosen document. One of the most common active learning approaches is called *uncertainty sampling*, a process where documents are chosen for labeling based on how uncertain the model is about the correct classification category for each document in the corpus.[11] Thus, an active learning process using uncertainty sampling alternates between estimating the probability that each document belongs to a particular classification outcome, sampling a subset of the documents[12] that the model is most uncertain about for labeling, then estimating the probabilities again using the information from the newly labeled documents. This process continues until the researcher is satisfied with

[10] Another benefit of generative models is that they can yield better estimates of how certain we are about the relationship between the outcomes and features. This is the case when a researcher uses an inference algorithm like Markov Chain Monte Carlo (MCMC) that learns the entire distribution for each of the parameters, rather than only point estimates.

[11] This is just one of many possible approaches. See Settles (2011) for a broad review.

[12] Exactly how many objects to select and label at each active iterations is also a matter of debate. When both the document selection scheme and the model fit are fast, labeling a single document (that is, the document the model is most uncertain about) is optimal. On the other hand, if the model takes some time to fit, and there are a large number of documents that potentially need labeling, a batch approach is justifiable.

		Predicted	
		Yes	No
Actual	Yes	TP	FN
	No	FP	TN

Table 1: **Confusion Matrix:** A confusion matrix compares the results of a classification model to documents' true labels. The upper-left quadrant is the count of True Positives (TP), the number of documents that the model predicts are the positive classification outcome that are in fact labeled as such. Correspondingly, the bottom-right quadrant is the count of True Negatives (TN), the number of documents that the model predicts to be negative which are in fact labeled negative in the validation set. The upper-right and bottom-left quadrants provide counts of False Negative (FN) and False Positives (FP), respectively.

the predictions generated by the model.

B.4 Model Evaluation

But how does a researcher decide whether she is satisfied by the predictions generated by the model? In most circumstances, the best way to evaluate the performance of a classification algorithm is to reserve a subset of the corpus for validation, which is sometimes referred to as a validation and/or test set. At the very beginning of the classification enterprise, a researcher should put aside and label a set randomly chosen documents that the active learning algorithm does not have access to.[13] Then, after training the model on the remainder of the documents (often called the training set), the researcher should generate predictions for the documents in the validation set using the trained model. By comparing the predicted labels generated by the model to the actual labels, the researcher can evaluate how well the model does at predicting the correct labels.

A common tool for comparing the predicted labels to the actual labels is a *confusion matrix.* In a binary classification setting, a confusion matrix will be a 2 by 2 matrix, with rows corresponding to the actual label, and the columns correspond to the predicted label. Table 1 shows a confusion matrix. The upper-left quadrant is the count of True Positives (TP), the number of documents that the model predicts are the positive classification outcome that are in fact labeled as such. Correspondingly, the bottom-right quadrant is the count of True Negatives (TN), the number of documents that the model predicts to be negative which are in fact labeled negative in the validation set. The upper-right and bottom-left quadrants provide counts of False Negative (FN) and False Positives (FP), respectively. A false negative occurs when the model classifies a document as negative, but according to the validation set the document is classified as positive. Similarly, a false positive occurs when the model classifies a negative document as positive.

Using the confusion matrix, the researcher can calculate a variety of evaluation statistics. Some of the most common of these are accuracy, precision, and recall. Accuracy is the most

[13] It is important to use a set aside validation set for testing model performance, rather than a subset of the documents used to train the model, in order to avoid *overfitting*.

straightforward measure of model performance, as it is simply the proportion of documents that have been correctly classified. Precision is used to evaluate the false positivity rate, and is the proportion of the model's positive classifications that are true positives. As the number of false positives increase, precision decreases; conversely, as the number of false positives decrease, precision increases. Recall is used to evaluate the false negativity rate, and is the proportion of the actual positive documents that are true positives. As the number of false negatives increase, recall decreases, and *vice-versa*. Accuracy, precision, and recall can be formally calculated as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$

When the proportion of positive and negative labeled documents in a corpus is balanced, accuracy is an adequate measure of model performance. However, it is often the case in text classification exercises that the corpus is unbalanced, and the rate of positively labeled documents is low. When this is the case, accuracy does a poor job at model evaluation, and precision and recall should be considered. Consider the case when 99 percent of documents belong to the negative class, and 1 percent to the positive. A model which simply predicts that all documents belong to the negative class would have an accuracy score of 0.99, but would poorly suited to the actual classification task. Conversely, the precision and recall scores of model that predicts all negatives would be 0, which would accurately signal to the researcher that the model does a very poor job at classifying positive documents.

Precision and recall are not perfect measures of model performance, however. A model that classified all of the actual positives correctly (i.e., when there are no false positives) but classified all of the actual negatives incorrectly would get a perfect precision score. Similarly, a model that classified all documents as positive would have a perfect recall. Note, however, that in the case with the perfect precision score, recall would be extremely low. And in the case with the perfect recall score, precision would be extremely low. These examples illustrate the fact that there is a fundamental trade-off involved in controlling the false positivity and false negativity rates: you can have few false positives if you are content with an extremely high number of false negatives; and you can have few false negatives if you are content with an extremely high number of false positives.

Recognizing this trade-off, researchers often combine precision and recall scores in an effort to find a model that has the optimal balance of the two. One common way of combining the two is an F1 score, which is the harmonized mean of precision and recall. Formally, the F1 score is calculated as:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

The F1 score evenly weights precision and recall, and so a high F1 score would indicate that both the false negativity and false positivity rate are low. It is worth noting these evaluation measures (accuracy, precision, recall, and the F1 score) are computed using labeled data ("ground truth"), which in practice, are available only for a limited subset of the records. However, as described below, an additional advantage of probabilistic modeling is that it allows to estimate sample counterparts for these evaluation measures based on model parameters.

With all these concepts in mind, in the next section we describe our proposed approach, with a special focus on its flexibility to balance the tradeoffs of working with labeled and unlabeled data and informing parameter estimation via auxiliary information.

C Detailed explanations about the EM algorithm to estimate parameters

Let $\mathbf{D}^{lp}$, $\mathbf{D}^{ln}$ and $\mathbf{D}^{u}$ be the document feature matrices for documents with positive labels, documents with negative labels, and unlabeled documents, respectively. Also let N^{lp}, N^{ln} and N^{u} be the number of documents with positive labels, negative labels, documents without labels. Likewise, $\mathbf{C}^{lp}$ and $\mathbf{C}^{ln}$ be the vectors of positive and negative labels. Then, the observed likelihood is the following:

$$\begin{aligned}
&p(\pi, \boldsymbol{\eta} | \mathbf{D}, \mathbf{C}^{lp}, \mathbf{C}^{ln}) \\
&\propto p(\pi)p(\boldsymbol{\eta})p(\mathbf{D}^{lp}, \mathbf{C}^{lp}|\pi, \boldsymbol{\eta})p(\mathbf{D}^{ln}, \mathbf{C}^{ln}|\pi, \boldsymbol{\eta})\Big[p(\mathbf{D}^{u}|\pi, \boldsymbol{\eta})\Big]^{\lambda} \\
&= p(\pi)p(\boldsymbol{\eta}) \times \prod_{i=1}^{N^{lp}} p(\mathbf{D}_i^{lp}|Z_i = 1, \eta)p(Z_i = 1|\pi) \times \prod_{i=1}^{N^{ln}} \Big\{p(\mathbf{D}_i^{ln}|Z_i = 0, \eta)p(Z_i = 0|\pi)\Big\} \\
&\quad \times \left[\prod_{i=1}^{N^{u}} \Big\{p(\mathbf{D}_i^{u}|Z_i = 1, \boldsymbol{\eta})p(Z_i = 1|\pi) + p(\mathbf{D}_i^{u}|Z_i = 0, \boldsymbol{\eta})p(Z_i = 0|\pi)\Big\}\right]^{\lambda} \\
&\propto \underbrace{\Big\{(1-\pi)^{\alpha_0 - 1}\prod_{v=1}^{V}\eta_{v0}^{\beta_{0v}-1}\Big\} \times \Big\{\pi^{\alpha_1 - 1}\prod_{v=1}^{V}\eta_{v1}^{\beta_{1v}-1}\Big\}}_{\text{prior}} \times \underbrace{\prod_{i=1}^{N^{lp}}\Big\{\prod_{v=1}^{V}\eta_{v1}^{D_{iv}} \times \pi\Big\}}_{\text{positive labeled doc. likelihood}} \\
&\quad \times \underbrace{\prod_{i=1}^{N^{ln}}\Big\{\prod_{v=1}^{V}\eta_{v0}^{D_{iv}} \times (1-\pi)\Big\}}_{\text{negative labeled doc. likelihood}} \times \underbrace{\left[\prod_{i=1}^{N^{u}}\Big\{\prod_{v=1}^{V}\eta_{v0}^{D_{iv}} \times (1-\pi)\Big\} + \Big\{\prod_{v=1}^{V}\eta_{v1}^{D_{iv}} \times \pi\Big\}\right]^{\lambda}}_{\text{unlabeled doc. likelihood}}
\end{aligned} \tag{1}$$

We weight the part of the observed likelihood that refers to the unlabeled document with $\lambda \in (0, 1)$. This is done because we typically have much more unlabeled document than labeled documents. By downweighting the information from the unlabeled document (i.e., setting λ to be small) , we can use more reliable information from labeled documents than from unlabeled documents.

Algorithm 1: EM algorithm to classify text

```
Result: Maximize $p(\pi^{(t)}, \boldsymbol{\eta}^{(t)} \mid \mathbf{D}^l, \mathbf{Z}^l, \mathbf{D}^u, \boldsymbol{\alpha}, \boldsymbol{\beta})$
if In the first iteration of Active learning then
    Initialize $\pi$ and $\boldsymbol{\eta}$ by Naive Bayes;
        $\pi^{(0)} \leftarrow \mathrm{NB}(\mathbf{D}^l,\ Z^l, \boldsymbol{\alpha})$;
        $\boldsymbol{\eta}^{(0)} \leftarrow \mathrm{NB}(\mathbf{D}^l,\ \mathbf{Z}^l, \boldsymbol{\beta})$;
else
    Inherit $\pi^{(0)}$ and $\boldsymbol{\eta}^{(0)}$ from the previous iteration of Active learning;
end
while $p(\pi^{(t)}, \boldsymbol{\eta}^{(t)} \mid \mathbf{D}^l, \mathbf{Z}^l, \mathbf{D}^u, \boldsymbol{\alpha}, \boldsymbol{\beta})$ does not converge do
    (1) E step: obtain the probability of the class for unlabeled documents;
        $p(\mathbf{Z}^u \mid \pi^{(t)}, \boldsymbol{\eta}^{(t)}\mathbf{D}^l, \mathbf{Z}^l, \mathbf{D}^u) \leftarrow$ E step($\mathbf{D}^u$, $\pi^{(t)}$, $\boldsymbol{\eta}^{(t)}$);
    (2) Combine the estimated classes for the unlabeled docs and the known classes for
     the labeled docs;
        $p(\mathbf{Z} \mid \pi^{(t)}, \boldsymbol{\eta}^{(t)}, \mathbf{D}^l, \mathbf{Z}^l, \mathbf{D}^u) \leftarrow$ combine($\mathbf{D}^l$, $\mathbf{D}^u$, $Z^l$, $p(Z^u \mid \pi^{(t)}, \boldsymbol{\eta}^{(t)}, \mathbf{D}^l, \mathbf{Z}^l, \mathbf{D}^u)$);
    (3) M step: Maximize $Q \equiv \mathbb{E}[p(\pi, \boldsymbol{\eta}, \mathbf{Z}^u \mid \mathbf{D}^l, \mathbf{Z}^l, \mathbf{D}^u, \boldsymbol{\alpha}, \boldsymbol{\beta})]$ w.r.t $\pi$ and $\boldsymbol{\eta}$;
        $\pi^{(t+1)} \leftarrow$ argmax $Q$;
        $\boldsymbol{\eta}^{(t+1)} \leftarrow$ argmax $Q$;
    (4) Check convergence: Obtain the value of $p(\pi^{(t+1)}, \boldsymbol{\eta}^{(t+1)} \mid \mathbf{D}^l, \mathbf{Z}^l, \mathbf{D}^u, \boldsymbol{\alpha}, \boldsymbol{\beta})$;
end
```

We estimate the parameters π and η using EM algorithm Dempster et al. (1977) and our implementation is presented as pseudocode in Algorithm 1. Note that by taking the expectation of the log complete likelihood function (Q function),

$$
\begin{aligned}
Q &\equiv \mathbb{E}_{\mathbf{Z}|\pi^{(t)},\boldsymbol{\eta}^{(t)},D,C}[p(\pi, \boldsymbol{\eta}, \mathbf{Z}|\mathbf{D}, \mathbf{C})] \\
&= (\alpha_0 - 1)\log(1-\pi^{(t)}) + (\alpha_1 - 1)\log\pi^{(t)} + \sum_{v=1}^{V}\Big\{(\beta_{0v}-1)\log\eta_{v0}^{(t)} + (\beta_{1v}-1)\log\eta_{v1}^{(t)}\Big\} \\
&\quad + \sum_{i=1}^{N^{lp}}\Big\{\sum_{v=1}^{V} D_{iv}\log\eta_{v1}^{(t)} + \log\pi^{(t)}\Big\} + \sum_{i=1}^{N^{ln}}\Big\{\sum_{v=1}^{V} D_{iv}\log\eta_{v0}^{(t)} + \log(1-\pi^{(t)})\Big\} \\
&\quad + \lambda\left[\sum_{i=1}^{N^u} p_{i0}\Big\{\sum_{v=1}^{V} D_{iv}\log\eta_{v0}^{(t)} + \log(1-\pi^{(t)})\Big\} + p_{i1}\Big\{\sum_{v=1}^{V} D_{iv}\log\eta_{v1}^{(t)} + \log\pi^{(t)}\Big\}\right]
\end{aligned}
\tag{2}
$$

where p_{ik} is the posterior probability of a document i being assigned to the k th cluster, $k = \{0, 1\}$, given data and the parameters at t th iteration. If a document has a positive label, $p_{i0} = 0$ and $p_{i1} = 1$.

If a document has no label,

$$
\begin{aligned}
p_{i0} &= 1 - p_{i1} \\
p_{i1} &= \frac{\prod_{v=1}^{V}\eta_{v1}^{D_{iv}} \times \pi}{\prod_{v=1}^{V}\left\{\eta_{v0}^{D_{iv}} \times (1-\pi)\right\} + \prod_{v=1}^{V}\left\{\eta_{v1}^{D_{iv}} \times \pi\right\}}
\end{aligned}
\tag{3}
$$

Equation 3 also works as the prediction equation. The predicted class of a document i is k that maximizes this posterior probability.

In the M-step, we maximize the Q function, and obtain the updating equations for π and η. The updating equation for π is the following.

$$\pi^{(t+1)} = \frac{\alpha_1 - 1 + N^{lp} + \lambda \sum_{i=1}^{N^u} p_{i1}}{\left(\alpha_1 - 1 + N^{lp} + \lambda \sum_{i=1}^{N^u} p_{i1}\right) + \left(\alpha_0 - 1 + N^{ln} + \lambda \sum_{i=1}^{N^u} p_{i0}\right)} \tag{4}$$

The updating equation for η is the following.

$$\begin{aligned} \hat{\eta}_{v0}^{(t+1)} &\propto (\beta_{v0} - 1) + \sum_{i=1}^{N^{ln}} D_{iv} + \lambda \sum_{i=1}^{N^u} p_{i0} D_{iv}, \quad v = 1, \ldots, V \\ \hat{\eta}_{v1}^{(t+1)} &\propto (\beta_{v1} - 1) + \sum_{i=1}^{N^{lp}} D_{iv} + \lambda \sum_{i=1}^{N^u} p_{i1} D_{iv}, \quad v = 1, \ldots, V \end{aligned} \tag{5}$$

D EM algorithm for a model with multiple clusters

D.1 Summary

The model outlined above assumes that there are two latent clusters, each linked to the positive and the negative class. However, this assumption can be relaxed to link multiple clusters to the negative class easily. In the world of mixture models, the simplest setup is to let $K = 2$ since the classification goal is binary, and we can link each latent cluster to the final classification categories. A more general setup is to use $K > 2$ even when a goal is a binary classification. If $K > 2$, but our focus is to uncover identify one cluster, we can choose one of the latent clusters to be linked to the "positive" class and let the all other latent clusters linked to the "negative" class (see e.g., Larsen and Rubin 2001 for a similar idea in the realm of record linkage). In other words, we collapse the $K - 1$ latent clusters into one class for the classification purpose. Using $K > 2$ makes sense if the "negative" class consists of multiple sub-categories. For instance, suppose researchers are interested in classifying news articles into political news or not. Then, it is reasonable to assume that the non-political news category consists of multiple sub-categories, such as technology, entertainment, and sports news. Using the number of clusters $K > 2$ may help improve the classification performance in the situations like this. For instance, BBC corpus consists of 5 categories, politics, business, sports, technology, and entertainment, and the classification goal here is to identify documents with the politics category.

D.2 Model

This section presents a model and inference algorithm when we use more than 2 latent clusters. The model presented in the main paper is a special case of the following model where the number of latent clusters is 2, i.e. $K = 2$.

$$\begin{aligned}
\pi &\sim Dirichlet(\boldsymbol{\alpha}) \\
Z_i &\overset{i.i.d}{\sim} Bernoulli(\boldsymbol{\pi}) \\
\eta_{\cdot k} &\overset{i.i.d}{\sim} Dirichlet(\boldsymbol{\beta}_k), \quad k = \{1, \dots, K\} \\
\mathbf{D}_{i\cdot} | Z_i = k &\overset{i.i.d}{\sim} Multinomial(n_i, \boldsymbol{\eta}_{\cdot k})
\end{aligned} \tag{6}$$

Note that $\boldsymbol{\pi}$ is now a probability vector of length K, and it is drawn from a Dirichlet distribution.

Let k^* be the index of the cluster linked to the positive class. The observed likelihood is the following.

$$
\begin{aligned}
& p(\boldsymbol{\pi}, \boldsymbol{\eta} | \mathbf{D}, \mathbf{C}^{lp}, \mathbf{C}^{ln}) \\
& \propto p(\boldsymbol{\pi}) p(\boldsymbol{\eta}) p(\mathbf{D}^{lp}, \mathbf{C}^{lp} | \boldsymbol{\pi}, \boldsymbol{\eta}) p(\mathbf{D}^{ln}, \mathbf{C}^{ln} | \boldsymbol{\pi}, \boldsymbol{\eta}) \Big[p(\mathbf{D}^{u} | \boldsymbol{\pi}, \boldsymbol{\eta}) \Big]^{\lambda} \\
& = p(\boldsymbol{\pi}) p(\boldsymbol{\eta}) \times \prod_{i=1}^{N^{lp}} p(\mathbf{D}_i^{lp} | Z_i = k^*, \eta) p(Z_i = k^* | \boldsymbol{\pi}) \\
& \quad \times \prod_{i=1}^{N^{ln}} \sum_{k \neq k^*} \Big\{ p(\mathbf{D}_i^{ln} | Z_i = k, \eta) p(Z_i = k | \boldsymbol{\pi}) \Big\} \times \left[\prod_{i=1}^{N^u} \sum_{k=1}^{K} \Big\{ p(\mathbf{D}_i^{u} | Z_i = k, \boldsymbol{\eta}) p(Z_i = k | \boldsymbol{\pi}) \Big\} \right]^{\lambda} \\
& \propto \underbrace{\prod_{k=1}^{K} \left\{ \pi_k^{\alpha_k - 1} \prod_{v=1}^{V} \eta_{vk}^{\beta_{kv} - 1} \right\}}_{\text{prior}} \times \underbrace{\prod_{i=1}^{N^{lp}} \Big\{ \prod_{v=1}^{V} \eta_{vk^*}^{D_{iv}} \times \pi_k \Big\}}_{\text{positive labeled doc. likelihood}} \\
& \quad \times \underbrace{\prod_{i=1}^{N^{ln}} \sum_{k \neq k^*} \Big\{ \prod_{v=1}^{V} \eta_{vk}^{D_{iv}} \times \pi_k \Big\}}_{\text{negative labeled doc. likelihood}} \times \underbrace{\left[\prod_{i=1}^{N^u} \sum_{k=1}^{K} \Big\{ \prod_{v=1}^{V} \eta_{vk}^{D_{iv}} \times \pi_k \Big\} \right]^{\lambda}}_{\text{unlabeled doc. likelihood}}
\end{aligned}
\tag{7}
$$

The Q function (the expectation of the complete log likelihood) is

$$
\begin{aligned}
Q & \equiv \mathbb{E}_{\mathbf{Z} | \boldsymbol{\pi}^{(t)}, \boldsymbol{\eta}^{(t)}, D, C} [p(\boldsymbol{\pi}, \boldsymbol{\eta}, \mathbf{Z} | \mathbf{D}, \mathbf{C})] \\
& = \sum_{k=1}^{K} \left[(\alpha_k - 1) \log \pi_k^{(t)} + \sum_{v=1}^{V} \Big\{ (\beta_{kv} - 1) \log \eta_{vk}^{(t)} \Big\} \right] \\
& \quad + \sum_{i=1}^{N^{lp}} \Big\{ \sum_{v=1}^{V} D_{iv} \log \eta_{vk^*}^{(t)} + \log \pi_{k^*}^{(t)} \Big\} + \sum_{i=1}^{N^{ln}} \sum_{k \neq k^*} p_{ik} \Big\{ \sum_{v=1}^{V} D_{iv} \log \eta_{vk}^{(t)} + \log \pi_k^{(t)} \Big\} \\
& \quad + \lambda \left[\sum_{i=1}^{N^u} \sum_{k=1}^{K} p_{ik} \Big\{ \sum_{v=1}^{V} D_{iv} \log \eta_{vk}^{(t)} + \log \pi_k^{(t)} \Big\} \right]
\end{aligned}
\tag{8}
$$

The posterior probability of $Z_i = k$, p_{ik}, is

$$
p_{ik} = \frac{\prod_{v=1}^{V} \eta_{vk}^{D_{iv}} \times \pi_k}{\sum_{k=1}^{K} \Big[\prod_{v=1}^{V} \eta_{vk}^{D_{iv}} \times \pi_k \Big]}
\tag{9}
$$

M step estimators are The updating equation for π is the following.

$$
\hat{\pi_k} \propto \begin{cases} \alpha_k - 1 + \sum_{i=1}^{N^{ln}} p_{ik} + \lambda \sum_{i=1}^{N^u} p_{ik} & \text{if } k \neq k^* \\ \alpha_k - 1 + N^{lp} + \lambda \sum_{i=1}^{N^u} p_{ik^*} & \text{if } k = k^* \end{cases}
\tag{10}
$$

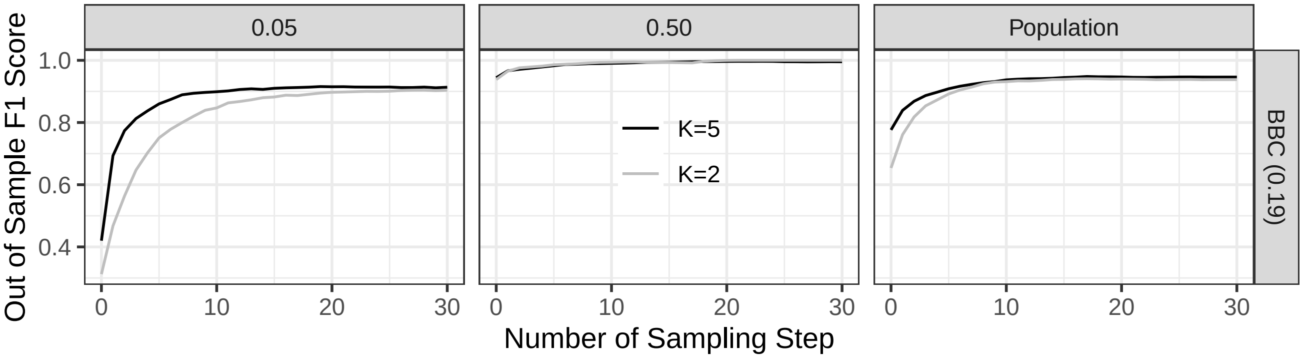

Figure 1: **Classification Results with 2 and 5 Clusters.** The darker lines show the results with 5 latent clusters and the lighter lines show 2 latent clusters. The columns correspond to various proportion of positive labels in the corpus. The y-axis indicates the out-of-sample F1 score and the x-axis show the number of sampling steps. Using multiple clusters improves the classification performance when the number of latent clusters matches the data generating process.

The updating equation for η is the following.

$$\hat{\eta}_{vk} \propto \begin{cases} (\beta_k - 1) + \sum_{i=1}^{N^{ln}} p_{ik} D_{iv} + \lambda \sum_{i=1}^{N^u} p_{ik} D_{iv} & \text{if } k \neq k^* \\ (\beta_k - 1) + \sum_{i=1}^{N^{lp}} D_{iv} + \lambda \sum_{i=1}^{N^u} p_{ik^*} D_{iv} & \text{if } k = k^* \end{cases} \tag{11}$$

Note that we downweight the information from unlabeled document by λ, to utilize more reliable information from labeled documents.

D.3 Results

Figure 1 shows the results of 2 and 5 latent clusters. The darker lines show the results with 5 latent clusters and the lighter lines show the results with 2 latent clusters. Overall, the model with 5 clusters performs better or as well as the model with 2 clusters. The gain from using 5 clusters is the highest when the proportion of positive label is small and when the size of labeled data is small.

Figure 2 shows the results when the multi-cluster and keyword upweighting approaches are combined.

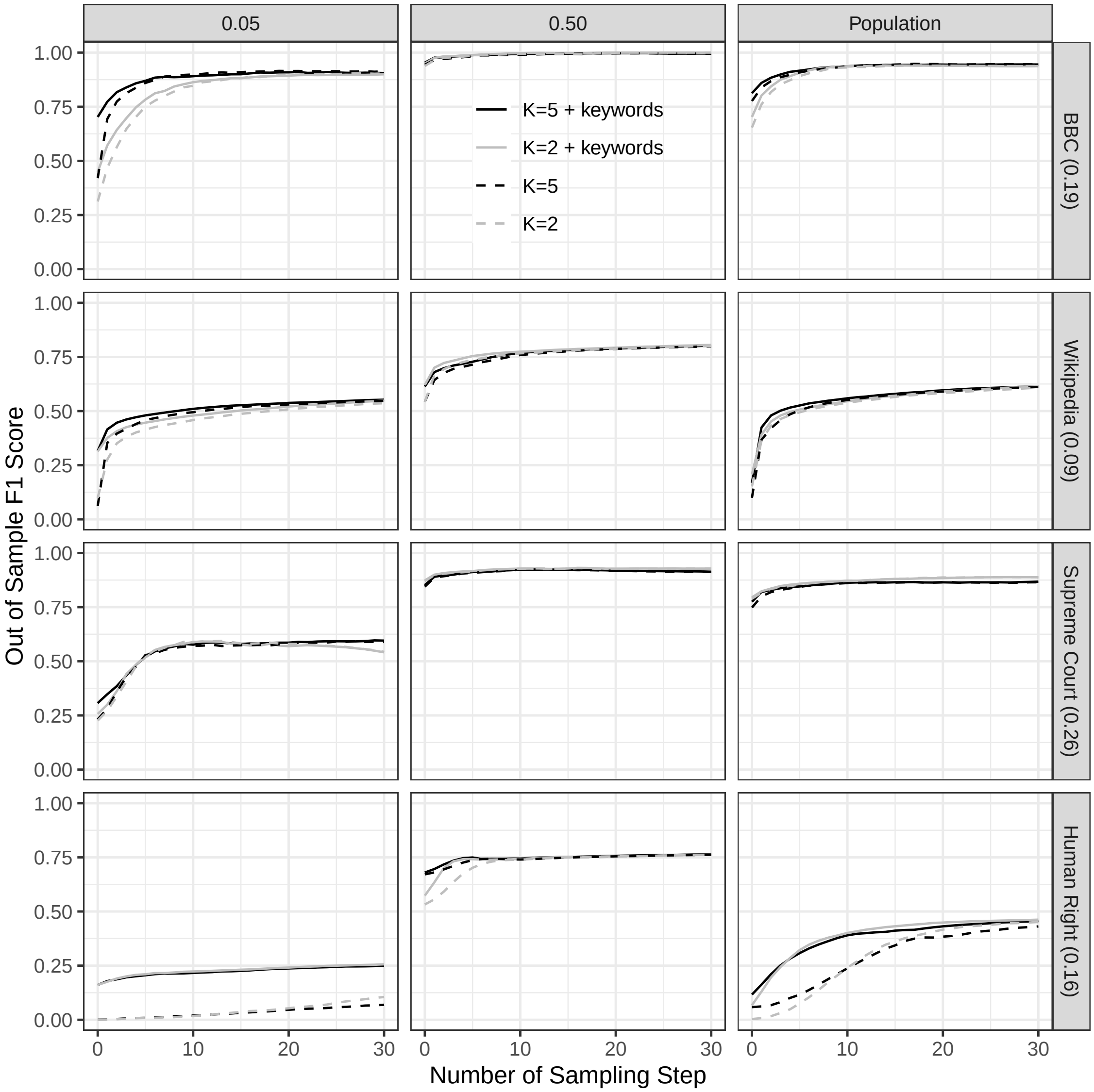

Figure 2: **Classification Results with Multiple Clusters and Keywords.** The rows correspond to different datasets and the columns correspond to various proportion of positively labeled documents in the corpus. The y-axis indicates the out-of-sample F1 score and the x-axis show the number of sampling steps. The linetype show whether keywords are supplied: the solid lines show the results with keywords and the dashed lines without keywords. The colors show the number of latent clusters in the mixture model: the darker lines show the results with 5 latent clusters and the lighter lines with 2 latent clusters. Using 5 clusters leads to as good or slightly better performance than using 2 clusters. The performance improvement is the largest with the BBC corpus, which consists of 5 news topic categories. Likewise, our mixture models with keywords leads to as good or better performance than the models without keywords. The improvement is the largest with the human rights corpus, where the number of words per document is the smallest.

E Additional Results

To complement the results presented in Figure 1 in the main text, Table E presents the results (across datasets) of fitting our model at the initial (iteration 0) and last active step (iteration 30). It is clear from the table that the improvements our approach brings in terms of F1-score are due to substantial gains in both precision and recall. Furthermore, after labeling 600 documents (20 per iteration), uncertainty sampling outperforms random sampling across evaluation metrics, which empirically validates the promise of active learning in terms of text classification.

Table 2: Classification Performance: Uncertainty vs Random Sampling with $\lambda = 0.001$

Dataset	Active Step	Uncertainty Sampling			Random Sampling		
		Precision	Recall	F1-score	Precision	Recall	F1-score
`Wikipedia`	0	0.71	0.13	0.22	0.71	0.13	0.22
	30	0.71	0.54	0.61	0.45	0.56	0.50
`BBC`	0	0.33	0.86	0.48	0.33	0.86	0.48
	30	0.92	0.96	0.94	0.92	0.94	0.93
`Supreme Court`	0	0.46	0.98	0.63	0.46	0.98	0.63
	30	0.85	0.91	0.88	0.75	0.96	0.84
`Human Rights`	0	0.61	0.01	0.02	0.61	0.01	0.02
	30	0.53	0.42	0.47	0.46	0.44	0.45

Similarly, and as noted in the main text, our results appear to be not too sensitive to the selection of the weighting parameter λ, provided that its value remains small. Figures 3 confirms this finding. After 30 active steps, the performance of our mixture model is better in terms of F1-score when $\lambda = 0.001$ if compared to $\lambda = 0.01$

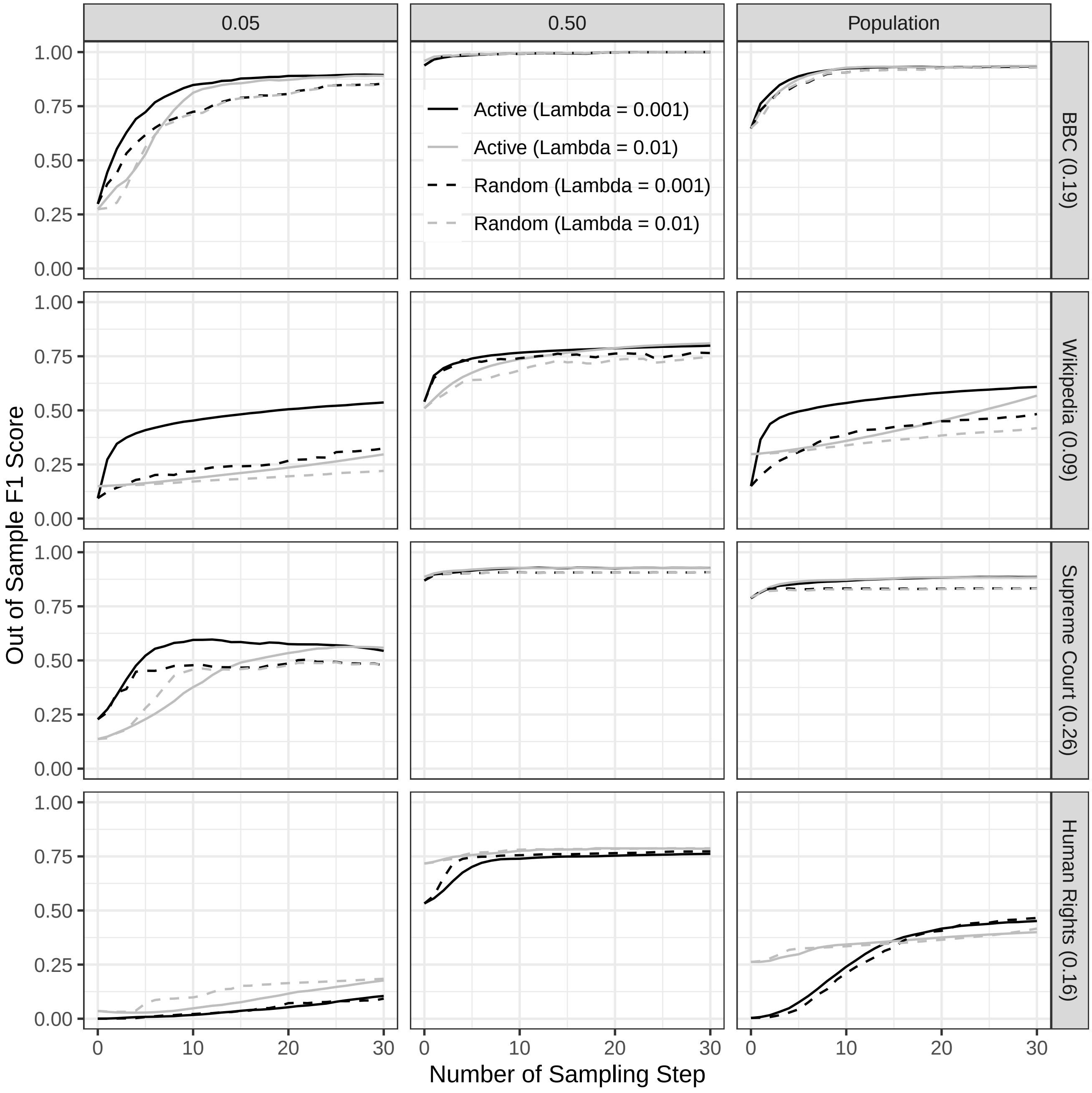

Figure 3: **Classification Results with 2 Clusters and $\lambda = 0.01$ vs $\lambda = 0.001$.** The darker lines show the results with $\lambda = 0.001$ and the lighter lines show $\lambda = 0.01$. The columns correspond to various proportion of positive labels in the corpus. The y-axis indicates the out-of-sample F1 score and the x-axis show the number of sampling steps. The smaller the value of λ the better the performance of our model.

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*, .

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977), "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1), 1–22.

Diermeier, D., Godbout, J.-F., Yu, B., and Kaufmann, S. (2012), "Language and ideology in Congress," *British Journal of Political Science*, 42(1), 31–55.

Gillion, D. Q. (2016), *Governing with words: The political dialogue on race, public policy, and inequality in America* Cambridge University Press.

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.

Hopkins, D. J., and King, G. (2010), "A method of automated nonparametric content analysis for social science," *American Journal of Political Science*, 54(1), 229–247.

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.

Larsen, M. D., and Rubin, D. B. (2001), "Iterative Automated Record Linkage Using Mixture Models.," *Journal of the American Statistical Association.*, 96(453), 32–41.

Lopez, J. C. A. D., Collignon-Delmar, S., Benoit, K., and Matsuo, A. (2017), "Predicting the Brexit vote by tracking and classifying public opinion using Twitter data," *Statistics, Politics and Policy*, 8(1), 85–104.

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

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.

Roberts, M. E., Stewart, B. M., Tingley, D., Airoldi, E. M. et al. (2013), The structural topic model and applied social science,, in *Advances in neural information processing systems workshop on topic models: computation, application, and evaluation*, Vol. 4, Harrahs and Harveys, Lake Tahoe, pp. 1–20.

Settles, B. (2011), *Synthesis Lectures on Artificial Intelligence and Machine Learning : Active Learning*, San Rafael: 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. 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.