A Sample Paragraph and Article Data

As noted in-text, the datasets used to fit the various feature extraction approaches examined in-text consist of subdivided constitutional documents. In particular, for the LDA, STM, TF-IDF, and LSI models used in-text, we fit models using *constitution paragraphs*, and for the word2vec models we use *constitution sentences*. In this Appendix, we provide examples of these subdivided documents.

To generate the paragraph dataset, we rely on hand-cleaned constitutional texts provided by Constitute. In the Constitute dataset, all constitutions are first subdivided according to their internal organizational structure (e.g. Articles and Sections in the U.S. Constitution). The lowest-level organizational units in each document are then subdivided into paragraphs (using line breaks as the division point). For word2vec, we further subdivide each paragraph into sentences (using the pretrained Punkt sentence tokenizer contained in NLTK (http://www.nltk.org/).) Examples of both the sentence and paragraph subdivision schemes are given in Table 1.

As suggested by Table 1, paragraph divisions frequently correspond with organizational subheaders contained in each constitution (e.g. Articles and Sections in the U.S. Constitution). However, this correspondence is not universal. Depending on writing styles, subheaders can contain a single sentence, a single paragraph, or many separate paragraphs. For simplicity, we use paragraphs to define our input documents when training LDA/STM/TF-IDF/LSI, but other subdivision approaches are certainly plausible.

Table 1: Example constitution dataset.

Paragraph	Text
preamble	We the People of the United States, in Order to form a more perfect Union, establish Justice, insure domestic Tranquility, provide for the common defense, promote the general Welfare, and secure the Blessings of Liberty to ourselves and our Posterity, do ordain and establish this Constitution for the United States of America.
1.1.1	All legislative Powers herein granted shall be vested in a Congress of the United States, which shall consist of a Senate and House of Representatives.
1.2.1	The House of Representatives shall be composed of Members chosen every second Year by the People of the several States, and the Electors in each State shall have the Qualifications requisite for Electors of the most numerous Branch of the State Legislature.
1.2.2	No Person shall be a Representative who shall not have attained to the Age of twenty five Years, and been seven Years a Citizen of the United States, and who shall not, when elected, be an Inhabitant of that State in which he shall be chosen.
1.2.3	Representatives and direct Taxes shall be apportioned among the several States which may be included within this Union, according to their respective Numbers, which shall be determined by adding to the whole Number of free Persons, including those bound to Service for a Term of Years, and excluding Indians not taxed, three fifths of all other Persons. The actual Enumeration shall be made within three Years after the first Meeting of the Congress of the United States, and within every subsequent Term of ten Years, in such Manner as they shall by Law direct. The Number of Representatives shall not exceed one for every thirty Thousand, but each State shall have at Least one Representative; and until such enumeration shall be made, the State of New Hampshire shall be entitled to choose three, Massachusetts eight, Rhode-Island and Providence Plantations one, Connecticut five, New-York six, New Jersey four, Pennsylvania eight, Delaware one, Maryland six, Virginia ten, North Carolina five, South Carolina five, and Georgia three.

Example constitution subdivision, showing the first five paragraphs of the U.S. Constitution (as defined by Constitute). "1.2.1" refers to Article 1, Section 2, Paragraph 1. Changes in highlighting color denote sentence breakpoints.

B Similarity Estimation Robustness Testing

B.1 Isotonic Regression

Based on suggestions from anonymous reviewers, we experimented with an alternate conceptualization and similarity estimation approach, which we document in this section. As described in §2.2, our goal in this paper is to learn a function $g(\mathbf{z}_i, \mathbf{z}_j) \approx \mathbf{Y}_{ij}$, where $\mathbf{Z}$ a matrix of features generated using the raw texts of $\mathcal{D}$ and $\mathbf{Y}$ the matrix of criterion pairwise similarity values. In the text of our paper, we find that the best-performing $g(.)$ function is a random forest with $n = 75$ documents used as training observations. However, as noted by reviewers, this approach forces our model to recover costly learning rules across a large number of auxiliary parameters, potentially degrading estimator performance.

To address this problem, one anonymous reviewer suggested an alternative problem framework. In particular, the reviewer proposed that we instead attempt to learn a univariate function $h(g(\mathbf{z}_i, \mathbf{z}_j)) \approx \mathbf{Y}_{ij}$, with $g(.)$ a simple unweighted distance function (e.g. cosine or Hellinger, as used in-text) and $h(.)$ a monotonic function (e.g. isotonic regression) relating unsupervised similarity values to the human-generated criterion. Under this approach, rather than attempting to use the textual features $\mathbf{Z}$ as inputs to a supervised learner, we instead collapse the features for each dyad into a single value, which we re-scale to approximate our criterion similarity values. This approach substantially reduces model dimensionality while (hopefully) incurring limited performance penalties.

We tested a version of this approach using scikit-learn (http://scikit-learn.org/stable/modules/isoton isotonic regression implementation (with test set values predicted by linear interpolation). To generate inputs for the model, we use a simple three-step procedure. First, for each feature set (word2vec, LDA, STM), dimensionality value, and training set size, we generate a set of pairwise similarity values $g(\mathbf{z}_i, \mathbf{z}_j)$, with $g(.)$ selected to fit

the constraints of the $\mathbf{Z}$ matrix (Hellinger for LDA and STM, cosine for word2vec). Second, we separate the dataset into train/test splits using the same block bootstrap procedure described in text. Specifically, for each split we select a set of countries and use all unique dyads within that set as our training set, and use all others as our test set. Finally, we train an isotonic regression model using our training set, and assess performance on the test set. We repeat this process 100 times for each training set size, feature extraction approach, and dimensionality value, and report means and ± 2 sample standard deviation ranges for each combination.

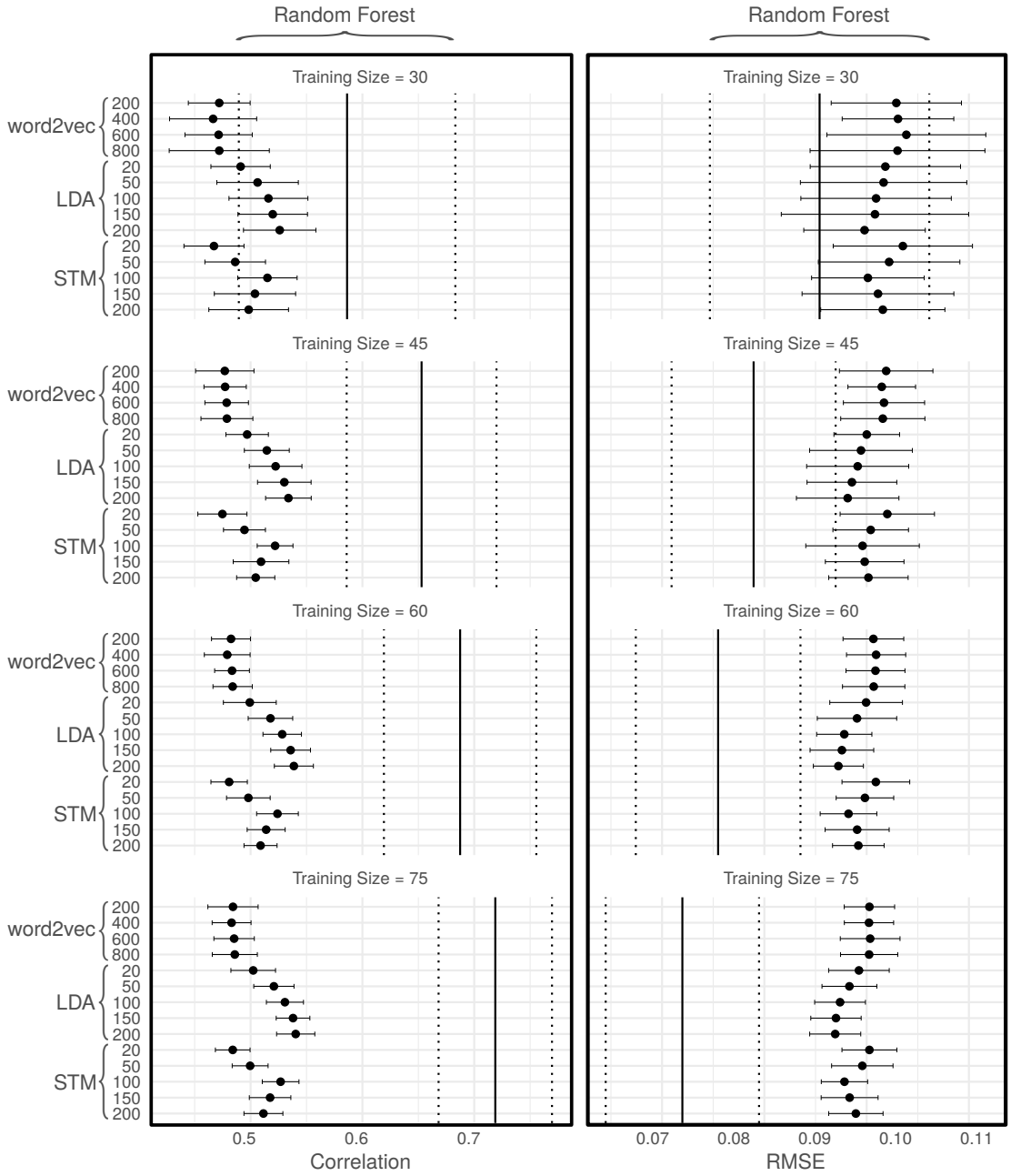
Unfortunately, as shown in Figure 1, this approach substantially underperforms the random forest approach presented in-text. Unsurprisingly, due to the smaller number of estimated parameters the values generated using isotonic regression generally display substantially less variance across training sets than the random forest baseline, but the performance gap between the two approaches is large enough to overwhelm these gains. Interestingly, in contrast to the models presented in-text, performance changes relatively little as sample size increases; in particular, though prediction variability decreases substantially as sample size increases, mean performance across all sample sizes is nearly constant.

To probe this latter finding, we experimented with an approach in which we allowed our isotonic regression learner to view *all* observations during training, and compared in-sample predictions to their human-generated counterparts. The results of this comparison are given in Figure 2, with the model fit from LDA_{200} (the best-performing feature set) visualized in Figure 3. As suggested by our previous results, the isotonic regression approach underperforms our existing random forest setup with $n = 75$ training documents even when predicting in-sample similarities with all observations used for training.

In our view, this gap highlights the challenges introduced by the lack of correspondence between $\mathbf{X}$ and $\mathbf{Z}$. Because the constraints, information content, and mapping

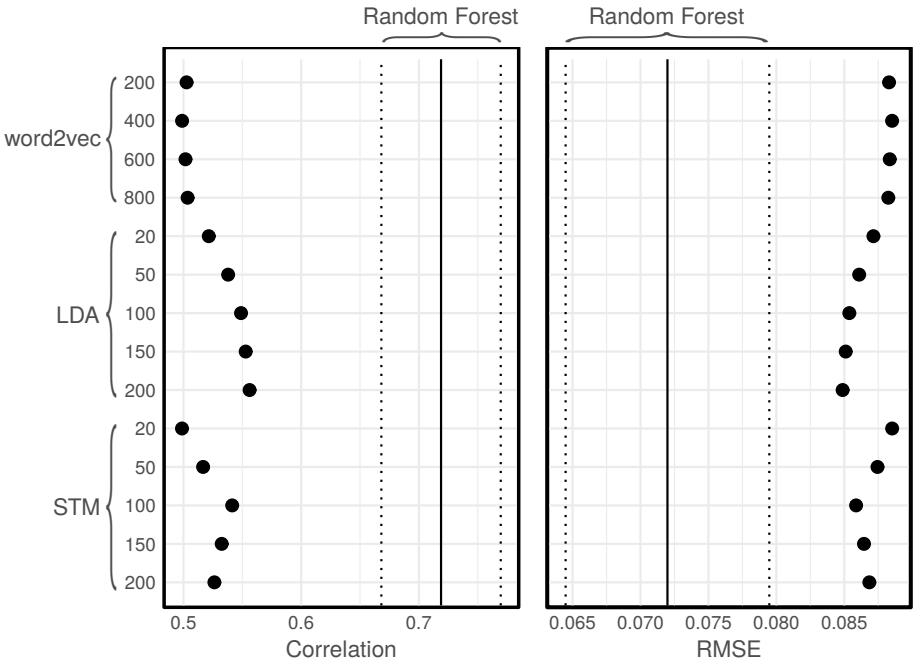
between these two sets of feature vectors potentially differs so substantially, collapsing the underlying features to a single dimension before training a learner appears to discard important information. At least for this application, then, more flexible approaches such as random forests appear to offer stronger performance.

Figure 1: Out-of-sample RMSE and correlation between predicted similarities generated through isotonic regression and human-generated values



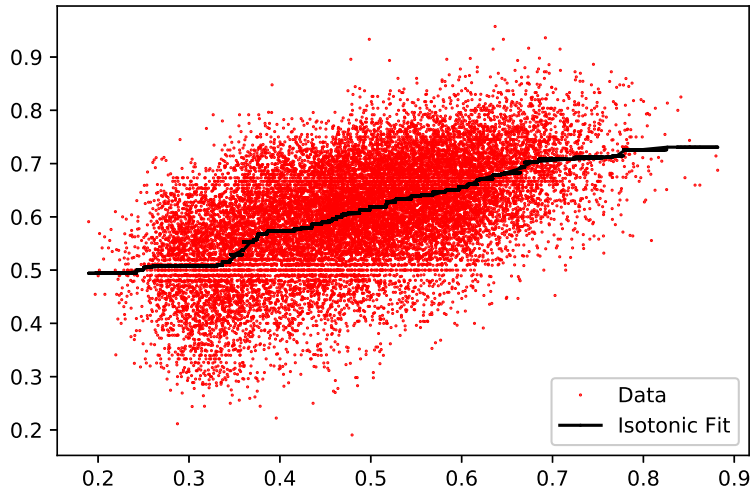
Solid horizontal lines represent ±2 sample standard deviations, estimated from 100 train/test splits. Solid and dashed vertical lines give mean and ±2 sample standard deviations for out-of-sample correlation/RMSE generated using concatenated LDA_{100} features at each training size.

Figure 2: In-sample RMSE and correlation between predicted similarities generated through isotonic regression and human-generated values



Dots represent in-sample correlation and RMSE values for similarities estimated using isotonic regression. Models were trained and evaluated on all observations, so no train/test splits were conducted. Solid and dashed vertical lines give mean and ± 2 sample standard deviations for out-of-sample correlation/RMSE generated using concatenated LDA_{100} features $n = 75$ training documents.

Figure 3: Isotonic model fit and machine/human-generated similarities, generated using LDA_{200} features.



Initial machine-generated similarities plotted against human-generated values, with isotonic model fit overlaid. Model was trained and evaluated on all observations, so no train/test splits were conducted.

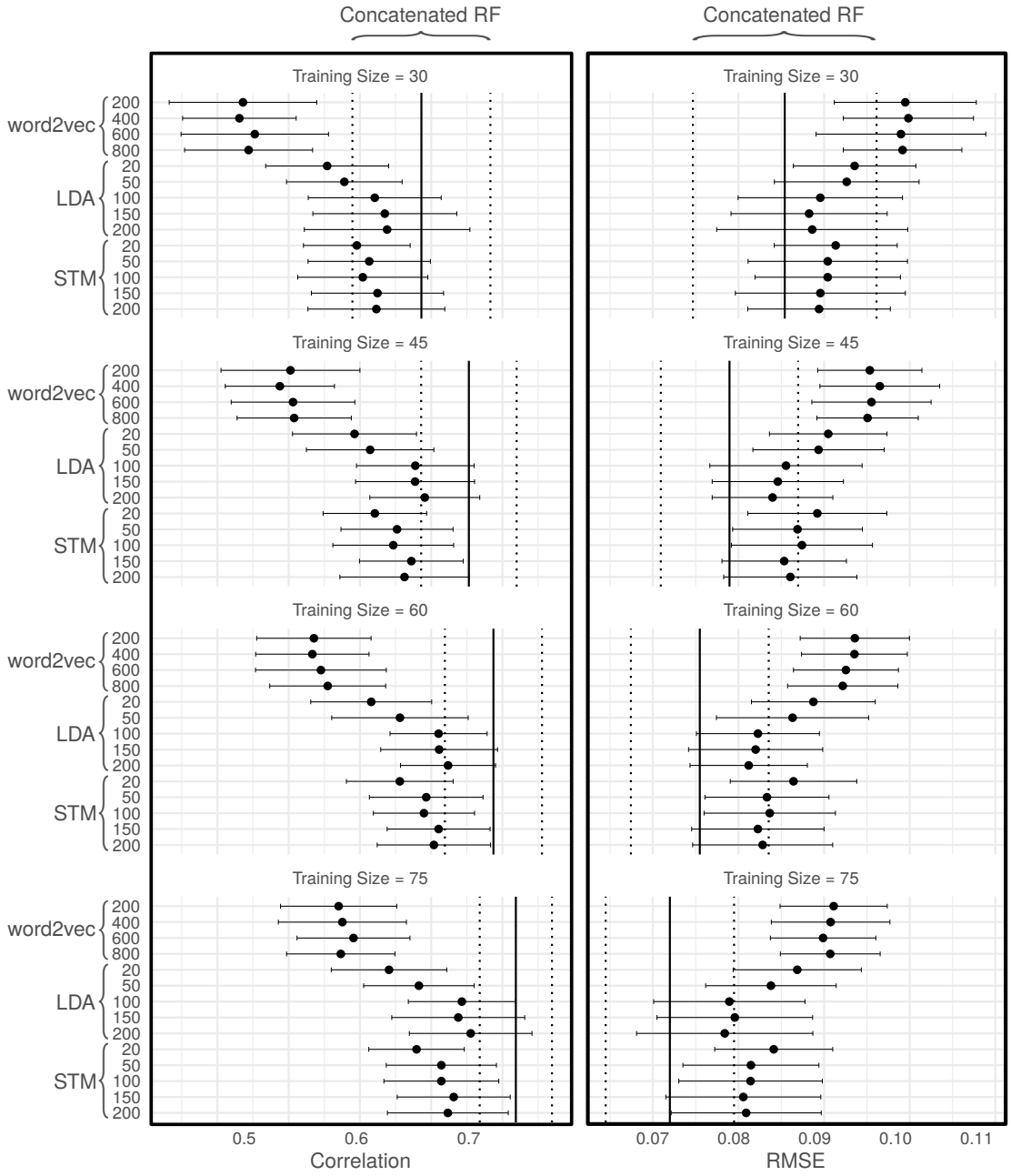
B.2 Differenced Feature Vectors

In our existing supervised learning setup, we use the concatenated feature vectors for each dyad as the inputs to a random forest model. As noted by an anonymous reviewer, this approach doubles the number of features under consideration and forces the model to learn potentially expensive feature combination rules (for example, to introduce splits when the difference between features in each half of the dyad is sufficiently large). Reducing the dimensionality of the input vector in some fashion, this reviewer suggests (e.g. by combining the feature vectors of each dyad), might substantially improve our results.

To experiment with this approach, we re-estimated the models presented previously. All parameter settings and modeling choices were identical to those described in-text. However, instead of training our random forest models using concatenated feature vectors for each dyad, we substituted element-wise absolute difference between each vector, $|\mathbf{z}_i - \mathbf{z}_j|$. Other combination methods (e.g. elementwise squared difference) also represent plausible candidates for comparison, but this approach seemed to offer a reasonable starting point for comparison.

The results of this comparison are given in Figure 4. Unfortunately, substituting differenced feature vectors for concatenated feature vectors appears to offer reduced performance at all training set sizes. Values generated based on word2vec features are particularly hard-hit, but performance in all cases suffers at least slightly compared with the in-text baseline. At least for our corpus, simple combination rules like differencing appear to discard an unnecessarily large amount of information, which the random forest models we employ are able to leverage.

Figure 4: Out-of-sample RMSE between predicted similarities generated using differenced feature vectors and in-text concatenated feature approach.



Solid horizontal lines represent ±2 sample standard deviations, estimated from 100 train/test splits. Solid and dashed vertical lines give mean and ±2 sample standard deviations for out-of-sample correlation/RMSE generated using concatenated LDA_{100} features at each training size.

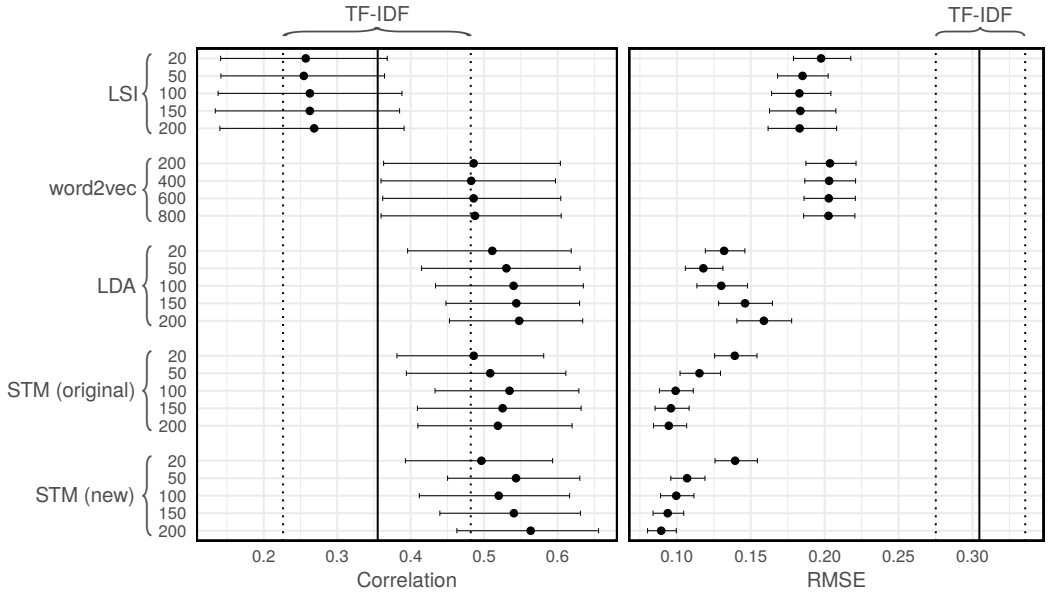
B.3 Additional STM Covariates

As noted in-text, the STM models we examine use a relatively sparse feature set. In particular, our covariate set for our STM models included a spline on the year of the constitution's enactment and a dummy variable indicating the constitution from which a given paragraph was drawn. In this section, we estimate an additional set of models which include region dummies as an additional set of covariates.

The results of this approach for our unsupervised and supervised experiments are given in Figures 5 and 6, respectively, with results from our main in-text figures included as a baseline. In the unsupervised comparison, similarity values estimated using STM features with the more expansive covariate set outperform our original STM specification slightly by both RMSE and correlation. However, the differences between these approaches are not significant at any dimensionality parameter value. In the supervised setting, these differences vanish entirely, with performance results based on the updated STM features matching the results presented previously.

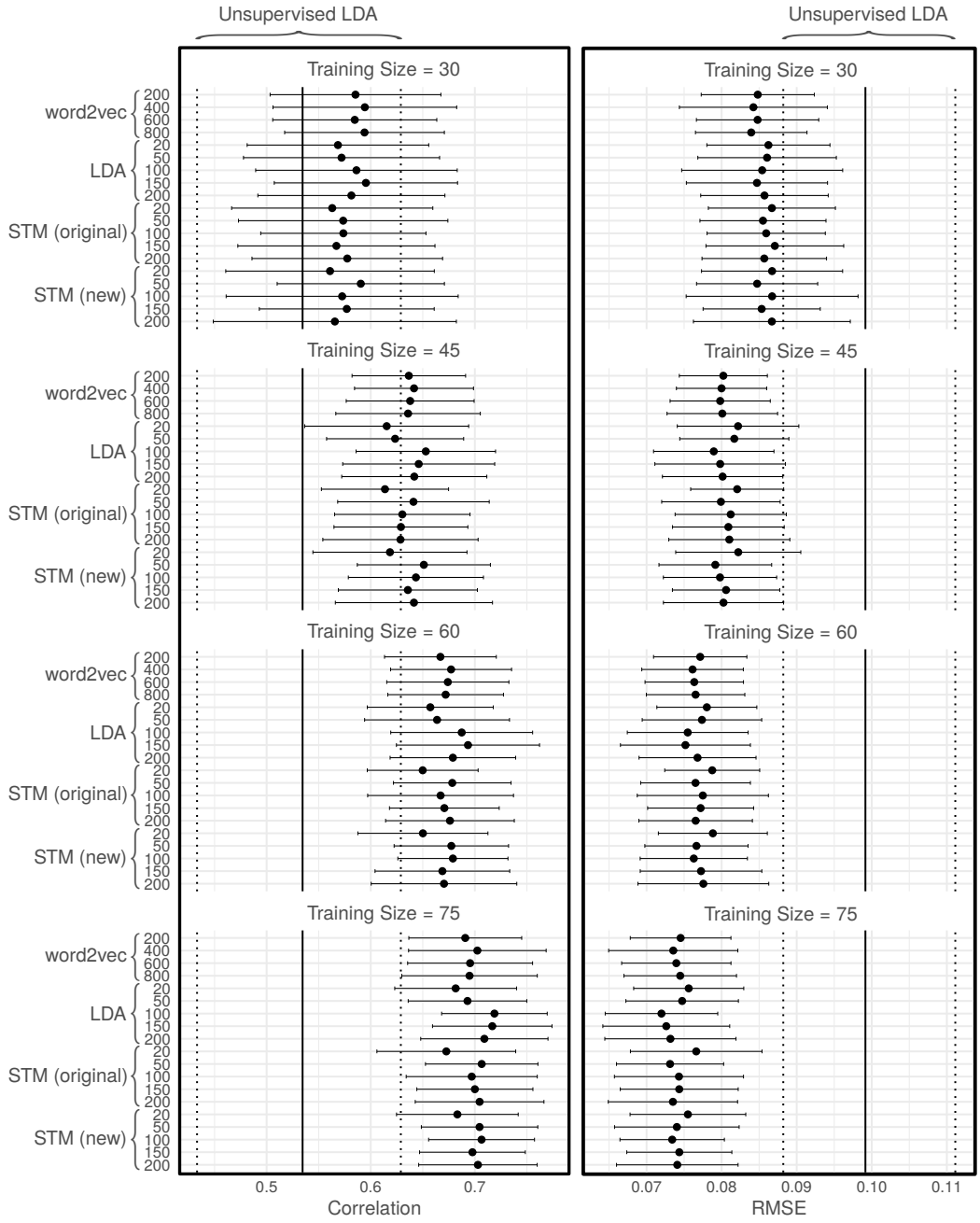
As mentioned in-text, our goal in this paper is to simulate a research scenario in which the researcher does not possess a particularly rich covariate set with which to train their feature extraction models. More expansive feature sets might improve performance, though we note that even our narrower STM feature sets presented in-text include some 193 dummy variables in addition to the year-of-enactment spline. However, because of our applied focus, we believe our choice to focus on a fairly narrow set of covariates is reasonable.

Figure 5: Correlations between machine- and human-generated similarity values



Results from in-text figures, with results from STM models estimated using region dummies appended as "STM (new)".

Figure 6: Out-of-sample RMSE between predicted similarities generated using differenced feature vectors and in-text concatenated feature approach.



In-text supervised results, with performance estimates drawn from STM models that include region dummies appended at each training set size as "STM (new)".

C QAP Supplementary Information and Robustness Testing

C.1 Coefficient Table for Human-Generated Model

Table 2: Linear model coefficient estimates produced using human-generated similarity values

	Estimate	P-value
Sqrt_Yeardiff	-0.012	≤ 0.001
East Asia	-0.037	0.122
Eastern Europe	0.122	≤ 0.001
Western Europe/North America	0.007	0.6660
Latin America	0.031	0.069
Middle East/North Africa	0.022	0.333
Oceania	-0.057	0.047
South Asia	0.128	0.002
Sub-Saharan Africa	0.006	0.667
Constant	0.657	≤ 0.001
Observations	18528	
Adjusted R^2	0.171	

P-values generated using a QAP null distribution.

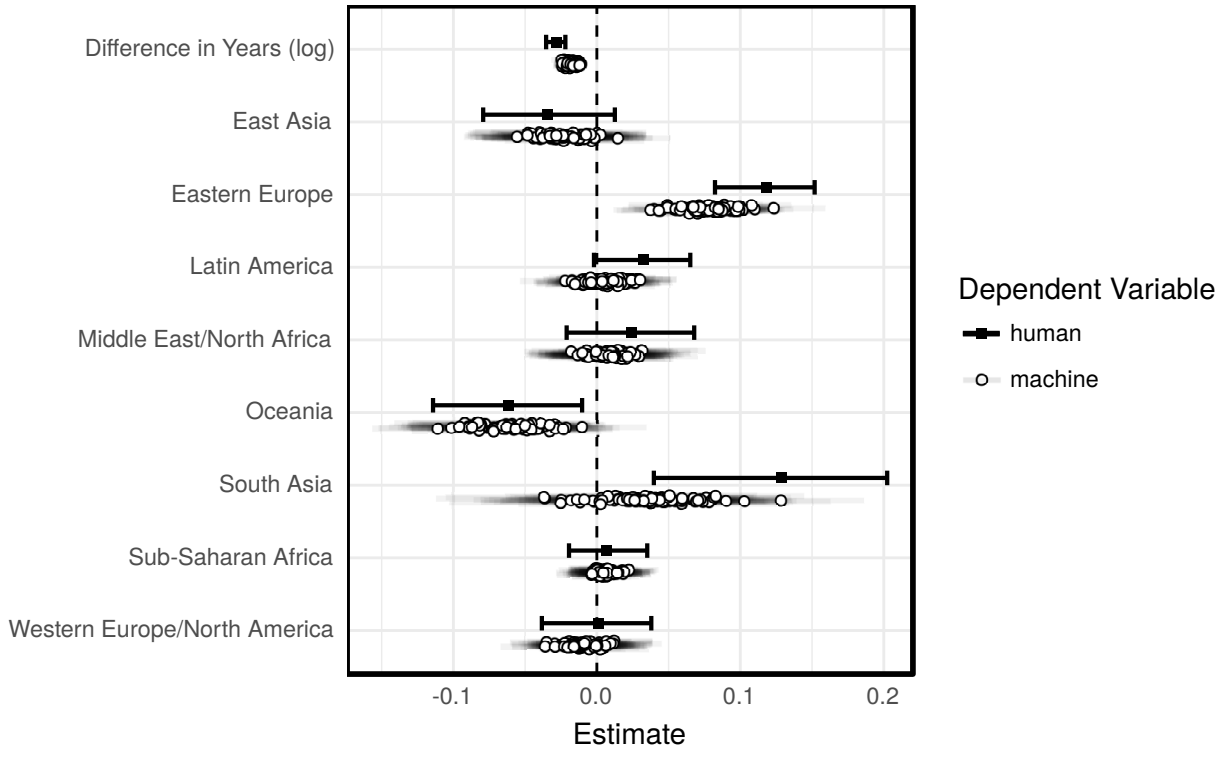
C.2 Alternate Time-Gap Variable Specification for Similarity Regressions

Based on suggestions from an anonymous reviewer, we include two alternate specifications for the linear model coefficients presented in-text. In particular, in this section we test two alternate measures for our time-difference variable. For reference, our original operationalization of this variable is $\sqrt{|t_1 - t_2|}$, denoting the years of enactment for each constitution in a given dyad as as t_1 and t_2.

Our first alternate approach - given in Figure 7 - replaces our original time-gap variable with a logarithmic specification, operationalized as $ln(|t_1 - t_2| + 1)$. We add a constant inside the logarithm in order to ensure that this variable is defined in cases where the two constitutions in the dyad were enacted in the same year. As shown in Figure 7, this operationalization produces essentially identical results to those given in-text. Just as in our main results, the year-gap coefficient is negative and significant, and all regional dummy variables return the same substantive conclusions. Performance results are also essentially identical, with some 87% of coefficients returning the same substantive conclusions as their human-generated counterparts (across 100 train/test splits).

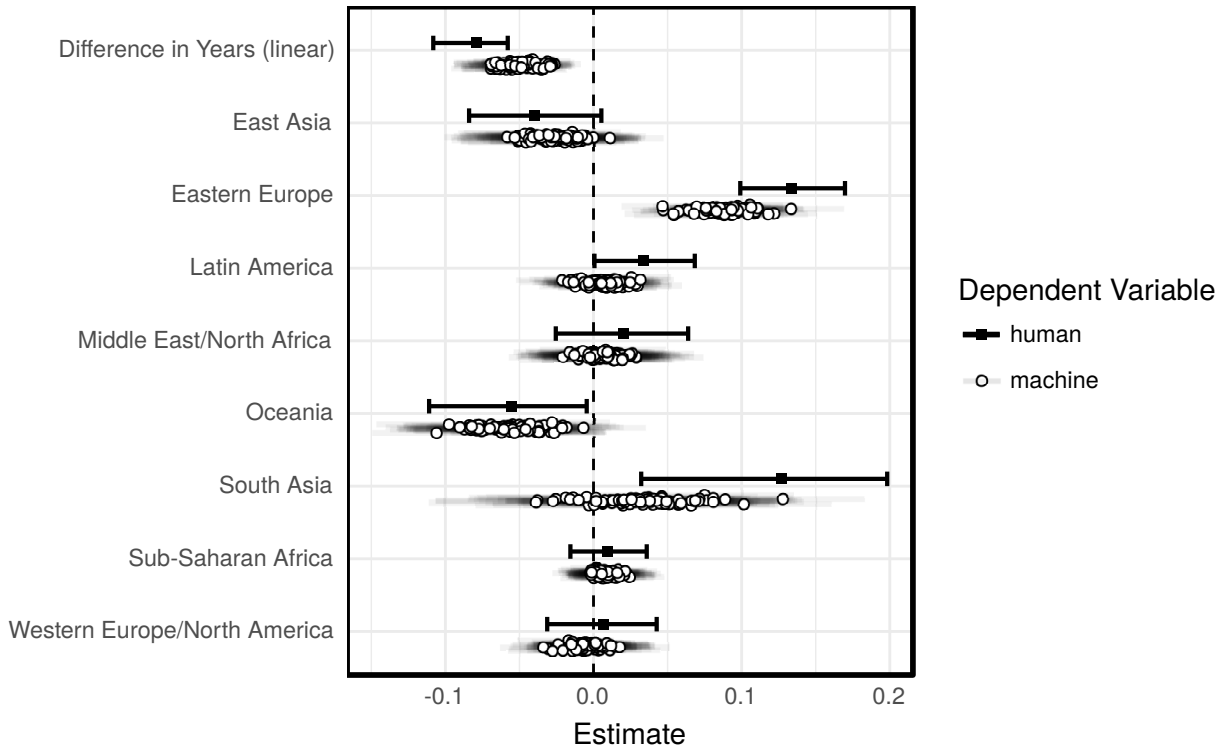
Our second alternate approach - given in Figure 8 - is identical to the first, but with the time gap variable operationalized as $\frac{1}{100}|t_1 - t_2|$. This specification follows that used by Cheibub et al. (2014), who use it to study similarity of constitutions with respect to their provisions regarding executive-legislative relations. In that study, Cheibub et al. (2014) report very similar findings to ours, with a gap of 100 years between date of enactment predicted to decrease constitutional similarity by approximately 0.05 to 0.15 points depending on era and model specification. As before, using this approach some 86% of coefficients returned the same conclusion as their human counterparts (across 100 train/test splits).

Figure 7: Linear model coefficient estimates generated using human and machine-produced similarity values



Critical values, confidence intervals, and p-values for each coefficient produced using a QAP null hypothesis as described in Dekker et al. (2003). Coefficient estimates and confidence intervals drawn from models estimated using machine-generated values are overlaid and jittered, and confidence intervals are faded. Intercept omitted for readability.

Figure 8: Linear model coefficient estimates generated using human and machine-produced similarity values



Critical values, confidence intervals, and p-values for each coefficient produced using a QAP null hypothesis as described in Dekker et al. (2003), and confidence intervals are faded. Intercept omitted for readability.

References

Cheibub, J. A., Elkins, Z., and Ginsburg, T. (2014). Beyond presidentialism and parliamentarism. *British Journal of Political Science*, 44(3):515–544.

Dekker, D., Krackhardt, D., and Snijders, T. (2003). Multicollinearity robust qap for multiple regression. In *1st annual conference of the North American Association for Computational Social and Organizational Science*, pages 22–25. NAACSOS.