Supporting information for:

"Categorizing topics versus inferring attitudes: a theory and method for analyzing open-ended survey responses"

William Hobbs and Jon Green

Table of Contents

A Data

A.1 Text pre-processing

- Probes removed (e.g., the interviewer writing down that they asked “anything else?”) and terms with 2 or fewer characters removed (the default ‘stm’ package setting – this ensures that the two-letter probes are removed)

- Names of presidents and presidential candidates removed (most important problem responses)

- Non-English language responses were removed

- All most important problem mentions were combined into a single text response (for years where mentions were asked and/or recorded separately), since they all came from the same prompt and we were unable to cleanly split responses in years containing first/second/third answers in one

- All party likes/dislikes were analyzed together (e.g., dimensionality reduction methods were run on both likes and dislikes) and then averaged, since they had slightly different prompts

- Automated standardization with “stm” package (lowercase, snowball stopwords removed, including stopwords written without apostrophes)

- Training/test splits for the ACA attitudes analysis (KFF and Pew responses formed the training sets and ISCAP panel responses were not included in training)

- Panel responses from the ANES are included in overall training (but not the 2016 only training added later): to simplify the 3 analyses in the main paper – test-retest, hand label correspondence, over time changes – each of which might justifiably have different training-test splits

- ANES years are equally weighted using survey weights (i.e., the much larger 2016 wave, which includes a large online sample, does not count more than earlier waves with only face-to-face data), with the exception of the topic models (as the ‘stm’ software, to our knowledge, does not allow use of weights in training)

A.2 Questions and samples

A.2.1 “Could you tell me in your own words what is the main reason you have (a favorable/unfavorable) opinion of the health reform law?”

2009 question (Pew): “What would you say is the main reason you (favor/oppose) the health care proposals being discussed in Congress?”
Hand labels: from the Kaiser Family Foundation surveys (2010-2015).

Years:	2009 (Pew) 2010-2015 (KFF) Jan ’16, Oct ’16, and Oct ’18 (ISCAP)
Number of responses:	14,278
Number of hand labeled responses:	11,094
Panel responses:	2,770
Panel respondents:	1,094

Table A.1: ACA attitudes open-ended response sample sizes.

A.2.2 “Is there anything in particular that you (like/dislike) about the (Democratic/Republican) party? What is that?”

Hand labels: 1984 - 2004 (every two years 1984-1992, every four years 1996-2004).

Included years:	1984 - 2004 (every two years 1984-1992) 2008 - 2020
Excluded years:	none
Number of responses:	62,798
Number of hand labeled responses:	21,850
Panel responses: **Panel responses:**	514 (1992-1996, 1-4 per respondent), 5,543 (2016-2020)
Panel respondents:	193 (1992 to 1996); 2,053 (2016 to 2020)

Table A.2: Party likes/dislikes open-ended response sample sizes.

A.2.3 “What do you think is the most important problem facing this country today?”

Hand labels: 1984 - 2000 (every two years, other than 1994).

Included years:	1984, 1986, 1988, 1990, 1992, 1996, 1998, 2000, 2004, 2008, 2012, 2016
Excluded year(s):	2020 (pandemic)
Number of responses:	22,983 (and 7,214 in 2020)
Number of hand labeled responses:	11,776
Panel responses:	540 (’92-’96), 5,072 (’16-’20)
Panel respondents:	270 (’92-’96), 2,536 (’16-’20)

Table A.3: Most important problem open-ended response sample sizes.

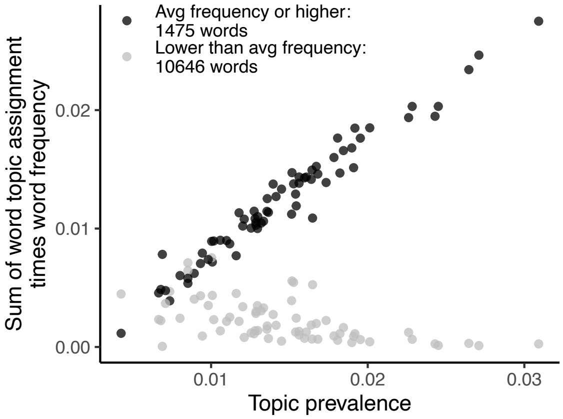


Figure B.1: This figure shows the contributions of common and rare words to topic prevalence in a topic model on the party likes and dislikes data (with k selected automatically using (Mimno and Lee 2014) as implemented in stm (Roberts et al. 2016)). Common words (here, those appearing more frequently than average) determine the prevalence – the frequency of more rare words is not associated with prevalence.

B Methods details

B.1 Topic models

- Uses “stm” package (Roberts et al. 2016)
- stm without covariates (a correlated topic model)
- $k = 2, 5, 10, 20, 30, 40, 50$ (in main paper)

In Figure B.1, we show that topic prevalence is strongly correlated with having a small number of frequent words in a topic.

B.2 Zero-shot labels and PCA

- Zero-shot classification of “This text is about ___.”
- Uses the BART language model (Lewis et al. 2020) fine-tuned on the MNLI corpus, as described in Yin et al. (2019) and implemented in the python package ‘ktrain’ (Maiya 2022).
- Labels were the top 1,000 words in the corpus
- Like the implied word method, PCA on matrix of square roots of probabilities

B.3 BERT embeddings and PCA

We observe no meaningful difference when using BERT embeddings in place of zero-shot classifications, and so these findings are not included in the main paper (they are retained only in SI figures).

- PCA on last layer of BERT sentence embeddings

B.4 Response distinctiveness

- BERT last layer sentence embeddings
- average embedding location for documents that contain a given word versus documents that do not
- response distinctiveness is the Euclidean distance between the contains word and does not contain word location averages
- these calculations and distances mirror embedding regression in Rodriguez et al. (2023), but where differences were instead calculated across groups for documents containing the same word

B.5 Supervised models

- trained on closed-ended ACA favorability (for the ACA attitudes analyses – favorable or not favorable) and trained on partisanship (for the ANES analyses – 7 point scale).
- ridge regression on document-term matrices, with lambda selected by cross-validation (as implemented in the R package 'glmnet' (Friedman et al. 2021))

B.6 Implied word method

This section present an unabridged version of the implied word method explanation contained in the main text, repeating text included there so that a reader does not have to go back and forth between the two.

Overall, the implied word method calculates a score that measures whether a document is 'about' a common word – whether or not the word was itself used – and then applying dimensionality reduction to summarize covariation in those 'implied word' scores. The goal is to estimate the extent to which one could substitute what the respondent happened to say with other statements without changing what they meant – and through this, infer what statements a respondent could have made consistent with the same general attitude.

More specifically, we compare a document-term matrix to a matrix that stands in for the respondent sampling distribution (the range of considerations to sample from) when a document is about an implied word. That matrix contains conditional distributions of co-ocurring words in all responses to the same or closely related prompt – i.e., across documents that contain the word ‘people’, what fraction of (unique) words in those documents was the word x.

To compare documents’ stated words to the implied words’ sampling distributions, we use Bhattacharyya coefficients, which measure overlap in probability distributions. We do so for every word in the corpus. That is, whether or not a document uses the word “people”, we still calculate whether the distribution of words in the document resembles the distribution of words for all other documents in the corpus that did use the word “people”.

Concretely, for a document i, stated word(s) j, and implied word k, the following calculation produces a document similarity score for word k in document i (a word that the document might be ‘about’):

$$m_{ik} = BC(d_i, g_k)_{ik} = \sum_{j=1}^{p} \sqrt{\frac{d_{ij}}{\sum_{j=1}^{p}(d_{ij})}} \sqrt{\frac{g_{jk}}{\sum_{j=1}^{p}(g_{jk})}}$$

where d_{ij} is an element in the original document-term matrix (whether word j was used in document i), g_{jk} is an element in the corpus conditional word co-occurrence matrix (approximately[7]: of respondents who used the word k, the fraction who also used the word j), and m_{ik} an element in the transformed document-term matrix (whether document i appears to be ‘about’ word k).

Below, we illustrate this process for a document that states: “they spend too much”. We compare this document to the conditional distributions of common words in the corpus, using the words people, issues, beliefs, candidates, help, and waste as these words (in our later analyses, we exclude stop words like ‘they’ and ‘the’). This calculation shows that, although this document does not explicitly use the word waste, the method identifies waste as the most likely ‘topic’ of the sentence.

Importantly, this approach will more heavily weight common words than rare words and do so across *all* of the comparisons. In the conditional distribution matrix, although each column is normalized to sum to 1, the rows are still strongly associated with word frequency.[8] To confirm the strong weighting toward frequent words, we illustrate in Figure B.2 that vectors for common words are more strongly correlated across the Bhattacharyya coefficient matrix and the original document-term matrix than more rare words.

This approach is not enough on its own to study attitudes. To better understand the broad associations of different common words across a corpus, we need to use some form of dimensionality reduction. This will provide a set of words that may be more recognizably symbolic as a group, *and* that more strongly vary across respondents.

We use singular value decomposition for that dimensionality reduction. Because this captures dominant sources of variation in the data, the singular vectors from this provide

[7] We do not zero out the diagonal of the word co-occurrence matrix. More precisely: across documents that contain the word ‘people’, what fraction of (unique) words in those documents was the word j.

[8] Note that the example table does not sum to 1 because it is a subset of the full matrix.

$$\sqrt{\frac{d_{ij}}{\Sigma_{j=1}^{p}(d_{ij})}} = \begin{array}{cccc} \textit{they} & \textit{spend} & \textit{too} & \textit{much} \\ \hline 0.5 & 0.5 & 0.5 & 0.5 \end{array}$$

$$\sqrt{\frac{g_{jk}}{\Sigma_{j=1}^{p}(g_{jk})}} = \begin{array}{l|cccccc} j's\downarrow\ k\text{'s} \rightarrow & \textit{people} & \textit{issues} & \textit{beliefs} & \textit{candidates} & \textit{help} & \textit{waste} \\ \hline \textit{they} & 0.17 & 0.16 & 0.15 & 0.16 & 0.18 & 0.16 \\ \textit{spend} & 0.03 & 0.03 & 0.02 & 0.02 & 0.03 & 0.08 \\ \textit{too} & 0.06 & 0.08 & 0.05 & 0.07 & 0.06 & 0.11 \\ \textit{much} & 0.06 & 0.06 & 0.06 & 0.07 & 0.06 & 0.11 \end{array}$$

$$\begin{array}{l|cccccc} & \textit{people} & \textit{issues} & \textit{beliefs} & \textit{candidates} & \textit{help} & \textit{waste} \\ \hline \boldsymbol{\textit{they spend too much}} & 0.16 & 0.16 & 0.14 & 0.16 & 0.17 & \mathbf{0.23} \\ \textit{they represent the middle class} & \mathbf{0.23} & 0.19 & 0.18 & 0.18 & \mathbf{0.24} & 0.17 \\ \textit{their stance on abortion} & 0.11 & \mathbf{0.19} & \mathbf{0.18} & 0.14 & 0.10 & 0.12 \end{array}$$

Table B.1: We illustrate calculations for the transformed document-term matrix. Each row of the transformed matrix is standardized (see text) prior to singular value decomposition. The leading dimension of this method will capture the number of words and use of more common words across documents, and the next will be the first substantive dimension.

the top candidate dimensions to correlate with different measures of attitudes (after the leading dimension, which captures only the number of words and how common they are), and for assessing stability over time.

Prior to applying SVD, we standardize the data: $\sqrt{\frac{m_{ik}}{\Sigma_{k=1}^{q} m_{ik}}}$. Since singular vectors correspond to the eigenvectors of $X^{\top}X$ and $XX^{\top}$, this standardization ensures that each respondent is weighted equally, and the square root here allows the first singular vector to more fully capture document length and word frequency, leaving subsequent dimensions to reflect more substantive variation. For data with weights, we can multiply the observations by the square root of the weight (and also use a weighted document-term matrix to create the conditional word distributions). Somewhat less important, though in keeping with our arguments on the importance of frequent words over rare words, we can further constrain the calculations to only q relatively common words (i.e., only calculating that documents are about common words rather than all words), with q substantially smaller than the total number of words p – for example, just the words that are used more than average. For our analyses, we restrict this to the number of words whose squared frequency is greater than the average squared frequency. We show in the Supplementary Tables and Figures section of the appendix that we see the same results for word frequencies simply greater than the

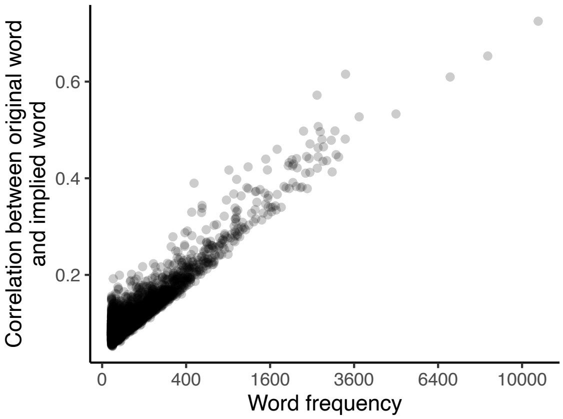


Figure B.2: This figure shows the correlation between words in the original document-term matrix and the 'implied' words in the transformed matrix. More rare words are not strongly correlated with their original use because they have been rescored to reflect the associations of the common words appearing in the same document.

average frequency. However, with that setting, the dimensions are sometimes highly correlated with each other, and have the potential to exaggerate the reliability of the findings (e.g., if all dimensions are moderately correlated with the first substantive dimension).

To score uncommon words on the same basis as common words – especially for documents that do not contain the common words used in the scaling – we multiply the transposed and standardized conditional word co-occurrence matrix by the right singular vectors, as if we had decomposed the (transposed and standardized) conditional word co-occurrence matrix rather than the implied word matrix. This has little effect on the scores for common words, while placing more rare words into the same scoring space.

Finally, we apply the scoring vectors to the original document-term matrix to produce document scores. Putting the scoring process together, with $\boldsymbol{D}$ representing the document-term matrix, $\boldsymbol{G}$ the standardized word co-occurrence matrix/sampling distribution matrix, and $\boldsymbol{V}$ the right singular vectors of singular value decomposition of $\boldsymbol{M}$, the transformed document-term/'implied word' matrix describe above, document scores are produced by the matrix multiplication $\boldsymbol{D}(\boldsymbol{G}^\top \boldsymbol{V})$ and then standardized so that each observation has a Euclidean norm of one. Word scores – for both common and rare words – are $\boldsymbol{G}^\top \boldsymbol{V}$.

When listing keywords, we multiply word scores by the square root of words' frequencies and then report the top words with the largest values on each side of the scale. This multiplication ensures that the keywords reflect the influence of common words on the scaling process, as illustrated in Figure B.2 – we treat common and rare words equally when assigning document scores.

C Implied word method: R code walk-through

Step 1. Calculate implied words

Our first step in the implied word method is to calculate the similarity between documents and common words – meaning, a word that is common in the provided set of open-ended responses.

For this, we calculate the similarity of **a)** the distribution of words *in a given document* and **b)** the distribution of words across all documents *that contain a given common word*.

The more the distribution of words in a document resembles the distribution of words across all documents (*in the prompt-specific corpus*) that contain a given focal word, then the more we say that document is 'about' / 'implies' that word – whether or not the focal word is itself used in the document. Note that in the calculation below similar use of frequent words will more strongly influence the 'implied word' similarity score than similar use of infrequent words (as we illustrated in the previous section).

1.1. Calculate co-occurrences of words from a document-term matrix. Rows of this matrix are documents (one row for each document) and columns are words (one column for each word). For the elements of this matrix, 1 indicates the presence of a word in an open-ended response and 0 its absence. In the R code, we use sparse matrices (and the packages `Matrix` and `RSpectra`) to speed up calculations – sparse matrices use empty values in place of 0's.

```r
cooccurrence_matrix <- Matrix::crossprod(document_term_matrix)
```

The diagonal of this matrix is the number of times that a word was used in the corpus and off-diagonals are the numbers of co-occurrences of (pairs of) words.

optionally, weight the document-term matrix prior to calculating the co-occurrence matrix:

```r
cooccurrence_matrix <- Matrix::crossprod(
  weight_matrix(document_term_matrix, w=weights)
)
```

1.2. Row-standardize the co-occurrence matrix to get the distribution of words (in each row of this matrix). These distributions represent the square root of probabilities used when calculating the Bhattacharyya coefficients.

`row_standardize_matrix()`, in effect, divides each row by its sum (for row-wise probabilities) and then applies an element-wise square root (for later input into a Bhattacharyya coefficient calculation). The function is written slightly differently than this explanation to speed up computation.

```r
standardized_cooccurrence_matrix <- row_standardize_matrix(cooccurrence_matrix)
```

$$\sqrt{\frac{g_{kj}}{\sum_{j=1}^{p}(g_{kj})}}$$

This step does not affect the final output when all elements in the document-term matrix are 1 or 0, as they are here.

```r
standardized_document_term_matrix <- row_standardize_matrix(document_term_matrix)
```

$$\sqrt{\frac{d_{ij}}{\sum_{j=1}^{p}(d_{ij})}}$$

1.3. Subset standardized co-occurrence matrix to common words:

First, find words above the frequency cutoff

```
word_counts <- colSums(document_term_matrix)
common_words <- word_counts^2 >= mean(word_counts^2)
# or word_counts >= mean(word_counts)
```

And then truncate the standardized co-occurrence matrix, leaving only (square roots of) the distributions of common words. Note that we truncate here to avoid excessive calculations – we could also have truncated our implied word matrix, truncating the implied word matrix to the top q implied/common words.

```
truncated_cooccurrence_matrix <- standardized_cooccurrence_matrix[common_words,]
```

1.4. Calculate Bhattacharyya coefficients – matrix multiply standardized document-term matrix and transpose of truncated co-occurrence matrix:

```
implied_word_matrix <- standardized_document_term_matrix %*% t(truncated_cooccurrence_matrix)
```

$$m_{ik} = BC(d_i, g_k)_{ik} = \sum_{j=1}^{p} \sqrt{\frac{d_{ij}}{\sum_{j=1}^{p}(d_{ij})}} \sqrt{\frac{g_{jk}}{\sum_{j=1}^{p}(g_{jk})}}$$

Step 2. Find dominant variation in implied words

The next step is to find dimensions in the standardized implied word matrix that explained the largest variance in that matrix. We are not only interested in common words – we are specifically interested in common words that strongly vary/co-vary across respondents.

For this, we use singular value decomposition.

2.1. Standardize the implied word matrix, so that documents influence the decomposition more equally (with unequal influences coming in through the optional weighting below):

```
standardized_implied_word_matrix <- row_standardize_matrix(implied_word_matrix)
```

2.2. Run singular value decomposition:

```
svds <- RSpectra::svds(standardized_implied_word_matrix, k=10)
```

optionally, weight matrix prior to decomposition:

```
svds <- RSpectra::svds(
  weight_matrix(
    standardized_implied_word_matrix,
    w = weights
  ),
  k = 10
)
```

Step 3. Score documents

After finding the dimensions that explain the largest variance in implied word usage, we score the original documents on those dimensions.

3.1. Extract word scores:

```
word_score_matrix <- svds$v
```

optionally (but recommended), if many respondents do not use common words in their responses, use right singular vectors to score all words based on their standardized co-occurrences with common words:

```
word_score_matrix <- standardized_cooccurrence_matrix[,common_words] %*%
  svds$v
```

There is no transpose in this step as in the main text only because we aligned our description there with column-oriented notation – note the transpose and switch in notation from steps 1.2 to 1.4 above – and we wrote this R code to consistently run row-wise standardizations.

3.2. Apply word scores to documents:

```
scored_documents <- document_term_matrix %*% word_score_matrix
```

3.3. Standardize scored documents to have a Euclidean norm of 1 (similar to document scores summing to 1 in a topic model):

```
standardized_scored_documents <- euc_row_standardize_matrix(scored_documents)
```

We use the `standardized_score_documents` matrix for our analyses. The first column of this matrix, which we name `X0`, is closely related to word frequency. The second column, `X1`, is a substantive dimensions – e.g., the "issues/positions/ideology" versus "groups/performance/candidates" dimension in the partisan likes and dislikes analysis.

Step 4. Get keywords

Last, we can extract keywords using a combination of a word's polarity on a dimension and its frequency. We number dimensions starting at 0, since dimension 0 captures word frequency. Later dimensions capture variation in word use beyond mere frequency.

This code assumes that matrix columns are named (with their corresponding words).

```
vocab <- colnames(document_term_matrix)
# distance from word score mean times square root of word frequency
dimension_number <- 1
frequency_weighted_and_centered_word_scores <- scale(
  word_score_matrix[,dimension_number+1], center = TRUE, scale = FALSE
) * # polarity
  sqrt(word_counts) * # frequency
  common_words # common word subset (from Step 1)

vocab[order(frequency_weighted_and_centered_word_scores, decreasing = T)][1:15]
vocab[order(frequency_weighted_and_centered_word_scores, decreasing = F)][1:15]
```

Standardization and weighting functions

We have reduced these functions to their bare-bones functionality, and explain their function in comments because they are written for fast sparse matrix calculations with the `Matrix` R package.

```r
row_standardize_matrix <- function(m) {
  #### this function standardizes the rows of a matrix
  #### these calculations are equivalent to standardizing each row to sum to 1
  #### and then taking the square root of each value/probability
  if (class(m)=="dsCMatrix") {
    m <- as(m, "dgCMatrix")
  }
  ## transpose so that later operations are on columns
  ## (which are rows of the original matrix)
  m <- Matrix::t(m)
  ##
  row_norm <- sqrt(Matrix::colSums(m))
  m@x <- sqrt(m@x) /
    rep.int(row_norm, diff(m@p))
  ## transpose back to original format
  m <- Matrix::t(m)
  return(m)
}
```

```r
euc_row_standardize_matrix <- function(m) {
  #### this function standardizes the rows of a matrix
  #### so that each row has a Euclidean norm of 1
  if (class(m)=="dgeMatrix") {
    m <- as(m, "dgCMatrix")
  }
  ## transpose so that later operations are on columns
  ## (which are rows of the original matrix)
  m <- Matrix::t(m)
  ##
  row_norm <- sqrt(Matrix::colSums(m^2))
  m@x <- m@x /
    rep.int(row_norm, diff(m@p))
  ## transpose back to original format
  m <- Matrix::t(m)
  return(m)
}
```

```r
weight_matrix <- function(m, w) {
  #### this function multiplies each row
  #### by the square root of the observation weight
  ##
  m <- Matrix::t(m)
  ##
  m@x <- m@x *
    rep.int(sqrt(w), diff(m@p))
  ##
  m <- Matrix::t(m)
  return(m)
}
```

Putting all scoring steps together

```r
# input: document_term_matrix -- rows are document, columns are words
# 1 or 0 for occurrence of word in document
# use sparse representation for computational efficiency
library(Matrix)
library(RSpectra)

# output matrix: word co-occurrences
cooccurrence_matrix <- Matrix::crossprod(
  weight_matrix(document_term_matrix, w=weights)
)

# output matrix: word distributions, square roots of probabilities
standardized_cooccurrence_matrix <- row_standardize_matrix(cooccurrence_matrix)
standardized_document_term_matrix <- row_standardize_matrix(document_term_matrix)

# output matrix: square roots of probabilities, truncated to common words
word_counts <- colSums(document_term_matrix)
common_words <- word_counts^2 >= mean(word_counts^2)
truncated_cooccurrence_matrix <- standardized_cooccurrence_matrix[common_words,]

# output matrix: implied words
implied_word_matrix <- standardized_document_term_matrix %*%
  t(truncated_cooccurrence_matrix)
standardized_implied_word_matrix <- row_standardize_matrix(implied_word_matrix)

# SVD
svds <- RSpectra::svds(
  weight_matrix(
    standardized_implied_word_matrix,
    w = weights
  ),
  k = 10
)

# output matrix: word scores
word_score_matrix <- standardized_cooccurrence_matrix[,common_words] %*%
  svds$v

# output matrix: scored documents
scored_documents <- document_term_matrix %*% word_score_matrix
standardized_scored_documents <- euc_row_standardize_matrix(scored_documents)
```

Step 5. Adding embeddings (optional)

We can use implied word document scores and embeddings to create an embedded version of the implied word method. The logic of this process is much like using embeddings for the term "positive" minus embeddings for the term "negative" as a zero-shot sentiment classifier (see, for example, a similar approach here: https://platform.openai.com/docs/guides/embeddings/use-cases). Here, we use the positive and negative ends of the implied word document scores, after centering, to find a contrast embedding.

5.1: multiply document implied word scores by embeddings and average to get implied word embedding. The example below calculates this embedding for only dimension 1 of the implied word method (numbering starts at 0, since 0 is a frequency dimension).

```r
# inputs:
# documents scored with implied word method (documents on rows, dimensions in columns)
# document embeddings (documents on rows, embedding dimensions in columns)
# label embeddings (labels on rows, embedding dimensions in columns)

library(text2vec)

# output: implied word embedding for dimension 1 (one row, n embedding dimensions columns)
implied_word_embedding_1 <- colMeans(
  (scored_documents[,2] - mean(scored_documents[,2])) * document_embeddings
)
```

optionally, subtract the weighted mean and calculate weighted column-wise means:

```r
implied_word_embedding_1 <- apply(
  (scored_documents[,2] - weighted.mean(scored_documents[,2], w=weights)) *
    document_embeddings,
  2,
  weighted.mean,
  w = weights
)
```

5.2: calculate the cosine similarity of each document with the implied word embedding:

```r
# output: document-level implied word embeddings for dimension 1
# (documents in rows, one column for embedded implied word score)
scored_documents_1 <- text2vec::sim2(
  document_embeddings, implied_word_embedding_1,
  method = "cosine"
)
```

5.3: calculate top labels for poles of implied word dimension using cosine similarity: In the main paper, top labels are the labels with the 100 highest and 100 lowest cosine similarities.

```r
# input: label embeddings (labels in rows, embedding dimensions in columns)

# output: label scores (one row, n labels columns)
scored_labels_1 <- text2vec::sim2(
  implied_word_embedding_1, label_embeddings,
  method = "cosine"
)
```

Collected guidance for using the implied word method

This user guide collects guidance provided in the main text of the article into one convenient location.

Users of the implied word method can install the `impliedWords` package from `https://github.com/wilryh/impliedWords`, and use that R package to run the method. The github site will also host this user guide (/ an updated version of it).

Before using the method, open-ended survey data must be processed into a document-term matrix format. See the R package for an example on how to do this using the `stm` package.

Hyperparameters

The implied word method has few hyperparameters for users to choose from, and we recommend that users mostly rely on the settings used for the main analysis in this paper.

Those settings are:

- Setting ‘common words’ equal to the words whose squared frequency is greater than the average squared frequency
    - this can alternatively be set to frequency greater than average frequency, if a user thinks that important words may have been left below the cutoff
    - however, we have rarely observed meaningful differences when changing this setting – and any differences may merit investigation (e.g., the introduction of outliers when lowering the cutoff)
- Projecting rare words – meaning that all words will be scored by the method, rather than just the common words
    - this can alternatively be set to score only rare words, if a user would like to assess sensitivity to scoring only common words

Pitfalls

After running the implied word method, users should assess whether there are large outliers influencing the output.

Outliers

Outliers can have a large influence on the scored dimensions. We have encountered a few main scenarios where this can happen:

- the corpus includes responses in different languages
    - for example, if a corpus contains both English and Spanish – but primarily English – then the Spanish words will tend to be very large outliers and have a substantial influence on the scored dimensions

- respondents have copied identical text from the web into their response
    - for example, many respondents copying parts of their answers from the same Wikipedia page will tend to lead to large clusters of outlying word scores
- insufficient data cleaning has left clusters of responses that contain the same text indicating “no response” – for example, “Respondent did not answer”

To check for outliers, we recommend plotting the word scores using the `plot_keywords` function in the `impliedWords` package. When there are outliers, this plot will show words that are very far from the rest of the words (typically in the 1st and 2nd dimensions) and/or the majority of word scores in only half or a corner of the plot (with the outliers being very small or below the function’s word frequency cutoff).

Corpus ‘context size’

The context covered in some corpora may be too large for our method to work well – for example, when what is contextually common for one subset of the data is not contextually common for the other. For this, we can split the data on some variable that may drive overly large context size (and a contrast that we know we do not want the method to identify – e.g., pandemic versus non-pandemic years) and assess the correlation in word frequencies across that contrast. The `impliedWords` package will soon implement this check.

Interpretation and in-depth validation

We recommend an in-depth validation of the implied word method’s output when using it in research. This validation can focus primarily on an accurate and transparent interpretation of the output.

Interpreting dimensions and keywords in context

To interpret dimensions, we recommend that users plot word scores and list keywords, as well as read a sample of documents by their associated document scores. A sample of documents by document scores should also be provided in articles that use the method (see, for example, the tables in SI Section K of this article).

Users can further incorporate the generative AI and embedding descriptive approach we use in the main text of this paper (with full details in the SI). This labeling can be helpful prior to more time-intensive and costly hand labeling, or in combination with it. Without pre-existing hand labels, it may be challenging to efficiently describe the contents of a dimension to the reader of an article who may not have access to many of the texts to read themselves, since our method relies on context-specific and potentially symbolic meanings of words. Lists of keywords may not always be informative on their own, and information contained in lists of documents may be difficult to efficiently convey to readers.

In validating interpretations, we recommend users assess associations with existing hand labels or create a set of hand labels that can more or less reproduce the output of the implied word method. Users can refer to research on best practices for coding potentially complex constructs from, for example, Benoit et al. (2016) and Tanweer et al. (2021). Here, we caution that coders may need to be given appropriate context to code texts appropriately, and that researchers will likely need to critically assess crowd worker performance in context – potentially iterating on provided context and instructions.

Last, users should assess dimensions' associations with metadata covariates (e.g., year, gender, education, party). Dimensions can be associated with a covariate and still be valid (some constructs *should* be associated with covariates like party and education). However, dimensions are not guaranteed to conform to researcher expectations. it is possible for the primary variation in an open-ended response to be communication style, just as a closed-ended responses can be dominated by social desirability or acquiescence bias. Associations with covariates can provide useful information for accurately and transparently interpreting dimensions.

Hand labeling

The implied word method is useful primarily for uncovering dimensions in text that tend to be highly stable over time and so more likely to represent stable attitudes. We anticipate that the vast majority of useful constructs recovered by the implied word dimension will be codeable by hand, and perceptible to a human reader (*after* the dimension has been identified).

Because of this, we recommend that researchers create a codebook to reproduce and validate the automated output of the implied word method whenever possible. Researchers can code the dimension directly or indirectly (through, for example, coding complex constructs using combinations of simpler and easier to code features in text). As also mentioned above, coders may need to be given appropriate context to code texts appropriately, and researchers will likely need to critically assess crowd worker performance in context – potentially iterating on provided context and instructions.

The process of creating and implementing the codebook should clarify the interpretation of dimensions, and allow for replication of findings across different measurement approaches (with potentially different sources of error).

Use of multiple open-ended responses

We can use multiple open-ended questions to assess possible style effects. If our first dimensions represent communication style only, then we might observe strong correlations over time between different questions. In our article, this test was not definitive – ideally, we would have a method that would be able to capture shared political content across multiple, potentially related political questions. Completely unrelated questions or a political versus non-political question would be preferable for this evaluation.

D Twitter analysis

D.1 Word frequency and month-to-month correlations in user word re-use

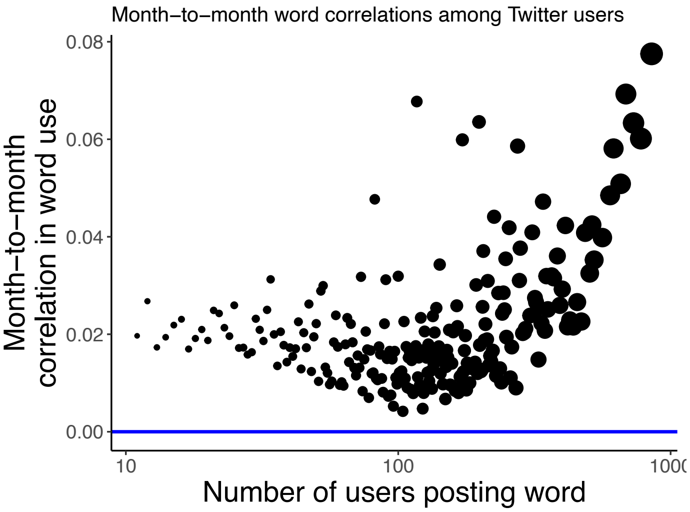


Figure D.1: *Correlation in word use over time in social media posts.* The blue line in this figure is a horizontal line at correlation equal to 0. Each point represents a group of words, excluding stop words from the English language 'snowball' stop word list. Words are grouped by word frequency, and we plot the average correlation within each bin. This data is drawn from a large sample of Twitter users who have been linked to voter records and whose tweets have been continuously collected since 2017 (Hughes et al. 2021). The correlations analyzed here cover the period January 2019 through June 2022 (given a tweet collection issue starting in July 2022 that is now being resolved) and, for computational reasons, a random 10% sample of the overall Twitter panel (approximately 100 thousand users).

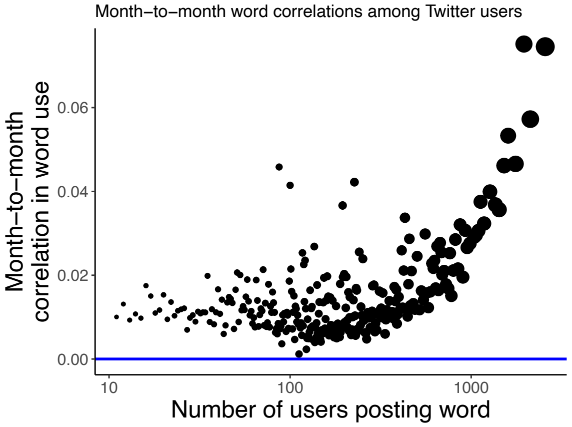


Figure D.2: *Correlation in word use over time in social media posts.* This figure repeats Figure D.1, limiting the sample to tweets containing 'political' keywords in the following categories (following keywords and categories in Green (2023)): class, climate, conservative, democrat, far-left, far-right, gender, guns, health, immigration, lgbt, liberal, progressive, race, reproductive health, republican, tax and spending policy, trade. Keywords are excluded from the correlation analyses.

D1 (-)	(+) D1	D2 (-)	(+) D2
health	racist	gop	gay
care	white	democrats	racism
climate	republican	republican	trans
via	republicans	republicans	racist
new	gop	trump	black
change	party	election	people
medicare	democrats	senate	white
crisis	gay	biden	women
join	shit	house	men
u.s	like	party	like
tax	democrat	democrat	community
medicaid	trump	vote	love
access	ass	via	can
workers	man	president	feminist
insurance	racists	impeachment	i'm

Table D.1: Implied word method keywords – 'political' content only. Keywords categories used in filtering: class, climate, conservative, democrat, far-left, far-right, gender, guns, health, immigration, lgbt, liberal, progressive, race, reproductive health, republican, tax and spending policy, trade. In addition to removing non-political content, this filtering also removes highly repetitive spam content (e.g. enter-to-win sweepstakes posts).

E Supplementary Tables and Figures

E.1 ACA attitudes: keywords, panel correlations, and hand label multiple R's

E.1.1 ACA attitudes: keywords

Implied word method

D1 (-)	(+) D1	D2 (-)	(+) D2
people	government	people	care
insurance	much	going	health
health	involved	lot	government
everyone	going	pay	access
coverage	cost	know	believe
afford	money	help	needs
access	everything	insurance	involved
conditions	want	money	everyone
helps	control	helps	affordable
affordable	run	just	everybody

Table E.1: Implied word method: top 2 substantive dimension keywords (ACA attitudes). Note that the first dimension of this method reflects word frequency, and we label it dimension 0.

Zero-shot PC's

D1 (-)	(+) D1	D2 (-)	(+) D2
helping	screwed	money	doesnt
positive	terrible	financial	without
helps	sucks	pay	dont
beneficial	failure	economic	illness
helpful	oppose	prices	opinion
helped	bad	economically	otherwise
good	poorly	paying	illnesses
help	opposed	budget	freedom
right	disagree	price	patient
providing	unfair	dollars	less

Table E.2: Zero-shot method: top 2 substantive dimension keywords (ACA attitudes)

E.1.2 ACA attitudes: panel correlations and hand label multiple R's

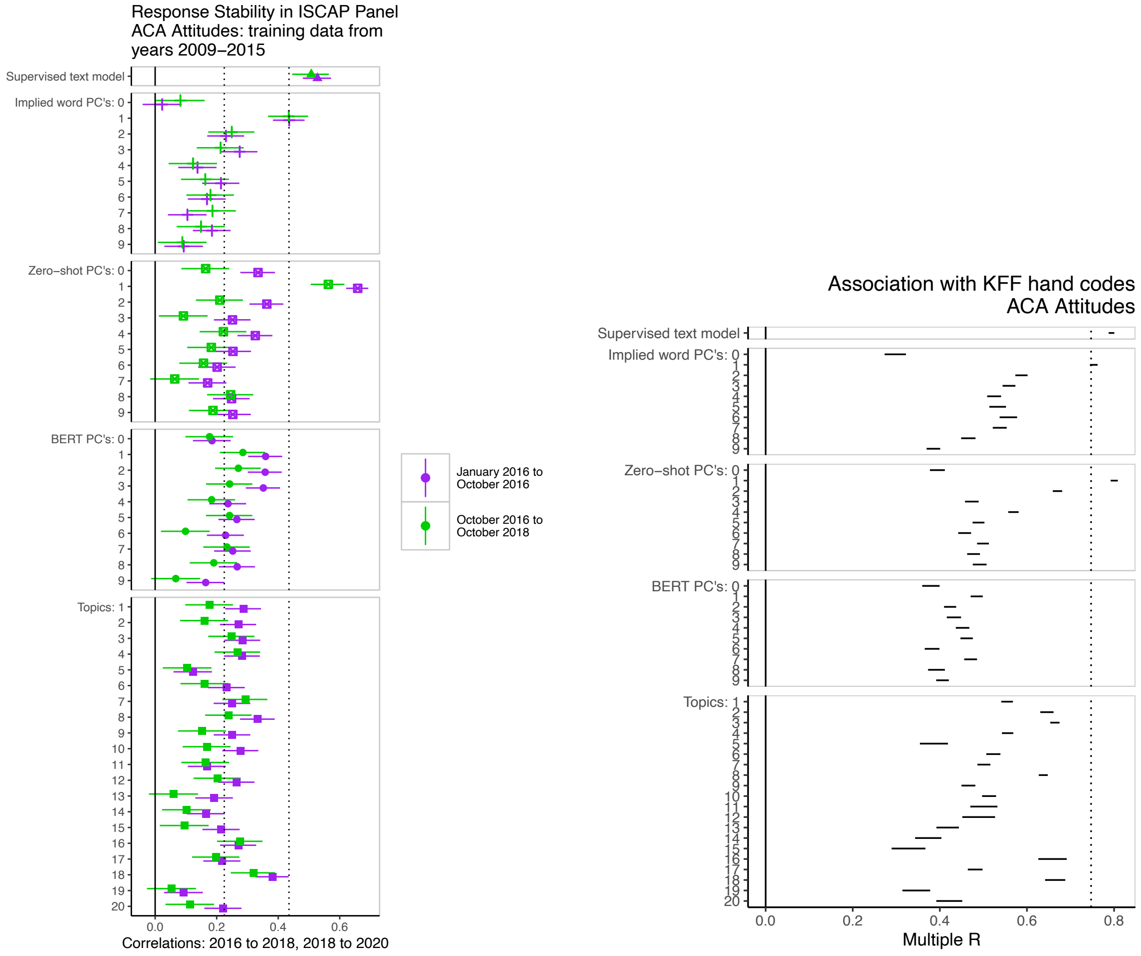


Figure E.1: The hand label multiple R analysis is limited to labels occurring at least 10 times and across 2 waves (this is lower than in the ANES analysis because there were fewer labels occurring in many waves).

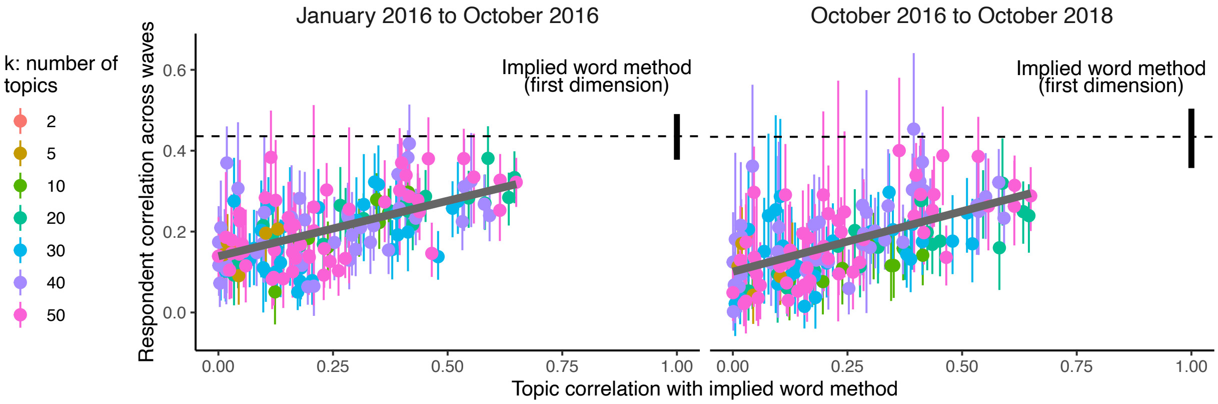


Figure E.2: Test-retest correlation and correlation with the first dimension of the implied word method for topic models across multiple settings of k (the number of topics): Affordable Care Act responses.

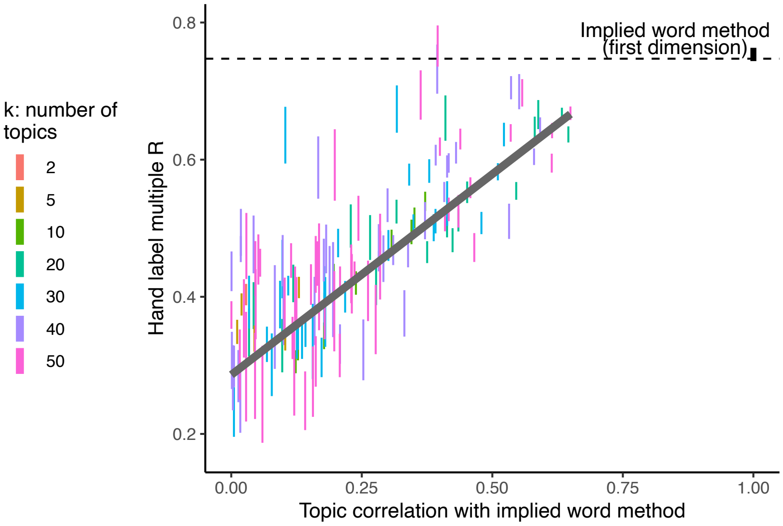


Figure E.3: Hand label multiple R's and correlation with the first dimension of the implied word method for topic models across multiple settings of k (the number of topics): Affordable Care Act responses.

E.1.3 ACA attitudes: alternate ‘common word’ cutoff

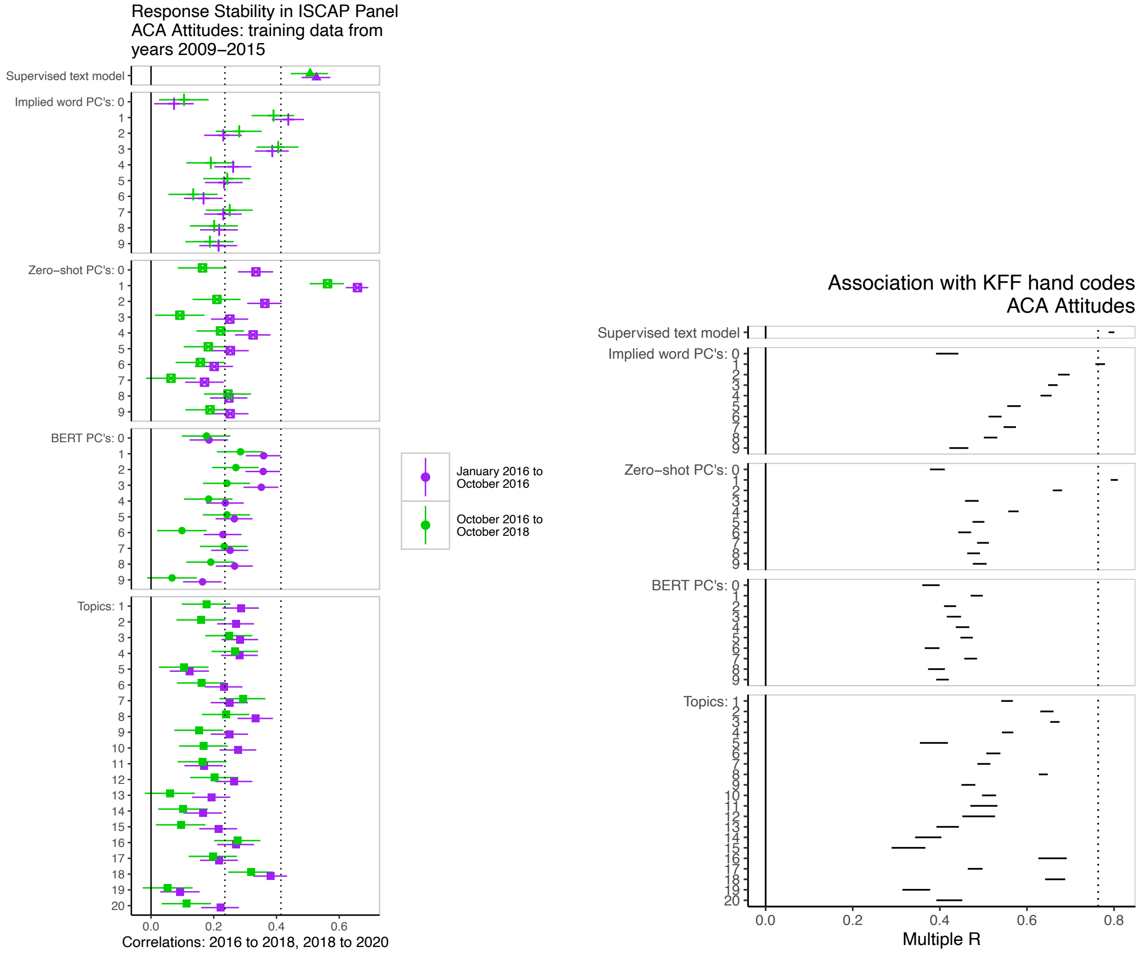


Figure E.4: This figure repeats the findings in Figure E.1 for ‘common words’ that have a frequency greater than the average frequency of words.

E.2 Party likes/dislikes: keywords and issue preferences on first dimension of implied word method

E.2.1 Party likes/dislikes: keywords

Implied word method

D1 (-)	(+) D1	D2 (-)	(+) D2
abortion	people	class	trump
rights	rich	middle	president
stance	poor	rich	party
gun	class	poor	candidate
pro	working	people	together
views	get	lower	parties
issues	help	tax	right
conservative	always	help	good
marriage	man	social	vote
gay	middle	working	republican

Table E.3: Implied word method: top 2 dimension keywords (party likes/dislikes). Note that the first dimension of this method reflects word frequency, and we label it dimension 0.

Zero-shot PC's

D1 (-)	(+) D1	D2 (-)	(+) D2
helping	negative	personal	especially
supportive	dislike	service	alot
positive	bad	lack	something
good	wrong	economically	among
right	opposed	working	strongly
helps	disagree	economy	appears
inclusive	dishonest	families	particularly
encourage	blame	wage	things
help	anti	fiscally	regarding
supporting	extreme	feeling	minded

Table E.4: Zero-shot method: top 2 substantive dimension keywords (party likes/dislikes).

E.2.2 Party likes/dislikes: issue preferences on first dimension of implied word method

Figure E.5 shows that the first dimension of the implied word method, people versus issues, is strongly associated with policy stances but not aligning with partisan stances on those issues. In Figure E.6, we show that some of this pattern is due to non-linearity. Most people who discuss issues and stances are more liberal on abortion, but those specifically mentioning abortion (and at the most extreme end of the dimension) are more conservative on the issue.

Issues here were chosen because they were included on a large number of ANES waves.

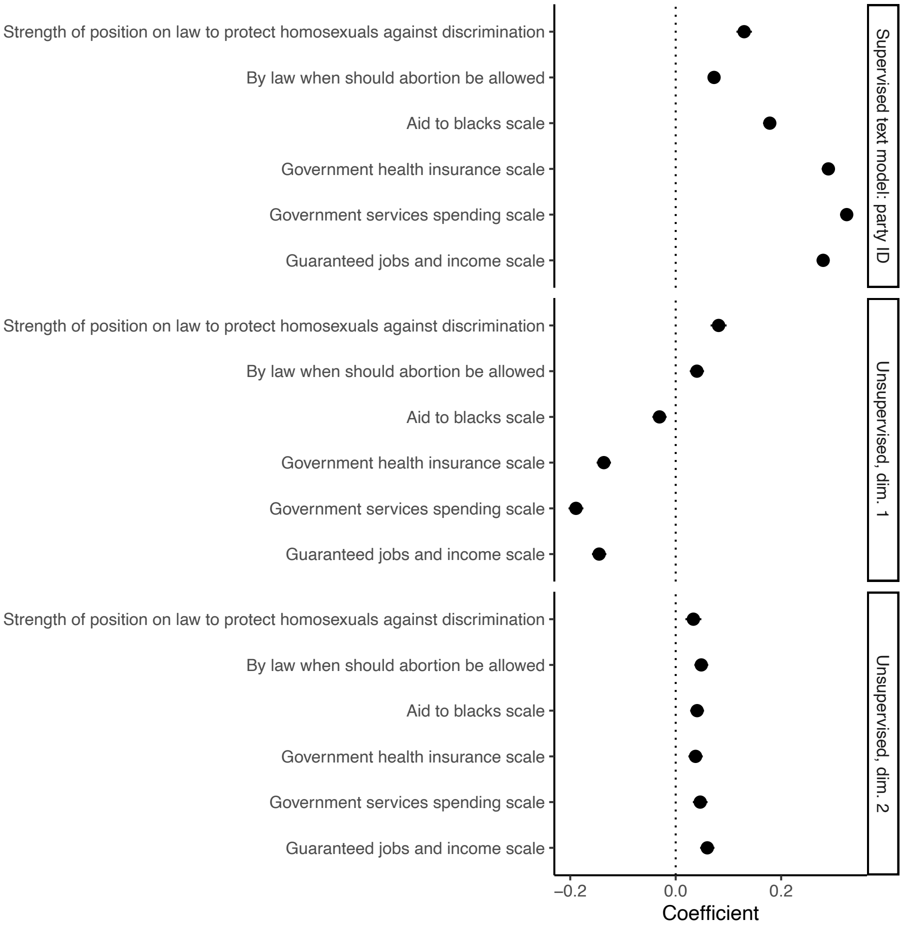


Figure E.5: Association between unsupervised implied word scores and policy stances compared to supervised model of party ID. All policy stances are coded so that higher values on the scale are more conservative.

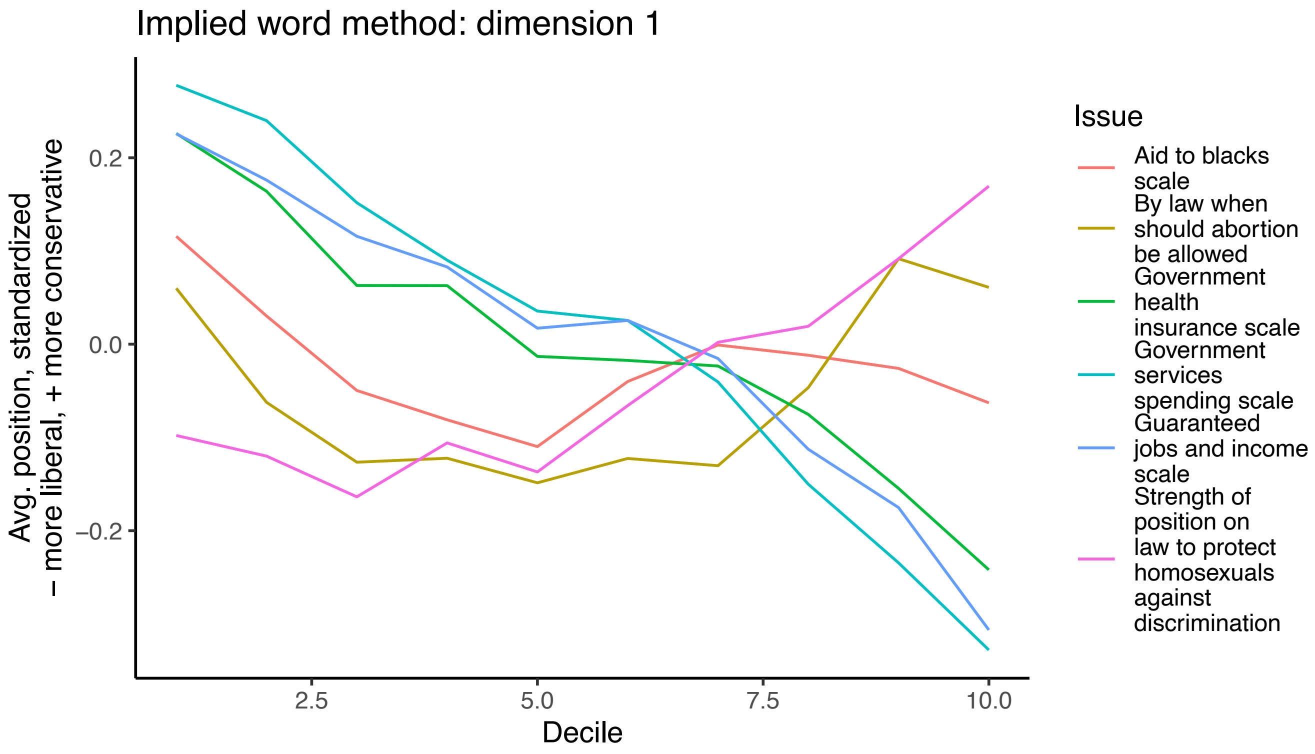


Figure E.6: This figure shows non-linearity in some issue preferences in the first dimension of the implied word method. For example, respondents tend to be more conservative on ‘By law, when should abortion be allowed’ when they either specifically mention abortion or talk generally about groups when answering what they like or dislike about the parties.

E.2.3 Party likes/dislikes: alternate ‘common word’ cutoff

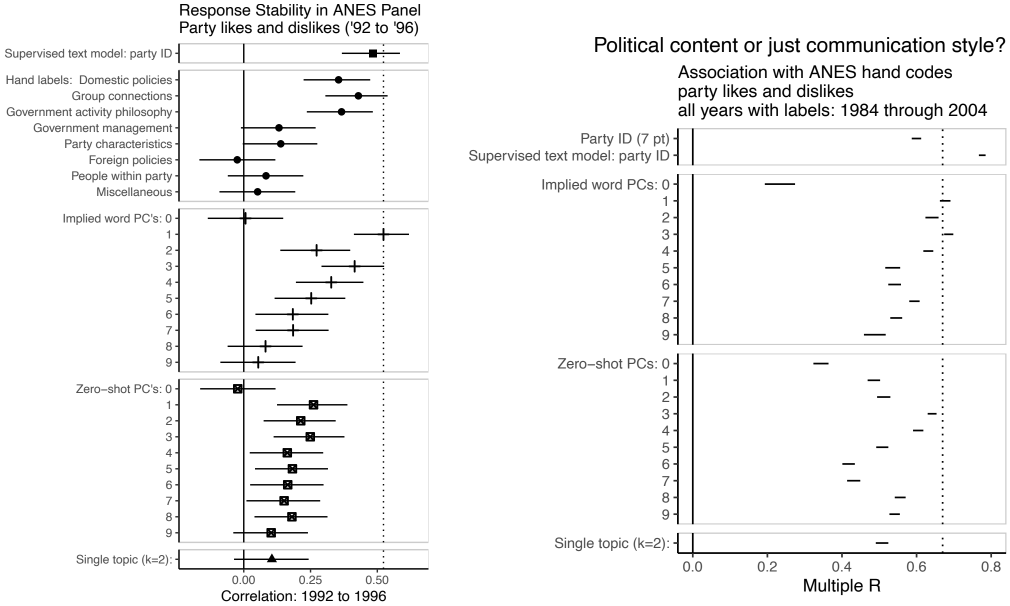


Figure E.7: This figure repeats the findings in Figures 3 and 4 for ‘common words’ that have a frequency greater than the average frequency of words. This specification was not used in the main results because many of the dimensions were highly correlated with the top substantive dimensions, potentially exaggerating reliability of lower dimensions.

E.3 Party likes/dislikes: illustration of coherence vs stability in topic models

To illustrate our point that topic models, as used in current practice, do not reliably produce categories with high test-retest reliability (and so more likely to reflect attitudes), we show below the correlations over time for hand selected topics that we perceived to have high coherence. These topics seemed to have a relatively cohesive set of keywords.

For this purpose, we ran a correlated topic model with the "stm" package (Roberts et al. 2016) as in our main analyses, but allowing the software package to select the number of topics automatically (following (Mimno and Lee 2014)). We came away with a model containing 69 topics. From there, we looked at each topic's keywords as selected by the high frequency and exclusivity value (Bischof and Airoldi 2012), using the package default of 0.5. We then selected a small number of topics that seemed to us to have the highest coherence, meaning that we would tend to group these words together (when thinking about politics, though not necessarily in the context of this question).

Last, we studied the 2016-2020 correlation in these topics. We focused on 2016-2020 because we have far more data than for 1992-1996, we need more data to study relatively sparse topics (from a model with 69 topics), and also because the topic models, in not accepting weights like our implied word method, may have better captured variation in responses during that period in which there were many more open-ended responses from the online sample.

The findings show that these topics widely varied in their correlations over time. Figure E.8 on the next page displays these correlations and Figure E.9 on the page after shows that these correlations are not driven by large shifts in *overall* topic prevalences from 2016 to 2020. To us (though of course we can begin to rationalize the results after seeing them), it was not obvious which of these clusters would be most strongly correlated over time when we only chose them based on keywords – other than our awareness of which ones seemed to most closely resemble the output of the implied word method.

Broadly, our point here is that 1) topic models produce many topical categories that are not strongly correlated over time, 2) we need panel data to understand the stability of responses, 3) the coherence of topic keywords can tell us little about the expected stability of a response, or an underlying, stable attitude, and 4) correlation with our implied word method (as we demonstrate in Figure 3 tends to be a better indicator of response stability than coherence.

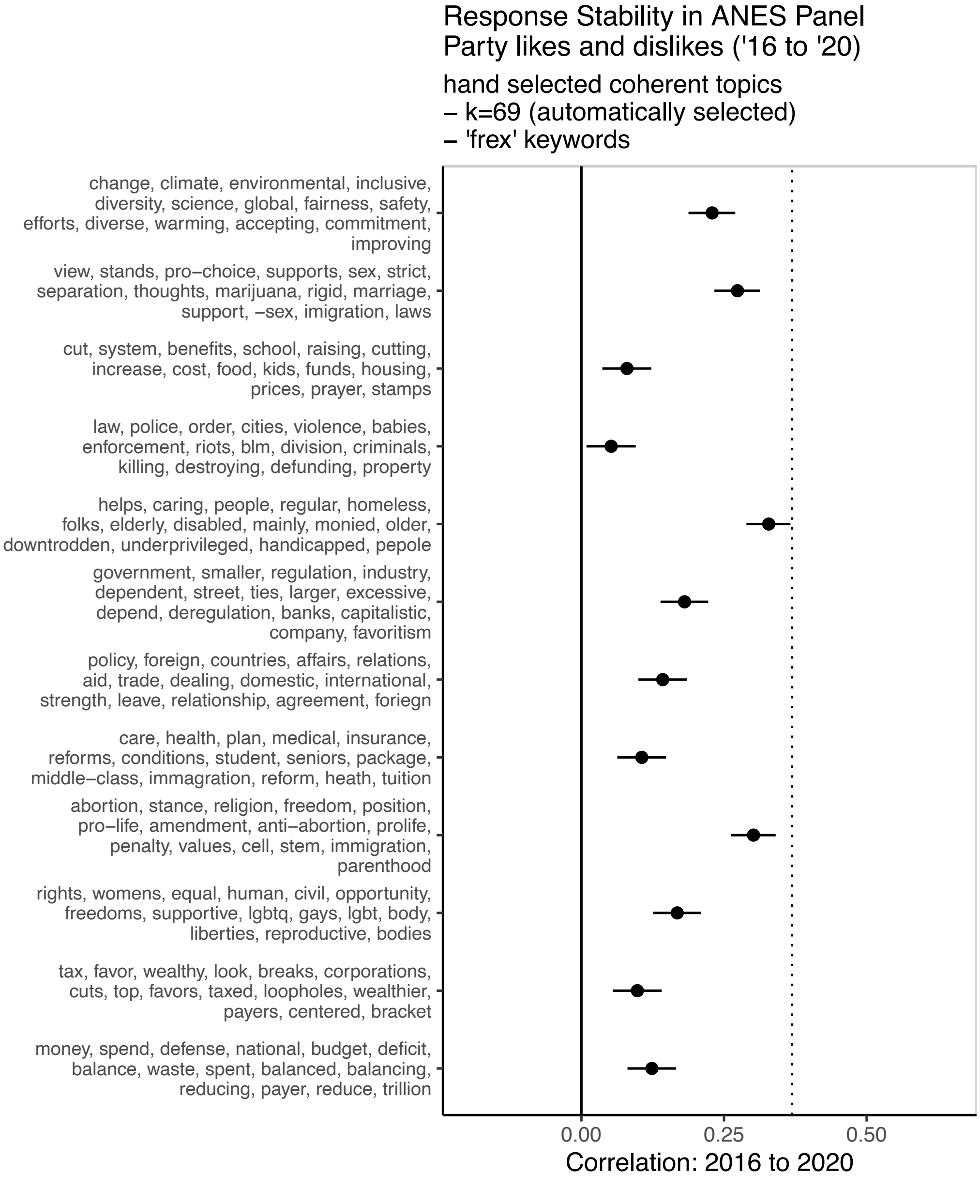


Figure E.8: 2016-2020 test-retest reliability of coherent topics from the party likes/dislikes data, ordered by topic number.

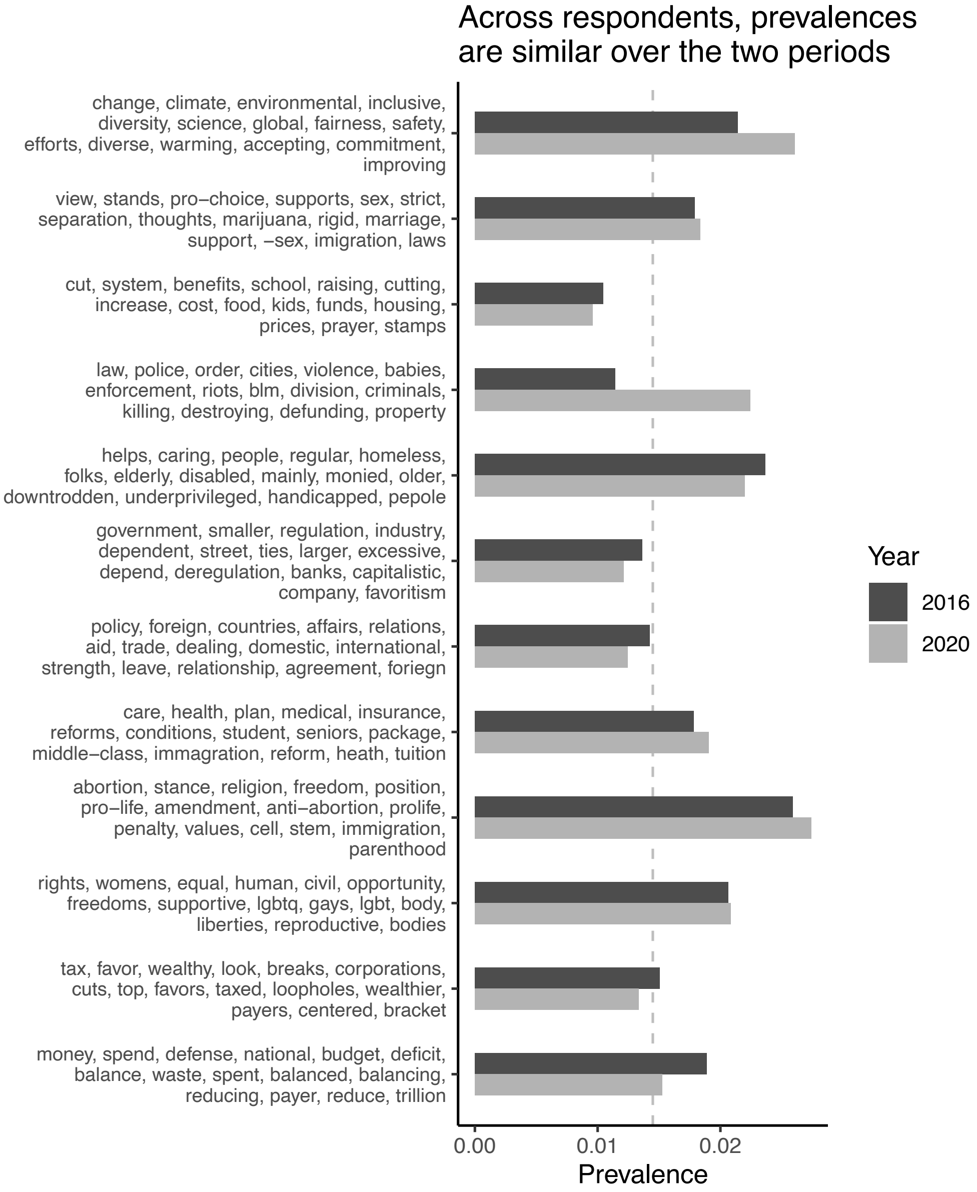


Figure E.9: 2016-2020 prevalences of coherent topics from the party likes/dislikes data. The vertical gray line indicates the value $\frac{1}{69}$.

E.4 Party likes/dislikes: response distinctiveness

We contrast the common word approach with an approach that emphasizes response distinctiveness. By response distinctiveness, we mean that responses could be categorized primarily by how different they seem compared to other responses in a corpus. For this, we use BERT sentence embeddings (Devlin et al. 2018) and, with linear regression, the difference between the average embedding location for documents that contain a given word versus the average for documents that do not. Response distinctiveness is the Euclidean distance between those averages. This follows the approach in embedding regression for studying differences in language use across groups (Rodriguez et al. 2023).

The left panel of E.10 shows that the most frequently-used words are among the likeliest to be re-used. These may reflect 'issue publics' (Krosnick 1990) or 'easy issues' (Carmines and Stimson 1980; Abramowitz 1995). The right panel of Figure E.10 shows the association between response distinctiveness for sentences containing a word, and that word's 2016-2020 correlation. Here, by contrast, we do not observe any relationship between response distinctiveness and word re-use. While common words alone are far from perfect indicators of stable attitudes, they are more informative than response distinctiveness.

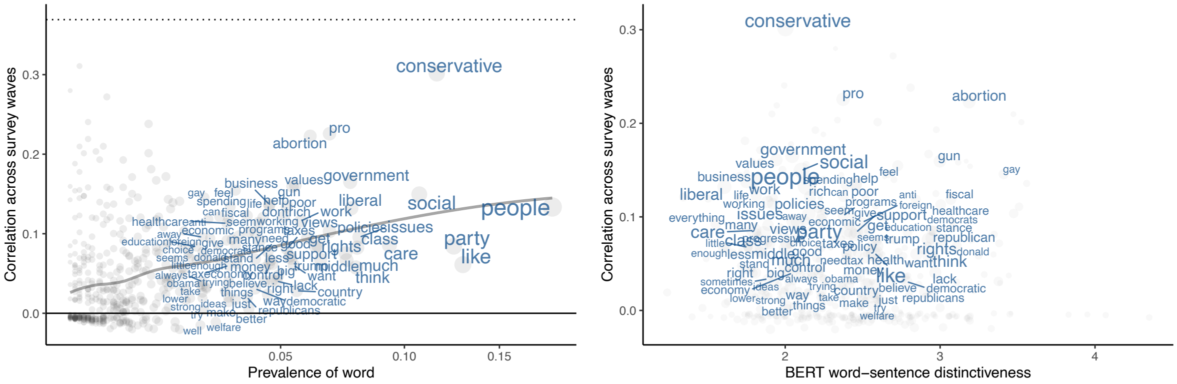


Figure E.10: Response distinctiveness versus word frequency.

E.5 Most important problem: keywords

E.5.1 Most important problem:: keywords

Implied word method

D1 (-)	(+) D1	D2 (-)	(+) D2
people	economy	war	health
children	deficit	nuclear	education
kids	war	arms	lack
get	budget	going	healthcare
school	terrorism	russia	care
schools	foreign	countries	racism
just	unemployment	get	immigration
work	east	now	crime
can	relations	reagan	affordable
pay	nuclear	know	insurance

Table E.5: Implied word method: top 2 dimension keywords (most important problem). Note that the first dimension of this method reflects word frequency, and we label it dimension 0.

Zero-shot PC's

D1 (-)	(+) D1	D2 (-)	(+) D2
support	destruction	economics	moral
helping	terrible	economic	abuse
improved	destroying	economy	attitude
improve	bad	supporting	morality
help	threats	improving	corrupt
improving	destroy	helping	opinion
provide	hurting	improve	conflict
assistance	crisis	important	unrest
benefits	trouble	worked	divisiveness
supporting	threat	together	mind

Table E.6: Zero-shot method: top 2 substantive dimension keywords (most important problem)

E.5.2 Most important problem: alternate ‘common word’ cutoff

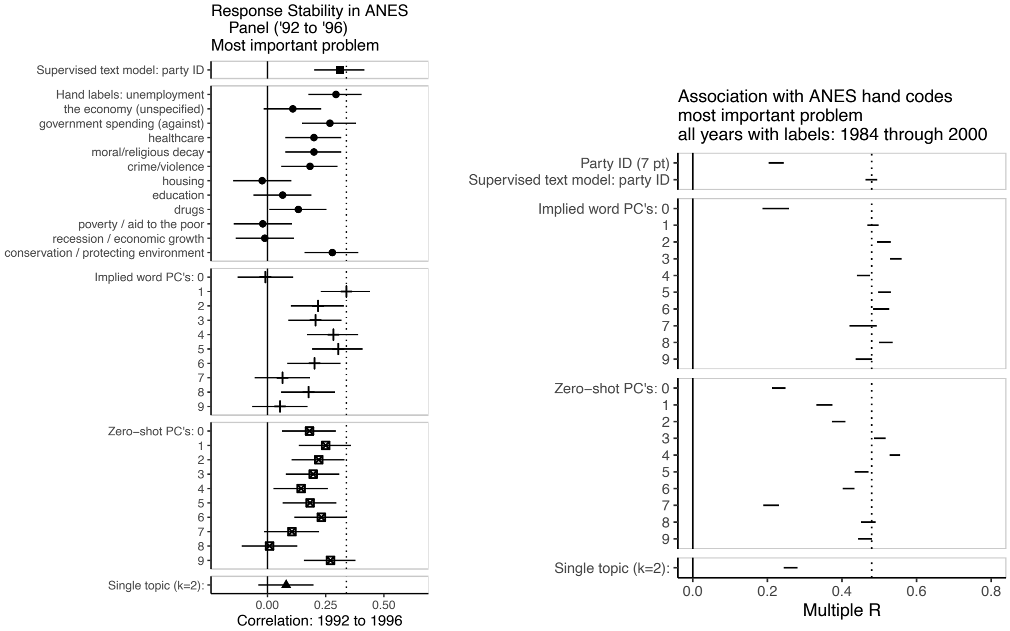


Figure E.11: This figure repeats the findings in Figures 3 and 4 for ‘common words’ that have a frequency greater than the average frequency of words.

F “Common” words?

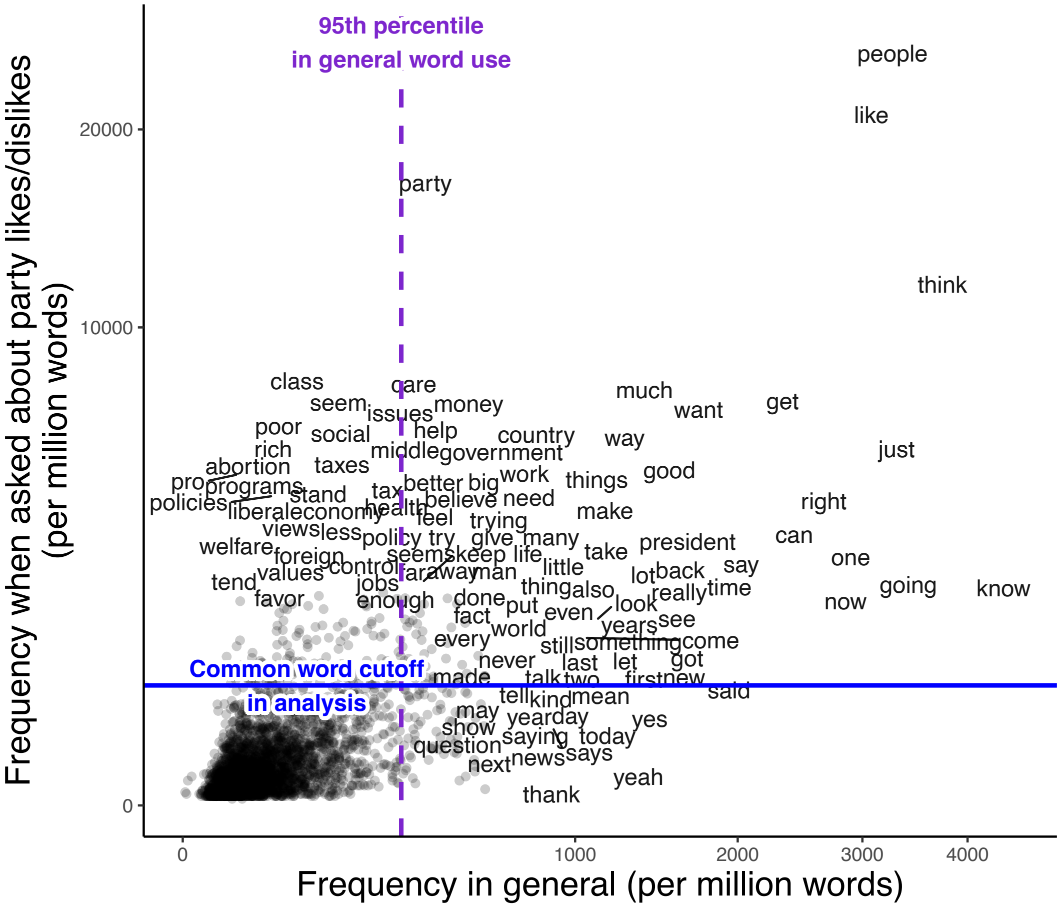


Figure F.1: Frequencies of words in response to the party likes and dislikes prompts versus frequencies of words in general use. Frequencies of general word usage are from the “spoken” genre of the Corpus of Contemporary American English (COCA) (Davies 2008). Words that are moderately common in general can be very common in response to a focused prompt. Corpus frequencies differ somewhat from Figure 3 because that analysis is limited to 2016-2020 ANES panelists, and because the scale here is by frequency per million words to align with the COCA frequency data.

G An example pair of responses about the Affordable Care Act

We argue that, to the extent to which sophisticated statements are informative of attitudes, it is typically through their use of common words, *or* those statements' resemblance to statements that *do* contain common words, that allow the listener (or in our case, the researcher) to place that statement in context. The use of highly idiosyncratic language – however sophisticated – is not by itself informative of attitudes and does not reliably provide additional, relevant information beyond words that are more commonly-used in the given context.

For example, consider two responses in our data concerning the Affordable Care Act from the same respondent in January and October 2016, respectively:

January 2016: *It allows me to continue to cover my daughter after college and I like the no pre existing condition part of it. Lastly, it gives many people a chance to get healthcare coverage they need.*

October 2016: *I like that it allows parents to cover there children longer, no pre existing conditions, clearer EOBs for patients to understand, EHRs, data exchanges, and insurance exchanges to promote insurance competition.* [sic]

Both of these relatively detailed statements (most respondents provide a single short reason for support or opposition) contain multiple words that are common in the particular context of discussing the Affordable Care Act – "conditions" and "coverage", for example, appear together on the pole of the first dimension from our method in Figure 2 – and some rare words (e.g., "EOBs," "EHRs," "exchanges"). Our point is that these contextually common words are more informative of the respondent's general attitudes about the Affordable Care Act – including the stable elements of their responses over time – than the more idiosyncratic rare words, even though they may signal sophistication on the part of the respondent.

Rare words do still matter in our analyses, however. Although they have less influence on dimensions of the implied word method, less common words such as "data" or "exchange" are still scored with respect to each dimension – based on their co-occurrence with more common words.

H Adding latest generation embeddings

H.1 Generating labels with GPT 3.5

We prompted GPT 3.5 (through Microsoft Azure) to return topical labels for every open-ended response across each of the questions analyzed in the main text. The goal of this process was only to produce a large number of labels that *could* be applied to the open-ended responses. We analyzed the labels using embeddings (see next section, Section H.2).

Like (Mellon et al. 2022), we prompted GPT 3.5 with 50 open-ended responses at a time. Responses were grouped randomly. To generate labels, we used the following prompts:

Affordable Care Act responses

> Here are open-ended responses to a question about the Affordable Care Act that asked “Could you tell me in your own words what is the main reason you have (a favorable/unfavorable) opinion of the health reform law?”:
>
> [50 survey responses, each on a new line]
>
> Please assign some topical categories to each open ended text response.
>
> GUIDELINES: Return all of the original survey responses, their ID numbers, and their most relevant categories. There are likely to be multiple relevant categories, many of which will not be words in the survey response itself. The number of relevant categories is likely to vary across responses. As an example response format, please return in this format:
> id:1|“survey response text”|“category1”,“category2”
> id:2|“survey response text”|“category3”,"category2”,“category5”,“category6”.

Party likes/dislikes responses

> Here are open-ended survey responses to a question about American political parties that asked “Is there anything in particular that you (like/dislike) about the (Democratic/Republican) party? What is that?”:
>
> [50 survey responses, each on a new line]
>
> Please assign some topical categories to each open ended text response.
>
> GUIDELINES: Return all of the original survey responses, their ID numbers, and their most relevant categories. There are likely to be multiple relevant categories, many of which will not be words in the survey response itself. The number of relevant categories is likely to vary across responses. As an example response format, please return in this format:
> id:1|“survey response text”|“category1”,“category2”
> id:2|“survey response text”|“category3”,"category2”,“category5”,“category6”.

Most important problem responses

> Here are open-ended survey responses to a question that asked “What do you think is the most important problem facing this country today”:
>
> [50 survey responses, each on a new line]
>
> Please assign some topical categories to each open ended text response.
>
> GUIDELINES: Return all of the original survey responses, their ID numbers, and their most relevant categories. There are likely to be multiple relevant categories, many of which will not be words in the survey response itself. The number of relevant categories is likely to vary across responses. As an example response format, please return in this format:
> id:1|“survey response text”|“category1”,“category2”
> id:2|“survey response text”|“category3”,"category2”,“category5”,“category6”.

GPT 3.5 would often not return labels for many of the texts that we submitted. We did not spend much time trying to fix this behavior because we only wanted a long list of labels, and did not need response level categories from this generative AI step.

H.2 Embedding labels and documents with the OpenAI v3 large embedding model

We used the OpenAI `text-embedding-3-large` model to embed each open-ended response as well as every category returned by GPT 3.5 (as described above in Section H.1). We also added text around the open-ended responses to provide minimal context. We added this same contextualization for the BERT comparisons included alongside the OpenAI embedding results. We did not contextualize embeddings for the GPT generated labels, since they are used for the purpose of assisting readers with interpreting implied word output (with limited context awareness). Before further analyses (i.e., embedding the implied word method as well as running principal component analysis on the embeddings), we averaged each respondent’s party likes/dislikes embeddings. All other data sets contained only one response per respondent. This averaging had the effect of removing information that indicated only which party likes/dislikes question a respondent was answering (and so without this averaging returning only the information we already had in closed-ended form).

For contextualization, we used the following prompts:

Affordable Care Act responses

> Here is an open-ended response to a public opinion survey question about the Affordable Care Act that asked “Could you tell me in your own words what is the main reason you have (a favorable/unfavorable) opinion of the health reform law?”:

> [1 survey response]
>
> The respondent gave this answer sometime between 2009 and 2018.
>
> Question: Broadly speaking, what is the main reason this respondent has a favorable or unfavorable opinion of the Affordable Care Act?

Party likes/dislikes responses

> Here is an open-ended response to a public opinion survey question that asked "Is there anything in particular that you (like/dislike) about the (Democratic/Republican) party? What is that?":
>
> [1 survey response]
>
> The respondent gave this answer in a United States presidential election year sometime between 1980 and 2020.
>
> Question: Broadly speaking, why does this respondent like or dislike the Democratic or Republican party?

Most important problem

> Here is an open-ended response to a public opinion survey question that asked "What do you think is the most important problem facing this country today?":
>
> [1 survey response]
>
> The respondent gave this answer in a United States presidential election year sometime between 1980 and 2020.
>
> Question: Broadly speaking, what does this respondent think is the most important problem or problems facing the United States?

H.3 Embedding the implied word method output

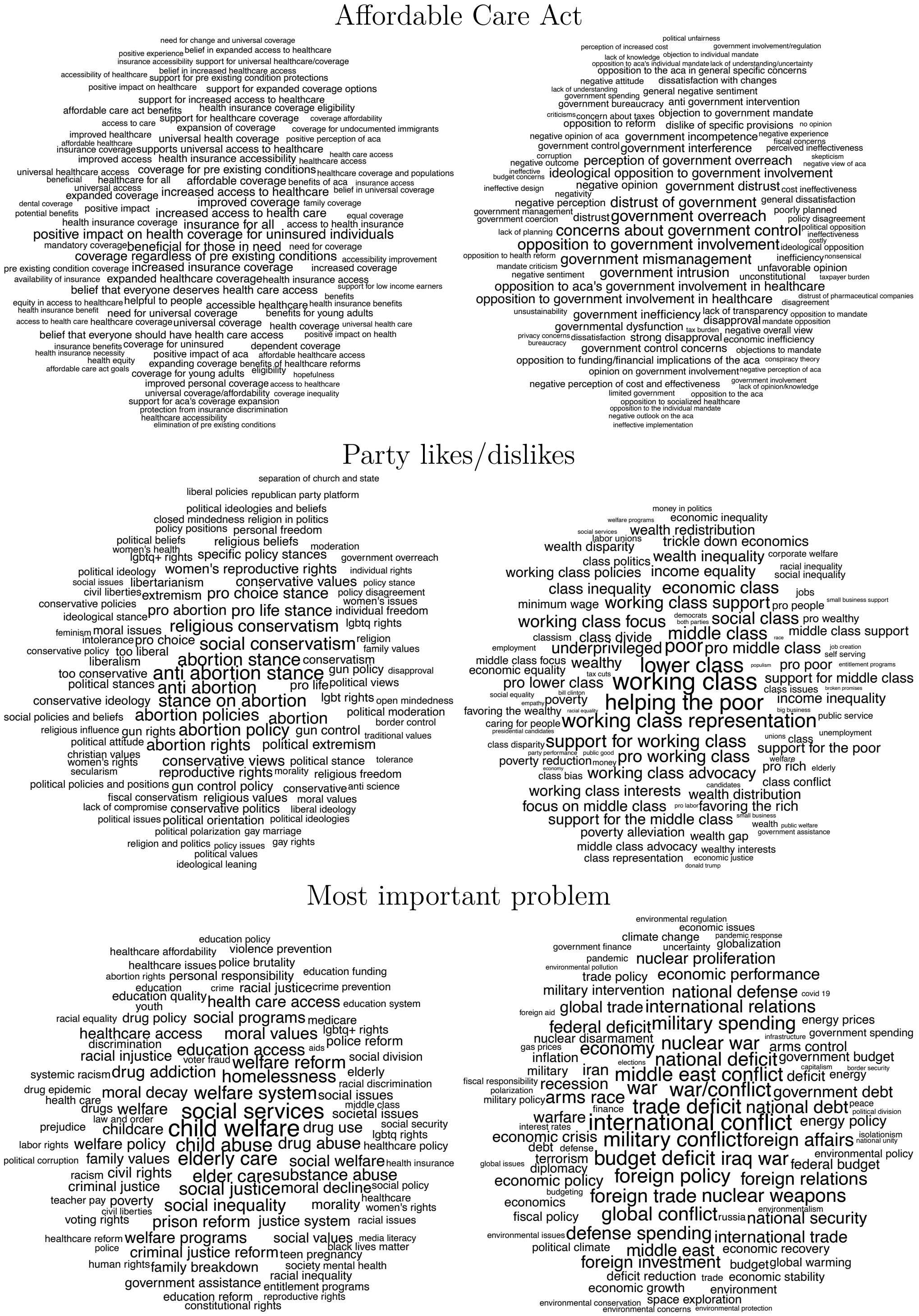
To create an embedding for each dimension of our implied word method, we centered the implied word document scores at each dimension's weighted mean (each ANES survey wave was weighted equally), and multiplied the implied word document scores by each document's embedding (see previous section, Section H.3). We then averaged the embeddings (i.e., we averaged each of the 3,072 dimensions for the OpenAI embeddings and 768 for BERT) to return an embedding for each implied word dimension. For analyses with a hold-out set in the implied word method training (e.g., the ACA analyses and the ANES analyses trained only on 2016 data), the same data was also held out for this dimension embedding step.

H.3.1 Scoring documents with embeddings

An implied word document score for the embedded version of the method is simply the cosine similarity between a given document's embedding and the implied word embedding (for a given dimension) described above. See the R code walk-through in Section C for an implementation this scoring process.

Similarly, the document score for a topical label (those generated by GPT 3.5 – for the analysis displayed in the bottom panel of Figure 6) is the cosine similarity between a given document's embedding and the label's embedding.

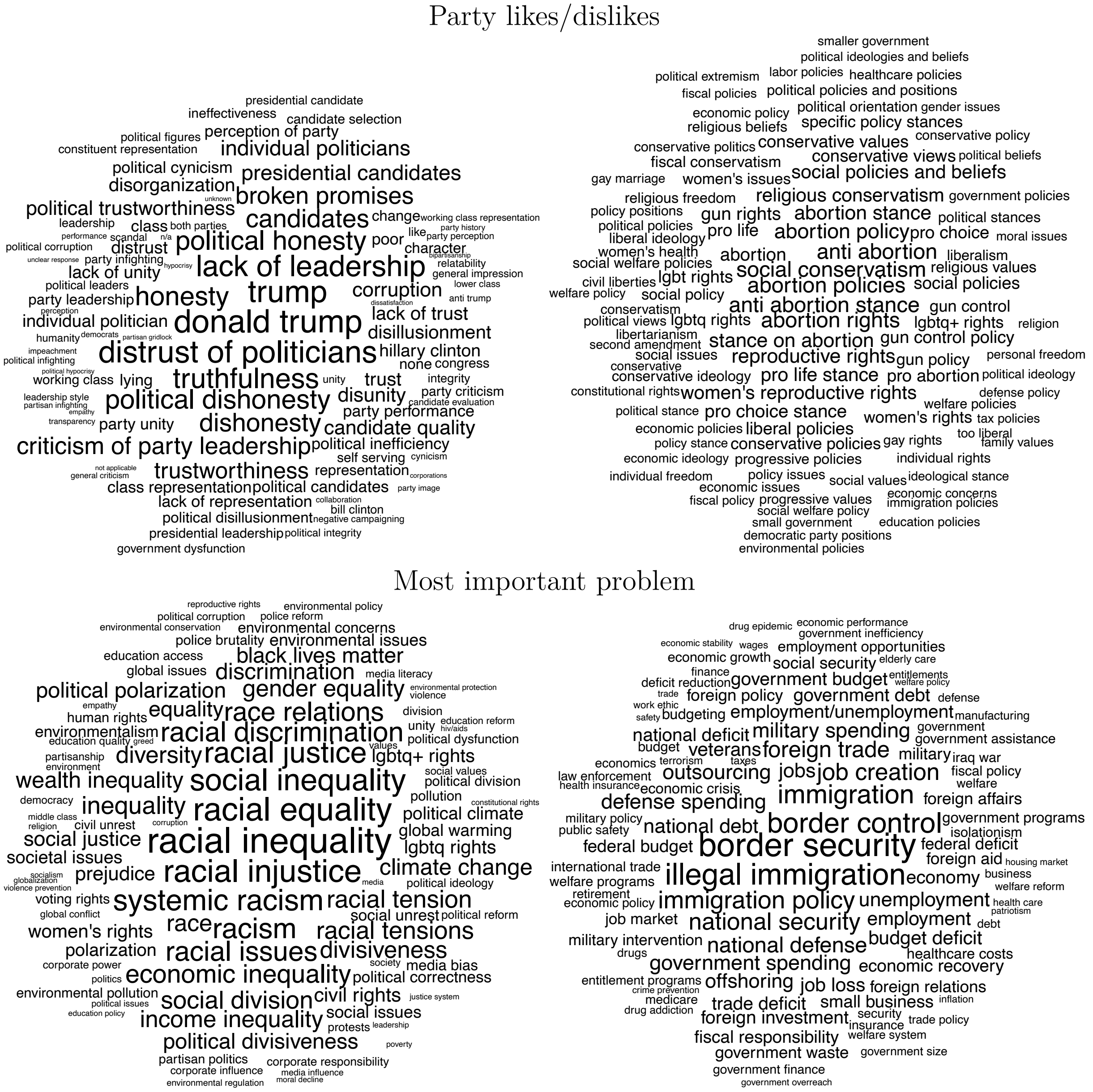
H.4 Top embedding labels for each open-ended question


Affordable Care Act

Party likes/dislikes

Most important problem

GPT 3.5 generated category labels with the 100 highest and 100 lowest cosine similarities for the first embedded implied word dimension of each open-ended question.



GPT 3.5 generated category labels with the 100 highest and 100 lowest cosine similarities for the first embedded implied word dimension of each open-ended question – trained on 2016 data only.

H.5 Embedding panel correlations

Below, we show the full results for the top 10 dimensions of the implied word method and the top 10 dimensions for PCA on the BERT and OpenAI embeddings.

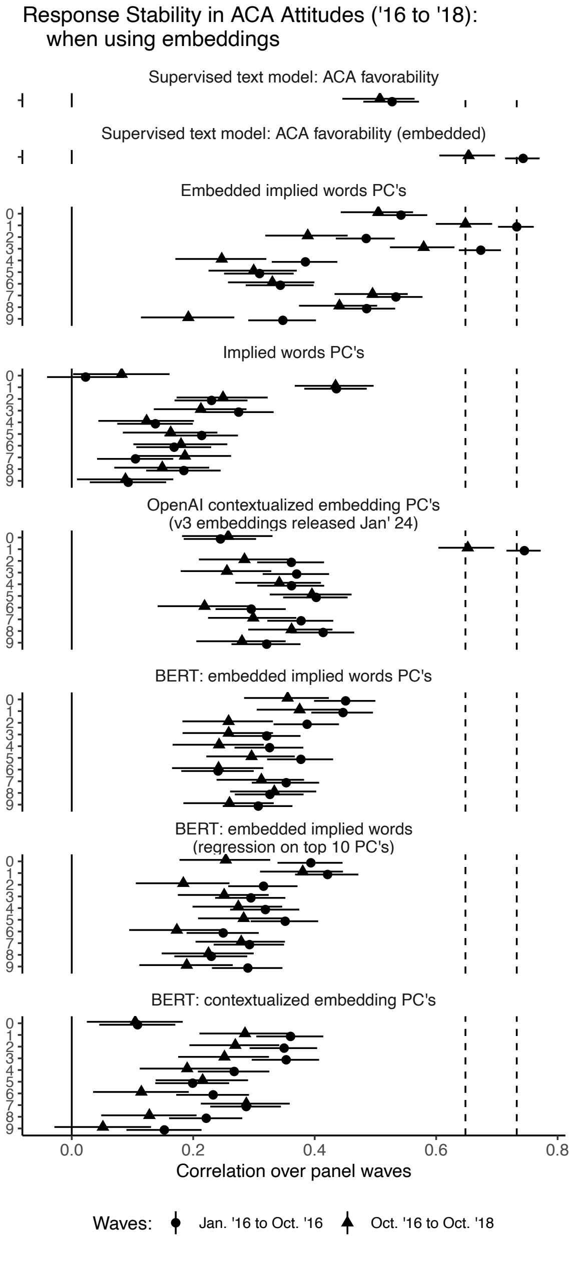


Figure H.3: Test-retest correlations for the embedded version of our method, along with PCA dimensions from BERT and OpenAI v3 embeddings – Affordable Care Act responses.

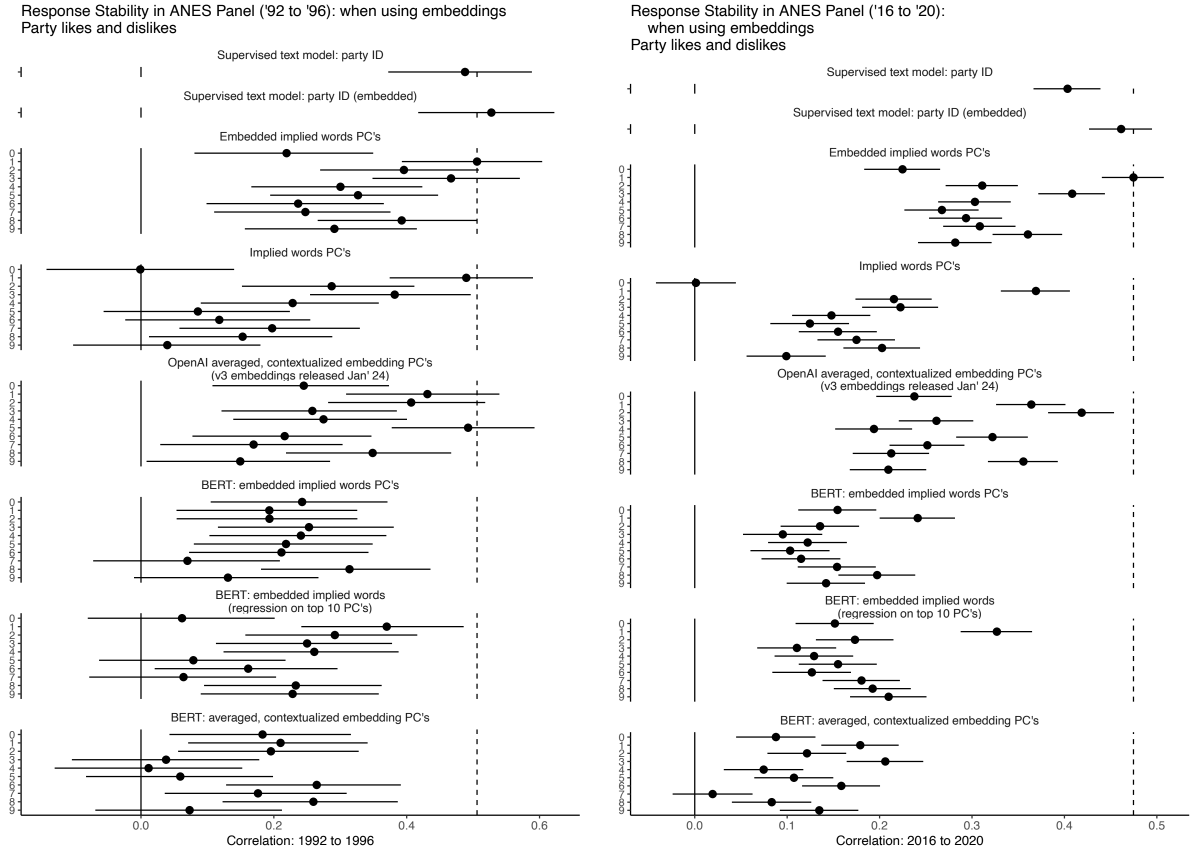


Figure H.4: Test-retest correlations for the embedded version of our method, along with PCA dimensions from BERT and OpenAI v3 embeddings – party likes/dislikes responses.

We speculate a few reasons that the new embeddings might out-perform BERT: 1) they are just bigger (e.g., 3,072 dimensions versus 768) and so can better capture variation in meaning (though this can plausibly also work against them), 2) they have more relevant training data (and the ability to better identify relevance), potentially including the ANES data itself (unfortunately, training data details are not public for these models, even to our knowleedge 'open-source' models, and 3) they are trained with and for a longer and more expansive context/context window and so, relevant to our task here, capture a broader sense of topics than more narrow windows.

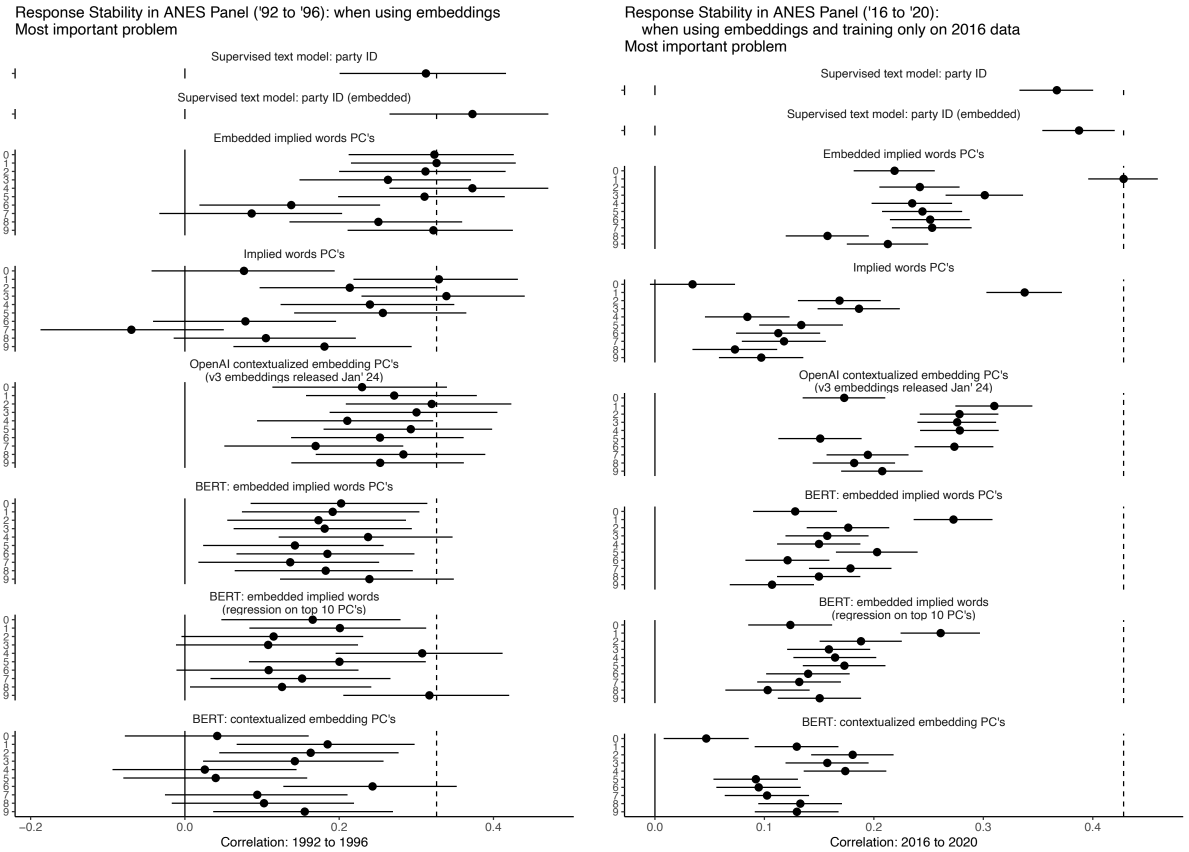


Figure H.5: Test-retest correlations for the embedded version of our method, along with PCA dimensions from BERT and OpenAI v3 embeddings – most important problem responses.

H.6 Embedding correspondence with hand labels

We show a hand label multiple R analysis for the embedding analyses below, following the same procedure we used to generate Figure 4.

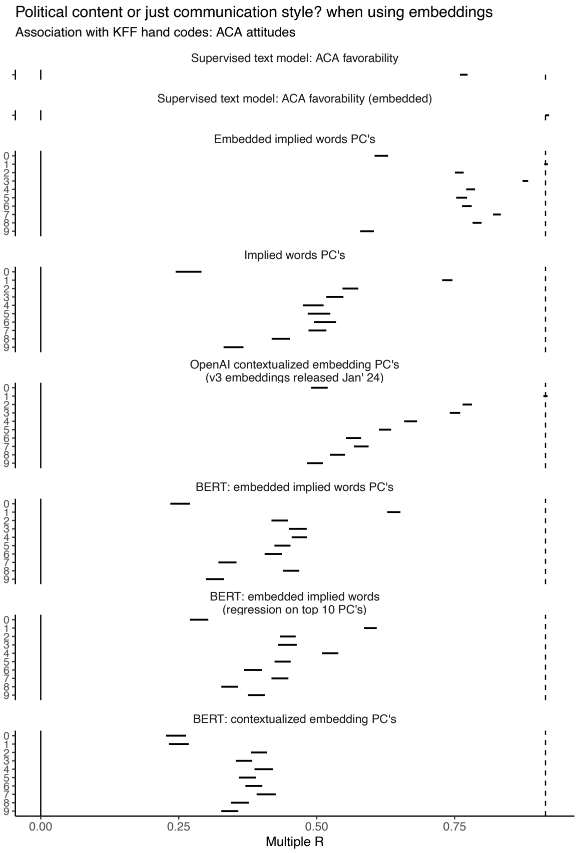


Figure H.6: Embedded method hand label multiple R's – Affordable Care Act responses. The first dimensions of the embedded implied word and OpenAI embedding PCA have very narrow confidence intervals and are on top of the dotted line.

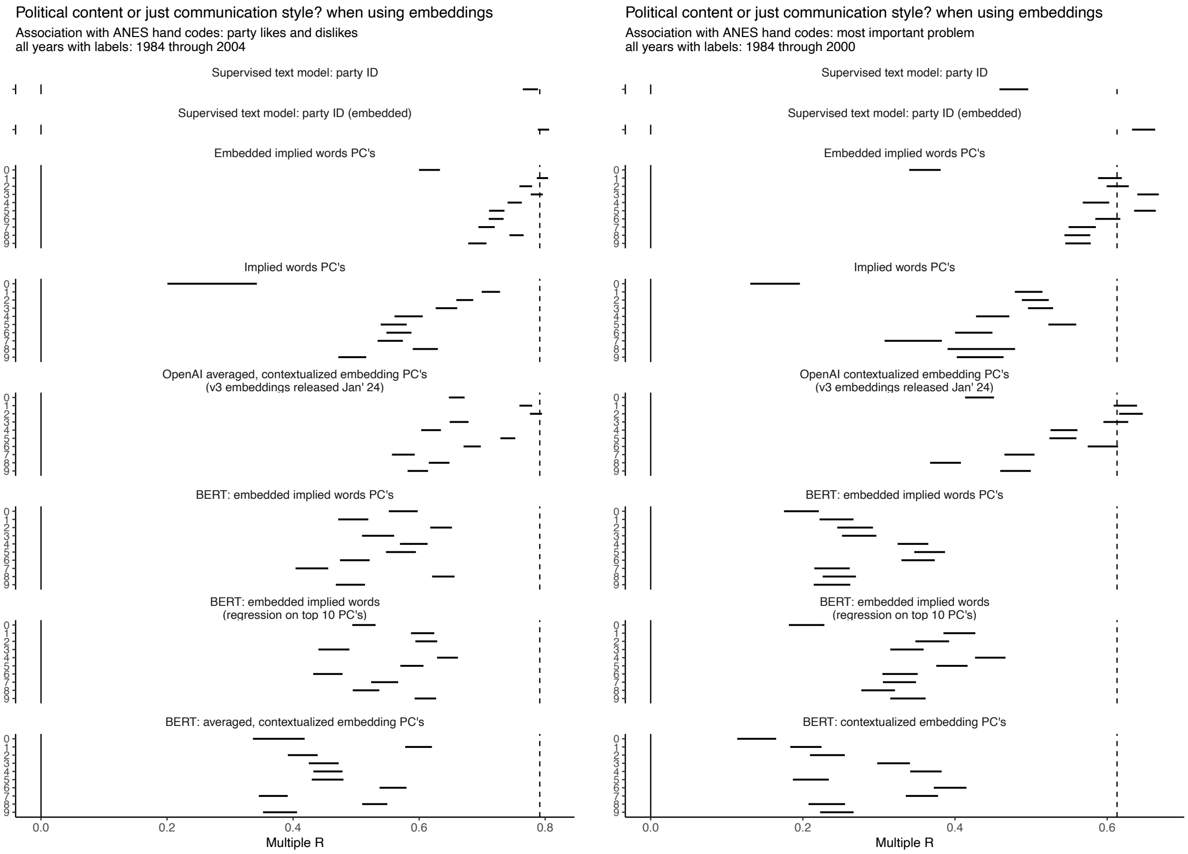


Figure H.7: Embedded method hand label multiple R's – party likes/dislikes and most important problem responses

I Assessing corpus ‘context size’

The context covered in some corpora may be too large for our method to work well – for example, when what is contextually common for one subset of the data is *not* contextually common for the other, or when the associations with those contextually common words differ substantially (and so, potentially, convey different symbolic meanings).

To assess this, we can split the data on some variable that may drive overly large context size and a contrast that *we know we do not want the method to identify* – e.g., pandemic versus non-pandemic years. We can then assess the correlation in word frequencies across that contrast.

Below, we show that over a shorter 4 year interval (rather than close to 40 years), even with a pandemic, 2016 (only) and 2020 do not meaningfully differ in terms of common word use (i.e., correlation in square root word frequencies). In this figure, black circles around the points indicate survey waves that are 4 years or fewer apart.

In the main text, we retrained our implied word method using just 2016 as training data for both ANES questions. This gave us an extra test for our added, latest generation embedding analyses, and also allowed us to better compare test-retest reliability and cross test-retest reliability.

Note that this test does not mean that the implied word method will *necessarily* be strongly influenced by the examined contrast. For example, we do not observe meaningful differences for the most important problem question when excluding 2008 (which asked a different question: “What do you think is the most important *political* problem facing the United States today?” – emphasis added). This appears to be because 2008 differs from the rest of the corpus primarily on the (first) dimension that the implied word method identifies even without that year included in training (i.e., this wording reduces the number of political issues mentioned that are also societal issues). 2020, on the other hand, includes much more novel information.

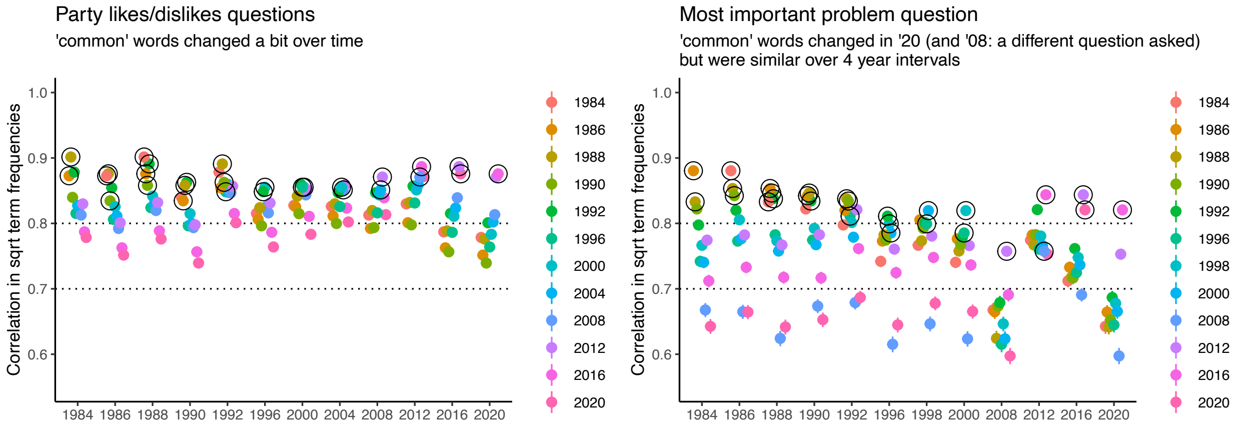


Figure I.1: Correlations in square root word frequencies, and so contextually common words and their associates, diverge in 2020 for the most important problem question, but are similar to those in 2016 – suggesting that a 40 year context size is too large for this question but a 4 year context size is not. Black circles around the points indicate survey waves that are 4 years or fewer apart.

J Response length and response stability

We were uncertain whether document length might be incorporated into our method in some way.

Longer responses could reflect more sincere or intensely-held attitudes, or they could reflect stylistic differences (such as verbosity), or they could reflect ambivalence. By the same token, intensely-held attitudes can often be expressed very succinctly. For example, respondents who like or dislike the parties' stances on abortion often write very short responses, and tend to have some of the most stable responses.

Beyond this, on the inference side, if a response is extremely short – and many of them are – it can sometimes be very difficult to interpret the response at all, much less infer the respondent's underlying attitude. And, in scoring documents with our method, we are able to average more word scores when a document is longer, likely increasing the reliability of our estimate (whether or not the underlying attitude is more intense and consistent) – even though the detailed, longer responses can be problematic when using unsupervised *training* to *find* attitude dimensions to score words on.

On the other hand, longer responses could contain a larger number of uninformative, filler words, even when the response as a whole can be clearly placed on a dimension.

To try to rule out some of these possibilities, we tested whether respondents who wrote longer documents in the 1st wave of 2 had higher test-retest reliability than shorter ones. These results are shown in Figure J.1. In this, we did not find that response length was consistently associated with more stability one way or the other.

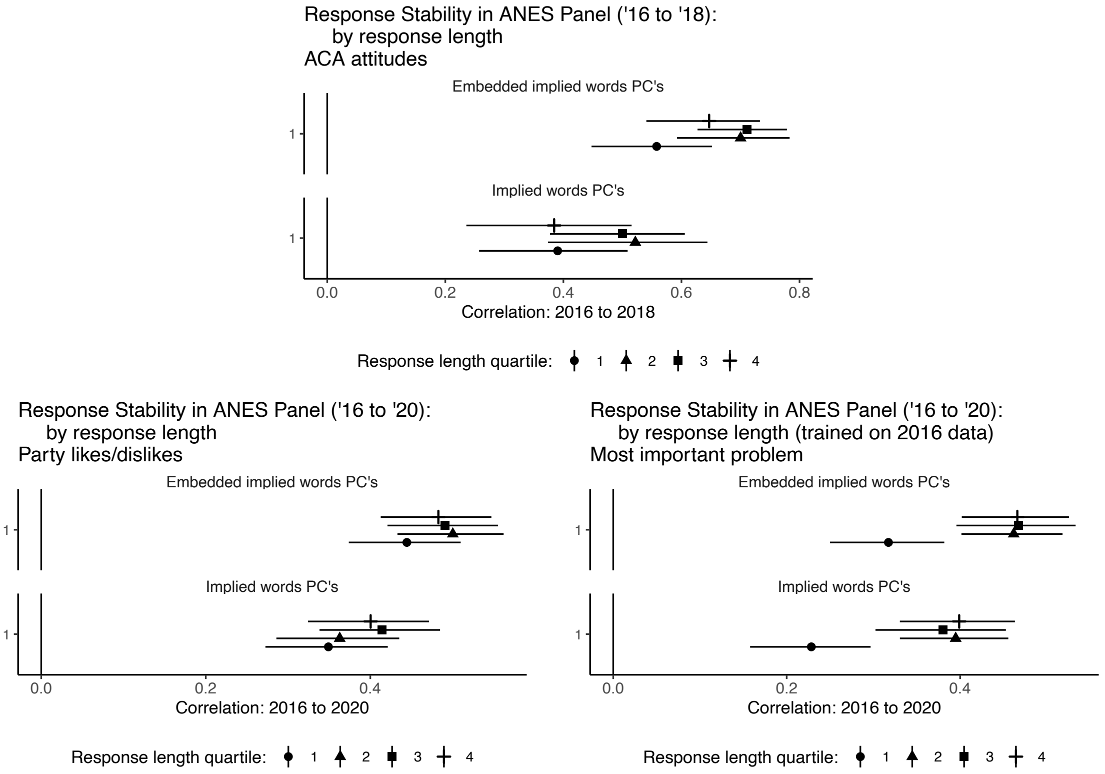


Figure J.1: *Test-retest reliability by response length quartile.* By and large, test-retest reliability was not different for the implied word scores across different quartiles of document lengths (as measured in the first wave of each comparison to avoid dropping respondents who wrote longer answers in one wave or the other). However, in the most important problem data, we do observe a very large difference between the first quartile of responses (very short responses) compared to the longer responses. In these analyses, we use the 2016-2020 ANES waves because they are much larger, and so we can split the data into quartiles without creating very small bins.

K Example responses and scores

First dimension in sd's	First dimension (embedded score) in sd's	Open-ended response, first 50 words stopwords and words <= 2 characters in gray were removed from the document-term matrix but not the embedded text
1.40	2.11	they're more for the people they try to help out the people the middle class
1.76	2.06	they're for the rich people
1.76	2.03	they are more for the rich people
0.59	1.47	i definetly feel they are the party of the working man basically more willing to give more money for the well being of children and people who arin more need of support both physcally and mentally
0.96	1.40	they try to do the best for the people have more help programs for people in us such as wefare aid to the underprivileged like food stamps and such
2.31	1.06	more for the rich
-1.11	0.90	they historical support and represent my values for me and most of us i believe they serve me and most of us better than republicians
0.69	0.89	they spend too much money don't know if it's the democrats raising federal employee wages or not they spend too much money to get voted in
0.70	0.70	i don't like richard nixon that still hangs in my mind lack of honesty republicans hold back can't tell if reagan is sincere
2.00	0.34	i'm a democrat because my mama and daddy was a democrat
0.72	0.14	it seems as though we always have wars and we have our wars when they are in power but maybe i'm wrong on that
0.56	-0.22	everything
-0.60	-0.59	it is a conservative thinking party weigh their problems and maintained a good relationship with foreign countries their import export policies are good the development plan for our country are good developing oil minerals science hope he balances the budget
0.00	-0.71	almost everything
-1.38	-0.87	the conservative ideas i like they are pro business and anti tax
-0.40	-1.05	their stand on a lot of issues that affect people not just working but personally gun control the general way they think we ought to be
-2.15	-1.08	they stand for family and moral values
-0.20	-1.32	i'm fairly conservative so i like the fact that they would not pass a law to let a woman walk around naked or show it on tv i wouldn't object to two people of the same sex who want to be partners but i don't think they should be allowed ... [truncated for this table]
-3.02	-2.17	same sex marriage support pro choice stem cell research support health care laws
-2.67	-2.46	they support abortion
-1.95	-2.66	approach to some social problems overly strict on personal rights like abortion

Table K.1: Example party likes/dislikes responses: randomly sampled by bin (cutoffs at ± 0.5, 1, 2) and ordered by first dimension of embedded implied word method. We have added dashed lines at 1 and -1 standard deviations of the scores. These redacted responses are no longer restricted use data: `https://electionstudies.org/data-center/restricted-data-access/`.

First	First dimension	Open-ended response, first 50 words
dimension	(embedded score)	stopwords and words <= 2 characters in gray were removed from
in sd's	in sd's	the document-term matrix but not the embedded text
3.16	1.12	they are crooks too they're all the same
2.14	1.90	they're more for the rich than the poor
2.06	1.92	they are more for the poor man
1.39	1.83	the raise the taxes for the low income and their not for the people with low income their for the richer people
1.35	1.75	they are out to help the middle classes and the poor
1.18	1.83	they are for the middle class every day people
0.84	-0.37	very much afraid that if mondale is elected we will end up in a war there always was a democratic president in office when we have gone to war
0.64	1.80	they like to make money for poor people they have jobs for poor people benefits healthcare
0.57	0.16	there are too many yes men in the demo party leave it at that
0.23	-0.37	seem to be a little more conservative they're conscious of budget spending money programs to save taxpayers money looking more at the deficit to balance the budget more aware of what's going on
-0.35	-0.29	not very unified
-0.38	-1.06	the way they have handled foreign relations
-0.51	-0.50	they are not humane enough
-0.94	-0.96	fore american people the economy business prolife freedom of speech
-0.99	-1.15	the way they handle the economy lower taxes more moderate in beliefs support 2nd ammendment
-1.04	-0.75	i think it's more respectful of individual rights abortion gay rights it demonstrates a belief that govt can do good in ways other than shoveling potholes the peace corps making more money for supporting college students i think it's more inclusive especially across racial lines
-1.30	-2.18	general conservatism stand on abortion era and defense spending a strong defense they aren't as willing to spend money
-1.75	-2.56	on some issues they're too liberal for me like abortion
-2.07	-2.61	mostly pro abortion or woman's choice
-2.30	-1.91	conservative values
-2.43	-2.34	i don't care for their stance on abortion gun control welfare

Table K.2: Example party likes/dislikes responses: randomly sampled by bin (cutoffs at ± 0.5, 1, 2) and ordered by first dimension of non-embedded (document-term matrix only) implied word method. We have added dashed lines at 1 and -1 standard deviations of the scores.

References

Abramowitz, A. I. (1995, February). It's Abortion, Stupid: Policy Voting in the 1992 Presidential Election. *The Journal of Politics 57*(1), 176–186.

Benoit, K., D. Conway, B. E. Lauderdale, M. Laver, and S. Mikhaylov (2016, May). Crowd-sourced Text Analysis: Reproducible and Agile Production of Political Data. *The American Political Science Review 110*(2), 278–295.

Bischof, J. and E. M. Airoldi (2012). Summarizing topical content with word frequency and exclusivity. In *ICML*.

Carmines, E. G. and J. A. Stimson (1980, March). The Two Faces of Issue Voting. *American Political Science Review 74*(1), 78–91.

Davies, M. (2008). Word frequency data from the corpus of contemporary american english (COCA).

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of NAACL-HLT*, 4171–4186.

Friedman, J., T. Hastie, R. Tibshirani, B. Narasimhan, K. Tay, N. Simon, and J. Qian (2021). Package 'glmnet'. *CRAN R Repositary.*

Green, J. (2023). The rhetorical 'what goes with what': Politicalpundits and the discursive superstructure of ideology in u.s. politics.

Hughes, A. G., S. D. McCabe, W. R. Hobbs, E. Remy, S. Shah, and D. M. J. Lazer (2021, September). Using Administrative Records and Survey Data to Construct Samples of Tweeters and Tweets. *Public Opinion Quarterly 85*(S1), 323–346.

Krosnick, J. A. (1990, March). Government policy and citizen passion: A study of issue publics in contemporary America. *Political Behavior 12*(1), 59–92.

Lewis, M., Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer (2020). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, pp. 7871–7880. Association for Computational Linguistics.

Maiya, A. S. (2022). Ktrain: A low-code library for augmented machine learning. *Journal of Machine Learning Research 23*(158), 1–6.

Mellon, J., J. Bailey, R. Scott, J. Breckwoldt, and M. Miori (2022). Does GPT-3 know what the Most Important Issue is? Using Large Language Models to Code Open-Text Social Survey Responses At Scale. *SSRN Electronic Journal.*

Mimno, D. and M. Lee (2014). Low-dimensional Embeddings for Interpretable Anchor-based Topic Inference. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp. 1319–1328. Association for Computational Linguistics.

Roberts, M., B. Stewart, and D. Tingley (2016). Stm: R Package for Structural Topic Models. *Journal of Statistical Software.*

Rodriguez, P. L., A. Spirling, and B. M. Stewart (2023, January). Embedding Regression: Models for Context-Specific Description and Inference. *American Political Science Review*, 1–20.

Tanweer, A., E. K. Gade, P. Krafft, and S. K. Dreier (2021, June). Why the Data Revolution Needs Qualitative Methods. *Harvard Data Science Review.*

Yin, W., J. Hay, and D. Roth (2019, August). Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach. *EMNLP*.