Online Appendix for Active Learning Approaches for Labeling Text:

Review and Assessment of the Performance of Active Learning Approaches[*]

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

April 29, 2019

---

[*] Work supported by NSF award SES 1523355. Replication materials are available as **?**

[†] Department of Methodology, London School of Economics and Political Science, Columbia House, Houghton Street, London WC2A 2AE (E-mail: blakeapm@gmail.com).

[‡] 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 Appendix

1.1 Precision Results

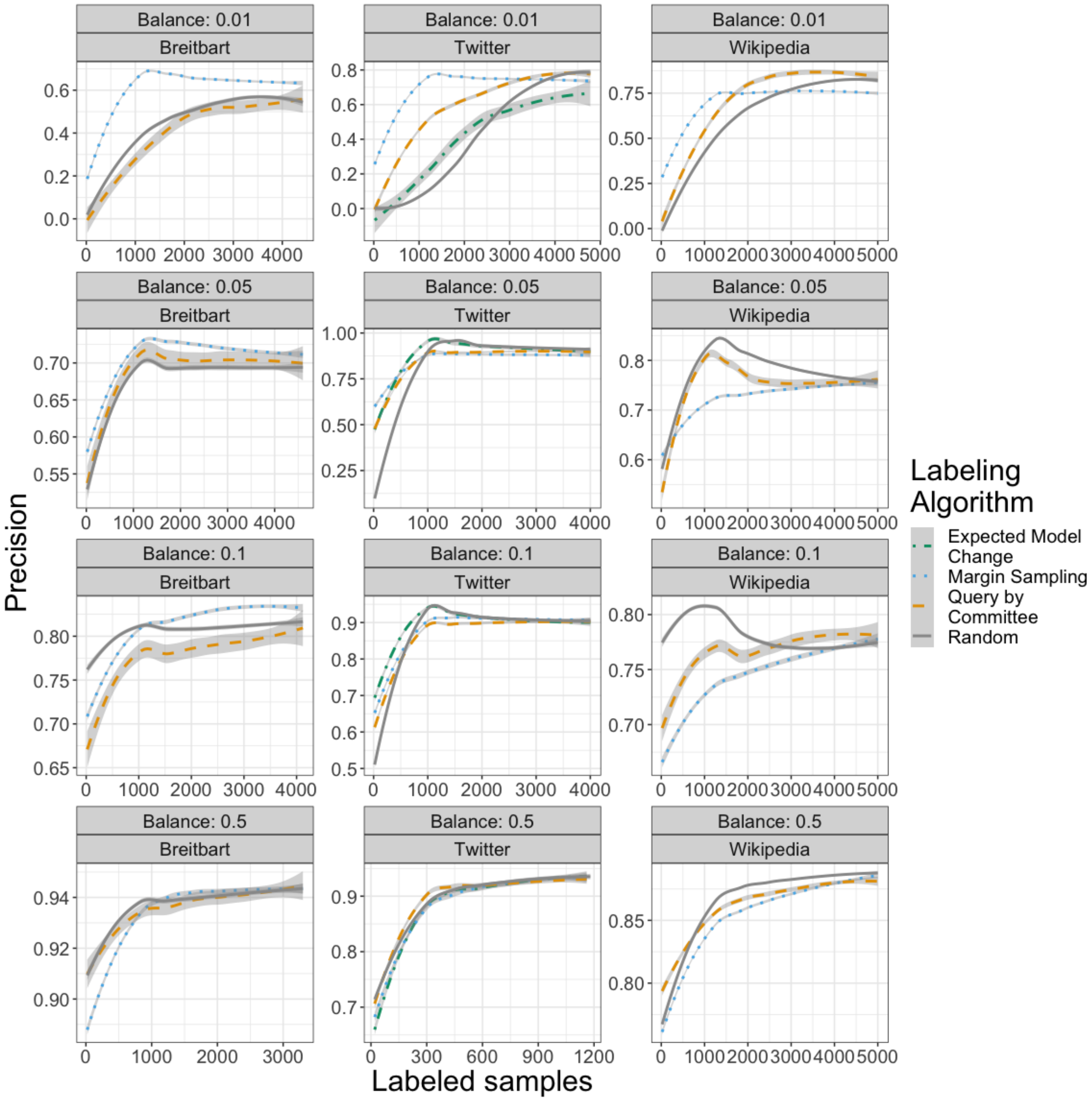

Figure 1: Precision score for experiments. The panel columns correspond to the datasets the rows to the different levels of class imbalance. Dots represent single replications of the experiment, smoothed lines are fits (and standard errors) of a generalized additive model.

1.2 Recall Results

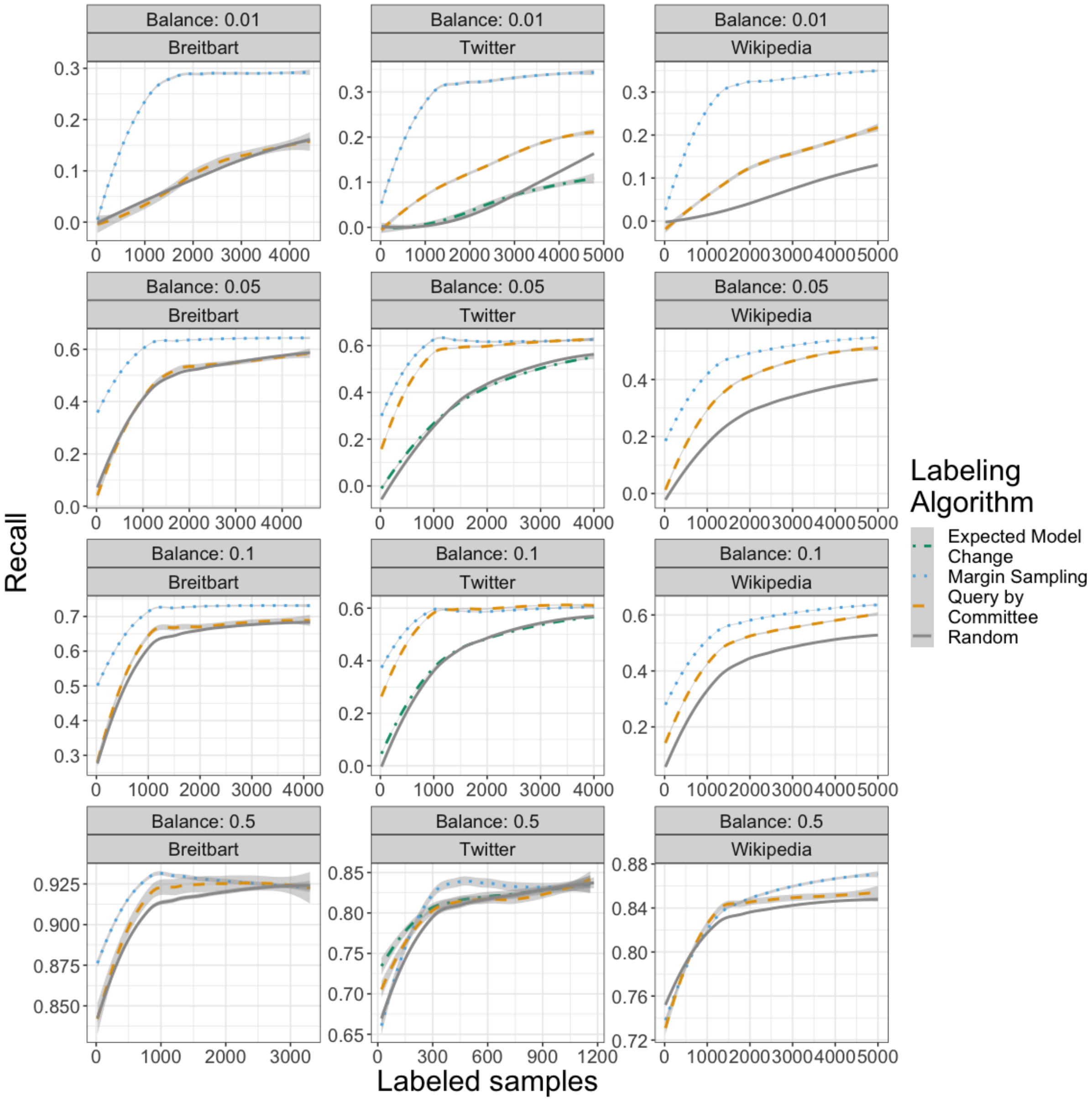

Figure 2: Recall score for experiments. The panel columns correspond to the datasets the rows to the different levels of class imbalance. Dots represent single replications of the experiment, smoothed lines are fits (and standard errors) of a generalized additive model.

1.3 Precision and Recall for Inter-coder Reliability Experiment

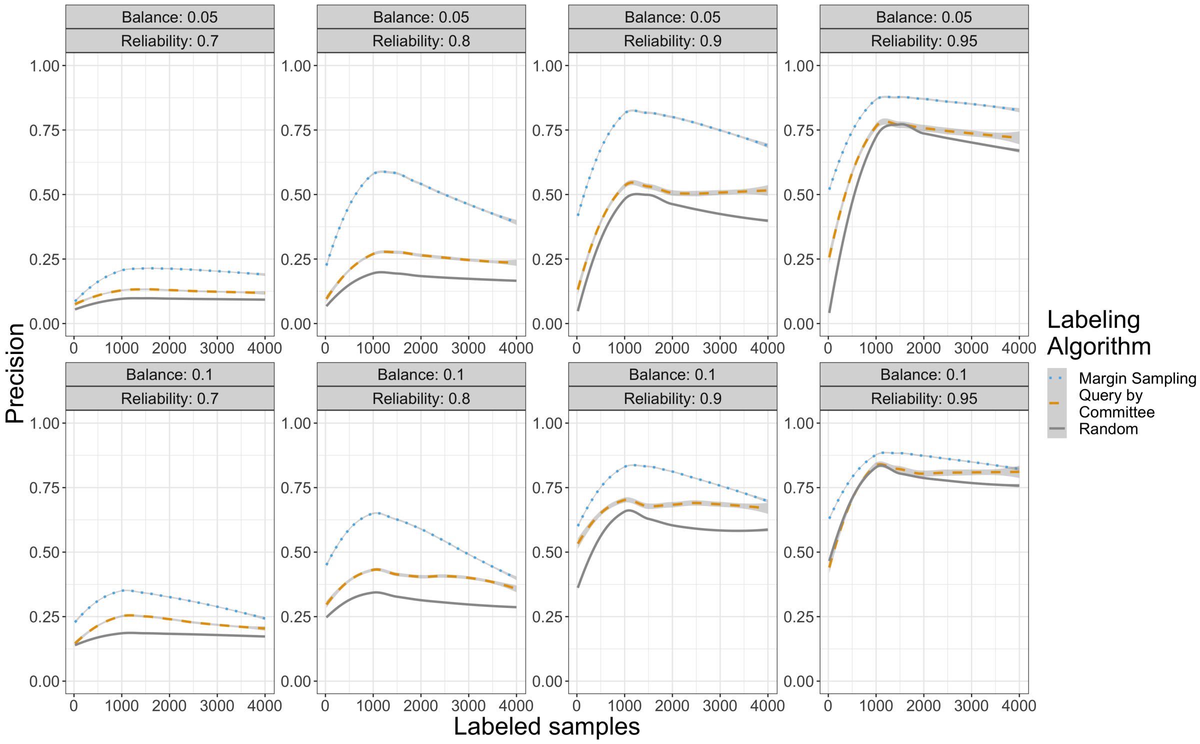

Figure 3: Active learning and passive learning performance on the Twitter dataset with different levels of simulated inter-coder reliability.

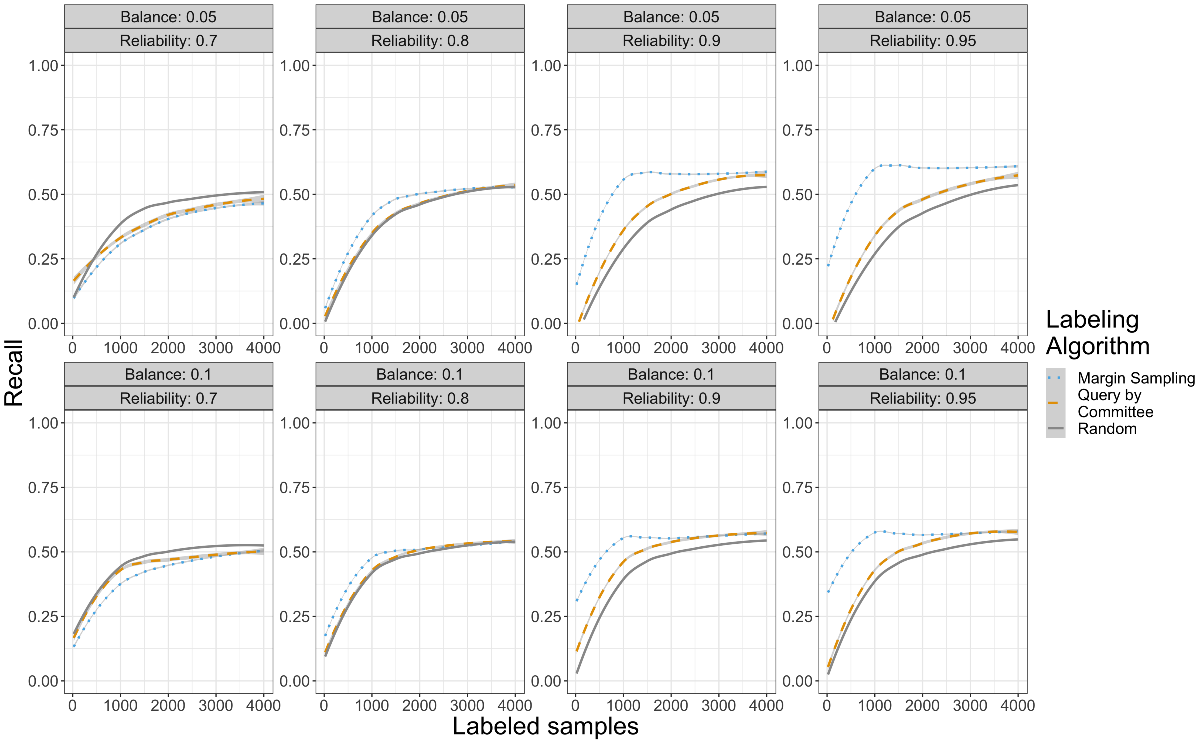

Figure 4: Active learning and passive learning performance on the Twitter dataset with different levels of simulated inter-coder reliability.

1.4 Active learning with partially randomly sampled data

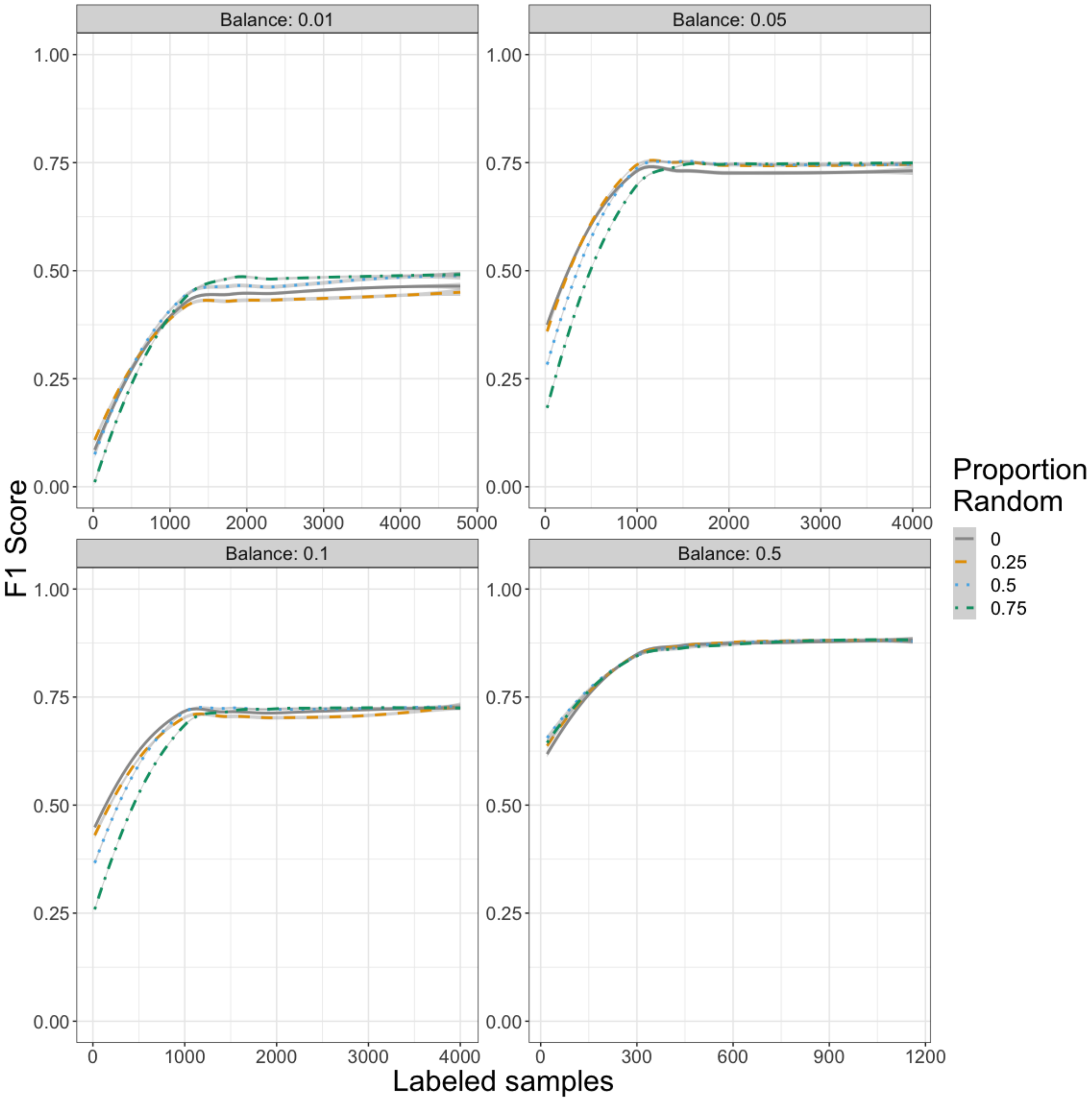

Figure 5: Performance on the held out data of active learning with margin querying strategy and additional randomly selected training data. Each line displays results for a different level of randomness. Results are from the Twitter dataset.

1.5 Pre-processing Choices

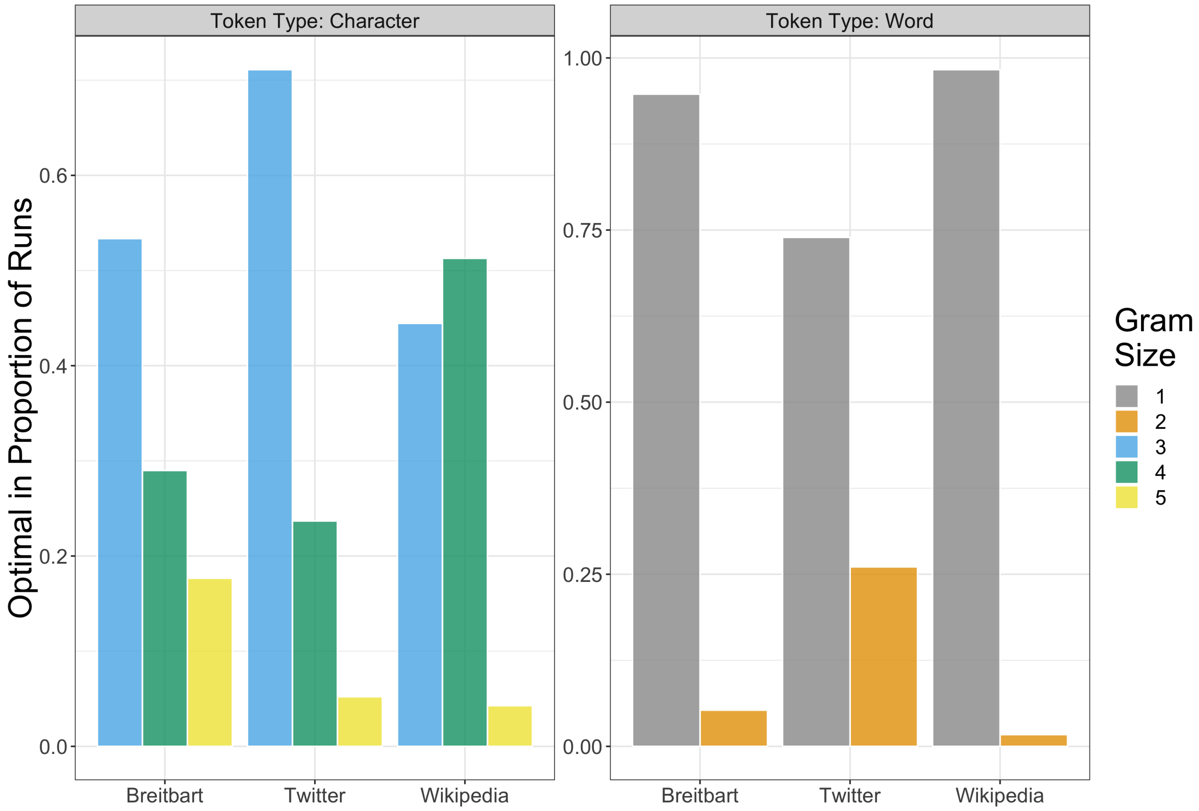

Figure 6: Distribution of n-gram sizes as chosen by the cross validation procedure across text corpora.

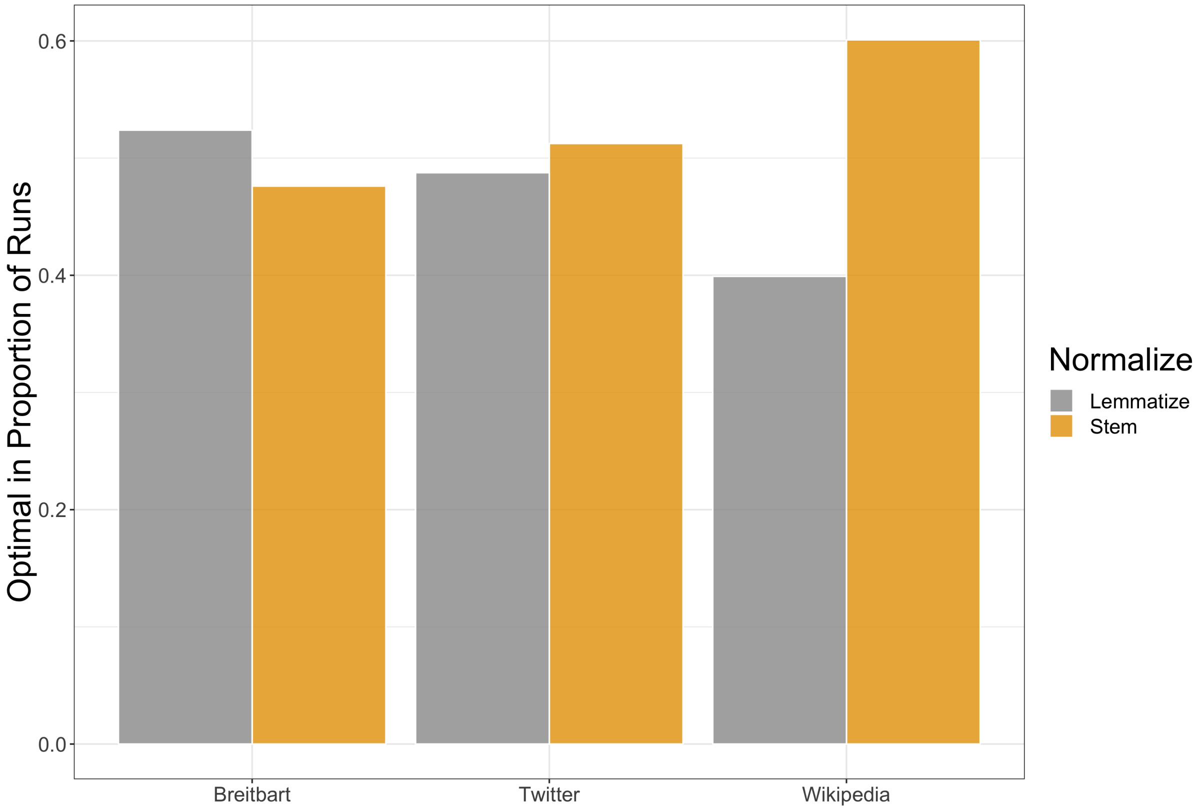

Figure 7: Distribution of token normalization technique as chosen by the cross validation procedure across text corpora.

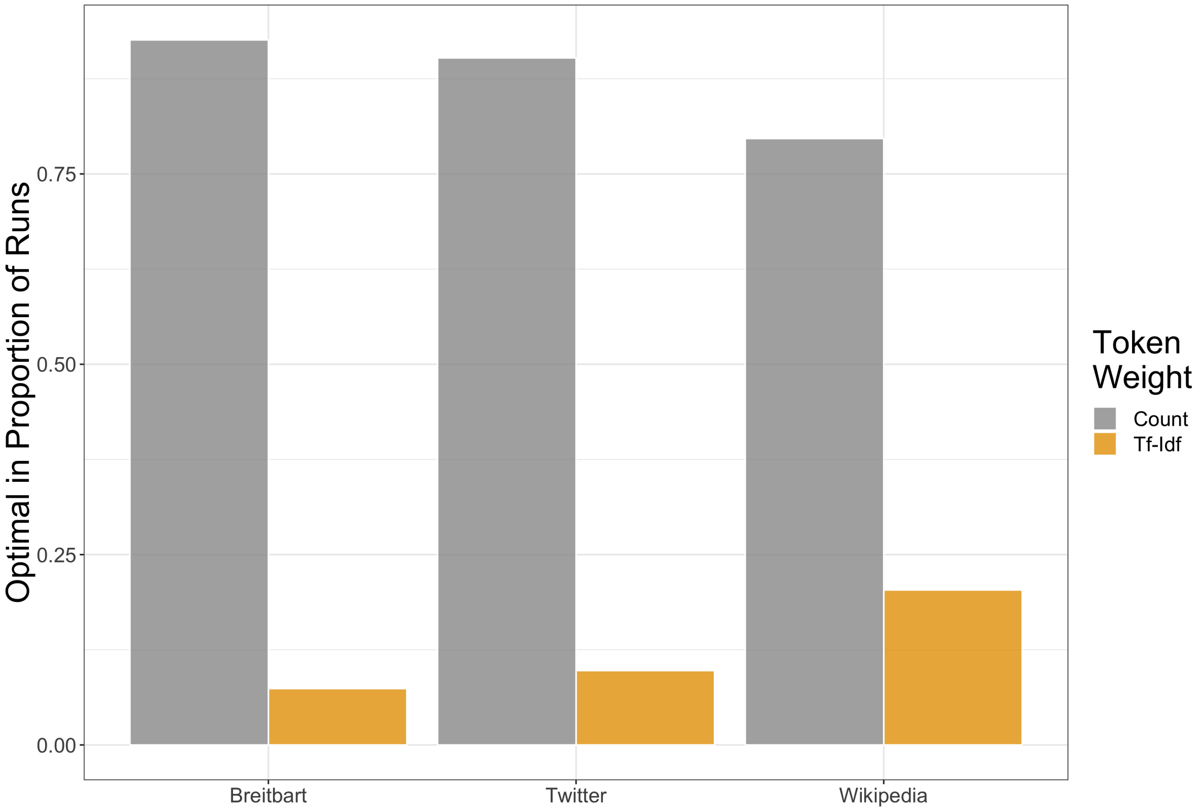

Figure 8: Distribution of token weight technique as chosen by the cross validation procedure across text corpora.

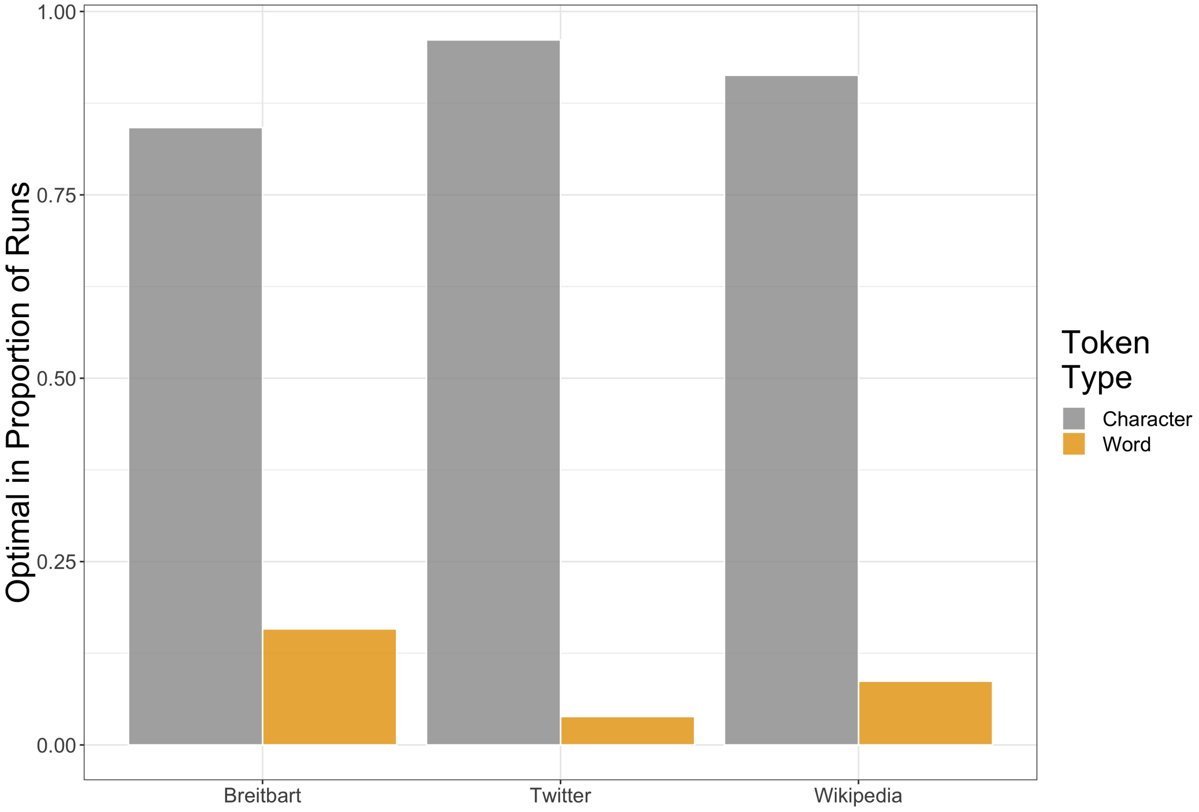

Figure 9: Distribution of token type technique as chosen by the cross validation procedure across text corpora.

1.6 Algorithms

1.6.1 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.

1.6.2 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: $\mathcal{C} = \theta^{(1)}, ..., \theta^{(C)}$[1]
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.

---

[1] 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.

1.6.3 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.