Active Learning Approaches for Labeling Text: Review and Assessment of the Performance of Active Learning Approaches[*]

Blake Miller[†] Fridolin Linder[‡] Walter R. Mebane, Jr.[§]

February 26, 2019

Abstract

Supervised machine learning methods are increasingly employed in political science. Such models require costly manual labeling of documents. In this paper we introduce active learning, a framework in which data to be labeled by human coders are not chosen at random but rather targeted in such a way that the required amount of data to train a machine learning model can be minimized. We study the benefits of active learning using text data examples. We perform simulation studies that illustrate conditions where active learning can reduce the cost of labeling text data. We perform these simulations on three corpora that vary in size, document length and domain. We find that in cases where the document class of interest is not balanced, researchers can label a fraction of the documents one would need using random sampling (or 'passive' learning) to achieve equally performing classifiers. We further investigate how varying levels of inter-coder reliability affect the active learning procedures and find that even with low-reliability active learning performs more efficiently than does random sampling.

[*] Work supported by NSF award SES 1523355.

[†] Program in Quantitative Social Science, Dartmouth College, Silsby Hall, Hanover, NH 03755 (E-mail: blake.a.p.miller@dartmouth.edu).

[‡] Department of Political Science, Social Media and Political Participation Lab, New York University, 431 19 West 4th Street, New York, NY 10012 (E-mail: fridolin.linder@nyu.edu).

[§] Professor, Department of Political Science and Department of Statistics, University of Michigan, Haven Hall, Ann Arbor, MI 48109-1045 (E-mail: wmebane@umich.edu).

1 Introduction

Supervised machine learning methods have gained increasing attention in political science research (e.g. Beck, King, & Zeng, 2000; Hill & Jones, 2014; Muchlinski, Siroky, He, & Kocher, 2016; Cranmer & Desmarais, 2017) and specifically in connection to text analysis (e.g. Collingwood & Wilkerson, 2012; Grimmer & Stewart, 2013; Drutman & Hopkins, 2013; Ceron, Curini, Iacus, & Porro, 2014; Workman, 2015; Wilkerson, Smith, & Stramp, 2015; Wilkerson & Casas, 2017). A core step in many research designs that involve text is the classification of documents into topical categories. Unsupervised topic models (Blei, Ng, & Jordan, 2003; Grimmer, 2010; Roberts, Stewart, Tingley, Airoldi, et al., 2013) have been arguably the goto method for political scientists for that purpose. In many contexts, supervised models would be more appropriate for the task at hand. However, supervised learning is costly because labeled data is required to train a classification model.

In this paper we introduce active learning, a data labeling framework developed in computer science that can greatly increase the efficiency of data labeling (Settles, 2012). We concentrate on text data as a specific application., but active learning can be used with any kind of data in a supervised context. Much of the cost that comes from labeling texts is a result of sampling documents for an expert to label when classes are imbalanced. For example a political scientist may wish to identify texts from a newspaper corpus that reference a terrorist attack. Because terrorist attacks are rare events we can expect that a random sample of newspaper articles will contain only a small number of articles about terrorist attacks, and an expert labeler will spend much of her time labeling irrelevant documents. In contrast, in active learning approaches, a learning algorithm suggests which observations the expert should label. With active learning, this suggestion is made according to a quantitative metric of the expected performance improvement that could be realized by each of the unlabeled observations in the data. Choosing data to be labeled in such a way reduces the labeling of documents that are not informative to the classification algorithm and thereby increases the efficiency of costly human labelers. We provide an introduction to the concept and an accessible starting point for political scientists who may benefit from this approach to labeling texts. We do not provide an exhaustive overview or theoretically rigorous exposition of the large number of varieties of active learning. For those who want to dive deeper into the theory and literature, see Settles (2012).

We conduct experiments that illustrate circumstances in which active learning produces the highest efficiency gains and show how common problems in social science—such as low inter-coder reliability—may affect active learning. We use three different text corpora and several classification models to compare active learning to the default approach to data labeling (random sampling or "passive learning") and vary the amount of available training data. Our aim is not to adjudicate among various active learning algorithms but to show that various approaches all compare favorably to random sampling. We find that active learning approaches reduce the costs of supervised learning in many scenarios. Under realistic conditions, active learning can produce an 8-fold decrease in the size of training data required to achieve the same performance as with passive learning. We find that in situations where the data are balanced—that is, the topic or class of interest makes up about half of the dataset—active learning provides no additional benefits. However, such balanced problems are relatively rare in political science (Ertekin, Huang, Bottou, & Giles, 2007; Sun, Wong, & Kamel, 2009). For an additional contribution that, to our knowledge, is novel even to the computer science literature on the topic, we induce varying levels of inter-coder (un)reliability. We find that even with fairly low inter-coder agreement, active learning performs more efficiently than passive learning, even though both methods' performance is reduced by labeling error (Mikhaylov, Laver, & Benoit, 2012).

2 Approaches to Automated Text Analysis

We start by introducing the theoretical background and the variety of methods that fall under the umbrella of active learning.

2.1 Unsupervised vs. Supervised Learning

Automated text analysis methods can be categorized as supervised or unsupervised (Grimmer & Stewart, 2013). Supervised models 'learn' from a subset of data that is annotated by experts, while unsupervised models require no annotation, instead learning from correlations and co-occurrences of text features. While unsupervised models are important research tools, supervised models are usually a better choice for measuring topical categories the researcher has defined a priori. Conversely, unsupervised models are a better choice for discovering the latent topics within large corpora in the absence of a priori knowledge about the structure of these corpora.

When deciding between supervised and unsupervised models, it is important to consider the promises and pitfalls of each approach. Unsupervised models are good for summarization and exploration of large corpora. However if the structure of the problem is well defined, that is, if there are specific topic categories defined by the researcher, supervised learning can be more useful than unsupervised strategies. Consider the example where a researcher is interested to find all social media posts related to a specific protest movement. In this case the structure of the problem is well defined, that is, the researcher knows a priori what the classes of interest are (posts relevant to the movement and posts not relevant to the movement), and given a post, she would be able to determine the class membership of said post. In contrast an unsupervised method, such as a topic model, would not be guaranteed to produce a topic that is congruent with the protest movement class. Furthermore if posts about this movement rarely occur in the data it is unlikely that a single topic emerges that captures this class. The supervised method would, therefore, be the more natural choice in such a situation.

Furthermore, as previous research has shown, unsupervised methods often lack agreed-upon model selection procedures (Wallach, Murray, Salakhutdinov, & Mimno, 2009), are highly unstable with respect to text preprocessing and hyperparameter choices (Denny & Spirling, 2018; Wilkerson & Casas, 2017), and require a great deal of human interpretation or post-hoc labeling (Chang, Gerrish, Wang, Boyd-Graber, & Blei, 2009). Conversely, model selection, hyperparameter/preprocessing choices, and model evaluation are all straightforward in supervised models.

Despite the many benefits of supervised approaches, their big disadvantage is the cost of labeling data—and this is likely a reason for the bigger popularity of topic models in political science (Quinn, Monroe, Colaresi, Crespin, & Radev, 2010). Huge text corpora are very easily accessible to researchers (e.g. social media data, websites, legislative documents), but categorizing subsets of them into relevant and non-relevant categories requires costly manual labeling.

2.2 Active vs. Passive Learning

In most supervised classification procedures, an expert[1] labels the class membership of a fixed set of observations in the data. These observations are usually a random sample drawn from the full dataset. Expert-labeled data are then used to "train" a learning algorithm to predict the labels of unlabeled observations in the dataset. We refer to this mode of data labeling and model training as "passive learning." While this approach to classification works quite well in many cases, we see two situations in which the cost of labeling data "passively" can be prohibitively large:

1. If the distribution of labels or classes in the corpus is imbalanced, that is, if one label occurs much more often than the other, it can take an enormous amount of labeled data to get enough information on the minority class for the algorithm to learn to recover it reliably.

2. If there are many very similar documents, drawing data for labeling at random can be very inefficient (a lot of data is needed to learn representations of all relevant areas of the feature space).

If a concept appears in only 1 percent of documents, to get a sample of 20 relevant documents, an expert labeler would have to label, in expectation, 2000 documents. Even with this time-intensive labeling effort, a machine learning algorithm trained on only 20 observations is unlikely to perform well. Equivalently, if there are large clusters of very

[1] Sometimes referred to as an "oracle," "labeler," or "coder."

similar documents, then if documents are sampled randomly for labeling the expert might spend a lot of time assigning labels to documents that are not informative to the classification algorithm because they differ so little in their features.

The computer science literature has consistently demonstrated that in these cases, active learning can be quite useful (see e.g. Schohn & Cohn, 2000; Dasgupta, Kalai, & Monteleoni, 2005; Tong & Koller, 2001; Roy & McCallum, 2001; Ertekin et al., 2007). In an active learning approach to classification, a learning algorithm suggests which observations the expert should label. This suggestion usually is made according to a quantitative metric of the expected performance improvement that could be realized by each of the unlabeled observations in the data. Documents are labeled in an iterative manner: After each iteration, a new classification model is trained and this model is queried for documents it is most uncertain about. This way only documents that would actually help the algorithm to improve its performance are selected for labeling. In the next section, we explain each step of the active learning procedure in more detail.

3 Active Learning

3.1 General Principles of Active Learning

Given a set of documents with known labels $\mathbf{y}$ and a set of features (these can be bag-of-words features but also document metadata like time stamps or author information) $\mathbf{X}$, the goal of text classification (or of machine learning in general) is to learn the function $y = f(\mathbf{X})$ that most accurately maps features to the labels beyond the specific set (training set) $\mathbf{y}, \mathbf{X}$ (see e.g. Friedman, Hastie, & Tibshirani, 2001, for an introduction). For simplicity of exposition we concentrate on binary classification in this paper (i.e. $y_i \in \{0, 1\}, \forall i$) but all methods described here are applicable to multi-class classification ($y_i \in \{0, 1, ..., k\} \forall i$) and continuous outcomes (see below).

There are many different implementations of the active learning framework. Each differs in what measure(s) are used to query unlabeled observations for an expert to label. Before discussing some of these variations in depth in Section 3.3, we give intuition on the logic of active learning using the least-complex variant, relying on logistic regression as the classification model and uncertainty sampling as the querying strategy (Lewis & Catlett, 1994).

We denote the set of unlabeled documents as $\mathcal{U}$ with all documents represented as $\mathbf{X}_{\mathcal{U}} = \{\mathbf{x}_i | i = 1, ..., N\}$, with $\mathbf{x}_i$ as a single document represented by a set of features (for example word counts or document metadata). Each document $\mathbf{x}_i$ has a corresponding true label $y_i \in \{0, 1\}$ that can be obtained by asking an expert labeler to label it. We denote the logistic regression model with a 0.5 probability threshold as $f(\mathbf{x}_i, \theta) = \mathbb{I}(1/(1 + e^{-\mathbf{x}\theta}) > 0.5)$, were $\mathbb{I}(.)$ is the indicator function that returns 1 if the condition is met and 0 otherwise. The most basic active learning algorithm is then:

1. Start with an initial set $\mathcal{L}$ of documents $\mathbf{X}_{\mathcal{L}}$ with known labels $\mathbf{y}_{\mathcal{L}}$
2. Train the model $f(., \theta)$ using the available training data $\mathbf{y}_{\mathcal{L}}, \mathbf{X}_{\mathcal{L}}$
3. Produce a predicted probability for each unknown document in unlabeled set $\mathcal{U}$: $\hat{\mathbf{y}} = f(\mathbf{X}_{\mathcal{U}}, \theta)$
4. Use the query function to obtain a new document for labeling: $z = q(\hat{\mathbf{y}}) = \text{argmin}_i \, |\hat{y}_i - 0.5|$
5. Obtain a label y_z for $\mathbf{x}_z$ from the expert labeler
6. Add y_z and $\mathbf{x}_z$ to $\mathbf{y}_{\mathcal{L}}$ and $\mathbf{X}_{\mathcal{L}}$, remove $\mathbf{x}_z$ from $\mathbf{X}_{\mathcal{U}}$
7. Repeat Steps 2 - 6 until a stopping criterion is reached

To summarize this algorithm: Starting with a population of documents, each of which is supposed to be classified, and an initial set of labeled data (this can be chosen at random or be produced from the domain knowledge of the researcher), a model is trained to separate the relevant from the non-relevant documents. Each document of the population is then assigned a predicted probability from the model and the document with a probability closest to 0.5—that is, the document the model is least certain about—is selected for annotation. In many cases it makes sense to select not a single document for labeling but the batch of least certain documents. The procedure is then repeated by re-training the model with the additional labeled data, querying more documents, labeling, etc. Usually the procedure is stopped when either the labeling budget is exhausted or a satisfactory classifier performance is reached (the definition of satisfactory, of course, depends on the application). Figure 1 displays a visualization of the algorithm. The bold line separating the observations in two-dimensional space corresponds to the 0.5 predicted probability threshold of the regression model. The active learning algorithm selects observations closest to this decision boundary in each iteration represented by each panel in the figure.

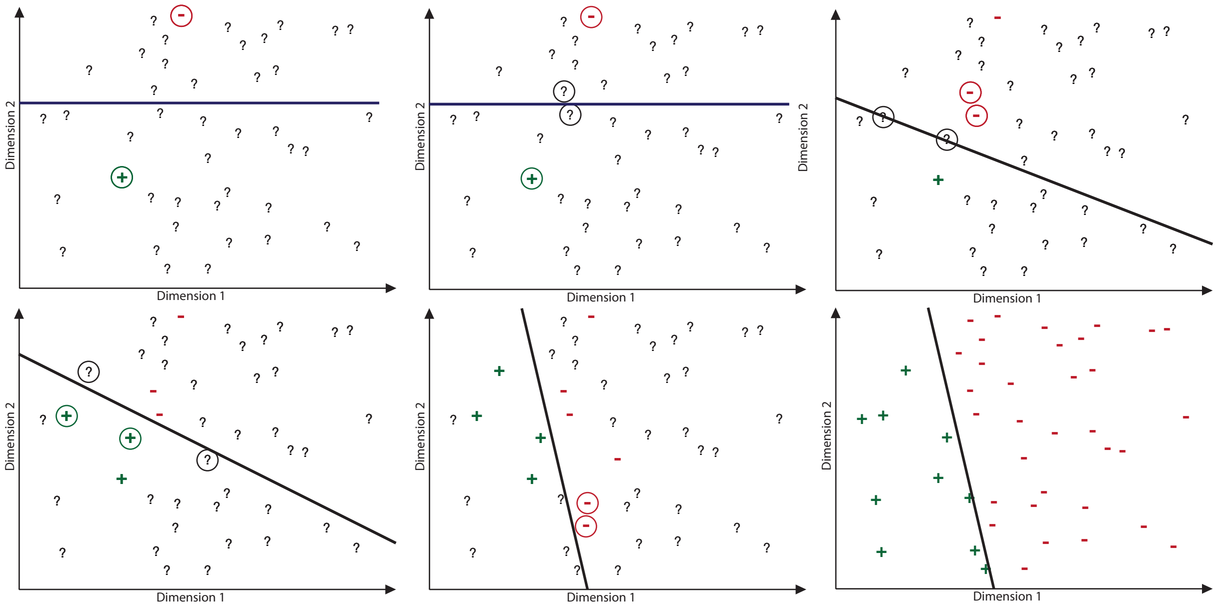

Figure 1: From left to right, top to bottom, the first 5 panels visualize the uncertainty sampling algorithm. Circles represent queries of unlabeled observations. Once an expert has labeled the data, they are represented by pluses for the positive class and minuses for the negative class. The last panel shows the ground truth, true labels of all labeled observations.

The algorithm can be easily adapted for use with nominal outcomes of more than two categories or continuous outcomes. For multi-class classification, one way of accommodating such outcomes would be to select the new document using entropy as a measure of classifier uncertainty.[2] Given a classification model that produces predicted probabilities for each class an observation could fall in, the entropy over these predicted probabilities is a measure of uncertainty of the classifier about this observation. For the case of continuous outcomes, a measure of the variance of the predicted value can be used to assess classifier uncertainty.[3] To reiterate, these options for accommodating

[2] E.g. Shannon-entropy, which is defined as $E(X) = -\sum_{i=1}^{n} p(x_i) logp(x_i)$ where X is the set of predicted probabilities over n possible classes.

[3] For example, when using a linear regression model as the classifier, the uncertainty about an observation's prediction is given by: $Var(\hat{\mathbf{y}}) = \hat{\sigma}^2 X_u (t(X)X)^{-1} X_u$ where X represents the features of the training data up to the current iteration and X_u represents the features of the so-far unlabeled data.

other outcomes are examples for the case of uncertainty sampling. For other query methods, slightly different approaches may need to be considered (see e.g. Körner & Wrobel, 2006; Settles, 2012, for more detailed guidance in such situations).

The strategy of actively selecting data to label intuitively reduces the number of labeled observations that are necessary to achieve a level of performance in the two situations described above. In the case of imbalanced classes, the model has initially little information about rare classes, given that the initial $\mathbf{y}$ most likely contains few observations belonging to the rare class. The model will be less certain about this class and therefore produce more samples for labeling from that class (this is assuming that there is some information about the class membership in the provided features $\mathbf{X}$).

In the second case, where many documents in $\mathbf{X}_{\mathcal{U}}$ are very similar, intuition on the potential benefits can be gained from step 3 in the algorithm above. Given three observations $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ with the distances the feature space: $d(\mathbf{x}_1, \mathbf{x}_2) \ll d(\mathbf{x}_1, \mathbf{x}_3) \approx d(\mathbf{x}_2, \mathbf{x}_3)$. If $\mathbf{x}_1$ is labeled, the model will produce a predicted probability closer to 0.5 for $\mathbf{x}_3$ as compared to $\mathbf{x}_2$ since it is very different from what the model has 'seen' in previous iterations. Because, $\mathbf{x}_3$ will provide more information than $\mathbf{x}_2$, the expert labeler will be presented with $\mathbf{x}_3$, thereby reducing inefficiency in the labeling.

3.2 Varieties of Active Learning

There is a large variety of ways to implement the active learning framework described in Section 3.1. Implementations can differ in several regards depending on the application: availability of unlabeled samples, the type of sample that is queried, the model and outcome, and the querying algorithm. These choices have produced a very large technical literature and make the choice for practical applications difficult. In this section, we will briefly describe the choices a researcher has, without going into technical depth (see Settles, 2012, for an excellent in-depth review of the varieties of active learning).

Availability of Samples: Depending on the data source, documents for annotation may be available in different forms. In the procedure described in Section 3.1 there exists a pool of documents from which single documents can be selected for querying and classification (*pool-based sampling*). In other applications, not all documents are available at any time but they are made available as a stream, for example in stream-based APIs such as the Twitter Streaming API. In this case, the active learning algorithm has to decide in real time which samples to select from the ones available at a certain time point (*stream-based selective sampling*) (e.g. Freund, Seung, Shamir, & Tishby, 1997). Finally, some authors have suggested producing a synthetic sample, that is constructed in such a way that the model learns best from it after it was labeled (*query synthesis*) (Angluin, 1988). This strategy can lead to multiple problems for text analysis in practice. Most importantly, a synthetically generated document will be difficult to label for an expert coder.

Model: In principle any classification or regression model can be used in an active learning approach. Some querying methods have specific requirements for the model. For instance, the uncertainty sampling strategy used in step 4 in the illustration above requires the model to produce a measure of uncertainty (for example a predicted probability), not just a discrete class prediction. There are also active learning varieties for quantitative outcomes. Uncertainty, in this case, is usually operationalized through the variance in the prediction of each unlabeled sample, so applications, in this case, are relatively straightforward.[4]

3.3 Querying Strategies

How to measure the potential information gain for the model from an unlabeled instance is probably the component of active learning that received the largest amount of attention. Besides the uncertainty sampling querying strategy described in Section 3.1, there are a variety of methods to obtain unlabeled documents from the model. They can be roughly grouped into three groups (Settles, 2012): 1) Algorithms that rely on the

[4] Though regression—statistical learning approaches with quantitative outcomes—may be of interest to researchers, we focus on qualitative outcomes in this paper, as text classification is more frequently the objective of social scientists. All the methods discussed in this paper however can be applied for regression. For specific guidance for regression problems, see Settles (2012).

uncertainty of one or multiple classifiers about unlabeled samples (uncertainty based); 2) algorithms that are based on estimating the expected model performance given a label was made available for unlabeled samples (model performance based); and 3) algorithms that exploit clustering or other structure in the data (structure based). The three most commonly used algorithms are margin sampling, query by committee and expected model change. We review these in more detail. We refer the reader to the literature for more details on clustering approaches (see e.g. Settles, 2012, for a more extensive review).

Margin sampling and uncertainty sampling are the algorithms most commonly used. We additionally discuss expected model change since it represents a different class of algorithms that are not focused on uncertain observations but rather try to optimize what the researcher might care about directly, the performance of the model. The details of how exactly we deploy these algorithms in our simulation studies is discussed in more detail in Section 4.2.

As is the case with other choices in machine learning such as the classification model or the method of tuning parameter optimization, it is difficult to make predictions about expected performance a priori, before interacting with the data.[5] It is therefore not possible to give direct guidance to researchers which of the querying algorithms is expected to perform best for the task at hand. This is the case because the function that is approximated by the classifier for text classification is highly complex, and the reasons for performance of various classifiers is not well understood. For practical applications it is therefore recommended to explore the robustness of an active learning approach by also investigating the performance of several querying strategies.

However there are other criteria besides robustness that applied researchers should consider when choosing a querying strategy. The various querying algorithms can vary drastically in their computational complexity and their ease of implementation. For instance, uncertainty sampling is a very simple approach that is easy to implement and computationally not very demanding. Predicting a label for each unlabeled observation

[5] This problem is sometime referred to as the “no free lunch theorem” and formalized in Wolpert (1996).

is computationally much less demanding than, for example, estimating a new model for every unlabeled observation, as is the case for the expected model change algorithm. This can become quite expensive as the number of observations in a dataset increase. For the "query by committee" approach, it is necessary to train a committee of several classifiers, making this approach considerably more expensive than other approaches that require only a single model.

3.3.1 Margin sampling

The algorithm used in section 3.1 to demonstrates the fundamental principle of active learning is called uncertainty sampling. That is, the algorithm queries the observations for which it is most uncertain about their label. Logistic regression, however, might not be the model of choice for text classification. Support vector machines (SVM) is a very commonly applied model for text classification. The simple margin algorithm detailed in Tong & Koller (2001) is a special case of uncertainty sampling.[6] Essentially the algorithm, similar to the logistic regression algorithm above, uses an uncertainty measure to query new observations for labeling by an expert labeler. Because the objective of the SVM model is to find a hyperplane that optimally separates classes in the feature space, observations that are closest to that hyperplane represent observations that the SVM is most uncertain of. For the simple margin classifier, at each iteration, an SVM model is trained and observations closest to the hyperplane are returned to an expert for labeling (as visualized in Figure 1 for the two-dimensional case).

3.3.2 Query by Committee

The query by committee approach follows principles similar to the uncertainty sampling strategy. However instead of relying on the uncertainty of a single model, uncertainty is measured using disagreements in classification decisions among a committee of models (Seung, Opper, & Sompolinsky, 1992).

[6] Though Tong & Koller (2001) find that the simple margin algorithm is not optimal in maximally reducing the "version space," and offer several superior alternatives, it still works quite well and is useful as an example due to its simplicity.

For the query by committee approach to active learning, a committee of models, $\mathcal{C} = \{\theta^{(1)}, ..., \theta^{(C)}\}$, is trained on the current labeled set $\mathcal{L}$. Each classifier in this committee represents a different hypothesis for how the version space[7] should be partitioned. Classifiers in the committee make predictions for each of the documents in the unlabeled set $\mathcal{U}$. The m documents where the committee disagrees the most are returned to the expert to label. This disagreement is usually computed using vote entropy[8] or Kullback-Leibler (KL) divergence (Kullback & Leibler, 1951).

In the original formulation Seung et al. (1992) uses perceptron learners, but there is no clear agreement on how to best choose a set of models for the committee. Some formulations of query by committee make use of ensemble methods such as bagging (Freund & Schapire, 1997) and boosting (Abe & Mamitsuka, 1998) to form committees. Others choose a diverse set of ensemble classifiers (Melville & Mooney, 2004). In a review of the literature Settles (2012) suggests that there is no clear guideline for size and model choice. The committee only needs to satisfy two conditions: 1) models in the committee must represent different regions of the version space; and 2) there must be some disagreement among committee members.

3.3.3 Expected Model Change

While margin sampling and query by committee select unlabeled observations based on an uncertainty measure, another class of active learning approaches seek to quantify the potential informational gain of each candidate unlabeled observation. These methods query the unlabeled observations that would result in the greatest change to the current model (expected model change), the greatest reduction of generalization error (expected error reduction) or the greatest reduction in output variance (variance reduction), if we knew its label (Settles, 2012; Roy & McCallum, 2001). These approaches are computationally expensive as they require that a model be fit for each candidate

[7] Version space refers to the area defined by all potential class partitions of data that are consistent with the current labeled training data (Mitchell, 1982).

[8] Vote entropy is defined as $VE = -\sum_{i}^{C} \frac{V(y_i)}{C} log \frac{V(y_i)}{C}$ where $y_i \in \{0, 1\}$, $V(y_i)$ is the number of votes that label receives, and C is the committee size.

observation in the unlabeled set. For text classification problems, it is common that the set of unlabeled observations is quite large, a potential drawback of this approach (Settles & Craven, 2008).

The expected model change approach as originally formulated by Settles, Craven, & Ray (2008) selects unlabeled observations with the largest expected gradient length, the average length of the gradient of the loss function ℓ_θ obtained by adding the training tuple $\langle x, \hat{y} \rangle$, predictors x from the unlabeled set $\mathcal{U}$ and $\hat{y}$, the expected label given the current model to labeled set $\mathcal{L}$. Unlabeled observations are queried according to $\underset{x}{argmax}\, P_\theta(y_i|x) \|\nabla \ell_\theta(\mathcal{L} \cup \langle x, \hat{y} \rangle)\|$. This essentially chooses the unlabeled observations that, if labeled, would maximally change the output of the model. Other slight variants of this approach have proven effective in many domains (Freytag, Rodner, & Denzler, 2014).

3.4 Biased Training Data, Generalization Error and Inter-Coder Reliability

The fundamental principle of active learning is to produce training datasets that are biased to include documents that the classification model is uncertain about. That is, the dataset used to train the classification model is *not* representative of the total corpus. This can have consequences for several aspects of the coding process.

Estimates of generalization error (that is, how well the classification model will perform on the general corpus) obtained from the training data can be biased or inconsistent estimates of the true generalization error. While we cannot claim to know the direction of bias in generalization error estimates in all circumstances, it has been established that it is a biased estimate in an active learning setting (Baram, Yaniv, & Luz, 2004). If precise estimates of generalization error are required or a choice between multiple active learning methods based on unbiased estimates of generalization error is desired, part of the labeling budget has to be spent on labeling data that has been selected randomly (Ali, Caruana, & Kapoor, 2014). In our simulation study we explore the gravity of this problem by comparing estimates of generalization error obtained from labeled data that was actively sampled with unbiased estimates.

Expert labelers are most likely presented with cases that are more difficult to label. This is a problem that has been, to our knowledge, omitted from the computer science literature.[9] This could, in turn, lead to the complication that the error rate of the expert labelers is likely to be higher as compared to randomly selected labeled data. An additional point to consider is that estimates of inter-coder reliability, analogous to the generalization error, cannot be extrapolated to the general corpus. In simulations described in detail below, we simulate varying levels of inter-coder reliability to investigate to what extent labeling error influences the performance of active learning and random sampling approaches.

4 Simulation Study

To investigate the performance, efficiency and practical applicability of active learning we conduct a simulation study on three selected text datasets, varying query strategy, class balance and inducing error in the labeling process (inter-coder reliability). In this section we detail our data, simulation design, and implementation.

4.1 Data

We use three corpora that vary by text length and text style. By varying the text domain, we hope to demonstrate that active learning works across many of the domains relevant to political scientists. We use a corpus of tweets, a corpus of Wikipedia talk page entries and a corpus of news articles from news website Breitbart. These corpora vary in document size from small, to medium, to large, respectively. The style of writing in each also varies, with social media, specialized/academic writing, and news articles respectively.

1. Twitter: This corpus is comprised of 24,420 tweets collected from a random sample of German Twitter users. The sample of tweets is a subset of a larger

[9] Zhao, Sukthankar, & Sukthankar (2011) studies the implications of labeling errors on the performance of active learning algorithms but not the other way around

random sample of all Tweets authored by about 80,000 users. Each tweet was labeled as being about the refugee topic or not by German-speaking CrowdFlower workers. Of the twenty-four thousand tweets about 700 are labeled as being about the topic. The dataset is used by Linder (2017) to study public reactions to refugee allocation in the German refugee crisis.

2. Wikipedia: This corpus of 159,571 Wikipedia talk page comments includes annotations of different kinds of toxic comments. The database was released as part of a machine learning competition, "Toxic Comment Classification Challenge" on the website Kaggle that is sponsored by ConversationAI, a team organized by Jigsaw and Google to build "tools to help improve online conversation." The goal of this competition is to classify the types of toxic comments using the provided expert annotations. For our simulations, we chose the label "toxic" because it has the most support of all represented classes. "Toxic" comments are aggressive comments, violent comments, personal attacks, etc. that do not contribute to a healthy and productive discussion on talk pages.

3. Breitbart: This corpus of 174,847 news articles represents the population of articles on the Breitbart news website. These articles each come with meta tags that are chosen by Breitbart authors and editors. For this dataset, we use the label "Muslim identity," which indicates whether a specific reference to Muslim identity was made in the article tags. This corpus is used to measure how moral and emotional frames in news media can increase support for violence against out-groups in Javed & Miller (2018). To make the simulations computationally more feasible we randomly sampled a subset of 25,000 articles for this study.

Even though we use datasets that realistically reflect data that are used in political science research, we must point out that data in a simulation study can never be a representative basis on which to make with certainty general statements about the performance of different machine learning algorithms on other tasks or in other data. The combination of possible data features or structure, of tasks' possibly diverse objectives and of algorithms' flexibility makes it impossible to generalize reliably.

4.2 Design and Implementation

We conduct two simulation studies. The first study compares the performance of active learning with three different querying strategies to passive learning on the datasets described above. In the second simulation study, we investigate whether and how error in the labeling process affects active and passive learning differentially.

Each study simulates the data labeling process in an iterative manner. At each iteration a batch of documents is selected—using the querying algorithm for active learning or random sampling for passive learning—and added to the training data available to the classifier at that iteration. We draw 20 documents at each iteration and stop once we have processed 25% of documents in the corpus or all positive samples are labeled. We refer to one labeling process completed in this manner as a run. In most cases we repeat each run 50 times in order to obtain Monte Carlo estimates of the performance statistics of the various algorithms.[10]

Figure 2 illustrates the design of a single iteration of the simulation. Before each run, the data are randomly split up into a dataset that is available for labeling and training and a held-out test set. The data that are available for training at each iteration are used to train a SVM classification model which is evaluated at each iteration on the held-out test set.

In addition to training the classifier at each iteration, a model and feature selection step is conducted. Using 5-fold cross-validation and randomized parameter search (Bergstra & Bengio, 2012), the tuning parameters of the SVM as well as the best feature

[10] Because of prohibitive computational cost, for the EMC algorithm, we only repeat each run 10 times. For the main experiment for Wikipedia and Breitbart, we repeat the experiment 15 times. For the inter-coder reliability experiment for query by committee, we repeat the experiment 30 times.

set are selected at each step.[11]

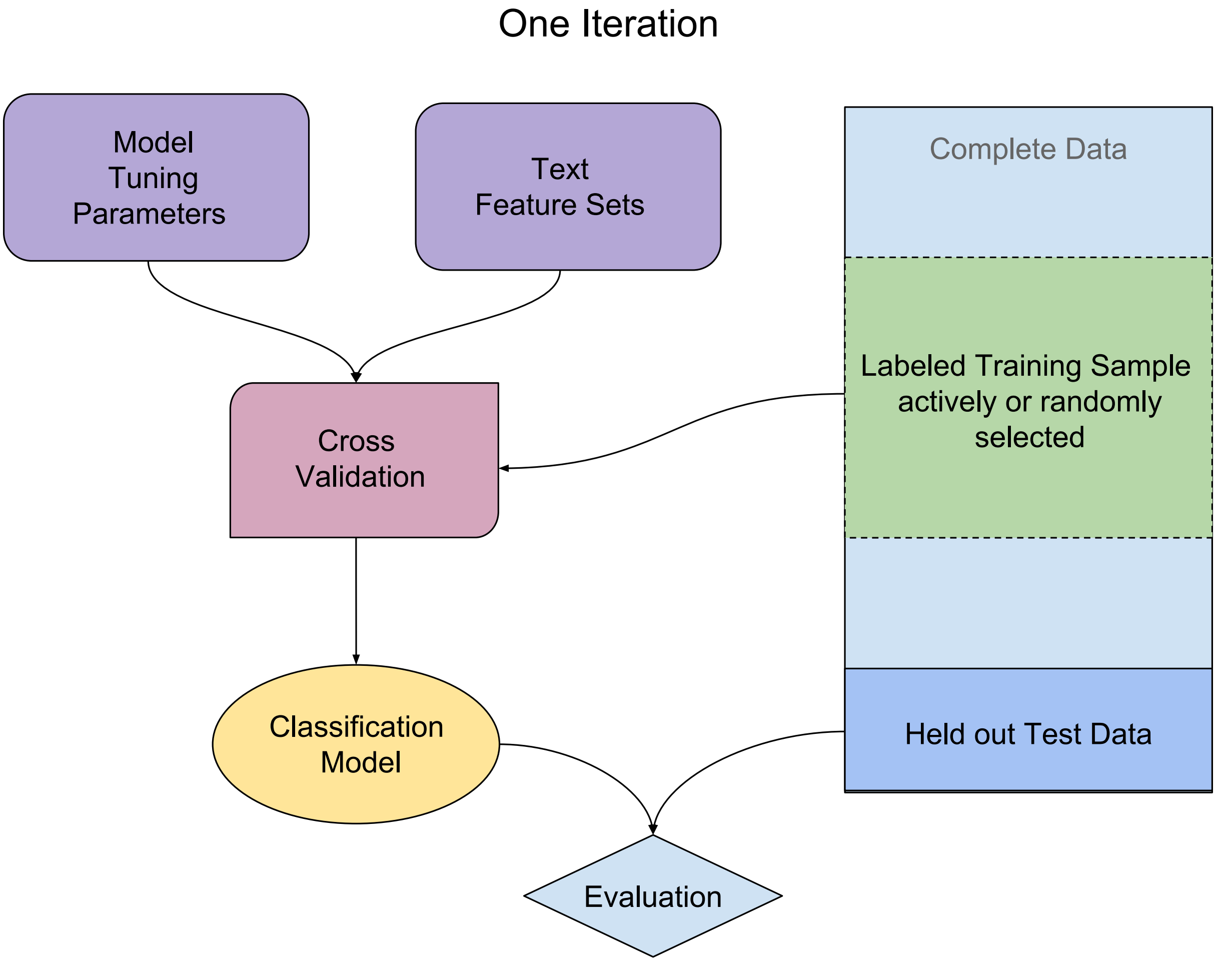

Figure 2: Design of the simulation study. In each iteration of the algorithm model tuning parameters and text feature sets are selected via cross validation on the training data available from previous iterations. The best performing classification model is then evaluated on a held-out test dataset.

In order to demonstrate the performance of the various algorithms under realistic conditions, we artificially induce various degrees of class imbalance into the datasets. We create new datasets with different levels of class imbalance by over- and undersampling the positive class from the original corpora. We use the following

[11] The main tuning parameter of the SVM is the C parameter. It is draw from a exponential distribution with $\lambda = 50$. For the feature set, we cross-validate over word n-grams of size 1 to 3 and character n-grams of size 3 to 5. Appendix Figures 12 to 15 display the choices resulting from this procedure in our experiment. As discussed in more detail above, model selection (algorithm and tuning parameters) with active learning is non-trivial. Ali et al. (2014) discuss the problem, that the observations selected with active learning are highly biased, therefore making generalization error estimates obtained from this training data (e.g. via cross-validation) invalid. Labeled observations are chosen to maximize the uncertainty of the classifier which leads to the performance of the classifier on this sample being much lower than on a true random sample from the population. In our experiments, we choose to still apply a standard model selection step because we did not intend to make generalization error estimates at each step. We also believed that this approach kept as many factors constant as possible between conditions.

positive class proportions: 0.01, 0.05, 0.10, and 0.50.

We compare four algorithms for selecting documents to label: active learning with margin sampling, active learning with query by committee, active learning with expected model change and randomly sampled labeled data. See Section 6.6 in the Appendix for the details of how we implemented various querying strategies. At each iteration after having selected the next 20 training instances we train the same SVM model for the final evaluation. We chose this procedure to compare the quality of the selection of training data while keeping the classification algorithm constant rather than confounding it with the performance of the classification model. For the query by committee algorithm, however, this means that a different model is used in the final classification step than in the querying step (margin sampling and expected model change are both based on SVMs in our implementation). This means that the query by committee algorithm might potentially perform better if the query algorithm and classifier for evaluation would match. This design does allow us to investigate whether the active learning algorithm is still providing benefits to the researchers, even when a different final classification model is used with the training data obtained. Having to "lock in" one specific classifier for a project might be a significant disincentive to using active learning, therefore assessing whether an actively sampled set of training data is superior to a randomly sampled one *independently* of the classification model is an important question for the practical applicability of the method.[12]

In addition to the main simulation study, we conduct an additional study, exclusively on the Twitter data, to test how varying levels of inter-coder reliability can affect the performance of active learning approaches. We induce random error in labeling assignment in the training data while leaving the held-out data used to evaluate performance unchanged, simulating coder error. We simulate varying levels of inter-coder reliability using the following algorithm:

For each level of inter-coder reliability $a_i \in \{0.7, 0.8, 0.9, 0.95\}$:

[12] Lewis & Catlett (1994) and Lu & Bongard (2009) find that active learning produces more informative samples than passive learning, even if the final learning algorithm is not known in advance.

1. For each label y_j in the training set $\mathbf{y}$, generate a uniform random number $u_j \in [0, 1]$;
2. If $u_j \geq a_i$, flip the label y_j from 1 to 0 or vice versa ($y'_j = |y_j - 1|$) resulting in transformed labels $\mathbf{y}'$;
3. Sample observations to 'label' using the maximal margin sampling algorithm with input $\mathbf{y}'$ and corresponding feature set $\mathbf{X}$

For the inter-coder reliability simulations we apply the margin sampling algorithm as well as the query by committee algorithm in order to evaluate if they could be affected differentially by the introduced noise.

4.3 Results

Figure 3 displays the classification performance on the held out test set for all three datasets for the first simulation study (no error in the labeling process). Each row of Figure 3 represents a level of class imbalance. "Balance: 0.01" means that there are 1% relevant labels and 99% non-relevant labels. In the columns are the three different datasets. The lines in the plots are generalized additive model fits across replications of the same conditions. The different lines show the classifier performance for the several labeling algorithms, that is, active learning with the margin, query by committee and expected model change algorithms and passive learning with randomly annotated data. All Figures display the performance of the model (defined as the F1-score[13]) for different amounts of available training data (ranging from 20 documents to 5000 documents). Note that there are different amounts of training data in the different balance levels. This is due to the fact that, for example, for a balance of 0.5, the total data set size is constrained to two times the number of relevant documents in the corpus. The original Twitter data, for example, has about 3% relevant tweets, which means that creating a balanced dataset reduces the total size of the data considerably.

The gains in performance and reduction in cost resulting from the active learning approach are clearly visible in the imbalanced conditions. The model trained on the randomly labeled data often does not achieve the same performance as the active

[13] The F1-score is a very common performance measure in machine learning. It is defined as the harmonic mean of precision and recall. Precision and recall results are presented separately in Figures 7 and 8 in the Appendix.

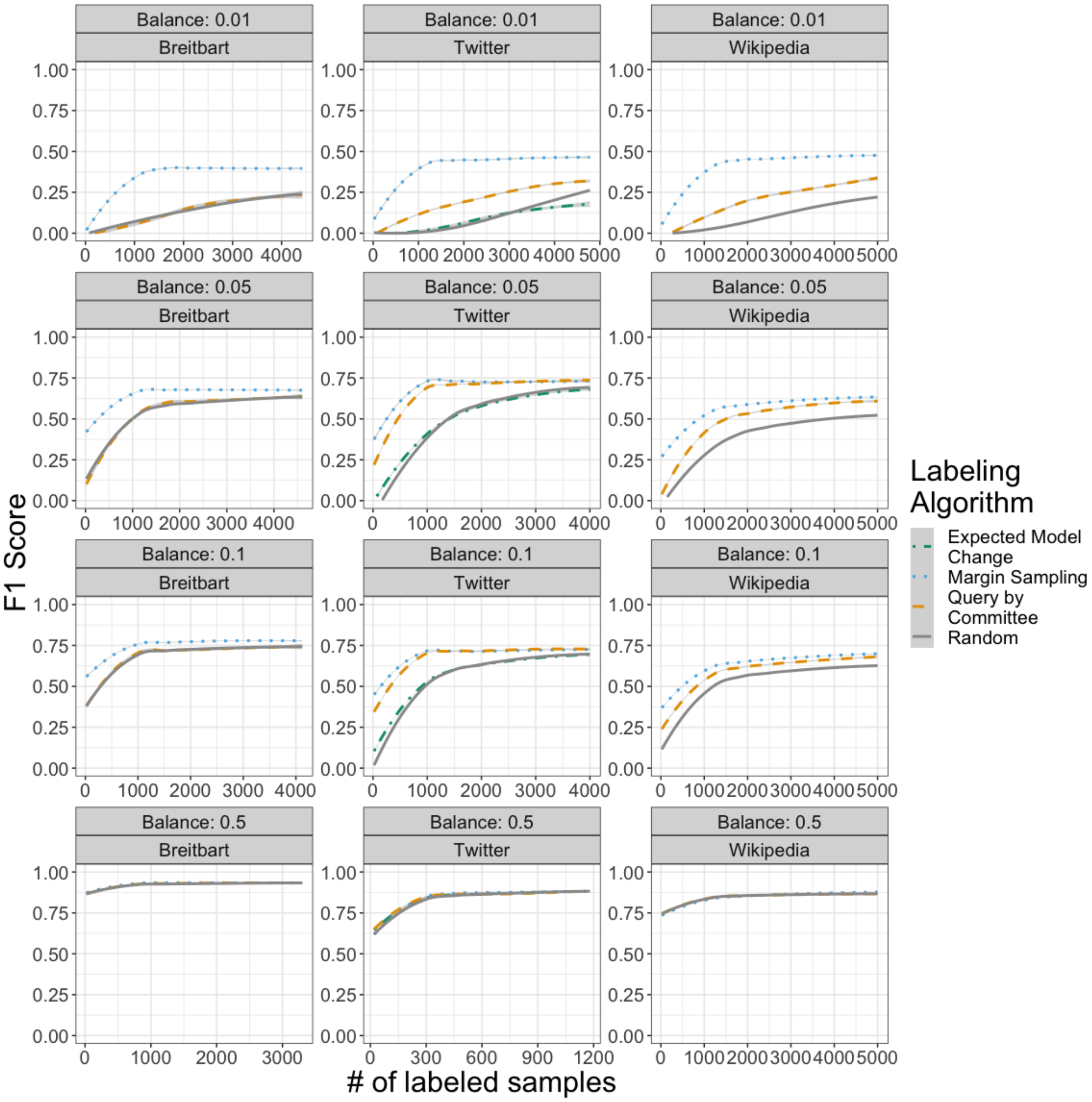

Figure 3: F1 score for experiments. The panel columns correspond to the datasets, the rows to the different levels of class imbalance. Dots represent single replications of the experiment, smoothed lines are fits (and standard errors) of a generalized additive model.

learning models, even with a quarter of the total corpus being labeled.[14] The best active learning algorithms reach their best performance very quickly with little increase in performance when adding additional training data. To quantify the gains achieved through active learning, Table 1 displays the average number of training samples that are required to achieve an F1-score of 0.1 for active learning with the margin algorithm and passive learning, as well as the ratio of the two. Depending on the balance of the

[14] Expected model change performs worse than random sampling does in the Twitter data with balance 0.01 once there are more than about 3500 labeled training observations.

data and the data set, passive learning requires the same amount of data (balanced data Wikipedia) or up to almost eight times the amount of training data (Twitter data with 1% relevant documents). The relationship of the performance differential with the balance of the data is striking. With 1% of the data being relevant the difference is dramatic, while for the balanced data, there is virtually no difference in performance.

Table 1: Number of training samples required to reach an F1-Score of 0.1 for active and passive learning.

Data Set	Balance	Samples Active	Samples Passive	Ratio
Breitbart	0.01	316.09	805.60	2.55
	0.05	77.50	142.04	1.83
	0.10	54.17	56.00	1.03
	0.50	20.00	20.80	1.04
Twitter	0.01	295.60	2305.92	7.80
	0.05	95.92	407.40	4.25
	0.10	73.33	212.80	2.90
	0.50	24.40	23.63	0.97
Wikipedia	0.01	314.44	1922.22	6.11
	0.05	103.46	379.69	3.67
	0.10	69.60	162.42	2.33
	0.50	23.20	21.80	0.94

We also observe significant differences between the various querying algorithms. Across all datasets and balance conditions, the margin algorithm performs best followed by query by committee and expected model change. Expected model change seems to produce no added performance compared to randomly annotated data. These results demonstrate that the choice of query algorithm can significantly impact the performance a researcher can expect from their classifier. As discussed in the theoretical sections, as is the case in many machine learning applications, it is difficult to choose a priori which algorithm will perform best for a given problem. In this sense the results of the different querying algorithms from this simulation study should not be interpreted as guarantees that generalize for any other application in text classification, but rather, as a demonstration that the choice of querying algorithm can matter and researchers should consider experimenting with different algorithms. Furthermore the query by committee algorithm had the disadvantage of the mismatch of classifier that was used to evaluate and the model involved in the query step.

The length and style of the documents seem to matter as well. The difference between the active and random labeling strategies is less evident for the Breitbart corpus than for the other two. The Twitter corpus, which is comprised of the shortest documents, show the largest differences.

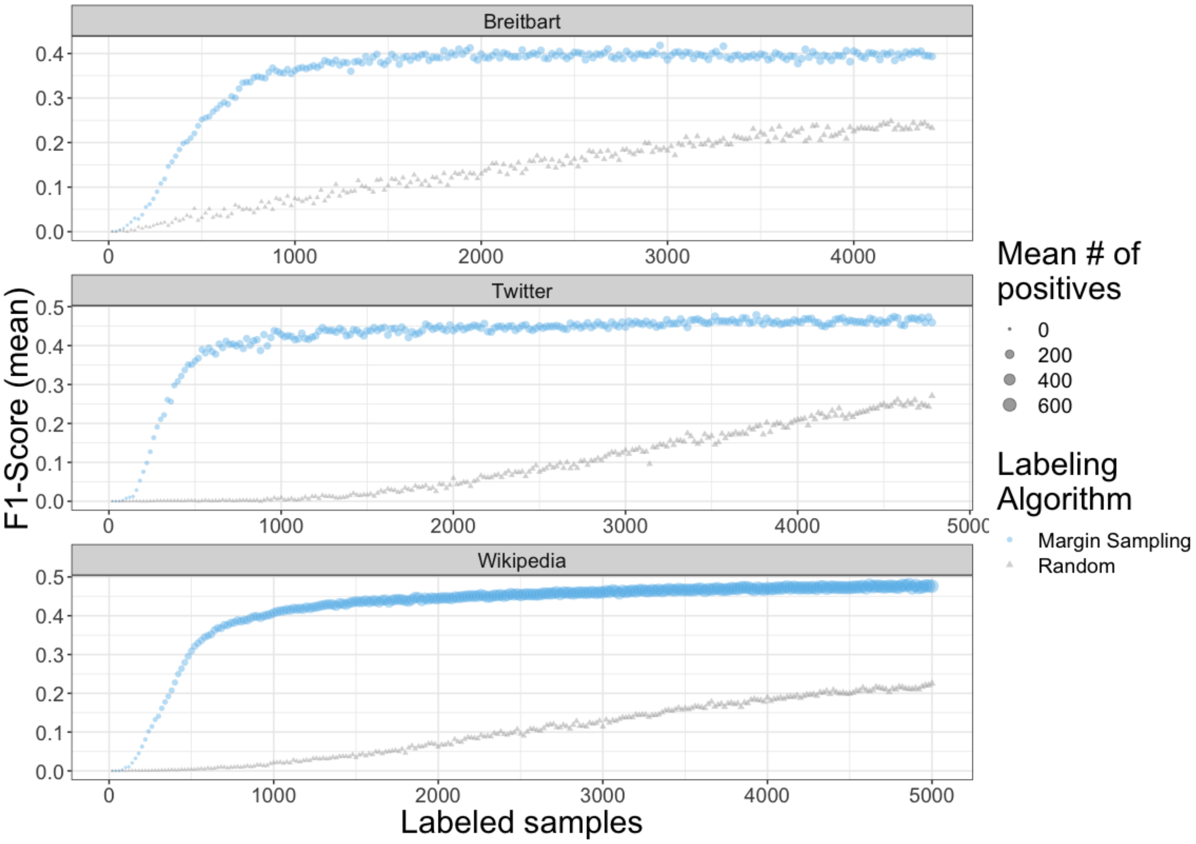

Figure 4: Performance by number of labeled examples for classifiers trained with active and passive learning (with class balance 0.01). Dots represent the average classifier performance across replications. Dot size is proportional to the average number of positively labeled observations in the training sample across replications.

To illuminate further why the active learner performs so much better in the imbalanced conditions, we zoom in on the 1% relevant data condition and additionally display the number of positive (or minority class) documents in the training data. These results are displayed in Figure 4. In this figure, each dot represents the average F1-score across experimental replications. The size of the dots represents the average number of positively labeled documents in the training data. The following pattern can be observed: The active learning algorithm acquires positively labeled documents (that is, documents belonging to the minority) much faster than the random learner. While the proportion of positive sample increases linearly—proportionally increasing with the size of the training data—for the random learner, the active learner acquires almost all its positive samples within the first 1000 training data samples. This explains the performance difference: The active learner has much more information on the minority class much earlier, allowing it to perform better more quickly. This is additionally evident when inspecting precision and recall separately (see Figures 7 and 8 in the Appendix). The availability of a more balanced training set (that is, more observations of the minority class), leads to drastically improved recall which in turn is pushing the F1-Score which is composed of precision and recall.

The better performance of active learning is mostly driven by this increased recall. In some instances when precision is higher for active learning than for random annotation, the results are mixed. In some cases in our simulations, the active learning algorithms produce even lower precision. The slightly reduced precision is likely due to the same mechanism: in very imbalanced situations where the algorithm has very little training data on the minority class, it is likely to simply predict the majority class for most observations—which can produce very high precision and very low recall. Therefore as the classifier gains more information on both classes false positives become more likely, reducing the overall precision. The reduction in precision is minor however and dwarfed by the increase in recall. For researchers that for some reason specifically want to improve precision this could be a concern.[15]

Figure 5 displays the inter-coder reliability results that were obtained by running the experiment with varying probability of error in the label assignment for the imbalanced condition (0.01) on the Twitter data. As expected, the overall performance of the classifier is affected by the amount of noise introduced into the training data by the simulated low coder reliability. Active learning outperforms random sampling even when the assignment of labels to the data is noisy. This holds for both the query by committee and margin algorithms. The margin algorithm seem to be affected more by additional samples added to the training data after the initial one to two hundred. We

[15] It should be noted, however, that in cases where precision is crucial, a different loss-function or classification threshold could be used to trade-off recall for precision.

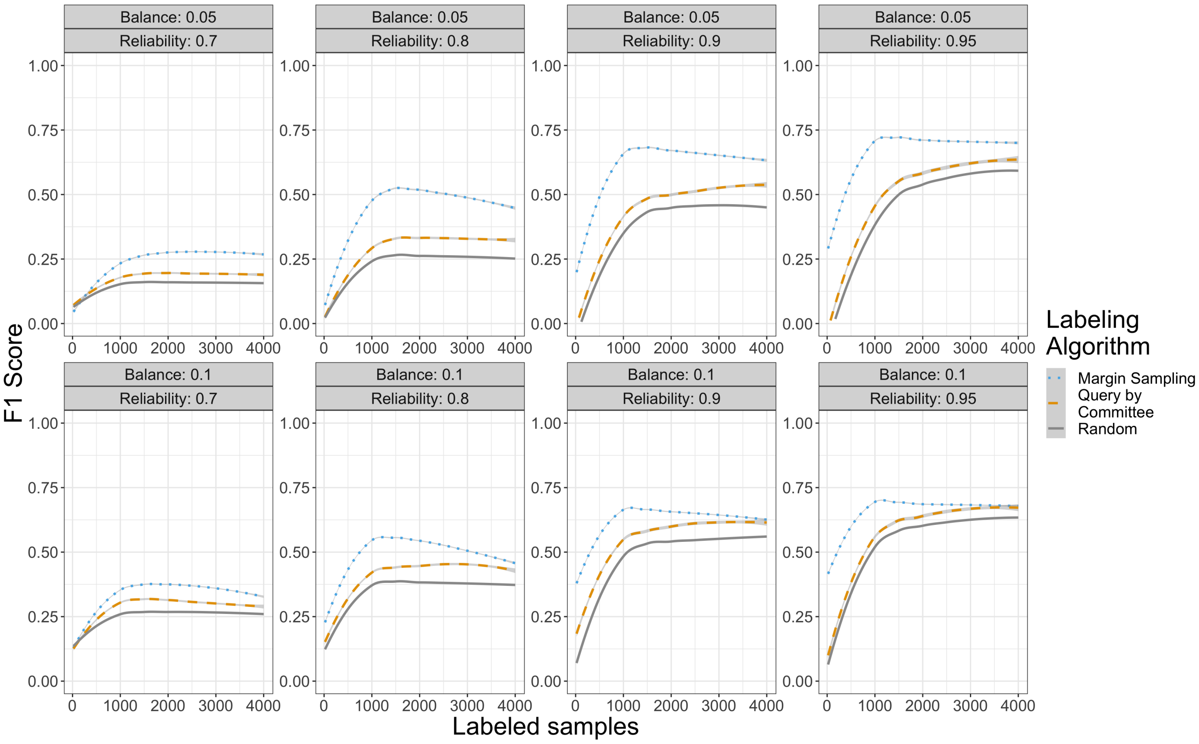

Figure 5: Active learning and passive learning performance on the Twitter dataset with different levels of simulated inter-coder reliability.

speculate that this has to do with the balance in the available training data. As shown above in Figure 4, the algorithm gets most of the minority class samples early (that is, within the first 1000 labeled observations). Therefore most of the additional labeled samples added to the training later are from the majority class and most labels for the minority class are results of the error process. That is, errors that are introduced in this later data affect only one class. This increase in false positives leads to a relatively significant decline in precision (see Figure 9 in the Appendix). The query by committee algorithm seems less susceptible to this decrease in precision even though not immune.

In Figure 6 we explore the topic of generalization error further. Each panel mirrors the panels in Figure 3. Since we are discussing the results in terms of error the y-axis displays 1 minus the F1-score. The red dashed line is the performance of the margin algorithm on the held out sample, and is identical to the corresponding line in Figure 3. The blue line displays the average of the average F1-scores obtained from cross validating the model on the available labeled data. That is, the blue line represents the estimate of the generalization error as it would be available to a researcher that does not

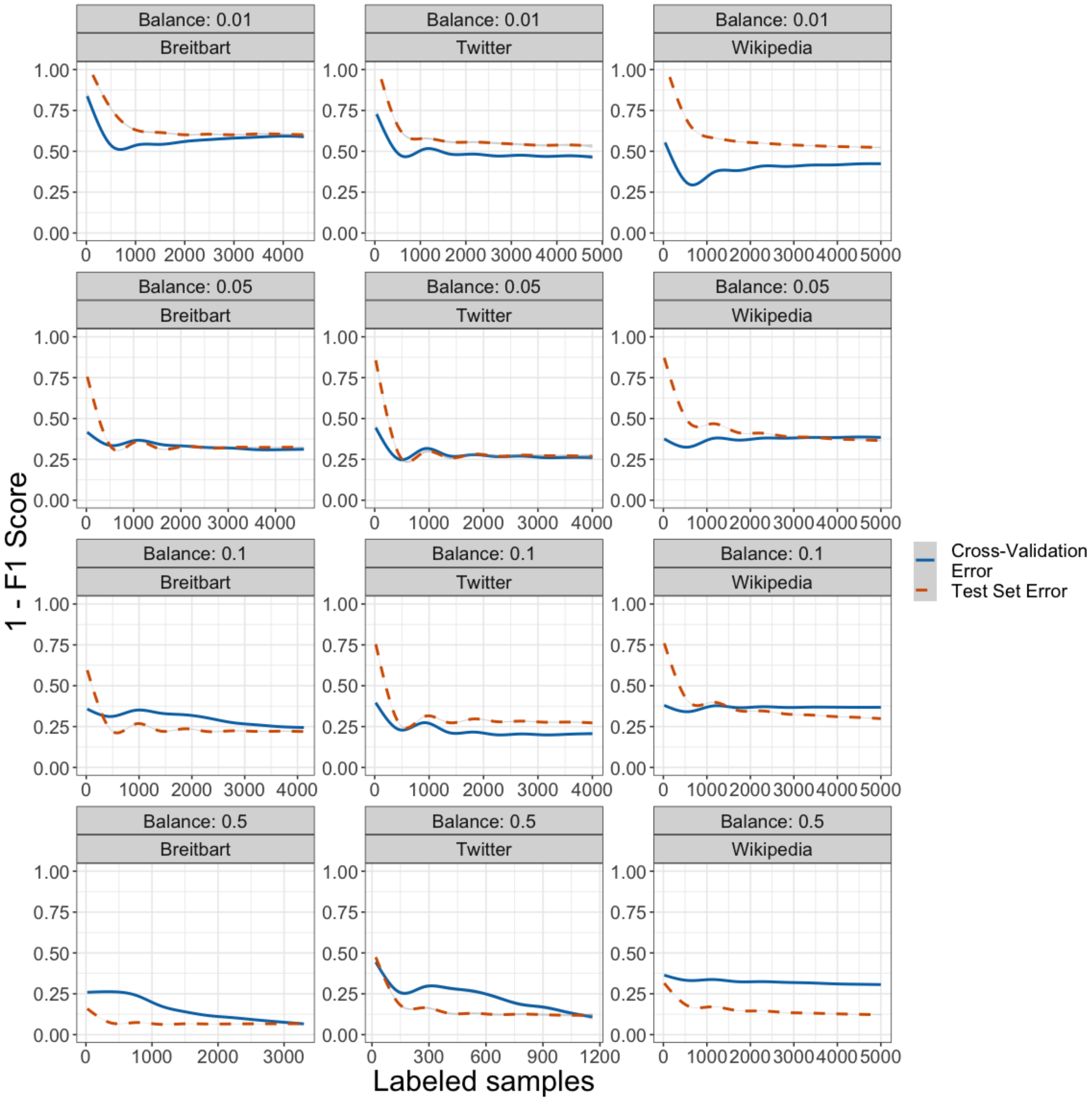

Figure 6: Comparison of generalization error (in terms of F1 score) in training and test set for active learning with margin querying strategy. Results are from the Twitter dataset.

have an unbiased ground-truth test set whereas the red line represents the true generalization error. We emphasize that, above, we established that active learning outperforms random sampling of the training data. In this section we only discuss the estimation of generalization error to the population of samples. The results in Figure 6 demonstrate that the estimated generalization error in the labeled data set is indeed different from the unbiased estimate. In the more balanced conditions the the error estimated from the labeled data tends to over estimate the error, whereas in the imbalanced conditions it tends to under estimate the error.

The lack of a precise estimate of generalization error is a disadvantage that has to be weighed against the improved performance of active learning. We expect that in most cases improved performance would be valued higher than confidence about the exact performance. But if a precise estimate of generalization error is required, we recommend researchers use part of their labeling budget to label randomly selected samples. This portion of the data can then be used to estimate this error.

There may be concern that the hyperplane fit in each iteration of the margin algorithm might get stuck in a suboptimal part of the version space. To investigate this we ran simulations varying the percent of each batch that is sampled randomly: 25%, 50% or 75% of documents were sampled randomly and the remaining sampled from proximity to the hyperplane. Including some purely random selection allows more exploration of the feature space and prevents the active learner from restricting its queries to a submanifold of our feature space. Performance was about the same across conditions, which suggests that, at least in the Twitter data, trapping suboptimal submanifolds are not a large concern.[16] Additionally, these simulations demonstrate that much of the gain in classifier performance is coming from the few observations in each batch that are closest to the hyperplane: decreasing the batch size may increase the efficiency of active learning if it is computationally feasible to do so.

5 Discussion

In this paper, we introduce the active learning paradigm and illustrate when it can be useful for labeling texts. We conduct several simulations with three datasets, varying the class balance, document length and sample type. We find that that the active learning approach—when compared to random sampling—reduces the costs of supervised learning in many realistic scenarios. Much of the extra cost with random sampling is due to the random approach's inefficient selection of documents to label when classes are imbalanced. Random sampling can produce very imbalanced training

[16] See Appendix Figure 11 for a visualization of performance at the different proportions of random observations per batch.

data that have too much information on some part of the feature space and not enough on others. For imbalanced datasets we find that passive learning methods can require up to 8 times as much training data to achieve the same performance as active learning. We also find that these results hold even in the presence of noise in the labeling process. The results, especially with regard to the balance of the dataset, confirm theoretical expectations that the active learning algorithms help to reduce inefficiency in labelling. We also demonstrate that the choice of querying method can affect the amount of labels required to achieve a specific level of classification performance. Given that general statements about the potential performance of different querying algorithms are difficult to make, we recommend that researchers experiment with different methods. Overall, active learning produces a biased training dataset that produces classification results that are better or at least as good as random sampling on out-of-sample data.

This bias in the training set, however, leads to some potential limitations that applied researchers should be aware of: relying solely on the actively sampled training data prohibits accurate estimation of the true generalization performance of the model. This can be a major concern in academic settings, where the error rate of the machine learning model is relevant to the confidence in substantive results that are based on the predictions of said model. However, this problem can easily be circumvented by dedicating a small amount of the labeling budget to a held out random sample that is exclusively used to estimate out-of-sample performance.

Not knowing the true performance of the model complicates model selection as well. If no randomly sampled data are available, the model cannot be optimized for prediction on out-of-sample data. Furthermore it is unknown whether the training data that were selected through active learning are optimal for other classification models as well. This may lead to concerns that the researcher has to commit to a specific classification model already at the labeling stage of the research project. It is important to note, however, that the evaluations in our simulation study were done on a randomly selected held-out dataset. That is, given our results and a large literature in computer science, applied researchers can expect that the active learning model performs better or at least as well in practice, even if additional effort would be required to obtain an unbiased estimate of the performance. Regarding the second concern about "lock-in" on a specific model, our results on query by committee suggest that active learning methods can outperform random sampling even in situations where a different final learning model is used with the training data.

It should be noted that the limitations that apply to "classical" machine learning for text classification apply to active learning as well. For instance, a classifier can only be as good as the training data it is based on. In cases where the language pertaining to the relevant class of documents changes (*concept drift*) the performance of the classifier can decrease drastically if applied to data from different time points. Furthermore, the classification methods that are used should be appropriate to the data at hand. For instance, if the data has known dependency structures like clusters or time-series structure, more appropriate classification algorithms might be used that do not assume a collection of independent observations.

A concern specific to active learning is that there may be an interaction between structure in the data and the dependence in samples. That is, in a case where the initial set of labeled data is an odd sample that only contains documents from one time period or cluster, could it happen that the model is led astray by this structure and never obtains data that is informative for other clusters? The results for the data in our experiments alleviates this concern. No guarantees can be made that generalize for all applications, but one can argue that active learning is a priori better suited to explore the relevant sections of the data. For instance if the initial sample is from a specific cluster or time period in the data, the active learner may be more likely to explore other clusters in the data because the observations there are different from the observations in the "known" cluster and therefore the model is less certain about them.[17]

Our results demonstrate that in many scenarios, active learning can provide large efficiency gains for supervised text analysis methods in political science (or machine learning applications in general). These efficiency gains are most dramatic when classes

[17] There are also algorithms that are built precisely on this intuition of using clustering structure in the data to explore it efficiently (Nguyen & Smeulders, 2004)

are highly imbalanced. Having imperfect coder reliability does not eliminate the efficiency advantage that active learning has over random sampling. We hope our findings will serve to promote the use of supervised learning approaches for automated text analysis by making the process less costly and less time-intensive for researchers.

References

Abe, H., & Mamitsuka, N. (1998). Query learning strategies using boosting and bagging. In *Machine learning: proceedings of the fifteenth international conference (icml98)* (Vol. 1).

Ali, A., Caruana, R., & Kapoor, A. (2014). Active learning with model selection. In *Aaai* (pp. 1673–1679).

Angluin, D. (1988). Queries and concept learning. *Machine learning*, *2*(4), 319–342.

Baram, Y., Yaniv, R. E., & Luz, K. (2004). Online choice of active learning algorithms. *Journal of Machine Learning Research*, *5*(Mar), 255–291.

Beck, N., King, G., & Zeng, L. (2000). Improving quantitative studies of international conflict: A conjecture. *American Political Science Review*, *94*(1), 21–35.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, *13*(Feb), 281–305.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, *3*(Jan), 993–1022.

Ceron, A., Curini, L., Iacus, S. M., & Porro, G. (2014). Every tweet counts? how sentiment analysis of social media can improve our knowledge of citizens political preferences with an application to italy and france. *New Media & Society*, *16*(2), 340–358.

Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J. L., & Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems* (pp. 288–296).

Collingwood, L., & Wilkerson, J. (2012). Tradeoffs in accuracy and efficiency in supervised learning methods. *Journal of Information Technology & Politics*, *9*(3), 298–318.

Cranmer, S. J., & Desmarais, B. A. (2017). What can we learn from predictive modeling? *Political Analysis*, *25*(2), 145–166.

Dagan, I., & Engelson, S. P. (1995). Committee-based sampling for training probabilistic classifiers. In *Machine learning proceedings 1995* (pp. 150–157). Elsevier.

Dasgupta, S., Kalai, A. T., & Monteleoni, C. (2005). Analysis of perceptron-based active learning. In *International conference on computational learning theory* (pp. 249–263).

Denny, M. J., & Spirling, A. (2018). Text preprocessing for unsupervised learning: why it matters, when it misleads, and what to do about it. *Political Analysis*, 1–22.

Drutman, L., & Hopkins, D. J. (2013). The inside view: Using the enron e-mail archive to understand corporate political attention. *Legislative Studies Quarterly*, *38*(1), 5–30.

Ducoffe, M., & Precioso, F. (2015). Qbdc: query by dropout committee for training deep supervised architecture. *arXiv preprint arXiv:1511.06412*.

Ertekin, S., Huang, J., Bottou, L., & Giles, L. (2007). Learning on the border: active learning in imbalanced data classification. In *Proceedings of the sixteenth acm conference on conference on information and knowledge management* (pp. 127–136).

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, *55*(1), 119–139.

Freund, Y., Seung, H. S., Shamir, E., & Tishby, N. (1997). Selective sampling using the query by committee algorithm. *Machine learning*, *28*(2-3), 133–168.

Freytag, A., Rodner, E., & Denzler, J. (2014). Selecting influential examples: Active learning with expected model output changes. In *European conference on computer vision* (pp. 562–577).

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* (Vol. 1). Springer series in statistics New York.

Grimmer, J. (2010). A bayesian hierarchical topic model for political texts: Measuring expressed agendas in senate press releases. *Political Analysis*, *18*(1), 1–35.

Grimmer, J., & Stewart, B. M. (2013). Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political analysis*, *21*(3), 267–297.

Hill, D. W., & Jones, Z. M. (2014). An empirical evaluation of explanations for state repression. *American Political Science Review*, *108*(3), 661–687.

Javed, J., & Miller, B. (2018). Mobilizing hate: Moral-emotional frames, outrage, and violent expression in online media. *unpublished*.

Körner, C., & Wrobel, S. (2006). Multi-class ensemble-based active learning. In *European conference on machine learning* (pp. 687–694).

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, *22*(1), 79–86.

Lewis, D. D., & Catlett, J. (1994). Heterogeneous uncertainty sampling for supervised learning. In *Machine learning proceedings 1994* (pp. 148–156). Elsevier.

Linder, F. (2017). Improved data collection from online sources using query expansion and active learning.

Lu, Z., & Bongard, J. (2009). Exploiting multiple classifier types with active learning. In *Proceedings of the 11th annual conference on genetic and evolutionary computation* (pp. 1905–1906).

Mebane Jr, W. R., Klaver, J., & Miller, B. (2016). *Frauds, strategies and complaints in Germany.* (Paper presented at the 2016 Annual Meeting of the Midwest Political Science Association, Chicago, April 7–10, 2016)

Mebane Jr, W. R., Pineda, A., Woods, L., Klaver, J., Wu, P., & Miller, B. (2017). *Using Twitter to observe election incidents in the United States.* (Paper presented at the 2016 Annual Meeting of the Midwest Political Science Association, Chicago, April 6–9, 2017)

Melville, P., & Mooney, R. J. (2004). Diverse ensembles for active learning. In *Proceedings of the twenty-first international conference on machine learning* (p. 74).

Mikhaylov, S., Laver, M., & Benoit, K. R. (2012). Coder reliability and misclassification in the human coding of party manifestos. *Political Analysis*, *20*(1), 78–91.

Miller, B. (2016). Automated detection of chinese government astroturfers using network and social metadata.

Mitchell, T. M. (1982). Generalization as search. *Artificial intelligence*, *18*(2), 203–226.

Muchlinski, D., Siroky, D., He, J., & Kocher, M. (2016). Comparing random forest with logistic regression for predicting class-imbalanced civil war onset data. *Political Analysis*, *24*(1), 87–103.

Nguyen, H. T., & Smeulders, A. (2004). Active learning using pre-clustering. In *Proceedings of the twenty-first international conference on machine learning* (p. 79).

Quinn, K. M., Monroe, B. L., Colaresi, M., Crespin, M. H., & Radev, D. R. (2010). How to analyze political attention with minimal assumptions and costs. *American Journal of Political Science*, *54*(1), 209–228.

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.*

Roy, N., & McCallum, A. (2001). Toward optimal active learning through monte carlo estimation of error reduction. *ICML, Williamstown*, 441–448.

Schohn, G., & Cohn, D. (2000). Less is more: Active learning with support vector machines. In *Icml* (pp. 839–846).

Settles, B. (2012). Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, *6*(1), 1–114.

Settles, B., & Craven, M. (2008). An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 1070–1079).

Settles, B., Craven, M., & Ray, S. (2008). Multiple-instance active learning. In *Advances in neural information processing systems* (pp. 1289–1296).

Seung, H. S., Opper, M., & Sompolinsky, H. (1992). Query by committee. In *Proceedings of the fifth annual workshop on computational learning theory* (pp. 287–294).

Sun, Y., Wong, A. K., & Kamel, M. S. (2009). Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence*, *23*(04), 687–719.

Tong, S., & Koller, D. (2001). Support vector machine active learning with applications to text classification. *Journal of machine learning research*, *2*(Nov), 45–66.

Wallach, H. M., Murray, I., Salakhutdinov, R., & Mimno, D. (2009). Evaluation methods for topic models. In *Proceedings of the 26th annual international conference on machine learning* (pp. 1105–1112).

Wilkerson, J., & Casas, A. (2017). Large-scale computerized text analysis in political science: Opportunities and challenges. *Annual Review of Political Science*, *20*, 529–544.

Wilkerson, J., Smith, D., & Stramp, N. (2015). Tracing the flow of policy ideas in legislatures: A text reuse approach. *American Journal of Political Science*, *59*(4), 943–956.

Wolpert, D. H. (1996). The lack of a priori distinctions between learning algorithms. *Neural computation*, *8*(7), 1341–1390.

Workman, S. (2015). *The dynamics of bureaucracy in the us government: How congress and federal agencies process information and solve problems.* Cambridge University Press.

Zhao, L., Sukthankar, G., & Sukthankar, R. (2011). Incremental relabeling for active learning with noisy crowdsourced annotations. In *Privacy, security, risk and trust* *(passat) and 2011 ieee third inernational conference on social computing (socialcom), 2011 ieee third international conference on* (pp. 728–733).

6 Appendix

6.1 Precision Results

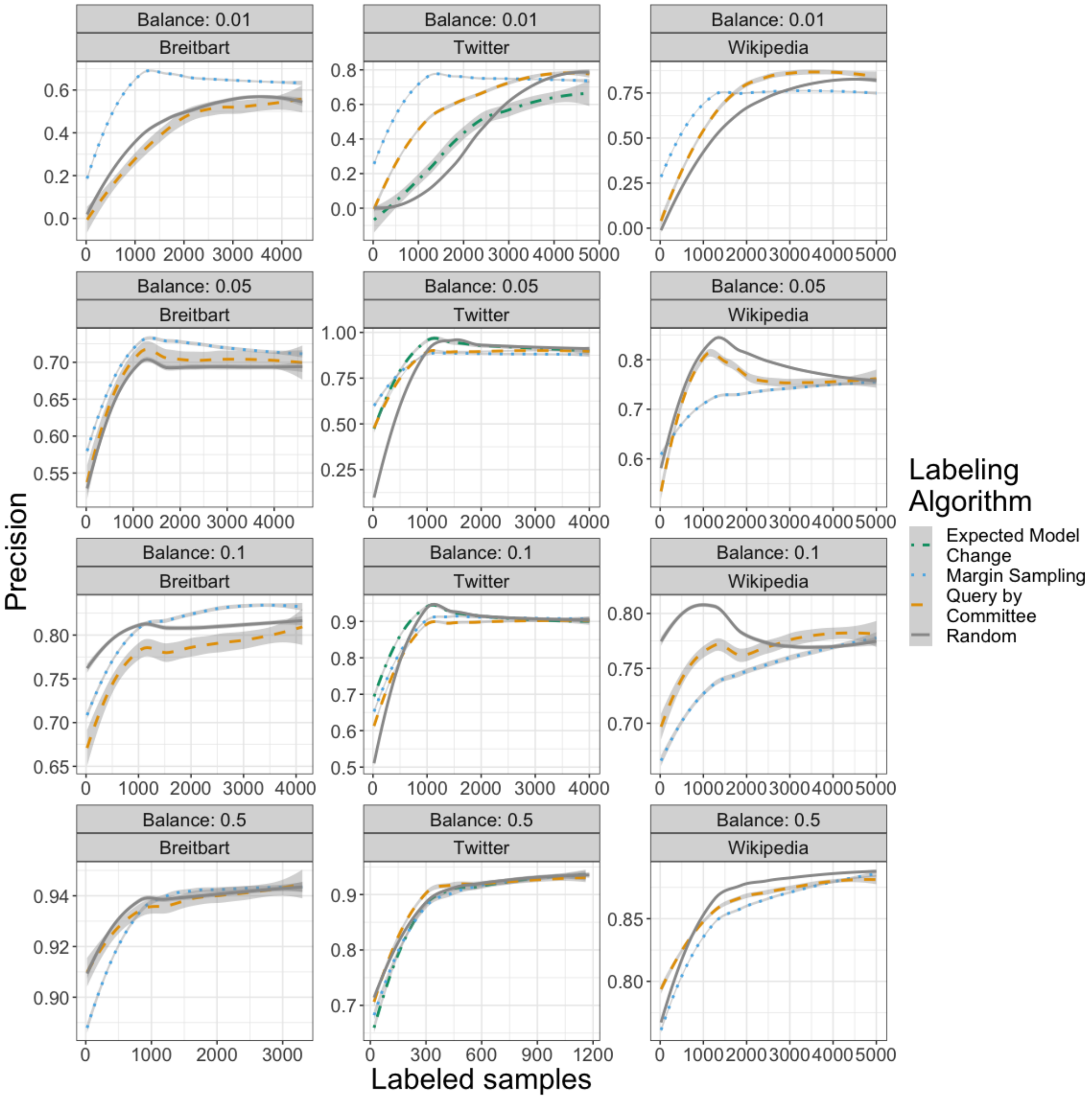

Figure 7: Precision score for experiments. The panel columns correspond to the datasets the rows to the different levels of class imbalance. Dots represent single replications of the experiment, smoothed lines are fits (and standard errors) of a generalized additive model.

6.2 Recall Results

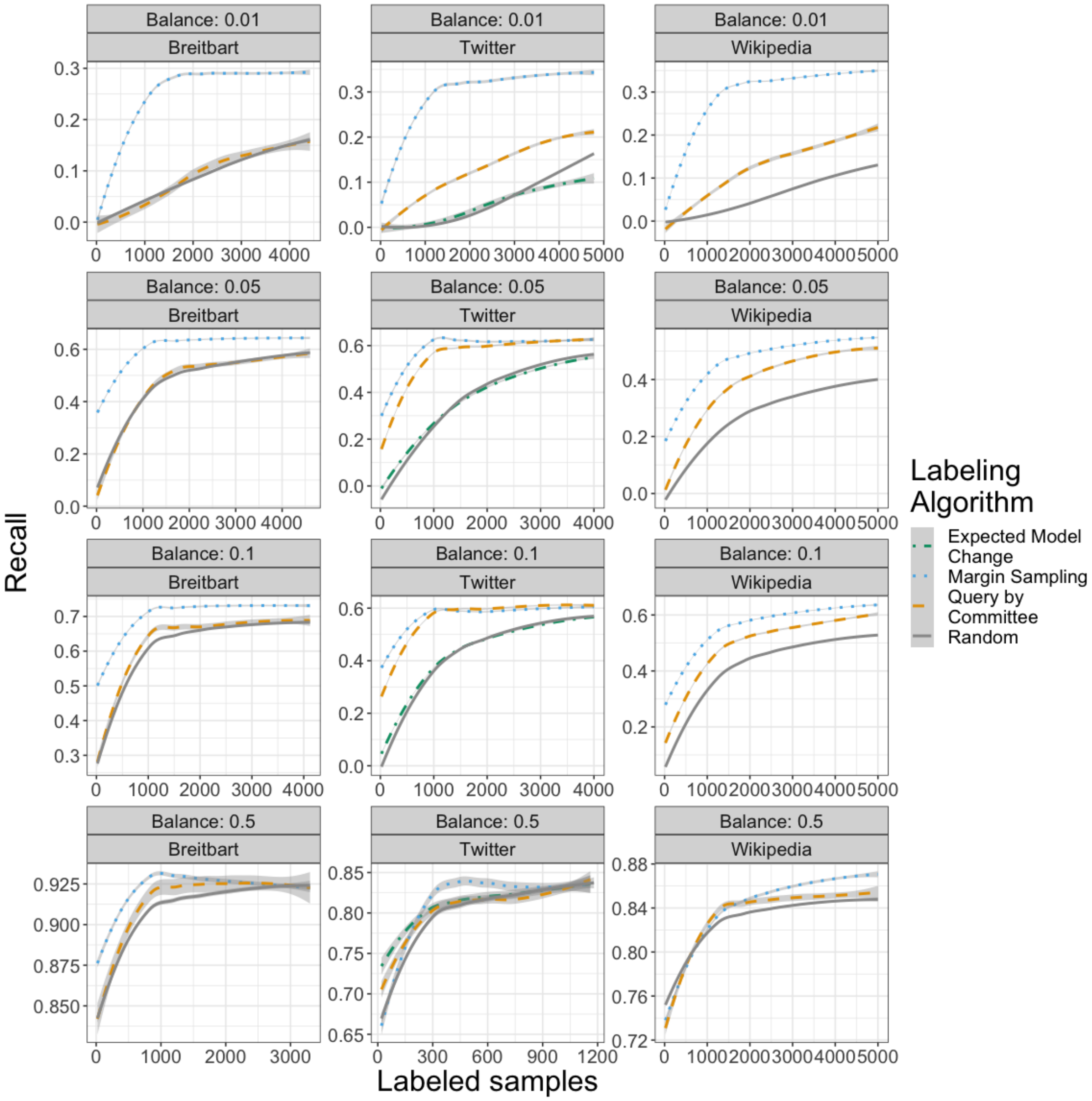

Figure 8: Recall score for experiments. The panel columns correspond to the datasets the rows to the different levels of class imbalance. Dots represent single replications of the experiment, smoothed lines are fits (and standard errors) of a generalized additive model.

6.3 Precision and Recall for Inter-coder Reliability Experiment

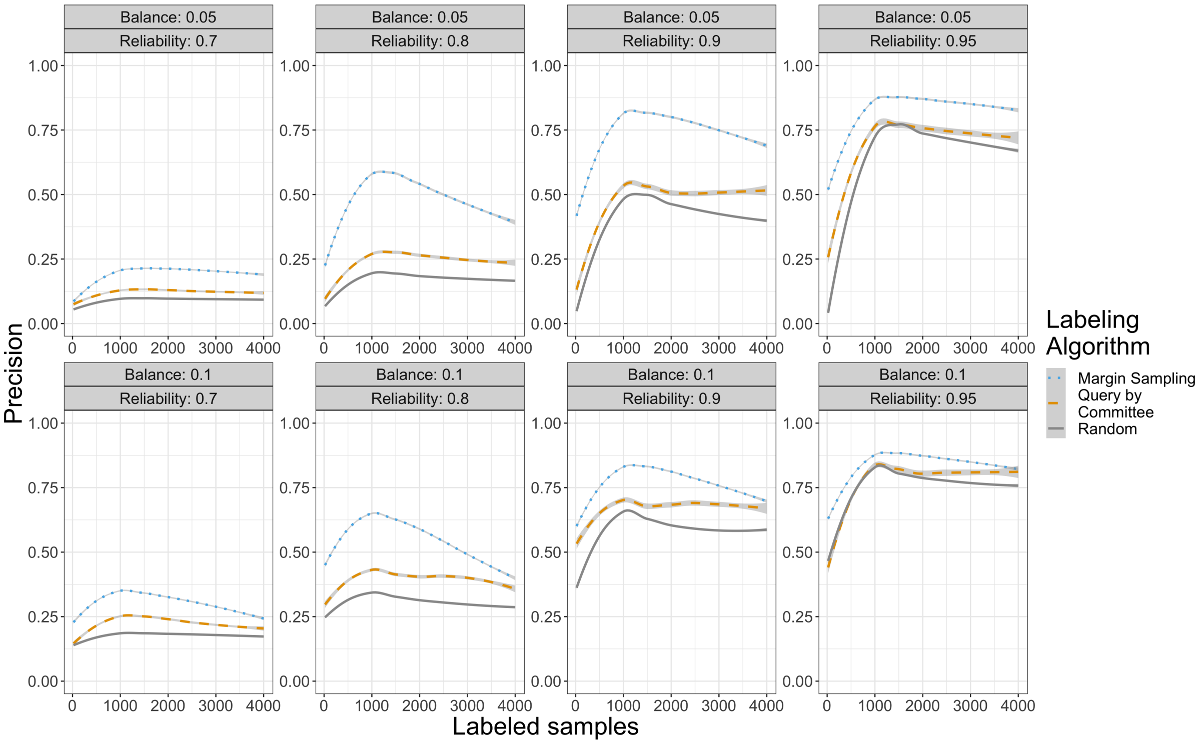

Figure 9: Active learning and passive learning performance on the Twitter dataset with different levels of simulated inter-coder reliability.

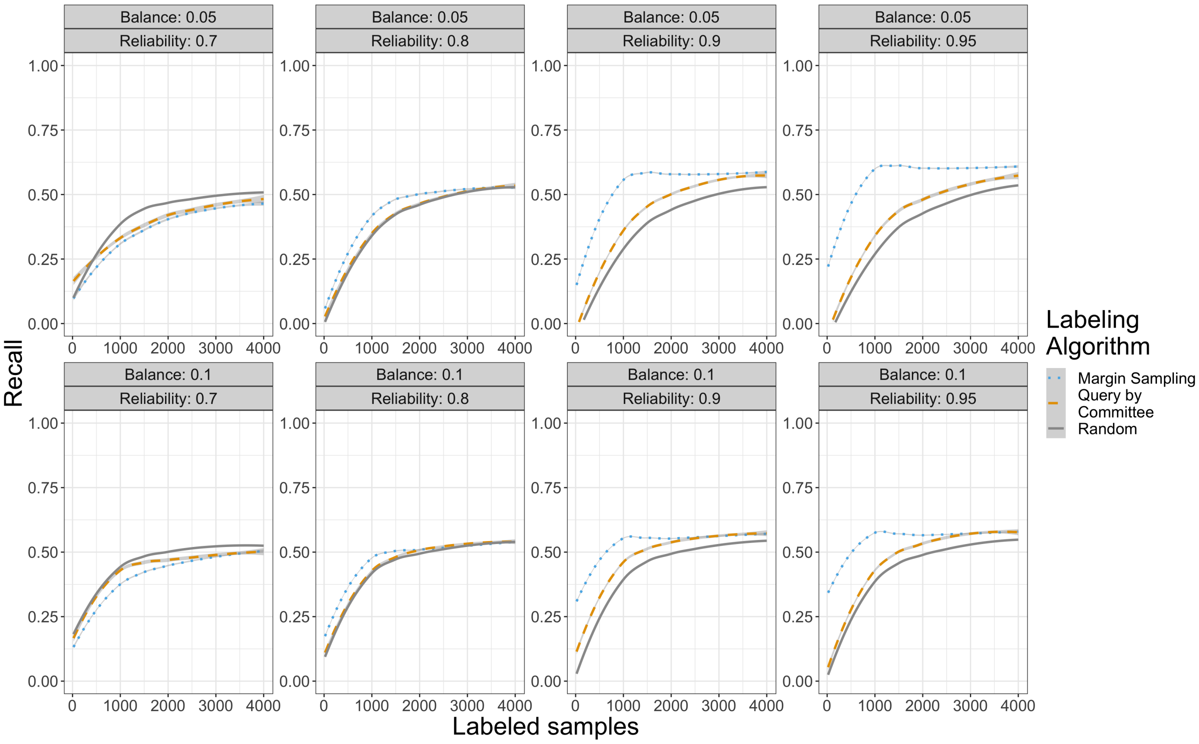

Figure 10: Active learning and passive learning performance on the Twitter dataset with different levels of simulated inter-coder reliability.

6.4 Active learning with partially randomly sampled data

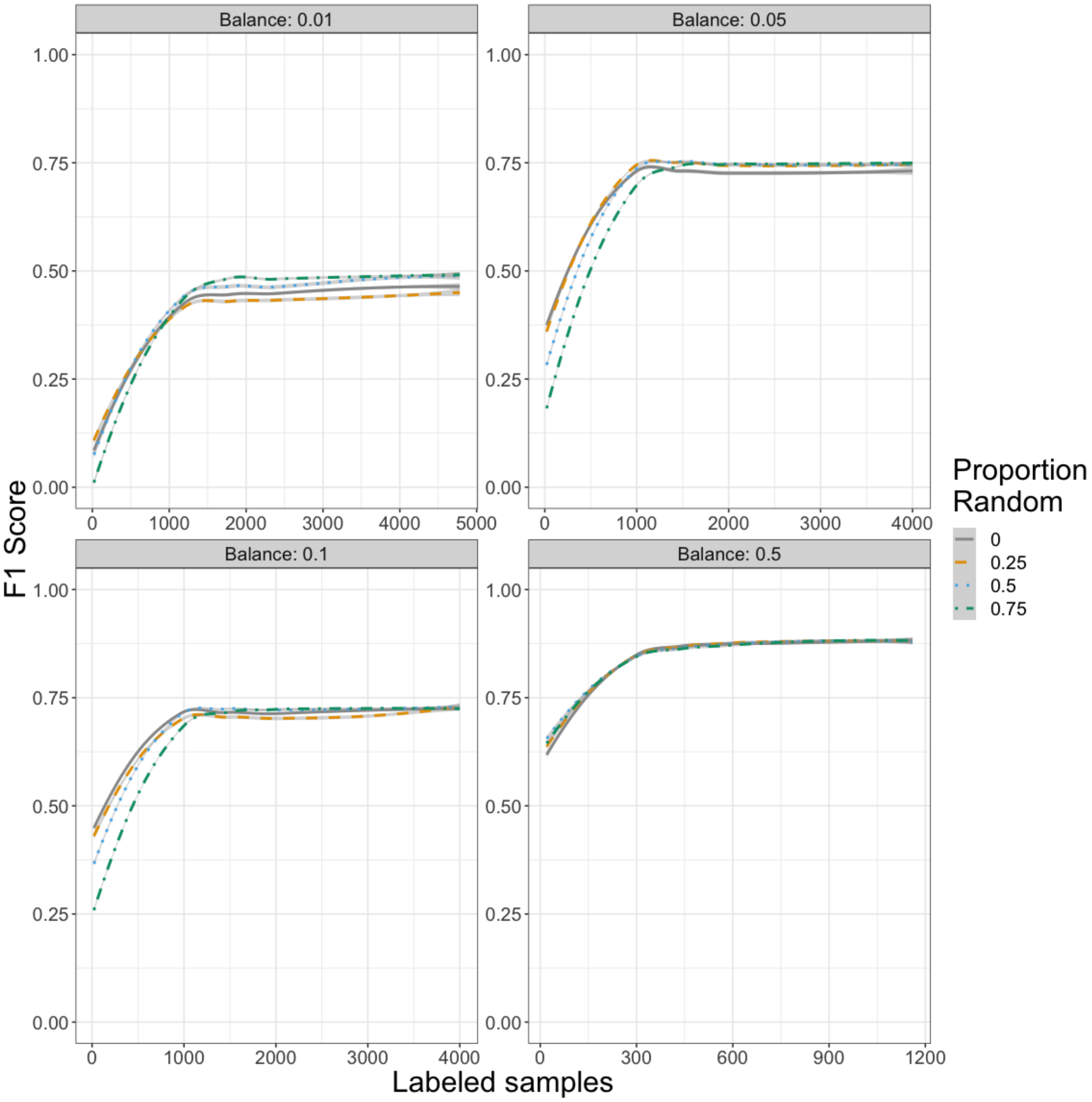

Figure 11: Performance on the held out data of active learning with margin querying strategy and additional randomly selected training data. Each line displays results for a different level of randomness. Results are from the Twitter dataset.

6.5 Pre-processing Choices

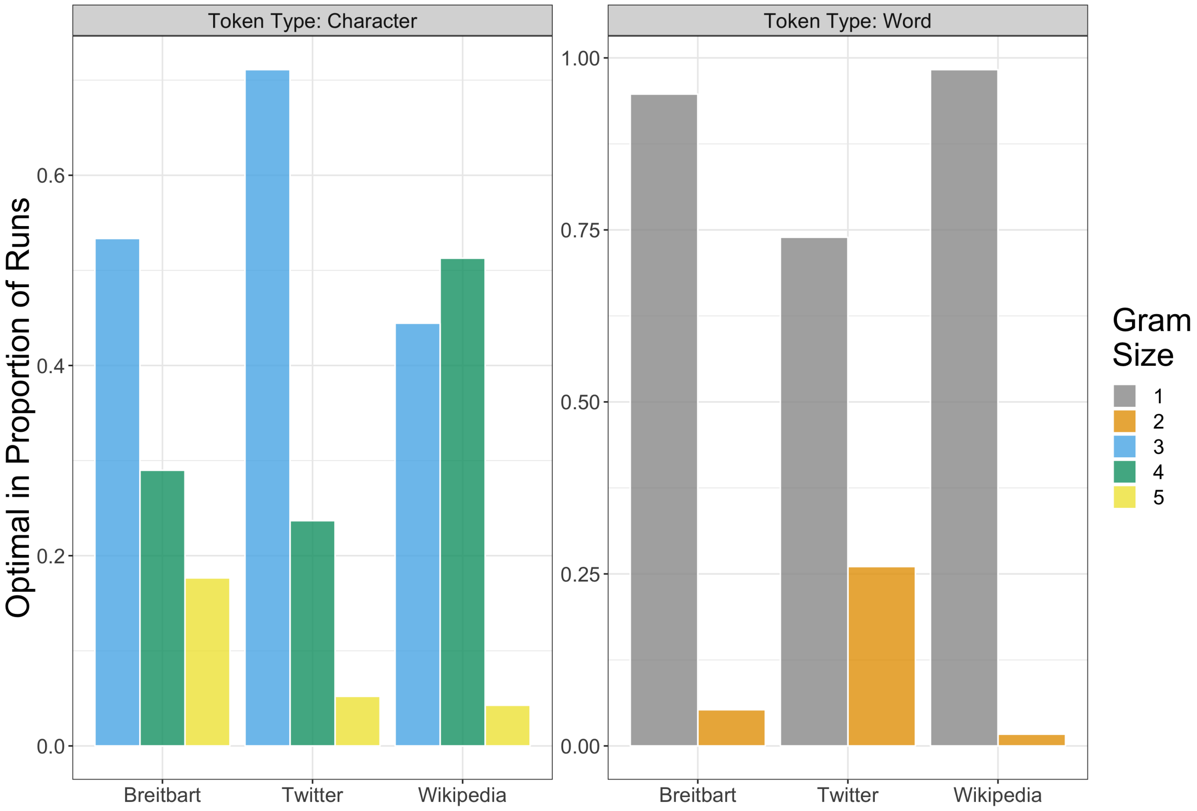

Figure 12: Distribution of n-gram sizes as chosen by the cross validation procedure across text corpora.

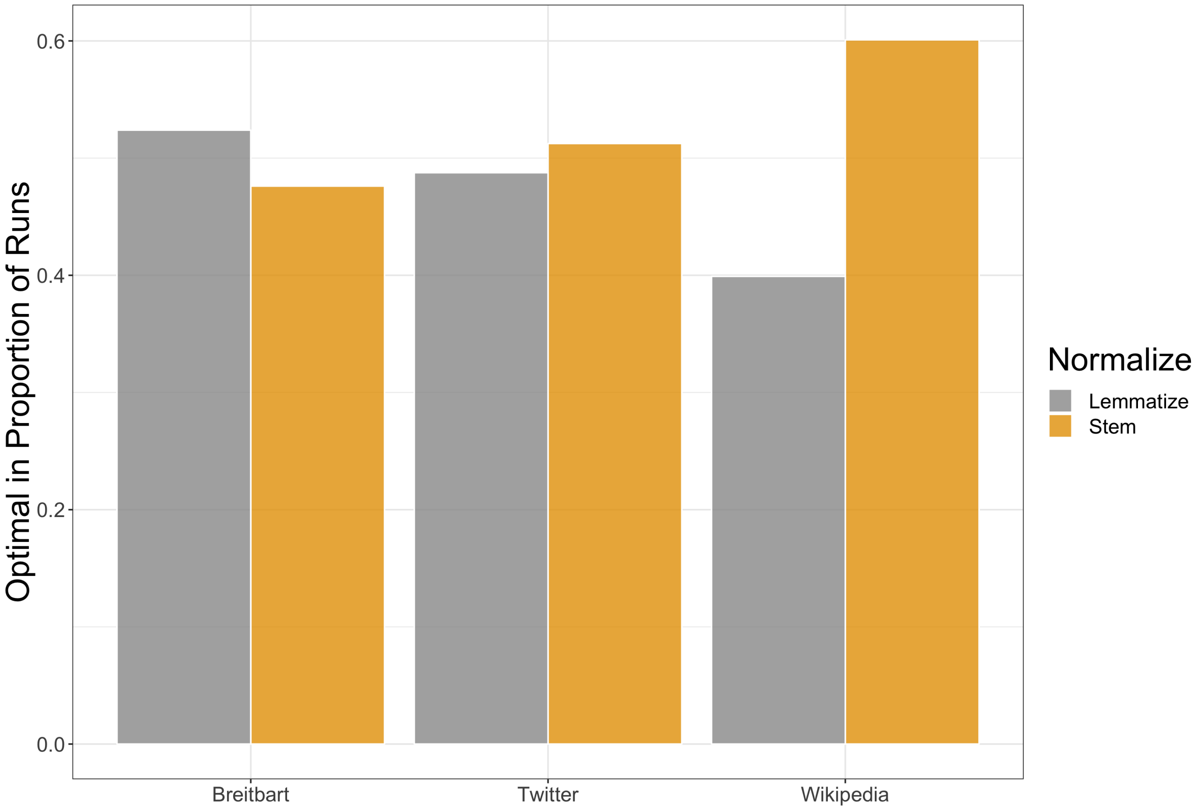

Figure 13: Distribution of token normalization technique as chosen by the cross validation procedure across text corpora.

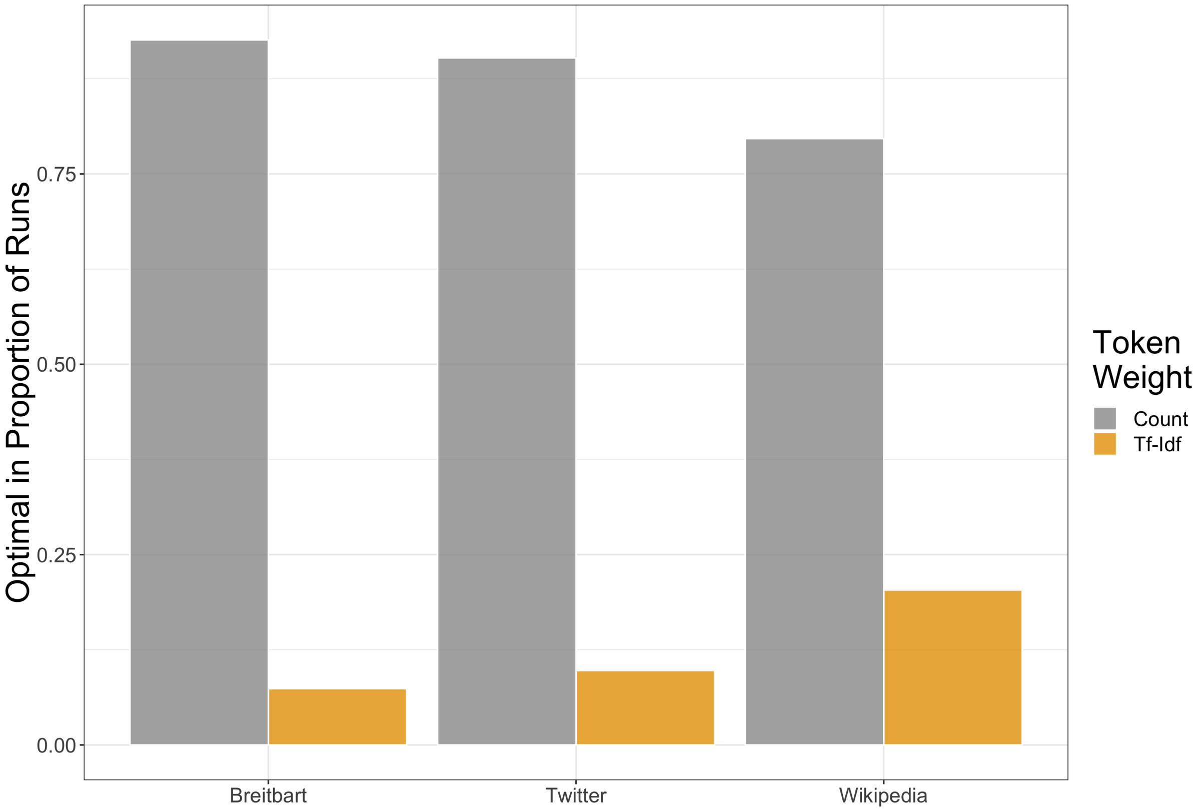

Figure 14: Distribution of token weight technique as chosen by the cross validation procedure across text corpora.

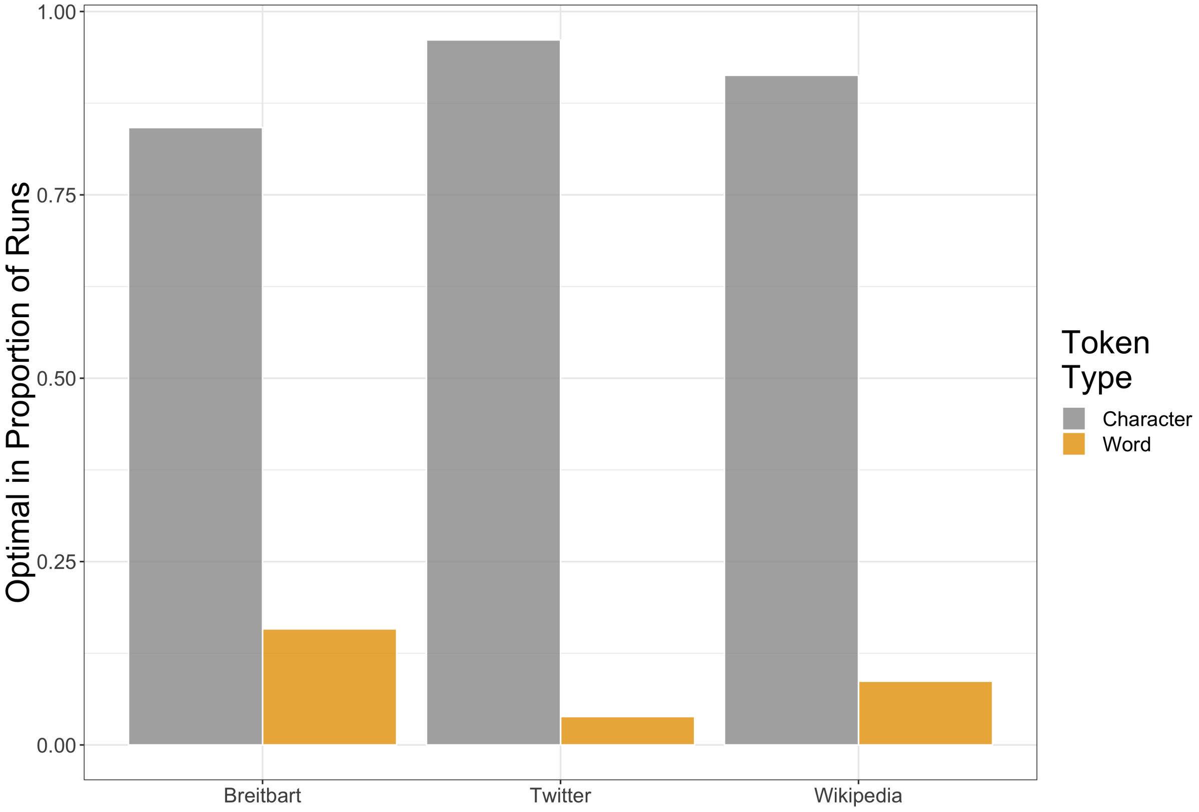

Figure 15: Distribution of token type technique as chosen by the cross validation procedure across text corpora.

6.6 Algorithms

6.6.1 Margin algorithm

1. Let $\mathcal{L}$ represent the set of labeled observations with predictors $\mathbf{X}_{\mathcal{L}}$ and labels $\mathbf{y}_{\mathcal{L}}$.
2. Train a regularized support vector machine (SVM) with L2 penalty
3. The class-separating hyperplane learned above $\mathbf{h} \subset \mathbb{R}^p$ is defined by $\mathbf{h} = \{\tilde{\mathbf{x}} : \mathbf{x}'\mathbf{w} + b = 0\}$
4. Calculate the distance between each unlabeled observation in $\mathcal{U}$ and the class separating hyperplane h learned in the previous step
5. Return a query set of the m (batch size) unlabeled observations closest to the hyperplane (m can be chosen to suit the specific coding task)
6. The expert labels each queried document.
7. Repeat Steps 2 - 6 with new labeled observations until a stopping criterion is reached.

6.6.2 Query by Committee

1. Let $\mathcal{L}$ represent the set of labeled observations with predictors $\mathbf{X}_{\mathcal{L}}$ and labels $\mathbf{y}_{\mathcal{L}}$.
2. Define a committee: $\mathcal{C} = \theta^{(1)}, ..., \theta^{(C)}$[18]
3. Using each model, predict the outcome all unlabeled observations $\hat{y}$
4. Calculate the vote entropy for each unlabeled observation $VE = -\sum_{i}^{C} \frac{V(y_i)}{C} log \frac{V(y_i)}{C}$ where $y_i \in \{0, 1\}$, $V(y_i)$ is the number of votes that label receives, and C is the committee size.
5. Return a query set of the m (batch size) unlabeled observations with the largest vote entropy VE.
6. The expert labels each queried document.
7. Repeat Steps 2 - 6 with new labeled observations until a stopping criterion is reached.

[18] For our simulations, we use 9 differently specified linear models (A naive bayes classifier and multiple models with different regularization choices: 2 logistic regression classifiers, 3 support vector machines, and 3 perceptron classifiers). When models have hyperparameters to tune, we choose the best performing classifier of 2 randomly specified models (hyperparameters drawn from exponential distributions at different scales) using cross-validated F1 score.

6.6.3 Expected Model Change

1. Let $\mathcal{L}$ represent the set of labeled observations with predictors $\mathbf{X}_{\mathcal{L}}$ and labels $\mathbf{y}_{\mathcal{L}}$.

2. Train a regularized support vector machine (SVM) with L2 penalty, $f(\mathbf{X}_{\mathcal{L}})$, using labeled observations $\mathcal{L}$

3. Using the model, predict the outcome all unlabeled observations $\hat{y}$

4. For each unlabeled observation:

    (a) Add the training tuple $\langle x_i, y_i \rangle$ from $\mathcal{U}$ to the labeled set $\mathcal{L}$

    (b) Retrain an SVM model with $\mathcal{L}^{+\langle x, \hat{y} \rangle} = \mathcal{L} \cup \langle x, \hat{y} \rangle$

    (c) Calculate a score for the model output change $s = sum(\ell_\theta(\mathcal{L}^{+\langle x, \hat{y} \rangle}; \theta))$ where ℓ_θ is the square loss function defined as 1 for $f(\mathbf{X}_{\mathcal{L}}) \neq f(\mathbf{X}_{\mathcal{L} \cup \langle x, \hat{y} \rangle})$ and 0 otherwise.

5. Return a query set of the m (batch size) unlabeled observations with the largest model change score s.

6. The expert labels each queried document.

7. Repeat Steps 2 - 6 with new labeled observations until a stopping criterion is reached.