Supplemental Information

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. 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 Alternative Modeling Approaches

B.1.1 Isotonic Regression

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, this approach forces our model to recover complex learning rules across a large number of auxiliary parameters, potentially degrading estimator performance.

To address this problem, we experimented with an alternate learning strategy, in which we 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's 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.

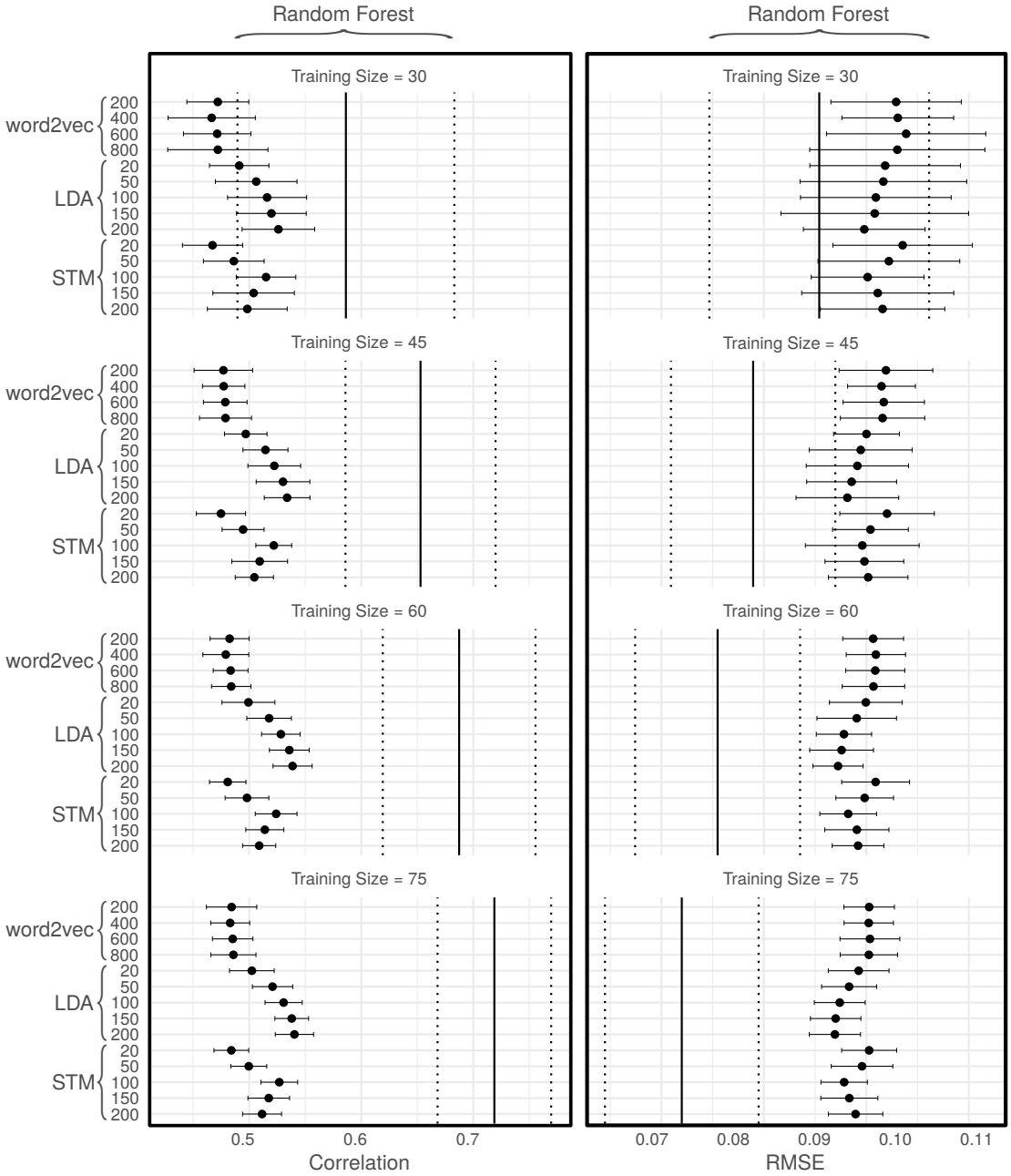
As shown in Figure 1, this approach substantially underperforms the random forest approach presented in-text. Unsurprisingly, due to the smaller number of estimated parameters the values generated using isotonic regression generally display substantially less variance across training sets than the random 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 differ so substantially, collapsing the underlying features to a single dimension before training a learner appears

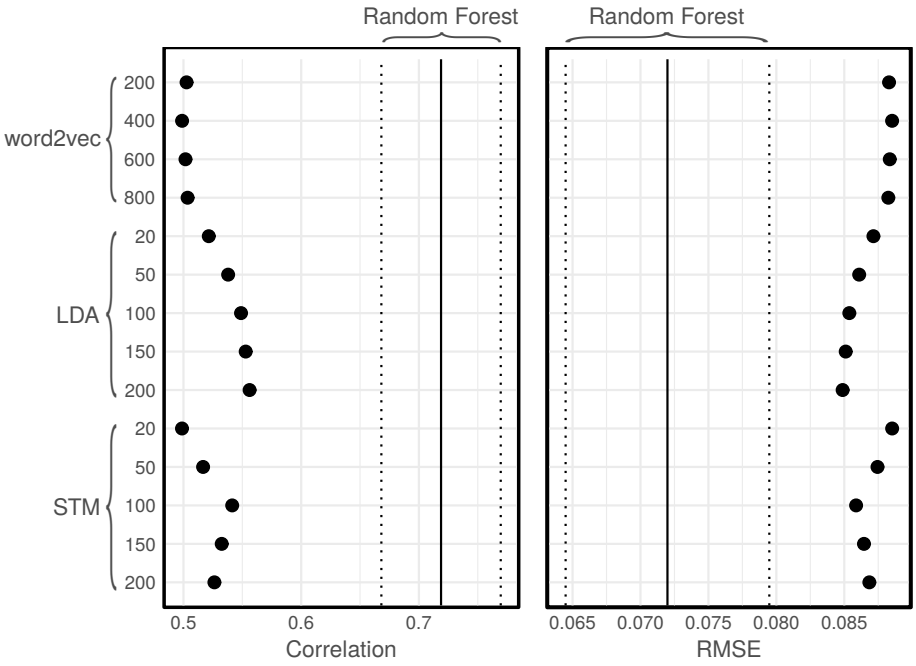
to discard important information. At least for this application, then, more flexible approaches such as random forests appear to offer stronger performance.

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



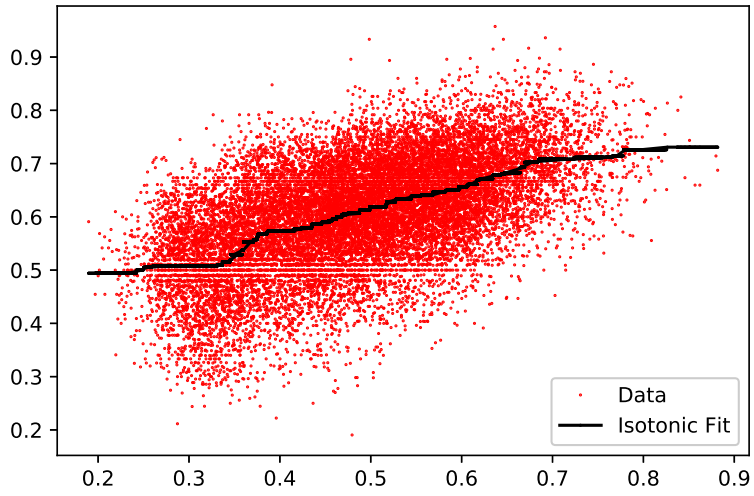
Solid horizontal lines represent ±2 sample standard deviations, estimated from 100 train/test splits. Solid and dashed vertical lines give mean and ±2 sample standard deviations for out-of-sample correlation/RMSE generated using concatenated LDA_{100} features at each training size.

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.1.2 Metric Learning

In addition to the random forest specification we present in the body of our paper, we also experimented with a more customized similarity learning algorithm. As described in-text, for all of our feature extraction approaches we began by generating an unsupervised baseline measure, which we produced by calculating a simple unweighted distance measure tailored to the constraints of each feature extraction approach. For LDA and STM, since the features produced by these methods are constrained to sum to one, we used an inverse discretized Hellinger distance as our baseline, defined as:

$$\begin{aligned} g_H(\mathbf{z}_i, \mathbf{z}_j) &= 1 - \frac{1}{\sqrt{2}}\sqrt{\sum_{k=1}^{K}(\sqrt{\mathbf{z}_{ik}} - \sqrt{\mathbf{z}_{jk}})^2} \\ &= 1 - \frac{1}{\sqrt{2}}\sqrt{(\sqrt{\mathbf{z}_i} - \sqrt{\mathbf{z}_i})^T(\sqrt{\mathbf{z}_i} - \sqrt{\mathbf{z}_i})} \end{aligned}$$

With $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, ...\mathbf{z}_n]^T$ an $n \times k$ feature matrix generated using an unsupervised feature extraction approach. Since this function is equivalent to a (rescaled) Euclidean distance between the square root of the i^{th} and j^{th} feature vector, we can define a weighted version of this function as:

$$g_{WH}(\mathbf{z}_i, \mathbf{z}_j, \mathbf{W}) = 1 - \frac{1}{\sqrt{2}}\sqrt{(\sqrt{\mathbf{z}_i} - \sqrt{\mathbf{z}_j})^T\mathbf{W}\mathbf{W}'(\sqrt{\mathbf{z}_i} - \sqrt{\mathbf{z}_j})}$$

Where $\mathbf{W}$ is a $k \times p$ matrix with with real, strictly positive diagonal elements and $p \leq k$. Since $\mathbf{W}\mathbf{W}'$ is guaranteed to be positive-definite, this expression represents a special case of a Mahalanobis distance, offering it a natural interpretation. The value p is a researcher-specified parameter that controls the rank of $\mathbf{W}\mathbf{W}'$, allowing researchers to select the dimensionality of their selected weights matrix.

Since our goal in this paper is to use $\mathbf{Z}$ to recover a set of human-generated similarity values $\mathbf{Y}$, this expression suggests a natural supervised learning algorithm.

In particular, our goal is to learn a $\mathbf{W}$ matrix satisfying:

$$\hat{\mathbf{W}} = argmin_W \sum_{i \in S} \sum_{j \in S \forall i<j} (\mathbf{Y}_{ij} - g_{WH}(\mathbf{z}_i, \mathbf{z}_j, \mathbf{W}))^2$$

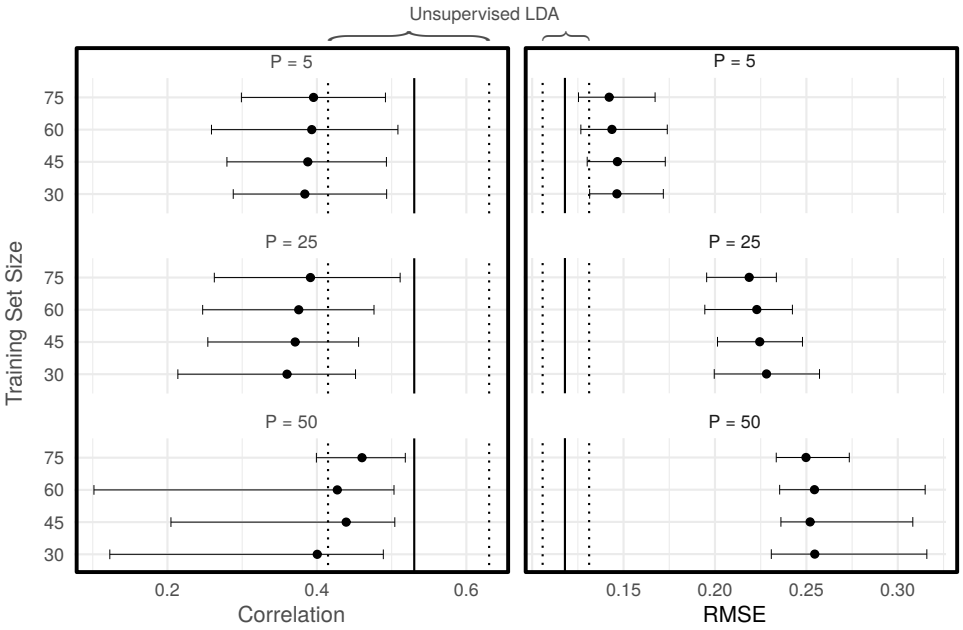
Where S represents the set of observation indices selected for inclusion in the training set. In words, our task is therefore to select a weights matrix $\mathbf{W}$ that minimizes the squared error loss between our human-generated gold standard similarity values and the weighted Hellinger distance between our unsupervised feature vectors. For the purposes of this section, we fit $\mathbf{W}$ using an L-BFGS algorithm with a numerical gradient approximation.[1]

To investigate the performance of this approach, we focused on the LDA_{50} features presented in-text. We selected this set of features for practical reasons. From a performance standpoint, in the supervised context all LDA and STM-based feature extraction approaches we examine performed essentially equivalently out-of-sample when used with the random forest specification we present in-text. As a result, if the method we examine in this section is competitive with our random forest specifications it should perform comparably when applied to any feature vector size. Since the algorithm we describe in this section involves estimating up to k^2 parameters, for testing purposes we therefore chose to focus on a feature extraction approach that produced a relatively small number of features.

The results of this comparison are given in Figure 5. Unfortunately, the metric learning approach we present in this section does not appear to perform well in our application. At all training set sizes and dimensionality values, this method underperforms the unsupervised baseline we present in-text, as measured by either correlation or RMSE. As measured by RMSE, performance is worst at the largest dimensionality values, suggesting that the optimizer we use may be failing to adequately explore

[1] As implemented in scipy. All parameter values not mentioned in-text left at their default values.

Figure 4: Out-of-sample correlation and RMSE for similarity values generated using a metric learning approach and LDA_{50} features.



Dots and horizontal lines indicate mean out-of-sample correlation/RMSE and 2.5%/97.5% percentiles for comparisons between CCP targets and metric learning equivalents, based on 100 train/test splits. Solid and dashed vertical lines represents the baseline unsupervised LDA_{50} scores, as described in-text.

the parameter space when a large number of parameters are present. However, at all sample sizes, the method performs poorly.

As in our isotonic regression investigation in the previous appendix, we attribute this performance gap to the lack of correspondence between $\mathbf{X}$ and $\mathbf{Z}$. Because the relationship between human and machine-generated similarity values is (potentially) highly non-linear, a more formalized metric learning approach does not appear to perform well in this setting. Optimizing this approach is a potential direction for future research; however, since the more straightforward random forest-based specification we present in-text seems to perform well without additional customization, we do not view these efforts as a priority for our current paper.

B.2 Alternative Random Forest Specifications

B.2.1 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. This approach has two potential drawbacks. First, training a supervised learner on a concatenated feature vector doubles the number of features under consideration, and forces the model to learn potentially complex feature combination rules.[2] This extra information likely reduces bias, but may increase variance substantially.

Second, by training the model using raw feature vectors, the concatenated feature engineering approach potentially allows the model to learn classification rules based on the *absolute* position of each document in the feature space, rather the *relative* position of each member of the dyad. Under this learning strategy, the model may learn classification rules based on average pairwise similarity between each member of the dyad and all other documents in the training set, rather than the the actual dyadic distance between the two observation under consideration. This extra information may help increase prediction accuracy; however, if the training set's clustering patterns are not representative of those present in the full dataset, this approach may reduce estimator performance (or, at least, increase dependence on the representativeness of the training set).

To investigate these issues, we re-trained the supervised learners we present in §4.1 using the element-wise absolute difference between feature vectors for each dyad, $|\mathbf{z}_i - \mathbf{z}_j|$.[3] This approach – which forces the random forest learner to unambiguously rely on *relative* rather than *absolute* document positions – allows us to investigate and quantify the impact of these concerns. If the performance differences between this approach and the concatenated approach presented in-text are large, then the two

[2]For example, in a similarity application, the model may need to introduce splits when the difference between features in each half of the dyad is sufficiently large.

[3]All other parameter settings and modeling choices were identical to those described in-text.

challenges we outline in this section may lead readers to be skeptical of the approach we employ.

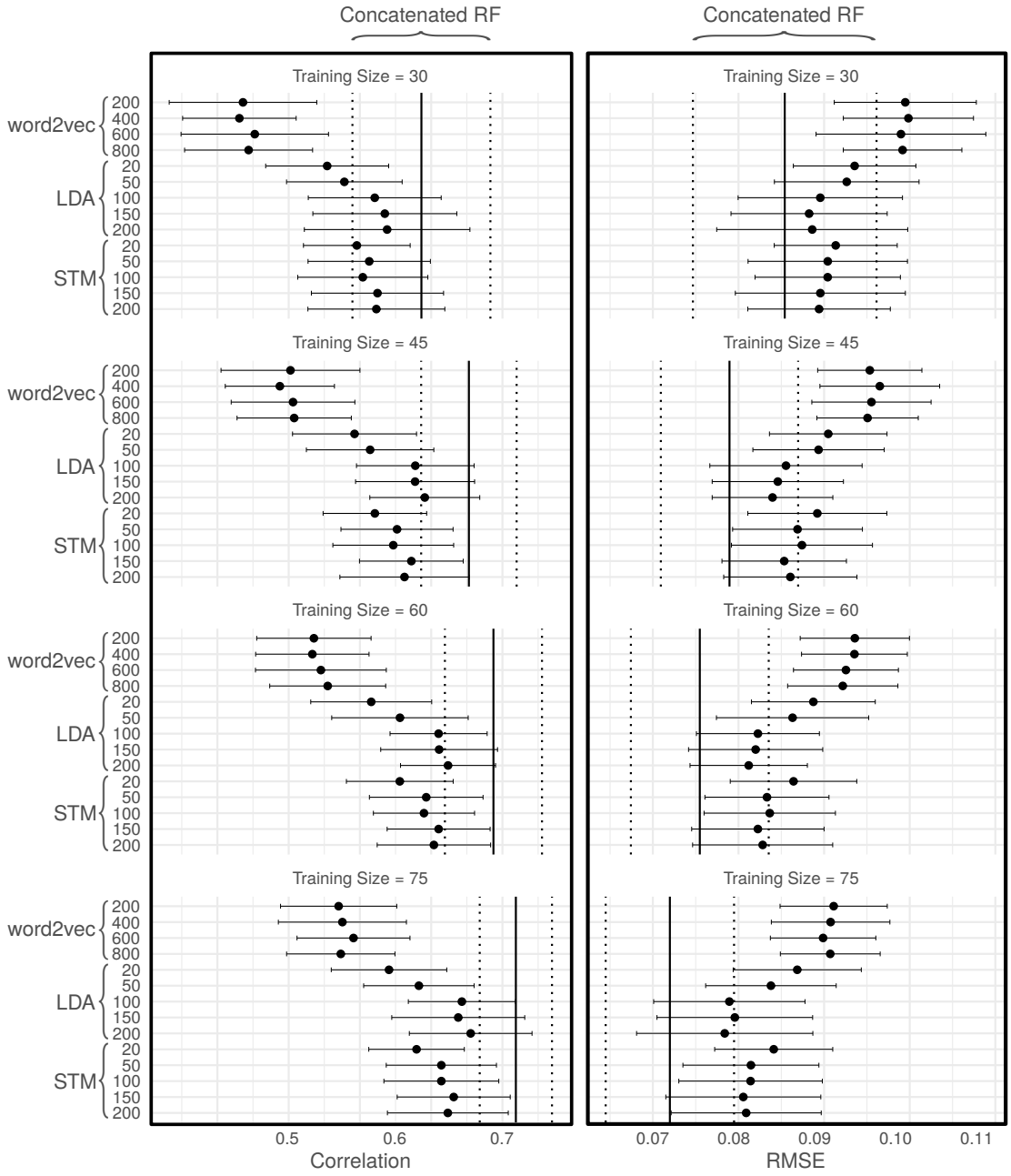
The results of this comparison are given in Figure 5. Compared to similarity values generated using LDA_{100} features (the best performer with the concatenated approach), values generated using the differenced procedures perform somewhat worse at most sample sizes and feature extraction approaches. Performance for values generated based on word2vec features are particularly hard-hit, while performance for those generated based on high-dimensional LDA and STM models are much closer to the concatenated baseline. Restricting the comparison to the best-performing model in each scenario (LDA_{100} for the concatenated approach, and LDA_{200} for the differenced approach), the concatenated feature vector approach exhibits an average out-of-sample RMSE value of 0.072 (compared with 0.078 in the differenced approach) and an out-of-sample correlation of 0.72 (compared with 0.66 in the differenced approach).

We draw two conclusions from these results. First, the concatenated feature approach outperforms the differenced approach, which suggests that the additional information provided by the former approach is indeed improving out-of-sample performance. Notably, we cannot say with certainty what *kind* of additional information contained in the concatenated feature vector is providing this benefit. The clustering possibility we raise above is one possibility, but others are also possible. For example, the simple differencing rule we examine in this section might be overly restrictive, and some other distance function or rule based on threshholding might be more effective in this scenario.

Second - and more importantly - the scale of the performance gap we observe between the concatenated and differenced similarity estimation approaches bounds the scale of the additional information contributed by the former strategy. As we describe above, if the concatenated feature vector approach we employ in-text is

inclined to learn prediction rules based on the absolute position of each observation in the feature space, then the performance of the concatenated approach will be highly dependent on the representativeness of the training set, which is not desirable for applied work. However, as we show in Figure 5, the difference between these two approaches is noticeable but small. Restricting our comparison to the best-performing model in each case (LDA_{100} for the concatenated approach, and LDA_{200} for the differenced approach), discarding the extra information included in the concatenated feature vector increases out-of-sample RMSE by 8.3% and decreases out-of-sample correlation from 0.72 to 0.66. Adopting the most pessimistic possible interpretation, these results suggest that the model's ability to consider the absolute position of each document in the feature space increases out-of-sample performance by less than 10%. Even for the most skeptical readers, then, these results should offer some reassurance regarding the validity of our results.

Figure 5: Out-of-sample correlation and RMSE 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.

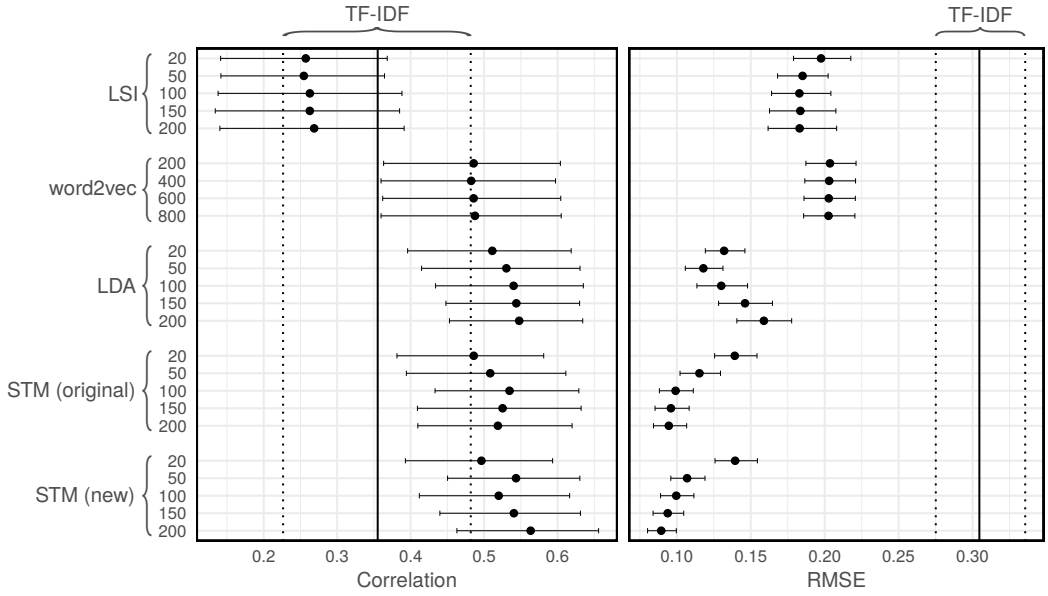
B.2.2 Additional STM Covariates

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 6 and 7, 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.

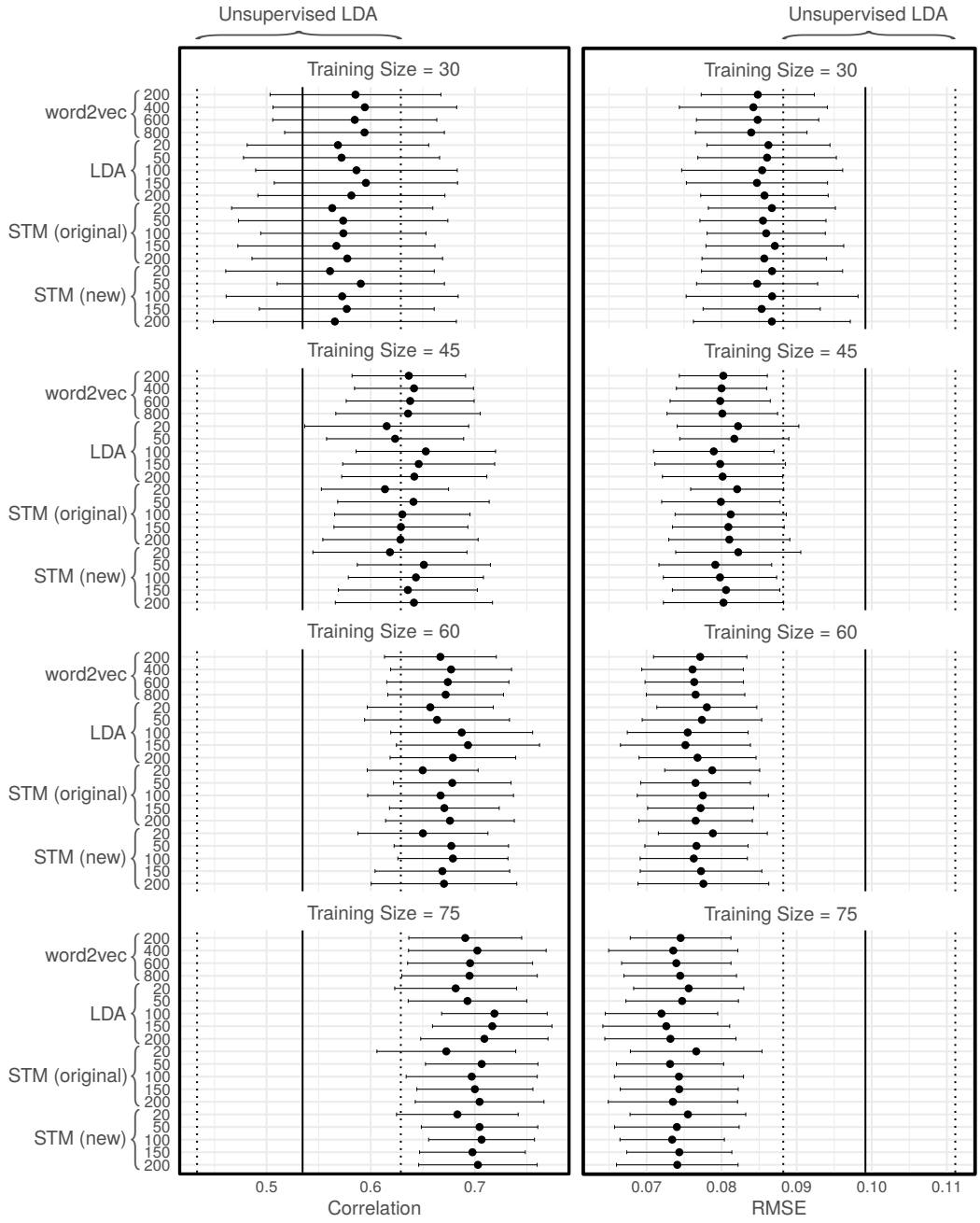
As mentioned in-text, our goal in this paper is to simulate a research scenario in which the researcher does not possess a particularly rich covariate set with which to train their feature extraction models. More expansive feature sets might improve performance, though we note that even our narrower 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 6: 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 7: 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 Out-of-Sample Inferential Performance

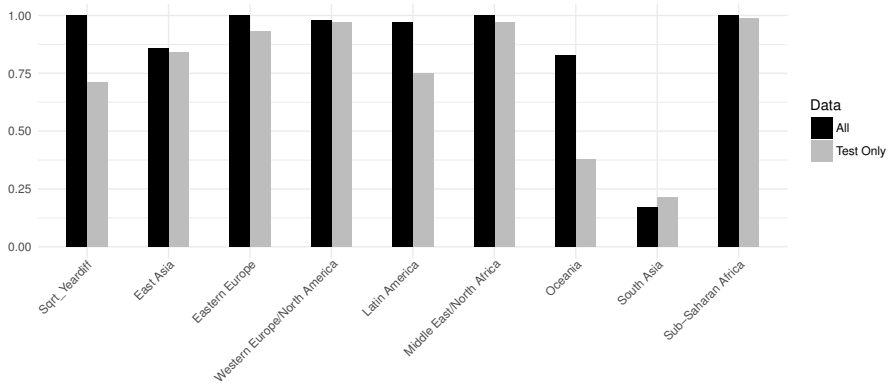
In §4.2, the models we estimate use a blend of human- and machine-generated similarity values for the dependent variable. In particular, we use human-generated data for the training-set dyads, and machine-generated data for dyads with one or both members drawn from the test set. Since this strategy most closely approximates the approach applied researchers would be likely to use in practice, we believe the results we offer in text offer the most useful information for applied readers. However, more methodologically-oriented readers might reasonably object that this approach produces a baseline that is too optimistic.

To address these concerns, we reestimated a set of models using only those dyads with neither country contained in the training set ($n = 6903$ with a training set of 75 countries), using the same 100 train/test splits used in §4.2. For each train/test split, we estimated two models: one using the human-generated similarity values drawn from the Comparative Constitutions Project, and one using our machine-generated approximations. We then compared the coefficient estimates generated using each model, and recorded the number of coefficients in each case in which our two models returned the same conclusion regarding sign and statistical significance (using $\alpha = 0.05$ as a significance cutoff). Note that, since this approach excludes all dyads with at least one member drawn from the training set, the underlying dataset (and resulting "criterion" human-generated coefficient estimates) changes for each train/test split. As a result, the "correct" substantive conclusion for each coefficient (i.e. the coefficient drawn when examining a model estimated using human-generated similarity values) may differ for each replicate.

The results of this experiment are shown in Figure 8.[4] Unsurprisingly, most coefficients perform somewhat worse in the test set-only models than those that include all

[4]In two train/test splits, the test set contained no countries drawn from the South Asia region, leading this coefficient to be excluded from the model. As a result, the sample size for this coefficient is 98 rather than 100.

Figure 8: Proportion of coefficients that return the same substantive conclusion for human- and machine-generated models, using all data and test-only dyads.



Proportion of model replicates which returned the same substantive conclusions for all covariates across 100 train/test splits. Values for models estimated using all data and models restricted to dyads with both countries are drawn from the test set are compared.

available data, with variables corresponding to small subgroups in the data or those whose "true" coefficients that are closest to the statistical significance threshold suffering the largest performance hits. However, even in this more stringent test, some 76% of coefficients return the same substantive conclusions when estimated using test set-only dyads, suggesting that our estimates still perform reasonably well in this set.

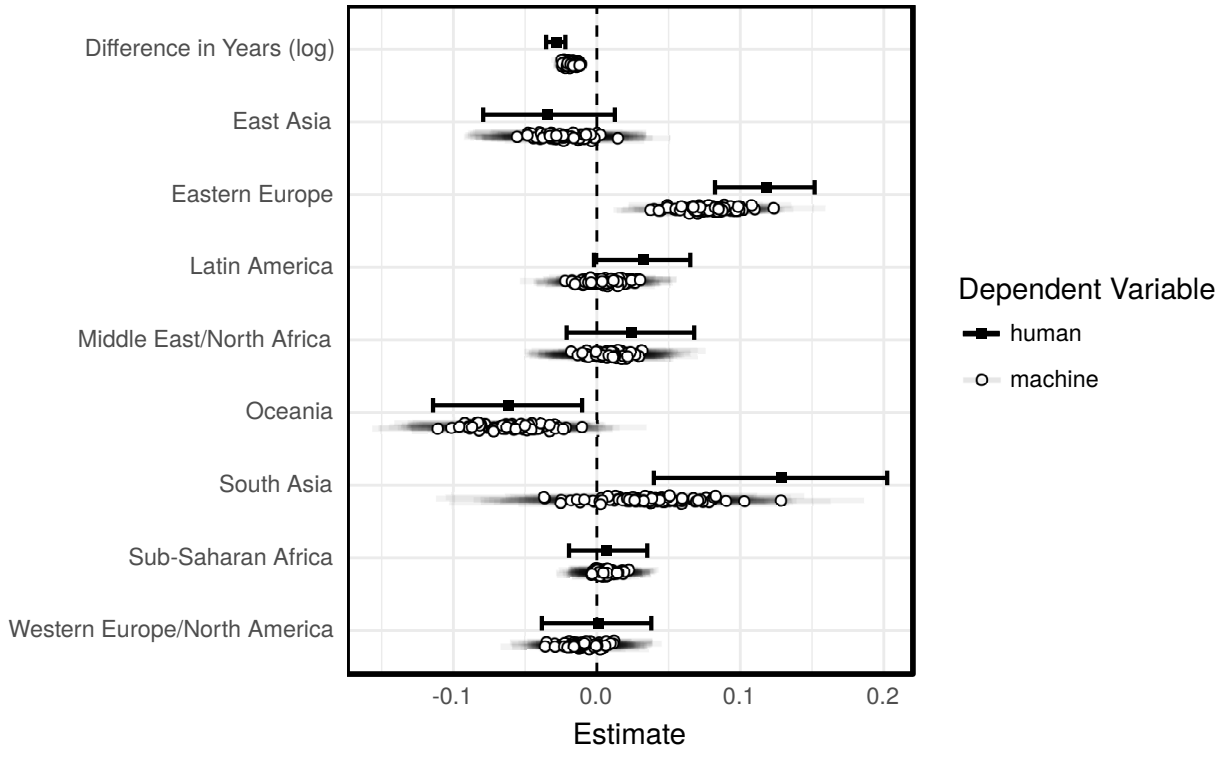
C.3 Alternate Time-Gap Variable Specification for Similarity Regressions

To explore alternate model specifications, in this section we re-estimate the QAP-adjusted models we present in §4.2 with two alternate specifications for the time-gap 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 9 - 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 9, 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).

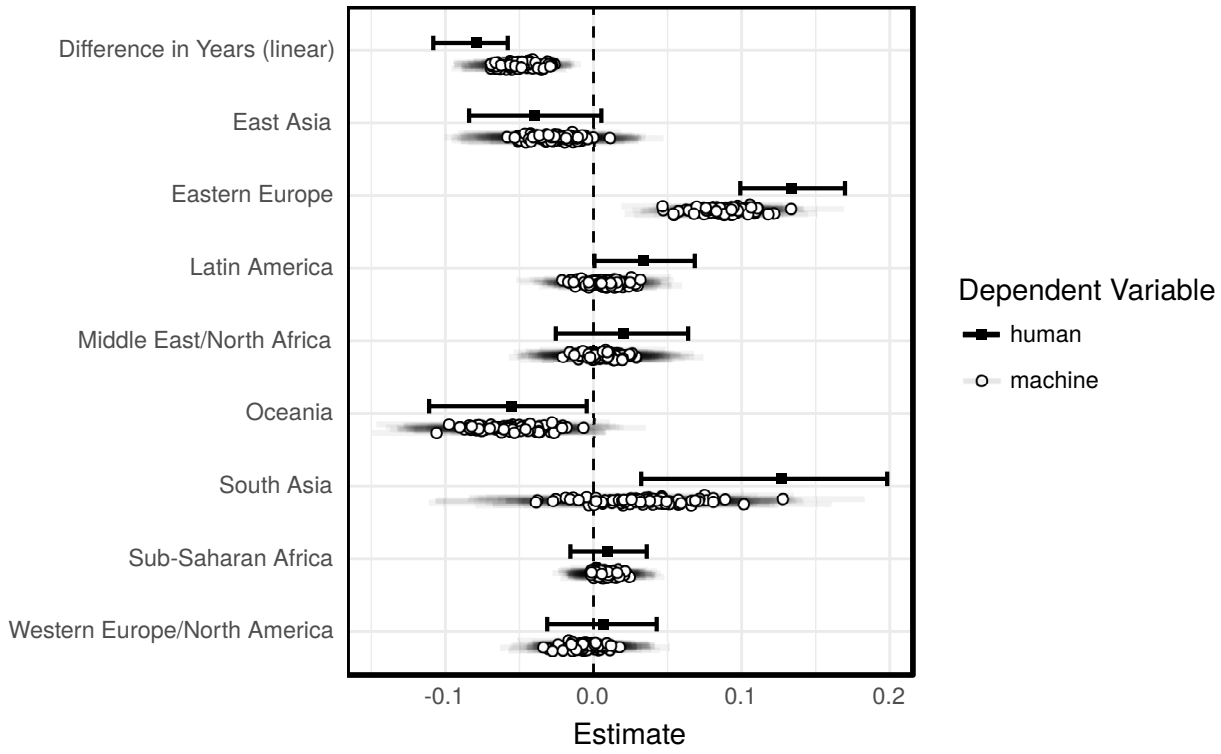
Our second alternate approach - given in Figure 10 - is identical to the first, but with the time gap variable operationalized as $\frac{1}{100}|t_1 - t_2|$. This specification follows that used by Cheibub et al. (2014), who use it to study similarity of constitutions with respect to their provisions 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 9: 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 10: 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.