Supplemental Materials:
Embedding Regression: Models for Context-Specific Description and Inference

Pedro L. Rodriguez Arthur Spirling Brandon M. Stewart

Contents (Appendix)

A Rodman: Details on Sample Sizes **2**

B The Presidential Transition in Meaning **3**

C Asymptotic Behavior **4**

D Benchmarking Embedding Regression against 'full' embeddings **6**

E Measuring Sentiment on the Backbenches **9**

F Experiments with decision variables **10**

G A Matrix uncertainty **19**

H Variation over time **21**

I Software **24**

J Regression outputs **25**

A Rodman: Details on Sample Sizes

A key challenge in Rodman's (2020) approach is that there is relatively little data (per time slice) to estimate embeddings from. Table A.1 presents the number of instances of each theme word for each period. Note that in almost 30% of the word-era combinations, there are fewer than 10 observations. Producing meaningful embeddings given these sample sizes is generally difficult.

	1855–	1880–	1905–	1930–	1955–	1980–	2005–
african_american	63	27	79	171	274	45	22
gender	4	41	374	560	460	258	284
german	1	2	62	512	13	2	2
race	5	15	76	188	190	34	38
treaty	3	1	143	216	30	3	1
Total Documents	71	111	496	1137	660	259	371

Table A.1: Number of instances of each category word in the Rodman corpus by 25 year time slice. All documents have the word equality. Many of the counts are quite low leading to a serious challenge for word embeddings.

B The Presidential Transition in Meaning

The meaning of Trump, the surname, underwent a transformation once Donald J. Trump was elected president of the United States in November 2016. This is a difficult case since the person being referred to is still the same entity, even though the meaning has shifted.

Using ALC, we embed a random sample (with replacement) of 500 mentions of Trump (the number of pre-2017 Trump mentions available) from 2001–2014 and 2017–2020, which we label celebrity Trump and president Trump, respectively. We do the same two cluster routine as above and inspect the 10 nearest neighbors—these are given in Table B.2. As we would expect, Trump in 2001–2014 is mentioned in the context of casinos and real-estate terms while Trump in 2017–2020 is mentioned in the context of terms associated with his presidency.

celebrity	trump, ivanka, ivana, melania, donald,
Trump	wynn, kepcher, condo, taj, mcgahn
President	president, cheney, assailed, clinton, bush,
Trump	presidents, assailing, impeaching, upbraided, alluded

Table B.2: Top 10 nearest neighbors of the transformed cluster centroids. Top row (unshaded) is 2001–2014. Bottom row (shaded) is 2017–2020.

In Figure B.1 we label the mentions of celebrity `Trump` and president `Trump`, respectively (results projected down to two-dimensions for visualization purposes), identifying the two clusters by their dominant word sense. We explicitly mark misclassifications with an x. While the two groups overlap, as would be expected given mentions are all of the same person, it is clear mentions of `Trump` tend to cluster by period.

C Asymptotic Behavior

In this exercise we evaluate the asymptotic performance of our approach. That is, we want to know whether—and how quickly—ALC embeddings converge to embeddings from a fully trained, full corpus `GloVe` model, as we increase the number of instances ALC has access to. Obviously, we would hope that as the sample approaches the whole corpus, ALC 'looks like' a full corpus model.

For our corpus we use the *Congressional Record*. We begin by estimating a full `GloVe` embeddings model and a corresponding transformation matrix $\mathbf{A}$. Next we select a set of 20 target words from the corpus vocabulary, including 10 politics terms and 10 randomly sampled terms, and estimate their corresponding ALC embeddings. We vary the number of instances, from 5 to the total number of instances of each term.[1] Finally, we compute

[1]The set of politics terms are: `democracy`, `freedom`, `equality`, `justice`, `immigration`, `abortion`, `welfare`, `taxes`, `republican` and `democrat`. The set of random terms are:

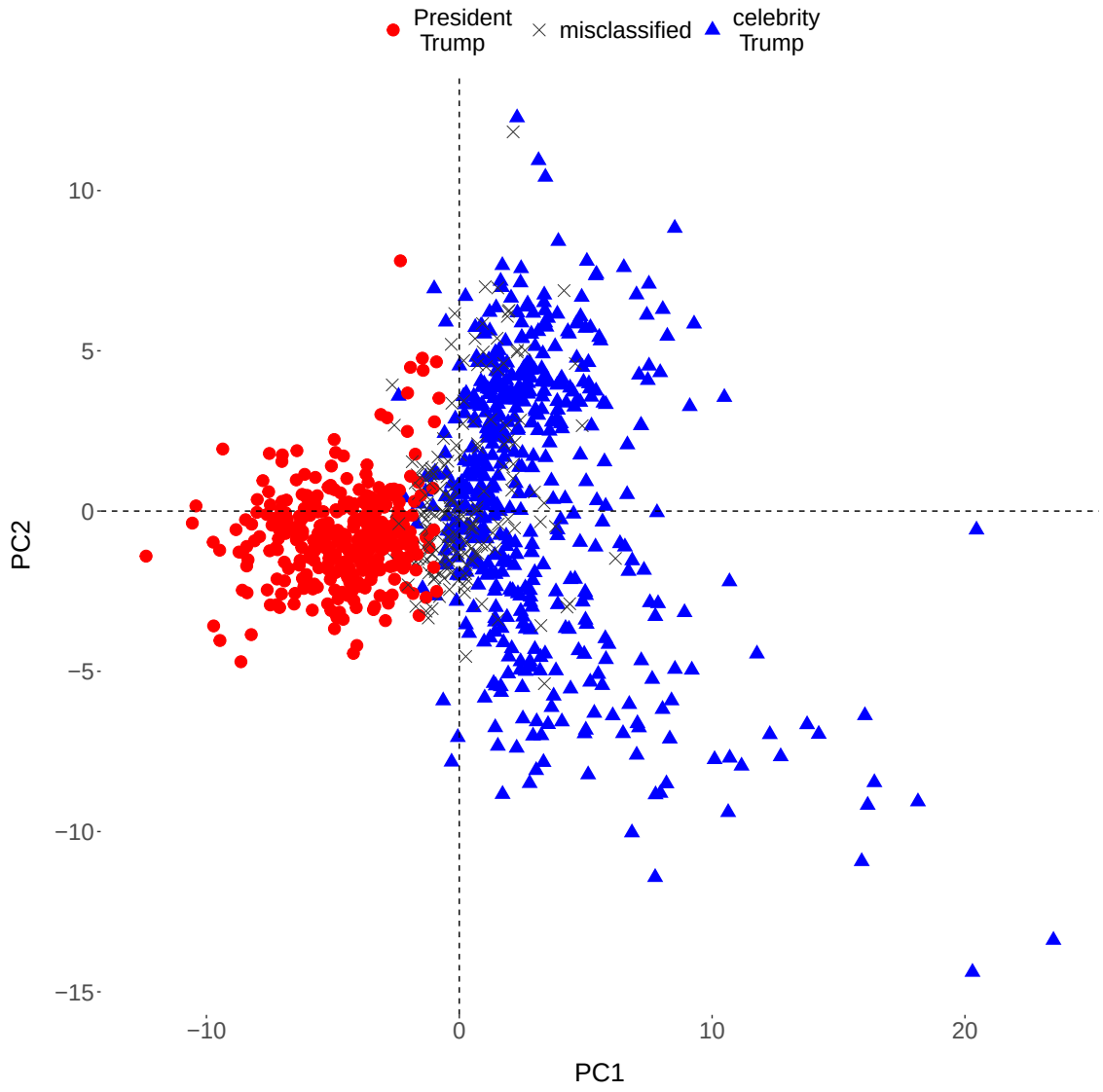
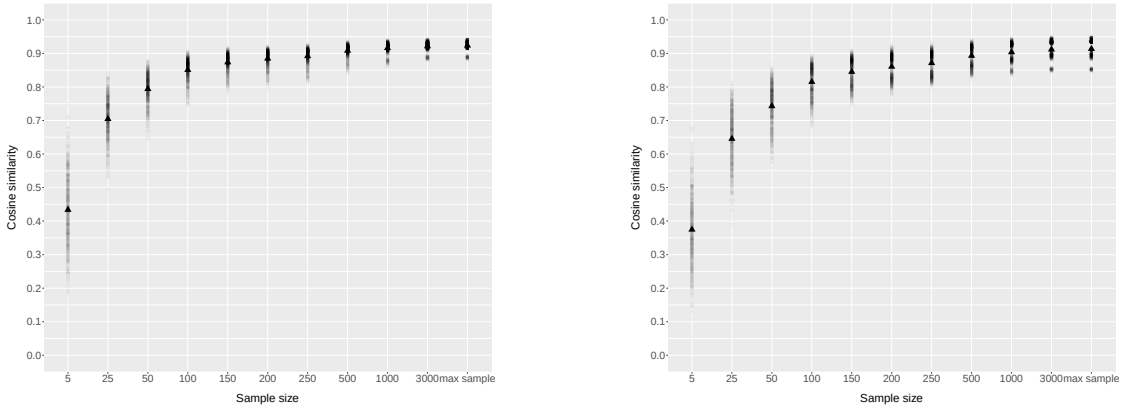


Figure B.1: Each observation represents a single realization of a context. Contexts for celebrity `Trump` include mentions of `Trump` in the New York Times during the period 2001-2014, while contexts for President `Trump` include mentions of `Trump` in the New York Times during the period 2017-2020.

the cosine similarity between each ALC embedding and its corresponding embedding in the full `GloVe` model. For each term and number of instances, we repeat this process with 100 random samples (with replacement). Figure C.2 plots the results separately for each sample

`surrounding`, `reasonable`, `money`, `expertise`, `finish`, `ago`, `ended`, `amazing`, `hours` and `volunteers`.

of the politics and random set of terms. We see that for both sets the ALC embeddings quickly converge to within a margin of error of the `GloVe` embeddings as the number of instances used to estimate the ALC embedding increases. This is expected and welcome behavior. In the case of the politics terms, with as few as fifty instances we see an average cosine similarity value of close to 0.8.[2]



(a) Mean over 10 politics terms

(b) Mean over 10 random terms

Figure C.2: Cosine similarity between a full `GloVe` (full corpus) embeddings model and ALC as a function of sample size.

D Benchmarking Embedding Regression against 'full' embeddings

An alternative to our *regression* approach to quantifying group differences is to estimate a full `GloVe` embeddings model for each group's use of a term. For any given word this can be done by tagging (literally, slightly altering) the word in the corpus such that it appears differently for each different group. Estimating a full `GloVe` model on this tagged corpus yields group-specific embeddings for the tagged words. We can then use these embeddings to quantify group differences. This is computationally costly but provides us with a straight-

[2]Note, we do not expect this value to converge fully to 1 as the transformation matrix A is itself a regression estimate.

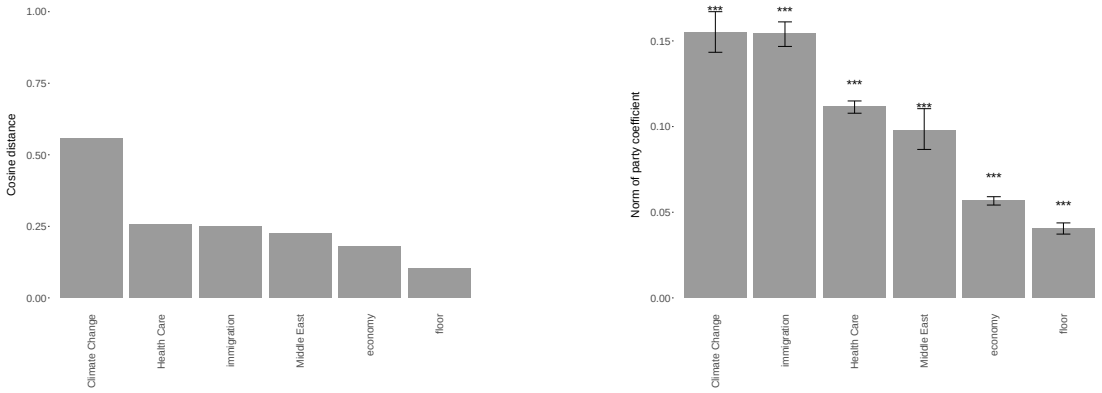
forward benchmark for our approach. Specifically we are interested in comparing inferences when applying both approaches to the following task: ranking a set of terms according to partisanship (in use).

For this exercise we use the Congressional Record corpus, sessions 111^{th}–114^{th} (the Obama years). As target words we use: `immigration`, `economy`, `climatechange`, `healthcare`, `middleeast` and, as a non-political control word we use `floor`.[3]

We tag every instance of a target word in the corpus with the party of its corresponding speaker, so for example, given a particular instance of `immigration` in a speech, we replace it with `immigrationd` if the author of the speech is a Democrat and with `immigrationr` if the author is a Republican. Given party specific embeddings for each target word we quantify partisanship using cosine distance, the higher the cosine distance, the more partisan the term. To quantify partisanship using our preferred approach we simply run a regression with party as a covariate and compute the norm of the resulting coefficient, the higher the norm of the party coefficient, the more partisan the term.

Figure D.3 plots both sets of results. Broadly speaking, the inferences one would draw from each are similar. On the one hand, `Climate Change` is clearly the most partisan issue while, as expected, our control term `floor` is the least partisan according to both models. `economy` stands out as the second least partisan according to both models. The remaining terms are similarly ranked except our approach suggests `immigration` is somewhat more partisan than `Health Care` and `Middle East`. All in all, the inferences from both approaches are not wildly different. In contrast to estimating a full `GloVe` embeddings model however, our approach is much faster, more stable—the solution does not vary across runs—and allows us to speak to the significance and sampling variance of our estimates.

[3]In the corpus we replace any mentions of `middle east` with `middleeast`, `health care` with `healthcare`, `immigrants` and `immigrant` with `immigration` and `climate change` with `climatechange`.

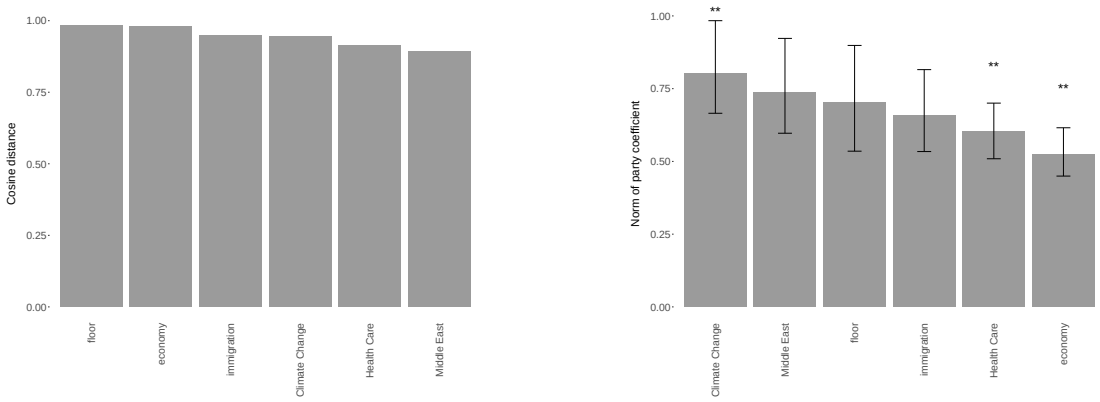


(a) `GloVe` 'regression' (b) `conText` regression

Figure D.3: Partisan differences using the Congressional Record corpus (Sessions 111$^{\text{th}}$ - 114$^{\text{th}}$). See SM Section J for full regression output.

Next we compare each model's performance with a significantly reduced sample, specifically one in which each target word appears in no more than five documents.[4] Our goal with this exercise is to compare how both methods fare in a small-sample world, relative to inferences using the full corpus. Figure D.4 plots both sets of results. In the case of the full GloVe model we see results are now flipped, with `floor` and `economy` showing the largest partisan differences. In contrast, the ALC results are comparable to the full-sample case. While `floor` shows a larger norm, it is not significant, and `Climate Change` remains the most significantly partisan of the target words. Combined, these results serve to highlight the added value of our approach, yielding similar inferences as the full embeddings model at a fraction of the cost and more robust in small-sample scenarios.

[4]To build this corpus we identify for each target word all documents containing the word and randomly sample five of these. We exclude from this sample any document containing multiple target words. Documents that do not contain any of the target words remain part of the corpus.



(a) GloVe (b) ALC

Figure D.4: Partisan differences using the Congressional Record corpus (Sessions 111$^{\text{th}}$ - 114$^{\text{th}}$), including only 5 instances of each target word. See SM Section J for full regression output.

E Measuring Sentiment on the Backbenches

To construct that sentiment estimate for the House of Commons, we take the following steps:

1. for the policy area of interest, designate the seed word (e.g. 'nhs' for the "Health")

2. embed that seed using ALC as described above (specifically using `GloVe` embeddings and the original Khodak et al. (2018) **A** matrix). We now have an embedding for every instance of the term. Aggregate those embeddings to party-rank-month (so if Tory backbenchers mention 'eu' twice in July 2015, we take the average embedding of those two mentions)

3. using inner product as a measure of similarity, compare that aggregate embedding to the embedding of words in a sentiment dictionary—i.e. the embeddings of words thought to connote positive or negative valence. The specific dictionary we use for this purpose is the Affective Norms for English Words (Warriner, Kuperman and Brysbaert, 2013), preprocessed and operationalized in the way described by Osnabrügge, Hobolt and Rodon (2021, ftn 9).

4. for a given (averaged) embedding for the party-rank-triple, calculate its valence as its mean similarity to the set of positive terms *plus* negative one multiplied by the mean similarity to the set of negative terms.

5. finally, we rescale those valences within party and term of interest (i.e. Tory-backbench and Tory-cabinet sentiment towards a given term over the time series is scaled 0-1, and the same is done for Labour-backbench and Labour-cabinet sentiment).

This general approach is inspired by the word embeddings association test (WEAT) of Caliskan, Bryson and Narayanan (2017) for measuring bias in text. Their approach uses cosine similarity as the measure of similarity rather than the inner product. Although widely used, this approach has been criticized for depending too heavily on the relative frequency of the seed word and the target words (Ethayarajh, Duvenaud and Hirst, 2019; van Loon et al., 2022). This dependence may arise in part due to the standardization by magnitude in the cosine measure. We might expect this problem to be less severe in our setting because we are comparing against two different embeddings of the same word, which—in this example—are used frequently by both groups.

At the time of writing, this is an active area of study and so out of an abundance of caution, we ran the study using both the conventional cosine similarity and the inner product (which does not standardize). We then presented the inner product results in the main paper as it has less well-defined patterns. Figure E.5 presents the results using cosine similarity. Researchers looking to use a similar design should consult the latest literature on the topic for guidance.

F Experiments with decision variables

While our approach does not require any active tuning of parameters, there are nevertheless choices to be made. To better guide practice, we ran a number of experiments. First, we briefly revisit our `Trump` vs. `trump` example from the main paper. Recall that our

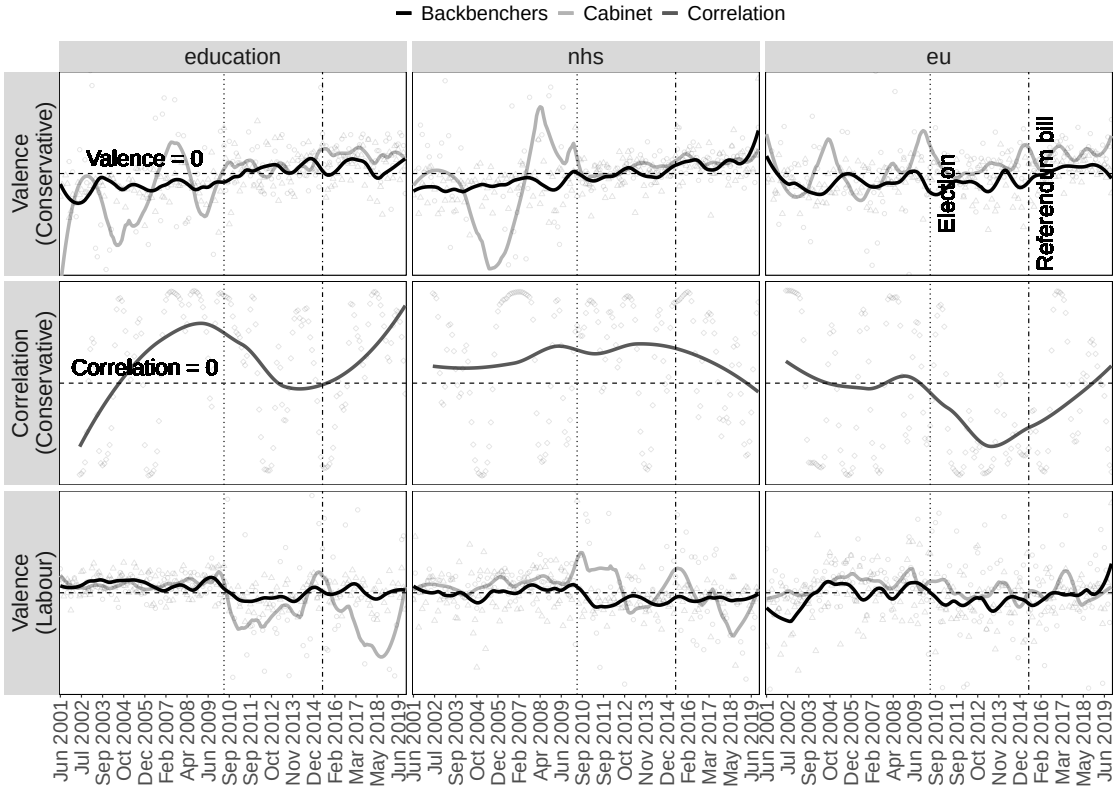


Figure E.5: The equivalent of Figure 9 using cosine similarity.

task is to classify ALC embeddings of individual mentions of the term `Trump/trump` in our NYTs corpus, into one of the two senses of the term—a *supervised task*. We use Stanford GloVe—window size of 6, 300 dimensions—as our pre-trained embeddings along with its corresponding **A** matrix as estimated by (Khodak et al., 2018). We further use k-means clustering to assign each single-instance ALC embedding into one of $k = 2$ clusters. To evaluate performance we use three common clustering metrics:

- **Homogeneity**: maximized ($h_i = 1$) when each cluster contains only members of a single class; minimized ($h_i = 0$) when each cluster contains a random assortment of members.
- **Completeness**: maximized ($c_i = 1$) when all members of a given class are assigned to the same cluster; minimized ($c_i = 0$) when members are randomly assigned to clusters.
- **V-measure**: a weighted combination –harmonic mean– of homogeneity and completeness; the more homogeneous and complete a given clustering, the higher this score – bounded between 0 (worst) and 1 (perfect).

And focus on two modeling choices:

1. **Context window size**: this refers to the window size of the contexts used to estimate the ALC embeddings which we set to 2, 6 or 12.
2. **Stopword removal**: we evaluate the effect of removing stopwords at the point of preprocessing the contexts used in estimating the ALC embeddings.[5]

Figure F.6 plots these results. First, we observe that irrespective of the size of the window or removal of stopwords, ALC is capable, with varying degrees of success, of distinguishing between the two senses. However, we do observe that for this particular task, larger windows

[5]We use `quanteda`'s list of stopwords.

and the removal of stopwords can be helpful, showing marginally higher scores across all three metrics.

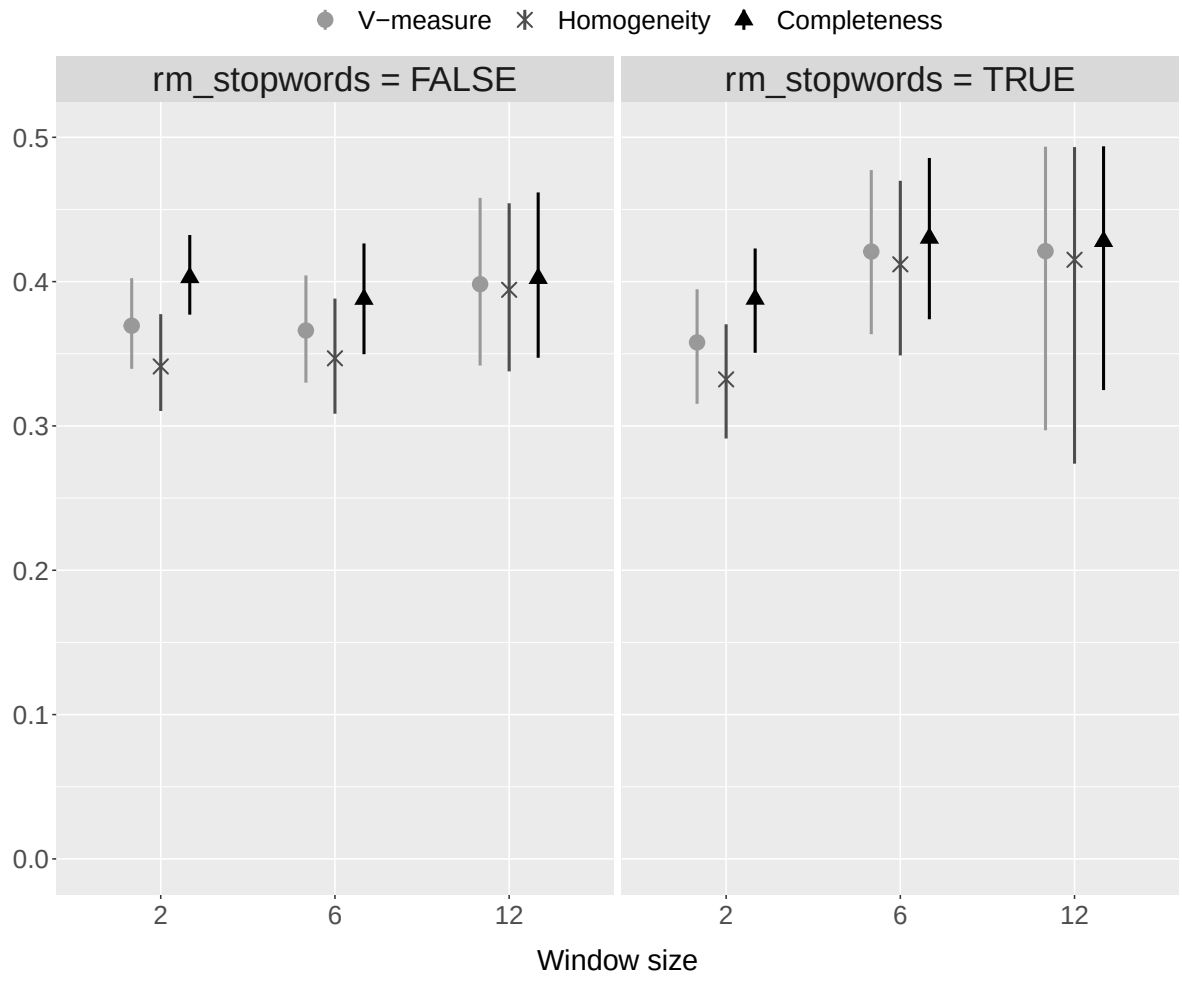


Figure F.6: Performance on a supervised classification task as a function of context window size and leaving/removing stopwords.

We next turn to our Congressional Records corpus–Session 107 - 114.[6] We evaluate performance along three metrics:

- **Reconstruction**: similarity between the estimated ALC embeddings of a set of terms

[6]We use a larger swathe of the Congressional Records than in our main example in the paper in order to train high quality locally-fit embeddings.

and their corresponding embedding in the set of pre-trained embeddings. The higher this similarity, the better ALC "reconstructs" the 'true' underlying embeddings (i.e. low bias).

- **Nearest neighbors**: overlap in nearest neighbors to the estimated ALC embeddings, as measured by the Jaccard Index, across the various model specifications. The higher this overlap, the lower the variance as a function of model specification.

- **Substantive**: temporal trends in partisan differences for a set of political terms. We define the partisan difference for a given term during a given session of Congress as the cosine similarity between the two party –Republican and Democrat– ALC embeddings. For each term we then have a time series spanning the eight sessions of Congress that constitute our corpus. The goal is to evaluate how these trends compare across model specifications. We quantify this using Pearson correlation. The higher the Pearson correlation, the lower the variance –in the substantive interpretation of results– as a function of model specification. We call these "substantive" in that they capture the types of relationships researchers are often interested in e.g. temporal variation in group differences.

We narrow our comparisons to a set of 'political' terms: `democracy`, `freedom`, `equality` `justice`, `immigration`, `abortion`, `welfare`, `taxes`, `republican`, `democrat`–as in Rodriguez and Spirling (2022). And focus on the same two modeling choices as in the previous experiment, with some differences in implementation:

1. **Context window size**: we train a separate "full" locally-fit `GloVe` embeddings model for each of three window size 2, 6 and 12 and estimate the corresponding $\mathbf{A}$ matrices. All models include any feature that appears at least 10 times in the corpus and are at least 3 characters long. Models are trained for 25 iterations.[7] We also evaluate

[7]We use the R package `text2vec` to estimate "local" GloVe embeddings.

Stanford GloVe (window of size 6, 300 dimensions) embeddings—downloaded from Stanford NLP's webpage—but in this case estimate a "locally-fit" **A** matrix i.e. using our Congressional Records corpus as input for the estimation rather than using Khodak et al. (2018)'s **A**.

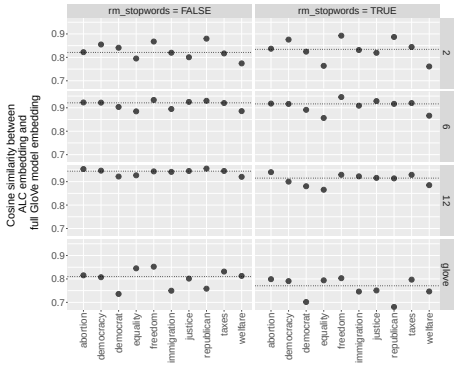
2. **Stopword removal**: as with the `trump`/`Trump` experiment, we evaluate the effect of removing stopwords at the point of preprocessing the contexts used in estimating the ALC embeddings. We leave stopwords in when estimating the full embeddings models as is standard practice.

Our results are summarized in Figure F.7. In Figure F.7a we observe that ALC performs its intended task well—in most cases extremely well—namely to reconstruct the original embeddings. Across all "local" model specifications we see an average cosine similarity across the ten terms of above 0.8 with larger window models (6 and 12) achieving an average above 0.9. Results suggest avoiding smaller windows (< 5) could be advantageous, although not strictly necessary. Even using `GloVe` pre-trained embeddings with a localized **A** matrix achieves excellent results. In other words, lacking enough data to train their own embeddings models, users can reasonably resort to pre-trained embeddings trained on (large and broad) corpora. Removing stopwords does not improve results, indeed if anything results are generally slightly worse. This is not altogether surprising in that stopwords were not removed when estimating the full embeddings model nor, more importantly, when estimating the **A** matrix. The latter takes care of reweighting dimensions in a way that mitigates the prevalence of stopwords.

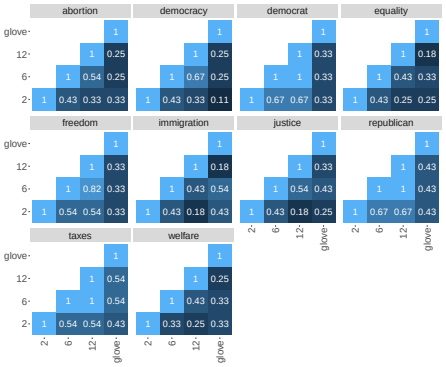
In Figure F.7b we observe the overlap in the top 10 nearest neighbors across the various model specifications as measured by the Jaccard Index. Results suggest significant overlap across models, even between Stanford `GloVe` pre-trained embeddings and local models.

Figure F.7d plots the trends in partisan differences over time. The appropriate way to read this plot is to compare for each term the time trends across model specifications—so compare plots within a column. We observe significant overlap across specifications for all

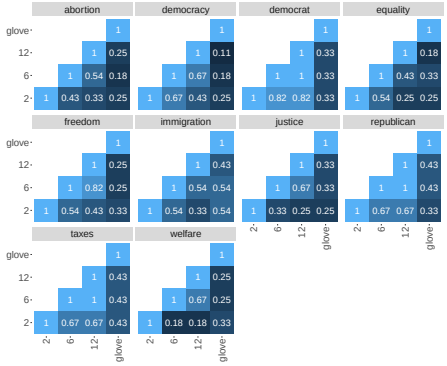
terms. We can further quantify this using Pearson correlation (see Figure F.7e). Again, we observe very high (above 0.9) correlations across all models, including between Stanford `GloVe` and the set of locally trained models.



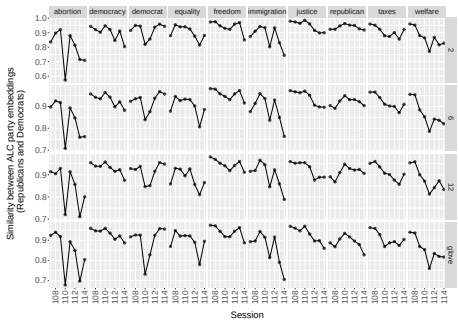
(a) Reconstruction



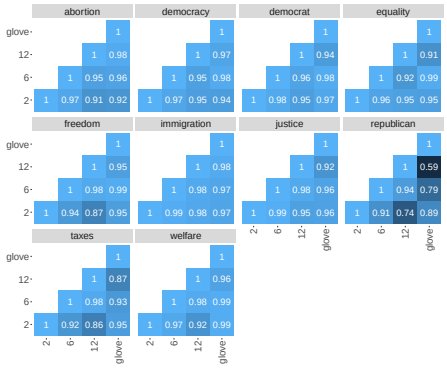
(b) Nearest neighbors (Jaccard Index)



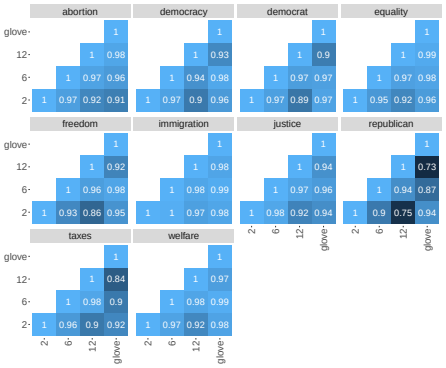
(c) Nearest neighbors (Jaccard Index) w/o stopwords



(d) Substantive (visualization)



(e) Substantive (correlation)



(f) Substantive (correlation) w/o stopwords

Figure F.7: Performance as a function of context window size and leaving/removing stopwords.

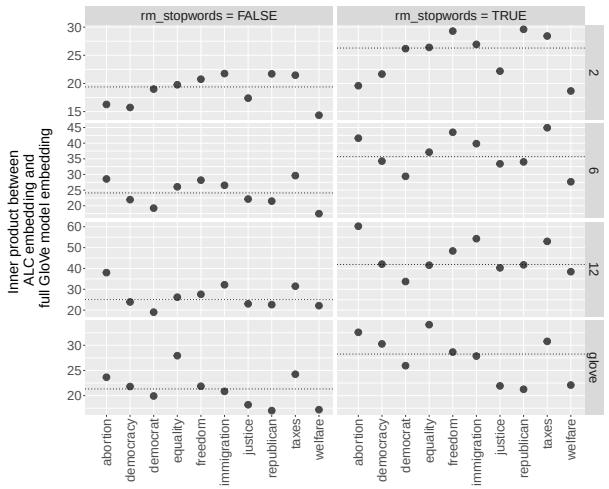
In addition to our experiments above, we also looked at the following decision variables:

- Similarity metric: cosine similarity versus inner product. The former only cares about the angle between vectors while the latter cares about the angle and magnitude. Indeed, cosine similarity is equivalent to normalizing the inner product by the magnitude of the vectors. It is uncommon to use inner product as a similarity metric due in part to its sensitivity to document length.

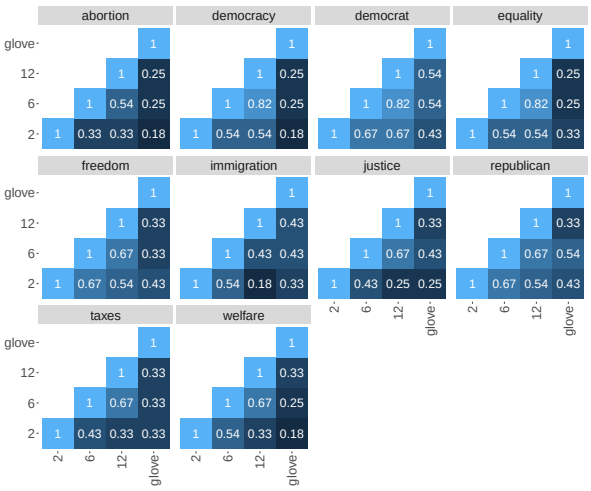
- Stemming: it's often the case that nearest neighbors show various terms with the same stem e.g. "enforcing" and "enforce". A user can easily group these by averaging the cosine similarities to nearest neighbors with the same stem.[8]

Figure F.8 summarizes results using the inner product as the similarity metric. Results are generally more variable then when using cosine similarity but we nevertheless still observe significant overlap in nearest neighbors and high correlations in temporal trends across model specifications. Unless there's a very problem-specific reason to use inner product we suggest users follow common practice and use cosine similarity as a metric.

[8]A practical matter to note here is that averaging across terms with the same stem may introduce noise through low-frequency misspelled terms with low cosine similarities to the target word. To avoid this we suggest subsetting candidate terms to correctly spelled words—this can be automated—when using stemming.

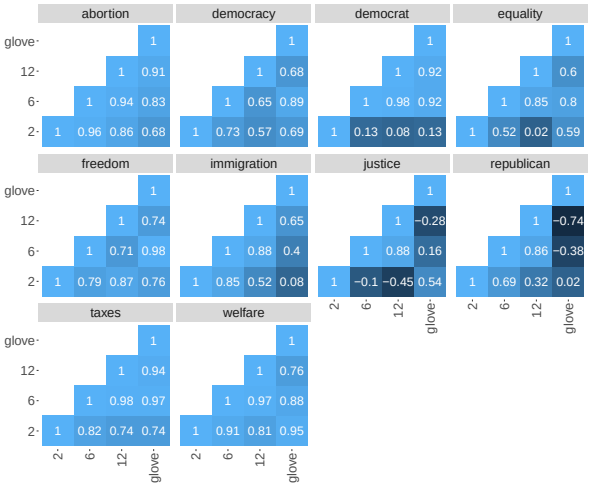


(a) Reconstruction



(b) Nearest neighbors (Jaccard Index)

(c) Substantive (visualization)



(d) Substantive (correlation)

Figure F.8: Performance as a function of similarity metric and stemming.

Figure F.9 summarizes results using stemming – in this case it only makes sense to look at our nearest neighbors metric. With some exceptions we observe significant overlap across models. While stemming may help group similar terms users should keep in mind it often comes at the cost of interpretability.

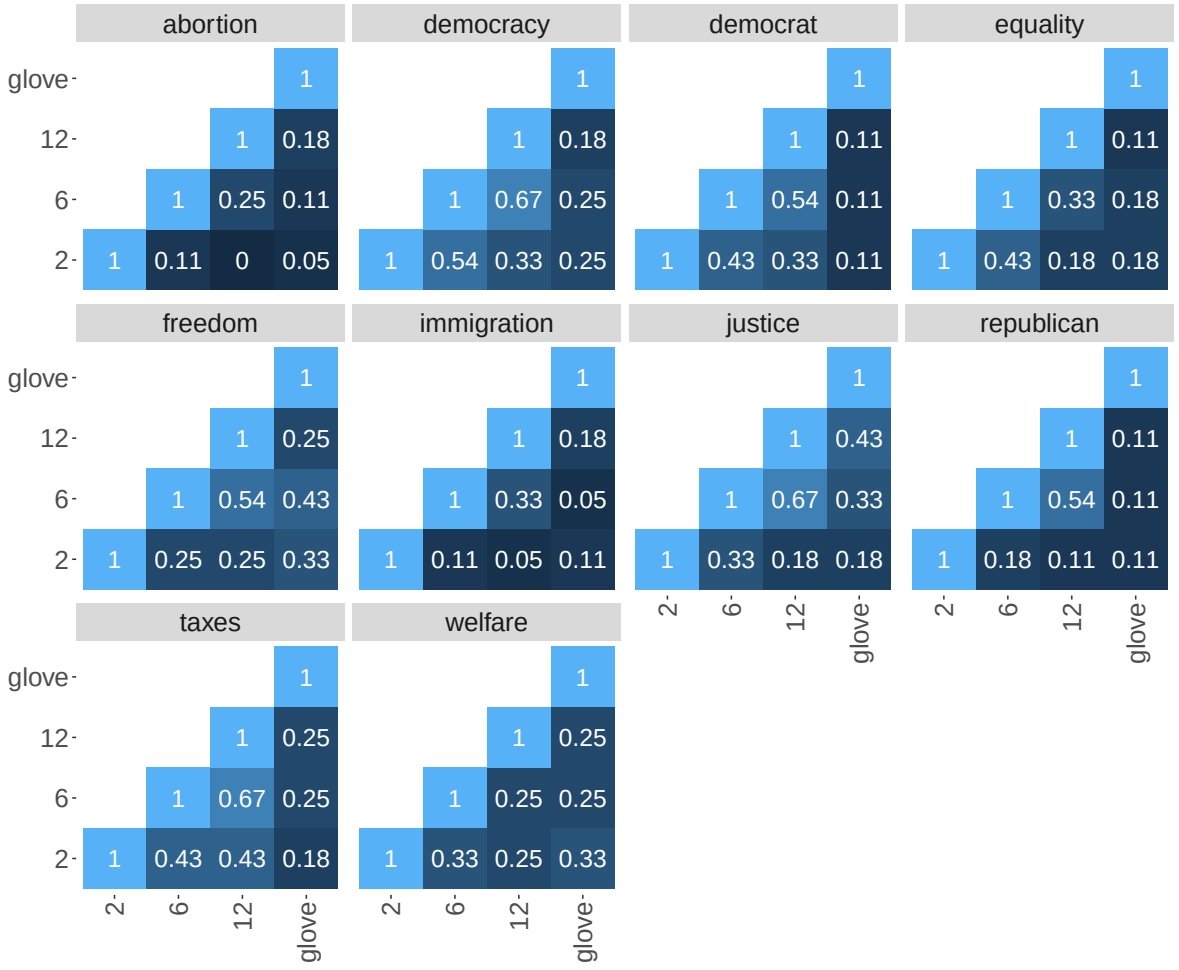
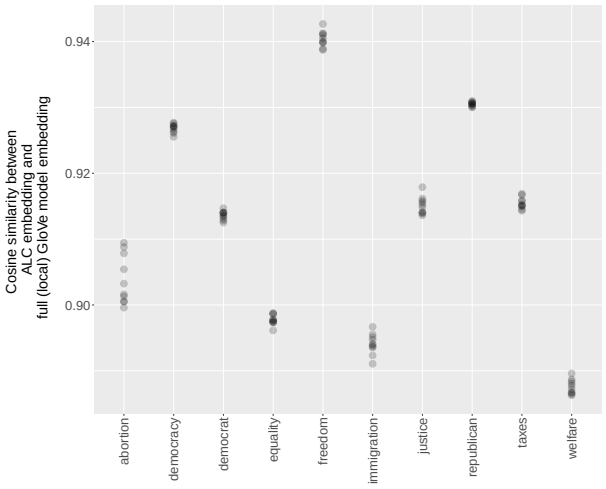


Figure F.9: Substantive (correlation)

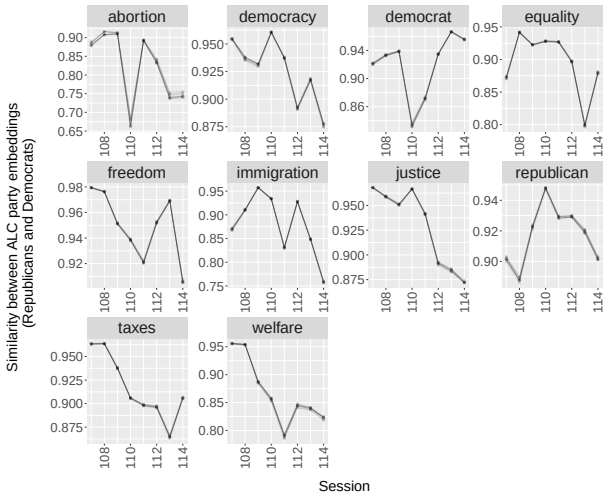
G A Matrix uncertainty

Throughout the examples in our paper we have assumed the transformation matrix **A** to be known and fixed, ignoring any uncertainty that may arise as result of having to estimate **A**. In the experiment that follows we evaluate how reasonable said assumption may be. We again use the Congressional Records (Sessions 107–114) as our corpus and a locally estimated GloVe model with context window size of 6. We next estimate 10 different **A** matrices for 10 bootstrapped samples of the corpus and apply the same evaluation framework as described in Section F. While our objective in Section F was to compare results across various

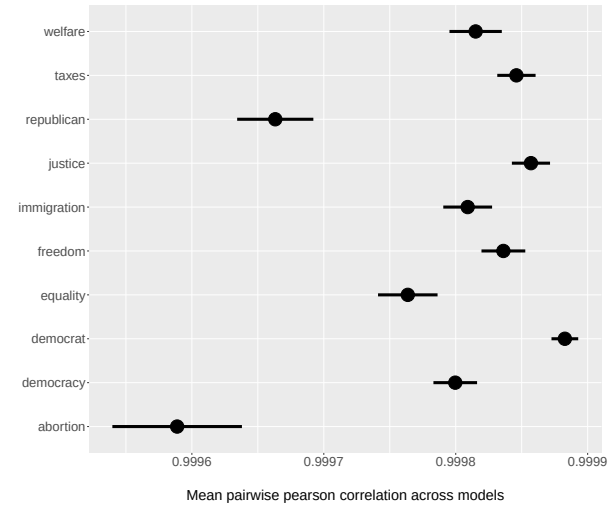
model specifications, in this case it is to compare results across the 10 estimates of **A**. Across all metrics—*reconstruction*, *nearest neighbors* and *substantive*—we see remarkably indeed negligible differences (see Figure G.10). Users should consider uncertainty in the calculation of the **A** matrix as a second-order concern, and unlikely to be consequential for topline results.



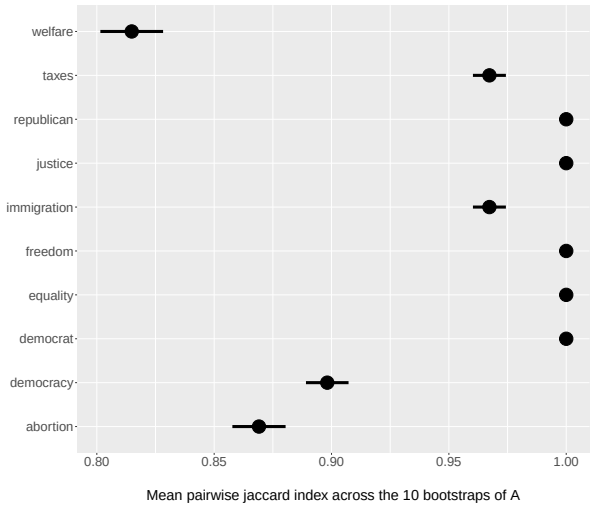
(a) Reconstruction



(b) Substantive (visualization)



(c) Substantive (correlation)



(d) Nearest neighbors (Jaccard Index)

Figure G.10: Performance as a function of **A** matrix estimation.

H Variation over time

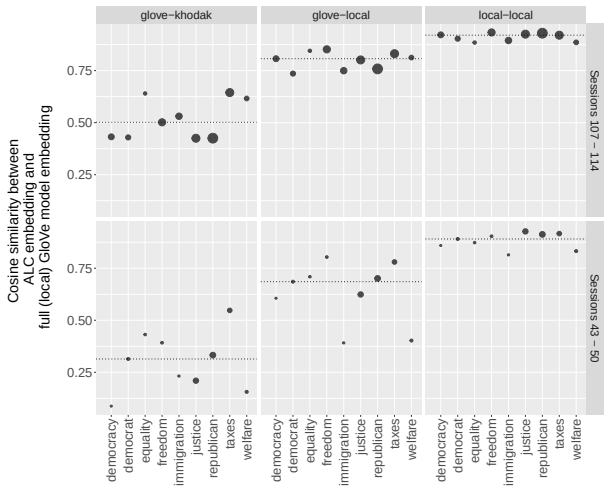
One concern users of the proposed approach may have is that pre-trained embeddings estimated on modern texts are ill-suited to study older texts. Take for example our replication of Rodman (2020) above where we use Stanford `GloVe` embeddings to study a corpus that includes data spanning 161 years. To shed light on this concern we look at two subsets of the Congressional Records: Sessions 43–50 (1873–1889) and Sessions 107-114 (2001–2017). For each subset we estimate a local embeddings model—context window size 6—and corresponding **A** matrix. We want to evaluate how well results using Stanford `GloVe` pre-trained embeddings match results using these local models with the expectation that we should observe larger differences when applying Stanford `GloVe` pre-trained embeddings to study the earlier sessions of Congress. To round off our comparison we evaluate differences using Stanford `GloVe` pre-trained embeddings with Khodak et al. (2018)'s **A** matrix and the same embeddings but with a locally estimated **A** matrix. So, for each period we have the following combinations of models:

- **local - local**: locally trained embeddings on the corresponding corpus and a locally estimated A matrix.
- **GloVe - local**: Stanford GloVe pre-trained embeddings and a locally estimated A matrix.
- **GloVe - GloVe**: Stanford GloVe pre-trained embeddings and the corresponding A matrix estimated by Khodak et al. (2018).

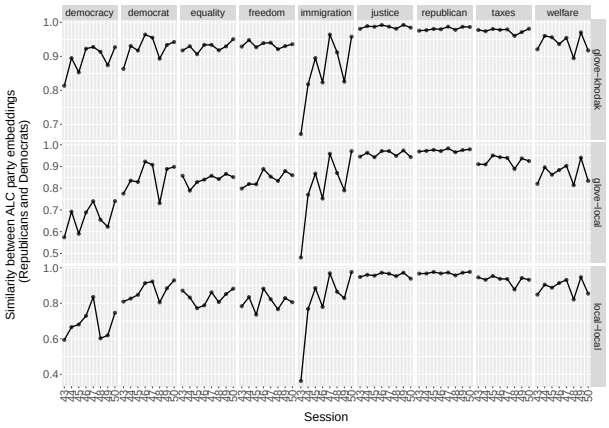
We again use the evaluation framework laid out in Section F. Figure H.11 summarizes our results. A couple of general patterns emerge. As expected, locally trained models with their corresponding **A** matrices show the best performance—irrespective of time period—across all metrics—reconstruction, nearest neighbors and substantive. Nevertheless, Stanford `GloVe` pre-trained embeddings with a locally estimated **A** matrix show remarkably strong

performance in terms of our reconstruction metric and nearest neighbors even for the earlier period—albeit somewhat worse than when employed to analyze more recent texts. As expected, Stanford `GloVe` pre-trained embeddings with Khodak et al. (2018)'s $\mathbf{A}$ shows somewhat lower performance in terms of reconstruction and nearest neighbors. However, turning to our substantive metric, with some exceptions correlations are high across all models, suggesting that even Stanford `GloVe` with Khodak et al. (2018)'s $\mathbf{A}$ performs well in capturing the general trends in the underlying data. What should readers make of these results? Given a large corpus, locally trained embeddings and corresponding $\mathbf{A}$ matrix is desirable. However, for smaller corpora, using large pre-trained embeddings models such as Stanford `GloVe` embeddings, will be more than adequate in most cases, ideally with a locally trained $\mathbf{A}$ matrix—given enough data—but not necessarily so.[9]

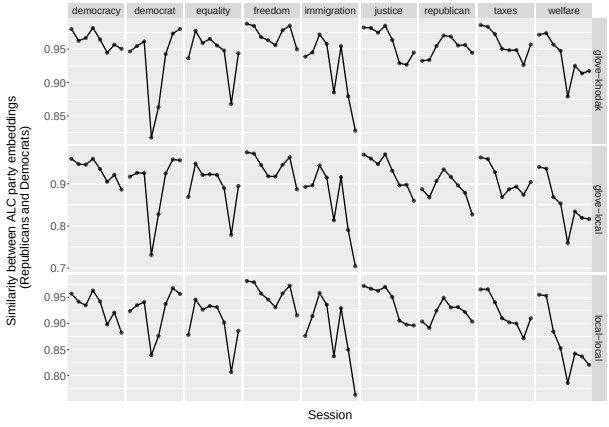
[9]Note, data requirements to train an $\mathbf{A}$ matrix are generally orders of magnitude lower than to train a full embeddings model. For a D dimensional embedding space, the former requires estimating $D \times D$ coefficients, whereas the latter requires estimating $V \times D$, with V representing vocabulary size and generally $V \gg D$ or there would be no reason to perform the embedding.



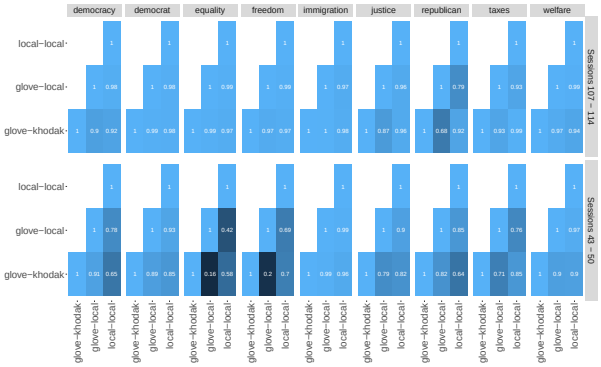
(a) Reconstruction



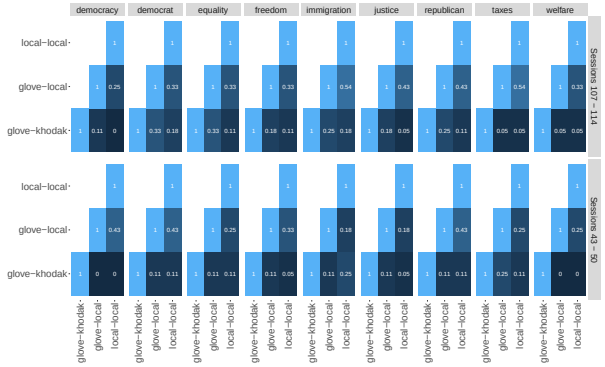
(b) Substantive (visualization)



(c) Substantive (correlation)



(d) Nearest neighbors (Jaccard Index)



(e) Nearest neighbors (Jaccard Index)

Figure H.11: Performance as a function of time.

I Software

To facilitate applying the methods presented in this paper we put together an R package – `conText`. The main function `conText` follows generic `R lm( )` and `glm( )` syntax in terms of $\sim$ operator. Please refer to the quick start guide to get started using the package. As with any package, we had to make a couple of design decisions that are worth noting here. First, ALC embeddings are computed using the available pre-trained context word embeddings. If a given context word is not available in the provide pre-trained embeddings, then that context word is simply ignored and the average is taken over the set of available context embeddings. Second, we've found that in practice limiting the candidate nearest neighbors to the set of words in the provided contexts, significantly reduces noise (non-sensical nearest neighbors such as misspelled words etc.). Whenever exploring nearest neighbors you can use the parameter `candidates` to delimit the set of nearest neighbors. Finally, we have made available—or simply more accessible—the GloVe pre-trained embeddings used in most of the examples in this paper along with their corresponding transformation matrix.[10] We are often asked when is it appropriate to use these pre-trained embeddings and their corresponding transformation matrix rather than estimate ones own. Unfortunately, there is no hard-and-fast rule for this, it comes down to how distinct you think your corpus is relative to the corpus used to train these embeddings (Wikipedia 2014 and Gigaword 5).

[10]The original GloVe embeddings computed by the Stanford NLP Group can be found here while the original transformation matrix computed by Khodak et al. (2018) can be found here.

J Regression outputs

Trump	0.490***
	(0.470, 0.510)
Post_Election	0.208***
	(0.194, 0.222)
Trump x Post_Election	0.446***
	(0.427, 0.467)
Observations	14,448

*p<0.1; **p<0.05; ***p<0.01

Note: Bootstrapped 95% confidence intervals in parentheses.

Table J.3: Regression output for Figure 4 in main text.

	and	but	also	abortion	marriage	immigration
Republican	0.036***	0.030***	0.050***	0.105***	0.126***	0.175***
	(0.035, 0.038)	(0.026,0.033)	(0.045,0.055)	(0.100,0.110)	(0.114,0.137)	(0.171,0.180)
Male	0.035***	0.041***	0.051	0.054***	0.107***	0.087***
	(0.033, 0.037)	(0.036,0.045)	(0.047,0.056)	(0.048,0.060)	(0.092, 0.121)	(0.080,0.094)
Observations	120,465	15,131	5,699	6,701	2,081	15,856

*p<0.1; **p<0.05; ***p<0.01

Note: Bootstrapped 95% confidence intervals in parentheses.

Table J.4: Regression output for Figure 5 in main text.

	immigration
Nominate Score (dim-1)	0.262***
	(0.252, 0.270)
Observations	7,526

*p<0.1; **p<0.05; ***p<0.01

Note: Bootstrapped 95% confidence intervals in parentheses.

Table J.5: Regression output for Figure 6 in main text.

	empire	Observations
British (1935-1936)	0.438***	2,964
	(0.417, 0.461)	
British (1937-1938)	0.379***	3,304
	(0.353, 0.401)	
British (1939-1940)	0.400***	1,741
	(0.376, 0.419)	
British (1941-1942)	0.480***	2,656
	(0.453, 0.505)	
British (1943-1944)	0.404***	2,843
	(0.380, 0.426)	
British (1945-1946)	0.406***	2,658
	(0.383, 0.430)	
British (1947-1948)	0.446***	2,551
	(0.426, 0.464)	
British (1949-1950)	0.4961009***	1,792
	(0.469, 0.521)	
British (1951-1952)	0.525***	1,370
	(0.499, 0.557)	
British (1953-1954)	0.622***	1,125
	(0.586, 0.653)	
British (1955-1956)	0.691***	1,519
	(0.650, 0.727)	
British (1957-1958)	0.802***	1,067
	(0.745, 0.854)	
British (1959-1960)	0.689***	1,073
	(0.641, 0.736)	
British (1961-1962)	0.714***	1,288
	(0.664, 0.756)	
British (1963-1964)	0.685***	1,138
	(0.640, 0.736)	
British (1965-1966)	0.592***	1,166
	(0.557, 0.622)	
British (1967-1968)	0.559***	890
	(0.518, 0.590)	
British (1969-1970)	0.564***	606
	(0.517, 0.606)	

*p<0.1; **p<0.05; ***p<0.01

Note: Bootstrapped 95% confidence intervals in parentheses.

Table J.6: Regression output for Figure 7 in main text.

	empire	Observations
British (1971-1972)	0.607***	651
	(0.564, 0.649)	
British (1973-1974)	0.522***	484
	(0.470, 0.564)	
British (1975-1976)	0.487***	526
	(0.446738, 0.525)	
British (1977-1978)	0.4954084***	494
	(0.451, 0.548)	
British (1979-1980)	0.532***	572
	(0.486, 0.574)	
British (1981-1982)	0.578***	532
	(0.500, 0.640)	
British (1983-1984)	0.594***	742
	(0.549, 0.640)	
British (1985-1986)	0.759***	956
	(0.711, 0.814)	
British (1987-1988)	0.538***	996
	(0.477, 0.588)	
British (1989-1990)	0.438***	1,022
	(0.405, 0.474)	
British (1991-1992)	0.453***	1,162
	(0.402, 0.492)	
British (1993-1994)	0.531***	812
	(0.489, 0.569)	
British (1995-1996)	0.558***	731
	(0.504, 0.606)	
British (1997-1998)	0.608***	720
	(0.560, 0.650)	
British (1999-2000)	0.591***	630
	(0.539, 0.650)	
British (2001-2002)	0.581***	628
	(0.530, 0.630)	
British (2003-2004)	0.650***	575
	(0.571, 0.729)	
British (2005-2006)	0.569***	573
	(0.515, 0.624)	
British (2007-2008)	0.623***	493
	(0.569, 0.675)	
British (2009-2010)	0.663***	463
	(0.620, 0.714)	

*p<0.1; **p<0.05; ***p<0.01

Note: Bootstrapped 95% confidence intervals in parentheses.

Table J.7: Regression output for Figure 7 in main text (continued).

	immigration	economy	climatechange	healthcare	middleeast	floor
Republican	0.154***	0.057***	0.155***	0.112***	0.098***	0.041***
	(0.147, 0.161)	(0.054,0.059)	(0.143,0.167)	(0.108,0.115)	(0.087,0.111)	(0.037,0.044)
Observations	7,686	24,790	2,092	29,134	2,153	36,717

*p<0.1; **p<0.05; ***p<0.01

Note: Bootstrapped 95% confidence intervals in parentheses.

Table J.8: Regression output for Figure D.3 in SM.

	immigration	economy	climatechange	healthcare	middleeast	floor
Republican	0.657	0.525**	0.802**	0.603**	0.737	0.702
	(0.534, 0.815)	(0.450,0.616)	(0.665,0.984)	(0.509,0.700)	(0.597,0.923)	(0.535,0.899)
Observations	17	26	12	21	10	13

*p<0.1; **p<0.05; ***p<0.01

Note: Bootstrapped 95% confidence intervals in parentheses.

Table J.9: Regression output for Figure D.4 in SM.

References

Caliskan, Aylin, Joanna J Bryson and Arvind Narayanan. 2017. "Semantics derived automatically from language corpora contain human-like biases." *Science* 356(6334):183–186.

Ethayarajh, Kawin, David Duvenaud and Graeme Hirst. 2019. In *Proceedings of the Association for Computational Linguistics (ACL)*. pp. 1696–1705.

Khodak, Mikhail, Nikunj Saunshi, Yingyu Liang, Tengyu Ma, Brandon M. Stewart and Sanjeev Arora. 2018. A la carte embedding: Cheap but effective induction of semantic feature vectors. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. Melbourne, Australia: Association for Computational Linguistics pp. 12–22.

Osnabrügge, Moritz, Sara B Hobolt and Toni Rodon. 2021. "Playing to the gallery: Emotive rhetoric in parliaments." *American Political Science Review* 115(3):885–899.

Rodman, Emma. 2020. "A timely intervention: Tracking the changing meanings of political concepts with word vectors." *Political Analysis* 28(1):87–111.

Rodriguez, Pedro L. and Arthur Spirling. 2022. "Word Embeddings: What Works, What Doesn't, and How to Tell the Difference for Applied Research." *The Journal of Politics* 84(1):101–115.

van Loon, Austin, Salvatore Giorgi, Robb Willer and Johannes Eichstaedt. 2022. Negative Associations in Word Embeddings Predict Anti-black Bias across Regions–but Only via

Name Frequency. In *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 16 pp. 1419–1424.

Warriner, Amy Beth, Victor Kuperman and Marc Brysbaert. 2013. "Norms of valence, arousal, and dominance for 13,915 English lemmas." *Behavior research methods* 45(4):1191–1207.