# Appendices

## Appendix 1. Additional methodology and validity tests

### *Appendix 1.1 Methodology of word vectors*

A full description of the approach to word vectors can be found in (Mikolov, Chen, et al. 2013) and some recent introductions in political science can be found in, e.g., (Rheault and Cochrane 2020; Rodman 2020), so we will be brief and focus on the elements that are relevant for our specific purpose. We largely follow the recent description by Zhang et al. (2020, Chapter 14). Refinements to the classical approach include the addition of so-called paragraph vectors, which allow for adding higher-order document features such as, for example, information about the political affiliation of the document author (Le and Mikolov 2014; Rheault and Cochrane 2020). Other refinements include so-called subwords, which capture, for example, the fact that different words might nonetheless be referring to (more or less) the same underlying construct (Bojanowski et al. 2017). Davidson et al. (2017)

Classical word embeddings can be trained either using a neural network or using more classical matrix optimizations (Pennington, Socher, and Manning 2014). We use the neural network approach since this allows us to take advantage of subwords, as further discussed below (Bojanowski et al. 2017).

The power of word vectors lies in the way the data is input to the network.[12]. For example, imagine a text that states: "I hate all Democrats and we should reelect President Trump." This text is split into 5-word-long chunks using a sliding window until we have reached the end of the text ("I hate all Democrats and", "hate all Democrats and we" etc). Next, the algorithm tries to predict the middle word, so if our context words are denoted $W_c$, we can write this as $W_c = \{w_{target-2}, w_{target-1}, w_{target+1}, w_{target+2}\}$ . The idea is then to try to predict a target word $\{w\}$ with the number of categories equal to the size of our vocabulary $V$ given the context of words $c$. An example could be, e.g., that we are trying to predict democrats given the contextual words {hate, all, and, we} i.e. $P(''democrats''|''hate'', ''all'', ''and'', ''we'')$.

The way a neural network is typically constructed is via a series of (hidden) layers, and this particular network only has an input layer (i.e., the [location of the] context words), an embedding layer and an output layer. If all words have an index $i$ in the used dictionary, the embedding layer ($z$) can in general terms be written as the product of the context word vector $c$ and the average of the embedding word vectors with window size m: $v = \frac{v_{target-2}, v_{target-1}, v_{target1}, v_{target2}}{2m}$:

$$z = c^T \cdot \bar{v}$$

The output of the embedding layer is then fed into an output layer that can handle multiple unordered outcomes and turn this into a probability. The typical choice is what in the deep learning literature is called a softmax, better known in the political science literature as a multinomial logit. This functional form can handle the V outcomes that are needed to predict all the different categories in the vocabulary. The input to the softmax is simply the output of the embedding layer, where we divide the sum of all dot products for our vocabulary:

$$P(w_t|W_c) = \frac{e^{c^T \cdot \bar{v}}}{\sum_{i \in V} e^{c_i^T \cdot \bar{v}}}$$

This function can be optimized using maximum likelihood estimation such that it is possible to find the weights that make generating a target word from the context words most likely. After training is complete, the only thing necessary to keep is the embedding layer, which represents the focal word vectors, i.e., each word gets projected into d-dimensions that represent each word's position in the vector space.

12. Here we present the CBOW model, but typically the so-called skip-gram approach produces very similar results (Zhang et al. 2020)

The super-unsupervised approach's only major departure from this fairly standard model is the inclusion of so-called *subwords* [13]. Some example subwords for the word "political" are "polit," "politi" and "itical." Subwords thus help reduce the number of words in the vocabulary since many similarly spelled words essentially represent a single construct. It is also extremely helpful in terms of words that are slightly misspelled, such as is often the case in social media data. In essence, the word vector for a given word thus becomes an average of the vectors of the subwords a given word consists of.

There are two important conclusions to be drawn from this exposition of estimating word vectors. First, words that tend to co-occur (i.e. often appear in the same window) will also have more similar word embeddings. The measure of similarity often used for vectors is the cosine similarity, which can be calculated using our newly trained word embeddings in the form of the dot product between them divided by their euclidian distance: $cos(\theta) = \frac{v_{e1} \cdot v_{e2}}{|v_{e1}||v_{e2}|}$ which ranges from -1 (complete opposites) to 1 (complete similarity), with a cosine similarity of zero indicating no relationship between the two vectors.

### *Appendix 1.1.1 Aligning word vectors*

Vector embeddings from from different models cannot be directly compared since they reside in different vector spaces (Di Carlo, Bianchi, and Palmonari 2019). There is an almost infinite number of approaches that can be used to align vector spaces (Søgaard et al. 2019) but here we will have used the so-called Compass-Aligned Distributional Embeddings (CADE) (Bianchi et al. 2020).[14]. This approach is useful since it can handle both changes over time, as well as across contexts (Bianchi et al. 2020) and is a natural extension of the basic word2vec methodology.

The basic problem is that distances in vector spaces cannot be directly compared, so even though we might be able to see that the word "trumpian" is closer to "fearmongering" for a word2vec model trained on tweets from Democrats compared to Republicans but exactly how much closer would be impossible to know without aligning the vector spaces. The CADE approach uses the distinction between the target word and context words introduced above. First a typical word2vec model is trained on the entire corpus; a "global" model. Then a model is trained separately for the timepoints or subgroups that we wish to study differences for; what in the CADE parlace is known as "slices". When training the slices the *target* embeddings are used from the global model, and not changed when training this slice; only the *context* embeddings are updated. This ensures that the vector spaces in fact reside in the same vector space while allowing for changes over time or across contexts. For a more detailed exposition we refer interested readers to (Di Carlo, Bianchi, and Palmonari 2019; Bianchi et al. 2020).

We use the CADE technique in all our applications except for our comparison between our own Twitter embeddings and those obtained from a pretrained model (see Appendix 1.5), since the CADE approach requires access to the corpus itself.

### *Appendix 1.1.2 API and model fitting*

The analyses of word embeddings presented in the main text are all done in the Python wrapper for Facebook's *fasttext* Python API. To test the robustness of the findings we also use the Python library Gensim's (Řehůřek and Sojka 2010) implementation of fasttext using similar settings as those in the *fasttext* API and obtain fairly similar results; see Table 1 below.

The fasttext model is run using the default settings simply to demonstrate that the technique does not require a lot of manual tuning to work. The defaults use a learning rate of .05, run for five

13. The approach also uses subsampling at the prespecified level of 0.0001 in fastText. Subsampling decreases the importance of frequent words and thus helps increase accuracy; see e.g. (Mikolov, Sutskever, et al. 2013)

14. The technique has previously been referred to as Temporal Word Embeddings with a Compass (TWEC) but is more general than only using it for studying changes over time so we use the more general name CADE

epochs (i.e. the whole corpus is trained on five times), and the size of word vectors is 100 dimensions. The sliding window size is 5 minimum number of word occurrences for a word to be included, only unigrams are used and our subwords can vary from 3 to 6 characters.

**Table 1.** Correlation table for the correlations between measures based on tweets (Political Hate, Political, Hate, Toxicity and Sentiment) and the measures from the survey data (Political interest, political knowledge, cynicism, trolling, gender) using the gensim library instead of fasttext

| | Toxicity score | Afinn score | Political hate vector | Political vector | Hate vector | Political interest | Political knowledge | Hostility | Female |
|---|---|---|---|---|---|---|---|---|---|
| Toxicity score | 1.00 | -0.61 | 0.46 | 0.30 | 0.62 | 0.24 | 0.23 | 0.33 | -0.06 |
| Afinn score | -0.61 | 1.00 | -0.19 | -0.23 | -0.27 | -0.20 | -0.19 | -0.25 | 0.03 |
| Political hate vector | 0.46 | -0.19 | 1.00 | 0.79 | 0.84 | 0.28 | 0.31 | 0.21 | -0.13 |
| Political vector | 0.30 | -0.23 | 0.79 | 1.00 | 0.41 | 0.32 | 0.30 | 0.19 | -0.21 |
| Hate vector | 0.62 | -0.27 | 0.84 | 0.41 | 1.00 | 0.24 | 0.27 | 0.23 | -0.03 |
| Political interest | 0.24 | -0.20 | 0.28 | 0.32 | 0.24 | 1.00 | 0.48 | 0.17 | -0.18 |
| Political knowledge | 0.23 | -0.19 | 0.31 | 0.30 | 0.27 | 0.48 | 1.00 | 0.17 | -0.30 |
| Hostility | 0.33 | -0.25 | 0.21 | 0.19 | 0.23 | 0.17 | 0.17 | 1.00 | -0.11 |
| Female | -0.06 | 0.03 | -0.13 | -0.21 | -0.03 | -0.18 | -0.30 | -0.11 | 1.00 |

### *Appendix 1.1.3 Preprocessing*

Differences in preprocessing in an unsupervised setting can often produce widely different answers to the research questions posed (Denny and Spirling 2018). There are clearer standards for preprocessing in the supervised setting, but the problem can also simply be turned into a question of hyperparameter tuning in this setting: We can simply vary the preprocessing steps and choose the model configuration with the best fit according to some fit metric such as e.g. accuracy or precision. This is the approach taken here. We vary a series of preprocessing steps to test the robustness of the method. In fact, the results seem unaffected by preprocessing. This is shown in Table 2.

Preprocessing is done using the Python library spaCy (Honnibal and Montani 2017). We extract the lower cased lemma for each word and remove stop words, punctuation, urls, emojis, hashtags and numbers and only include words with more than two characters.

**Table 2.** Correlation table for the correlations between measures based on tweets (Political Hate, Political, Hate, Toxicity and Sentiment) and the measures from the survey data (Political interest, political knowledge, gender).This correlation table shows the correlations without any preprocessing i.e. without removing stop words, punctuation, urls, emojis hashtags and numbers and without removing rare terms

| | Toxicity score | Afinn score | Political hate vector | Political vector | Hate vector | Political interest | Political knowledge | Hostility | Female |
|---|---|---|---|---|---|---|---|---|---|
| Toxicity score | 1.00 | -0.61 | 0.64 | 0.55 | 0.71 | 0.24 | 0.23 | 0.33 | -0.05 |
| Afinn score | -0.61 | 1.00 | -0.50 | -0.48 | -0.46 | -0.20 | -0.18 | -0.25 | 0.02 |
| Political hate vector | 0.64 | -0.50 | 1.00 | 0.97 | 0.95 | 0.40 | 0.45 | 0.29 | -0.12 |
| Political vector | 0.55 | -0.48 | 0.97 | 1.00 | 0.85 | 0.44 | 0.48 | 0.26 | -0.17 |
| Hate vector | 0.71 | -0.46 | 0.95 | 0.85 | 1.00 | 0.33 | 0.36 | 0.29 | -0.04 |
| Political interest | 0.24 | -0.20 | 0.40 | 0.44 | 0.33 | 1.00 | 0.48 | 0.17 | -0.18 |
| Political knowledge | 0.23 | -0.18 | 0.45 | 0.48 | 0.36 | 0.48 | 1.00 | 0.17 | -0.30 |
| Hostility | 0.33 | -0.25 | 0.29 | 0.26 | 0.29 | 0.17 | 0.17 | 1.00 | -0.11 |
| Female | -0.05 | 0.02 | -0.12 | -0.17 | -0.04 | -0.18 | -0.30 | -0.11 | 1.00 |

### *Appendix 1.2 Face validity*

Figure 1 shows the distributions of “Political Hate” scores based on our unsupervised-supervised approach and “Toxicity” scores (Google Perspecive API) for the a subset of the tweets. Figure 2 shows the distribution of “Political” and “Hate” scores separately.

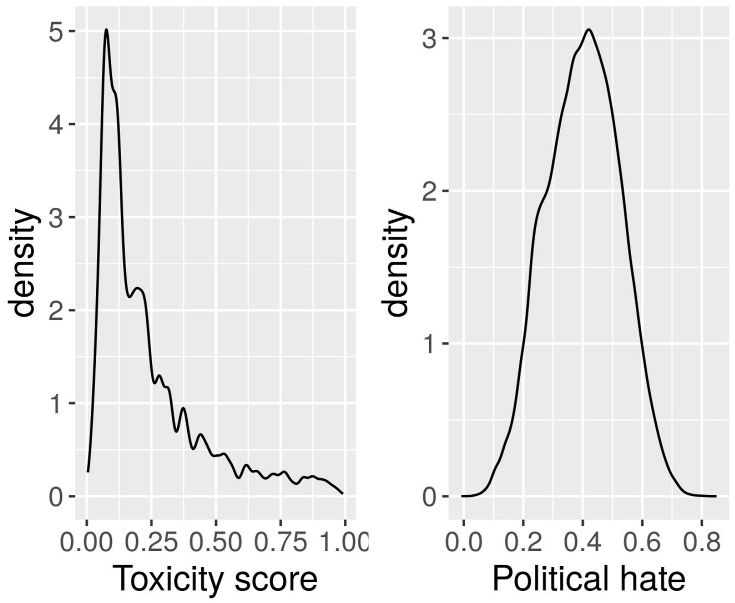

**Figure 1.** This figure shows the distributions for Toxicity scores and Political Hate scores for a subset of 100000 observations in the full (i.e. not grouped) dataset

**Table 3.** Mean values for the word vectors Political Hate, Political and Hate and mean values for Toxicity scores in the ungrouped dataset

| Construct | Mean | Standard deviation |
|---|---|---|
| Political word vector | 0.29 | 0.12 |
| Hate word vector | 0.36 | 0.11 |
| Political hate word vector | 0.40 | 0.12 |
| Toxicity score | 0.24 | 0.20 |

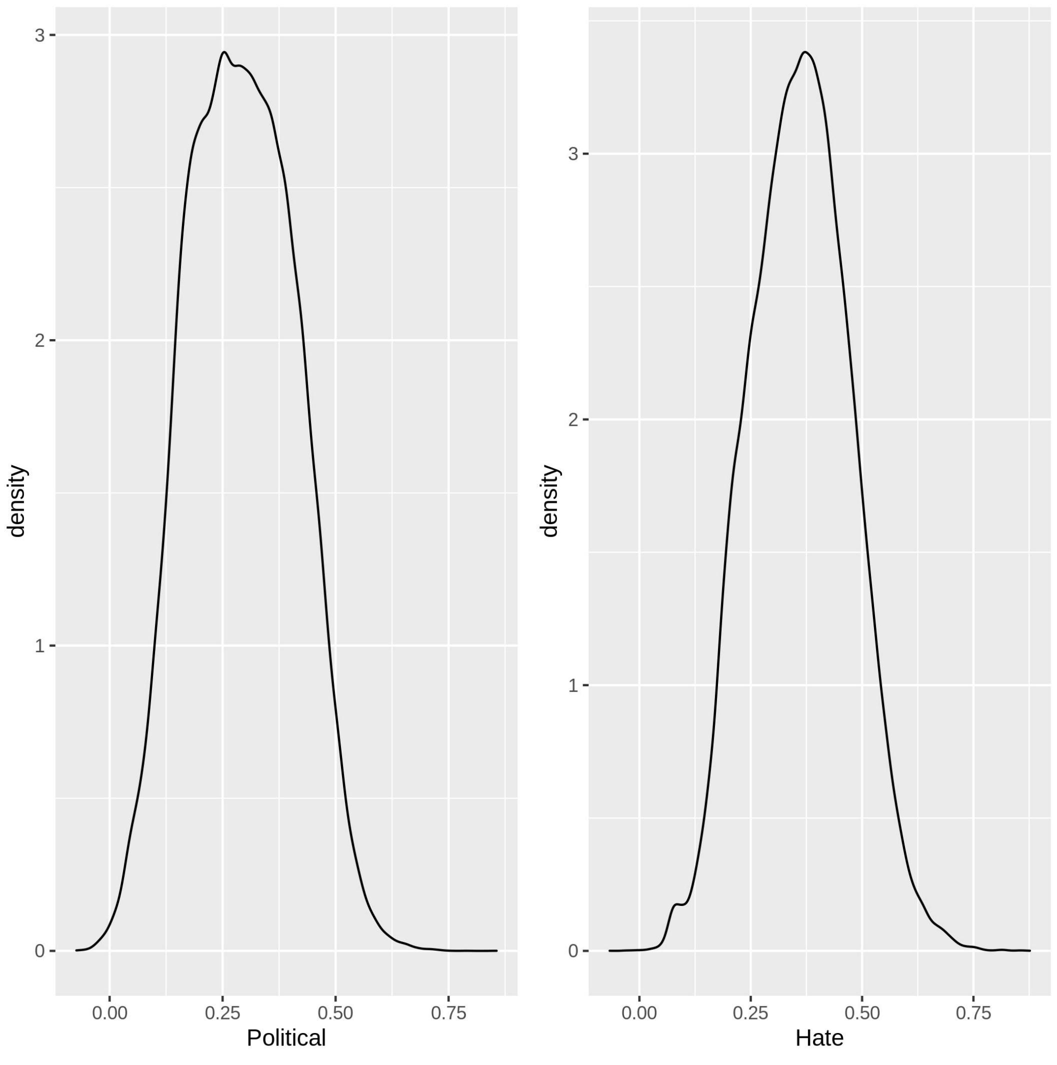

**Figure 2.** This figure shows the distributions for the Political word vector and the Hate word vector for a subset of 100000 observations in the full (i.e. not grouped) dataset

#### *Appendix 1.2.1 Methods and measures*

As Google's perspective API is the only publicly available, comprehensive classifier for online toxicity, it enjoys considerable popularity. That said, obviously it is not without its shortcomings (Salminen et al. 2018; Hosseini et al. 2017; Gröndahl et al. 2018). Most importantly for our case, it is admittedly agnostic to the topic of the discussion and has no features that could be helpful to distinguish between political and other forms of hate.

## *Appendix 1.2.2 Results*

**Table 4.** Rest of sample texts for Toxicity and Political Hate. The sample texts are 10 randomly selected samples among the first 40000 tweets scoring most highly on the particular dimension. The tweets are sorted separately for Toxicity score and Political Hate

| Number | Tweet |
|---|---|
| 0 | This is big. I hope that the wave of banning and forbidding these hate filled books and terrorist symbols continues across the country and around the world. If we could keep the white nationalist out of the white house, that would be radicalizing. |
| 1 | Do not watch this crime. I am. Stand withLeslie. |
| 2 | The plot of Footloose is playing out in Congress. Republicans don't like progressives enjoying life. It's not like she's is a real person like the rest of us. We want real people to represent us. There was no Russian robot. |
| 3 | When it comes to public policy, I'm a big fan of the man, but he's politically ill-informed and naive. I don't care who he supports. He is the greatest of all time. His political knowledge is sad but it is what it is. |
| 4 | Since people began talking about Trump's crime of "soliciting a foreign attack on our election", I have been trying to make this point. It was not a foreign attack. |
| 5 | "I hate Trump because he hates everything I love" should have been the 3rd line. I messed up a bit. |
| 6 | Donald Trump does not inspire people to do good things. He encourages people to commit acts of terror. The video is on his brand. |
| 7 | jsolomon reports Uh-oh! Trump is getting help from Latvia. |
| 9 | Is it possible that people who immediately blame Islam are exploiting attacks for pre-existing agendas? |
| 10 | I have had enough of white nationalism. America is a melting pot. We don't have to give up our identities in order to feel better. You are an American. |
| 14 | Elizabeth Warren is demonizing prosperous businessmen and women because of their success. It is disgusting. It is shameful. |
| 15 | Every time. Single. Democratic. There is a president. A candidate. Should. Be. Striking. Trump. For. Treasonous. There is behavior. For. Political. Gain. Like right now. |
| 16 | Beto taking time to visit a NH community one might consider insignificant in the grand scheme of this political arena meant a lot to our Muslim congregants. He showed up for us and proved that he is against racism and hatred. |
| 17 | The IDW is scattering into personal grievances, fringe treatments, political candidates without unity or rigour, and their final hurrah together seems to have been rallying around an incoherent defense of eugenics. It's sad! |
| 21 | They are useless. |
| 24 | Amazon corporate accounts are a royal pain. |
| 25 | Is it ridiculous for people to want a fair and honest election? Jamie Diamond's dick is sucks. |
| 26 | The stupidest man on this and all the other planets is jason garrett. It's amazing. |
| 28 | How to be a hypocrite. |
| 29 | Even if everyone else on the road doesn't, I will continue to use my blinker. |
| 30 | For a good laugh, watch the first part of the expositionsing metal bands. It's ignorant fucks! |
| 31 | Whistles are blowing. There are shoes that are dropping. The net is widening. The dam is cracking. The fat lady is singing. The Lying Criminal doesn't know what to do. |
| 34 | Here you go. You'll find the most idiotic thing on the internet today. |
| 35 | I'd rather fart on a cork and stick it in my mouth. |
| 37 | The whole street has to listen to your dog. |
| 38 | It's time for me to tell you that you're shit! He is a puppy. That's right, SCREW IT! FACTS ARE FACTS! Don't judge the messenger! |
| 39 | Wow, pat sajak, he sucks. |

### *Appendix 1.3 Convergent and discriminant validity*

#### *Appendix 1.3.1 Methods and measures*

The expert raters were given a brief common description of the coding task and the following definition of online political hate: "Degrading, harassing and intimidating online content grounded in political differences". The annotaters were asked to rate the "political" and the "hate" part separately. The hate rating was conducted on a 1-5 scale with the endpoints "Not at all hostile" to "Very hostile". The "political" scale was binary asking whether the tweet was "political" or "not political". We used the software doccano for the actual labeling (Nakayama et al. 2018). Each rater rated 500 common tweets and 500 unique tweets; the internal alpha reliability of the common ratings were .91 and they are thus to a very high extent measuring the same construct. The measure of political hostility for the experts is a simple summative score of each of these two scales where each variable is rescaled to range between zero and one. In addition to the correlations we also wanted to directly compare the two approaches if used for a binary classification task i.e. hate vs. not-hate. Since both measures are interval scaled we simply cut both into high and low at their respective means, to turn this into a binary classification. We then calculated the "accuracy" i.e. the number of times we get the same result across the two measures of online political hostility. We used this classification into "hostile" vs. "not hostile" as input to create a predictive model. More specifically we used the Python scikit-learn library (Pedregosa et al. 2011) to train a common machine-learning classifier in the form of a random forest algorithm. We used 1000 trees for the model but otherwise used the default settings in scikit-learn. We split the data into training and test and obtained a test accuracy of 77 for our classifier.

The most important feature of language models in the spirit of Google's famous BERT model is the idea of *transformers* (Vaswani et al. 2017). Explaining the details of this architechture goes beyond the scope of this paper but the take away is quite simple: The transformer architecture allows the computation of "context aware" vectors which borrow information from the surrounding words to calculate embedddings for any given word or any given sentence. This is obviously a huge step forward compared to the somewhat simple approach the super-unsupervised approach uses where each word has a fixed vector position no matter which words are surrounding it. We use a specific state-of-the-art transformer architecture that is trained specifically to calculate transformer based embeddings for sentences which we are interested in here: https://huggingface.co/sentence-transformers/all-mpnet-base-v2. We do not tweak the existing preprocessing steps for this architechture but simply follow the existing approach (Reimers and Gurevych 2019). We also tried out a pretrained language model called BERTweet, trained on Twitter data, but found that it did not work for our specific use case https://github.com/VinAIResearch/BERTweet. The correlations between "political hate", measured using this model, and the expert raters measure of political hate is 0.03 whereas it is 0.62, for the chosen model.

The marginal correlations all use pairwise deletion and Spearman correlations, since many of our constructs are not normally distributed. We also need to address outliers, as is typical in machine learning methods (Kuhn, Johnson, et al. 2013, Chapter 3). Since many of our constructs are skewed, we remove observations that are more than 1.5 medians above and below the median before calculating the correlations. The correlations without removing outliers can be found in Table 5. To account for the fact that some respondents in our dataset might tweet more often than others, we calculate all correlations at the individual level; i.e. we group the dataset at the individual level and thus calculate mean scores for our text data (i.e. mean scores for distances, sentiment and toxicity). The ungrouped correlations are quite similar to those obtained here, however, and can be found in Table 6.

The partial correlations are calculated in the statistical computing environment R (R Core Team 2019) using the psych package (Revelle 2019).

We rely on a binary indicator for gender as recorded by YouGov. We asked respondents' level of political interest relying on a standard ANES question measured on a 5-point scale ranging from "Extremely interested" to "Not interested at all." To measure political knowledge, we—again—relied on five factual questions from ANES and recorded the number of correct answers. To tap into partisan affect, we asked what respondents feel when they think about Democrats/Republicans, and list a series of six emotions (angry, frustrated, afraid, hopeful, enthusiastic, proud). Respondents can indicate on a 7-point scale the extent to which they feel each emotion, ranging from "Not at all" to "Very strongly." We aggregated the two scales by flipping negative emotions and taking the averages for each respondent. Finally, we included a 6-item measure of self-reported political hostility developed by Bor and Petersen (2019). Respondents were asked how often they did each of a series of listed behaviors in online political discussions on Twitter (e.g. "I posted or shared content on Twitter which could be taken as a threat or as harassment"; "I got blocked by another Twitter user"), with response options ranging from "1 = Never" to "4 = A few times a week" to "7 = Several times a day."

### *Appendix 1.3.2 Additional results for convergent validity*

As a robustness check, we ran the correlations in Table 4 for ten additional words for the "political hate" vector. The ten words were chosen such that they continually move further and further away from our chosen word in increments of 1000 (i.e. start with "political hate" and choose the word that is the 1000th closest word and next the 2000th closest word and so on). We demonstrate that you have to get 7,000-8,000 words away from the "political hate" vector for the results to reliably disappear. This is illustrated in SM Figure 2. This is comforting, since it implies our results do not hinge on the particular words chosen (i.e. "political" and "hate") but are quite robust across slightly different word choices. If we had chosen the words in Table 3, which are fairly close to the words "political" and "hate," we would thus have achieved quite similar results to those presented here. Conversely this also means that there are limits to how fine-grained analyses can be made using this technique. We are not able to clearly distinguish between the most closely related words in Table 3.

In addition the tables below show slightly different versions of the same correlation table.

**Table 5.** Correlation table for the correlations between measures based on tweets (Political Hate, Political, Hate, Toxicity and Sentiment) and the measures from the survey data (Political interest, political knowledge, gender). This correlation table has not removed any outliers.

| | Toxicity score | Afinn score | Political hate vector | Political vector | Hate vector | Political interest | Political knowledge | Hostility | Female |
|---|---|---|---|---|---|---|---|---|---|
| Toxicity score | 1.00 | -0.71 | 0.69 | 0.56 | 0.75 | 0.24 | 0.23 | 0.26 | -0.06 |
| Afinn score | -0.71 | 1.00 | -0.63 | -0.62 | -0.54 | -0.28 | -0.31 | -0.23 | 0.15 |
| Political hate vector | 0.69 | -0.63 | 1.00 | 0.94 | 0.90 | 0.41 | 0.44 | 0.19 | -0.13 |
| Political vector | 0.56 | -0.62 | 0.94 | 1.00 | 0.70 | 0.45 | 0.48 | 0.16 | -0.21 |
| Hate vector | 0.75 | -0.54 | 0.90 | 0.70 | 1.00 | 0.28 | 0.31 | 0.19 | -0.01 |
| Political interest | 0.24 | -0.28 | 0.41 | 0.45 | 0.28 | 1.00 | 0.48 | 0.10 | -0.18 |
| Political knowledge | 0.23 | -0.31 | 0.44 | 0.48 | 0.31 | 0.48 | 1.00 | 0.04 | -0.30 |
| Hostility | 0.26 | -0.23 | 0.19 | 0.16 | 0.19 | 0.10 | 0.04 | 1.00 | -0.11 |
| Female | -0.06 | 0.15 | -0.13 | -0.21 | -0.01 | -0.18 | -0.30 | -0.11 | 1.00 |

**Table 6.** Correlation table for the correlations between measures based on tweets (Political Hate, Political, Hate, Toxicity and Sentiment) and the measures from the survey data (Political interest, political knowledge, gender).This correlation table is based on the full i.e. not grouped dataset

| | Toxicity score | Afinn score | Political hate vector | Political vector | Hate vector | Political interest | Political knowledge | Hostility | Female |
|---|---|---|---|---|---|---|---|---|---|
| Toxicity score | 1.00 | -0.28 | 0.39 | 0.31 | 0.37 | 0.15 | 0.13 | 0.14 | -0.02 |
| Afinn score | -0.28 | 1.00 | -0.23 | -0.23 | -0.15 | -0.11 | -0.10 | -0.10 | 0.04 |
| Political hate vector | 0.39 | -0.23 | 1.00 | 0.88 | 0.82 | 0.32 | 0.29 | 0.22 | -0.08 |
| Political vector | 0.31 | -0.23 | 0.88 | 1.00 | 0.48 | 0.34 | 0.30 | 0.22 | -0.12 |
| Hate vector | 0.37 | -0.15 | 0.82 | 0.48 | 1.00 | 0.20 | 0.19 | 0.17 | -0.01 |
| Political interest | 0.15 | -0.11 | 0.32 | 0.34 | 0.20 | 1.00 | 0.52 | 0.23 | -0.08 |
| Political knowledge | 0.13 | -0.10 | 0.29 | 0.30 | 0.19 | 0.52 | 1.00 | 0.25 | -0.19 |
| Hostility | 0.14 | -0.10 | 0.22 | 0.22 | 0.17 | 0.23 | 0.25 | 1.00 | -0.22 |
| Female | -0.02 | 0.04 | -0.08 | -0.12 | -0.01 | -0.08 | -0.19 | -0.22 | 1.00 |

### *Appendix 1.3.3 Partial correlations with self-reported measures*

We notice that not only the "political hate" word vector but also the "hate" word vector is correlated with political interest and political knowledge in Table 4 in the main text. Conversely, the "political" vector is also somewhat related to hostility. This means, first of all, that politics, in the minds of the sampled Twitter users, is (to some extent) hate-filled and conflictual. To disentangle what is going on, we have calculated *partial* correlations between the self-reported measures and the word vectors, i.e. the correlations between our "political" word vector and e.g. "hostility," but taking the shared variance between "political" and "hate" into account (Table 7).

**Table 7.** Partial correlations between the word vectors "political" and "hate" for a series of constructs.The partial correlation refers to the correlation between, e.g., the "political" vector and the construct mentioned in the third column after accounting for the shared variance between the "political" and the "hate" word vectors

| Political vector | Hate vector | Construct |
|---|---|---|
| 0.09 | 0.15 | Hostility |
| 0.37 | -0.05 | Political interest |
| 0.38 | -0.04 | Political knowledge |
| 0.07 | 0.60 | Toxicity score |

The correlations between "hate" and political interest and political knowledge completely disappear after accounting for the shared variance between the "political" and the "hate" word vectors. This suggests that the only reason our "hate" word vector is correlated with these self-reported measures of political interest/knowledge is because the "hate" word vector itself is related to political interest. This pattern is also found for the toxicity score measurement.

Conversely, the correlations between self-reported hostility and our "hate" word vector are higher than the correlation between "political" and self-reported hostility after accounting for the shared variance between "political" and "hate."

## *Appendix 1.4 Criterion validity: Named Entity Recognition (NER) for political hate*

Another way to validate whether the "political hate" word vector is extracting the tweets that do in fact contain political hate is to extract the political figures and institutions around which American politics centers. If the tweets are about politics, we should also expect to extract political actors, such as "Trump" and "Democrats." To investigate whether our political hate vector does indeed concern politics, we therefore extract named entities from the 20,000 tweets with the highest scores on our "political hate" vector. This can be seen as a form of criterion validity since we are using an external criterion to validate our approach (Cronbach and Meehl 1955). In doing so, we compare the results with the named entities extracted from the 20,000 tweets with the highest toxicity scores. The results are remarkable: The political person Trump is e.g. mentioned 4163 times among the most politically hostile tweets compared with only 831 times for the top toxic tweets. For the full range of entities extracted see table 8.

### *Appendix 1.4.1 Methods and measures*

To investigate whether our "political hate" word vector indeed extracts political entities, we use spaCy (Honnibal and Montani 2017) to extract named entities. The input for this extraction uses the original texts as opposed to the preprocessed ones, since this would destroy the linguistic structure in the text that the spaCy model is trained on and uses to extract the entities.

The named entities recognition parser we use is *not* specifically trained to extract political figures, so in a sense this is a strong test of the technique: If we can find the patterns using even a general NER parser, the approach has high degrees of validity. Furthermore, we do not restrict the NER to *only* extract political figures and institutions from a predefined list; all NERs that are extracted are simply presented in Table 8.[15]

15. We removed these "extracted entities" "#" and "2" from the top toxicity NERs and "&amp" from the top political hate NERs since these are not words.

## *Appendix 1.4.2 Results*

**Table 8.** Named entity recognition (NER) for "political hate" and toxicity scores.We use the top 20,000 tweets that are closest to the "political hate" word vector and the top 20,000 tweets with the highest toxicity scores.

| Top NER political hate | Count political hate | Top NER toxicity score | Count toxicity score |
|---|---:|---|---:|
| Trump | 4163 | Trump | 831 |
| America | 2123 | one | 291 |
| Democrats | 1296 | Fuck | 269 |
| American | 1147 | America | 261 |
| Americans | 856 | Donald Trump | 243 |
| GOP | 698 | today | 205 |
| Republicans | 498 | @realDonaldTrump | 197 |
| one | 490 | Obama | 173 |
| US | 471 | American | 154 |
| @realDonaldTrump | 467 | NRA | 145 |
| Christianity | 465 | fuckin | 133 |
| Dems | 385 | FUCK | 129 |
| Democrat | 384 | GOP | 129 |
| Obama | 361 | first | 125 |
| Jews | 358 | Republicans | 124 |
| Donald Trump | 356 | Americans | 116 |
| Judeo-Christian | 343 | Twitter | 113 |
| Republican | 311 | Fucking | 95 |
| today | 300 | two | 88 |
| Christian | 292 | Nazi | 85 |

## *Appendix 1.5 External validity*

External validity concerns two separate but related investigations. First, there is the investigation of whether the embeddings obtained in this analysis are specific simply to this dataset and this context or whether a somewhat similar dataset would obtain similar embeddings. If the embeddings are general to other contexts, this would provide for more robust uses of them in different settings. If they are highly variable across time and contexts, this would imply that they either need to be retrained over time (again and again) or that they should only be used in very specific settings. We investigate this by comparing our embeddings to the publicly available TwitterGloVe embeddings.

To investigate the stability of word embeddings, we compare ours to the (by now somewhat old) Twitter embeddings from 2014 that were trained using the GloVe approach to training embeddings (Pennington, Socher, and Manning 2014). We use Procrustes rotation to align the two vector spaces using all common words as anchors, as also done by Hamilton, Leskovec, and Jurafsky (2016) in their interesting study of the changing meaning of word vectors over time.

The second analysis concerns the comparison to an already existing labeled dataset on political hate; i.e., can the super-unsupervised approach actually be used for classification? To make this test as difficult as possible, we use the publicly available dataset from Davidson et al. (2017). This dataset is constructed such that only tweets that are already potentially offensive are included. We are thus not trying to categorize tweets into "neutral" and "hate speech," a potentially much easier task, but instead attempting the more difficult task of trying to distinguish (potentially) hostile tweets into "offensive language," "hate speech" and "neither." If the approach succeeds in even this very difficult situation, chances are that it also works (and works better) in less demanding situations.

In a sense this is slightly beside the point, since the super-unsupervised approach circumvents the need for (a few) manual coders (in this case three) and it also relies on the people who write the tweets, not whether someone externally defines the tweet as "hate speech" or "offensive language." Almost all of the tweets in this dataset have some disagreement among coders, so this perhaps simply highlights the need for contextual information. For instance, an example of a tweet in the "neither" category is "The republican teabaggers hate this but screw em." It can certainly be debated as to whether this is the only category this tweet could belong to. We would thus not expect a 1:1 correspondence with the labeling done here and the original labeling, since the super-unsupervised approach is based on Twitter users' "self-labeling" and the Davidson dataset is based on three random people on crowd-flower labeling each specific tweet. The perception of what constitutes "hate speech" and "hate" might not correspond to the way Twitter users perceive hate in general.

### *Appendix 1.5.1 Classification*

First, we simply calculate the the mean levels of hate in the three categories using our trained model from above to calculate the distances from these external tweets to the "hate" vector. The hate scores increase in accordance with the external labeling: Those who are most hateful (according to our word vector score) are also those most likely to be classified into the "Hate speech" category.

**Table 9.** Mean "hate" vector scores for categories of hate speech

| | Mean scores for hate vector |
|---|---|
| Neither | 0.37 |
| Offensive language | 0.44 |
| Hate speech | 0.47 |

The second test concerns our ability to actually use the suggested super-unsupervised approach to *classify* tweets. We therefore simply split the dataset into the same percentiles as in the original dataset; i.e. the .06 most hateful tweets are assigned to the "hate speech" category (since only .06 are labelled as "hate speech") and the next .78 get assigned to "offensive language" and finally the last are assigned the "neither" category. The balanced accuracy is .47 against a baseline of 1/3. And even on the least likely category of "hate speech" (only .06 of the total number of categorized tweets), we are beating random chance by roughly a factor of three.[16]

**Table 10.** Confusion matrix for predicting hate speech using the "hate" vector

| | Neither | Offensive language | Hate speech |
|---|---|---|---|
| Neither | 0.42 | 0.12 | 0.10 |
| Offensive language | 0.55 | 0.83 | 0.72 |
| Hate speech | 0.03 | 0.05 | 0.18 |

### *Appendix 1.5.2 TwitterGloVe*

After aligning the embeddings across the two datasets, we simply create the word vector "political hate" for the TwitterGloVe dataset in the same way as we did for our own dataset and then calculate the distance from this combined word vector to each of the tweets, as also done above. Then we calculate the correlation between this newly defined "political hate" column in our dataset and our old "political hate" column. The correlation obtained here is .83. This suggests that the embeddings learned here are not overly sensitive to a particular time or context.

16. The total number of observations is 22,168 and there are 1252 tweets categorized as "hate speech." Under independence the expected cell count is $0.6 \cdot 0.6 \cdot 22168 = 79.8$. Since the observed number is 229, this means roughly three times better than random guessing, i.e. $\frac{79}{229} = 2.9$

## Appendix 1.6 Ecological validity: Context matters

### Appendix 1.6.1 Additional methods explanation for main text

To conduct the contextual analysis we use the super-unsupervised approach to train word vectors separately for self-identified Republicans and Democrats. The word vectors are then aligned using the approach described above in SM Section Appendix 1.5 so we are able to directly compare the distances in the two vector spaces.

In order to create the results in Figure 4 in the main text we first extracted tweets from each country using Twitter's research API access. We used the ability to extract tweets from different *languages*, i.e. Danish, Swedish, German and Italian, rather than using geo tagged tweets as this can be a source of bias (Malik et al. 2015). It is not directly possible to sample tweets so we created our own script to do so. Once the data was collected we followed the basic approach of the super-unsupervised approach. The words used to measure "political" across Denmark, Sweden, Germany and Italy were "politik", "politik", "politik", and "politica" and the words used to measure "hate" were "had","hata","hassen" and "odiare".

### Appendix 1.6.2 Additional results

First, we use so-called syntactic dependency parsing to automatically extract the grammatical relations between the words in each tweet; for a further elaboration and recent interesting example of how this can be used to study discretization and delegation in legislative texts, see (Vannoni, Ash, and Morelli 2020). We use this to extract who is a likely "actor" in the text by extracting who the nominal subject is ("nsubj" in spaCy's terminology). Let us use this tweet from a self-identified Democrat as an example: "There has never been a more corrupt, dishonest, anti-American threat to democracy than the modern day GOP. No crimes are important, so long as a Republican commits them." In this tweet, "Republican" is the nominal subject and is doing something perceived as politically hateful. An example tweet from a self-identified Republican can also help illustrate the point: "I hate when a Democrat who has been in his position for years tries to chastise President Trump when if we could get rid of you do nothing liberals we would have it!" In this sentence, "Democrat" is the nominal subject and is doing something perceived as politically hostile towards the in-group, i.e. Trump.

### Appendix 1.6.3 Methods and measures

In order to study who are the actors in the hateful tweets, we use syntactic dependency parsing in spaCy, i.e. break the tweet into its grammatical dependencies. We do this for the top 100,000 politically hostile tweets for each group separately. If the word vectors are picking up on systematic differences between Republicans and Democrats, we would expect the top hostile tweets for Republicans to more often depict the political rival as the main actor.

## *Appendix 1.6.4 Across countries*

Figure 3 illustrates the same development as figure 4 in the main text. The only difference is that this graph illustrates the effect not for our political hate vector but an additive index consisting of the vector "political" and of (negative) sentiment using the Afinn dictionaries for Danish and Swedish (Nielsen 2011), which are unfortunately not available in German or Italian. To keep the results comparable to the the rest of the manuscript we therefore only show results for these two countries.

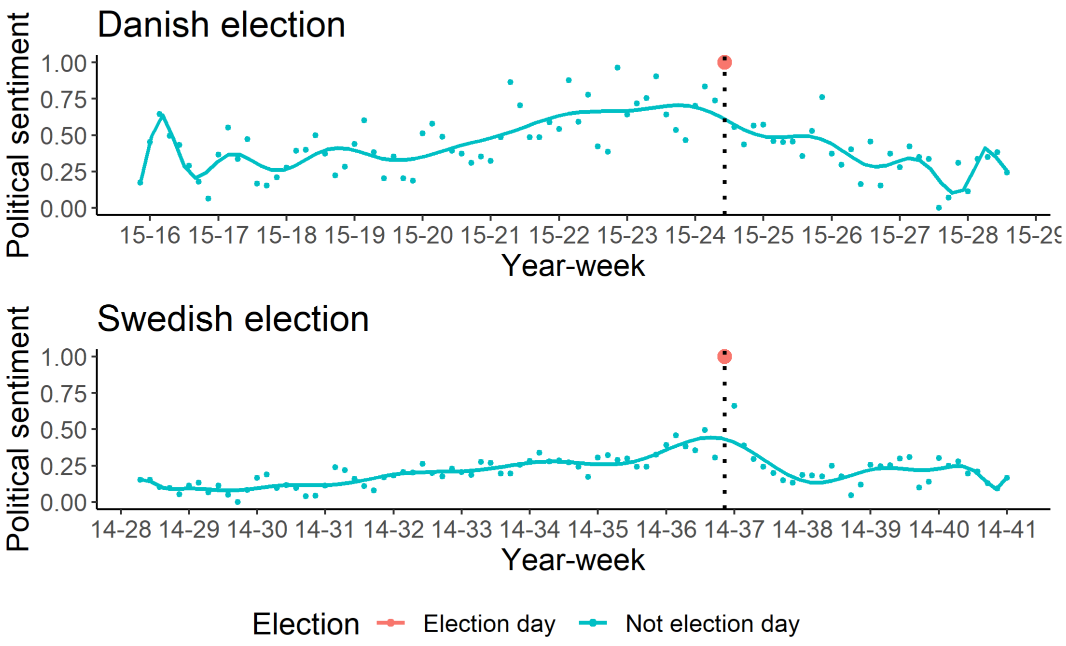

**Figure 3.** The figure illustrates the development of Political (negative) sentiment across Denmark and Sweden in the General Elections of 2014 and 2015 using the popular Afinn dictionary

### *Appendix 1.6.5 Dependency parsing*

As we saw in the example tweets above in Table 2, it is a common strategy to depict the enemy as the main "actor" in the tweet and then portray them as doing something negative. For instance, Tweet 6 states that: "Most presidents inspire people to do good things, but not Donald Trump. He inspires people to do evil things and commit violent acts of terror. The #TrumpVideo is on brand for him." Here the actor is "he," i.e. Trump, and he is acting in a negative way as perceived by this tweet's author.

This expectation is also quite consistently borne out: For those who are self-identified Republicans, "Democrats" and "Dems" and "Obama" are actors in 87, 86 and 75 percent of the most politically hateful tweets. Conversely "Republicans," "GOP" and "Trump" are actors in 71, 78 and 51 percent of the most hateful tweets for self-identified Democrats.

**Table 11.** Using dependency parsing to extract the "actor" in a given tweet.The nominal subject is often in a grammatical sense taken to be the main "actor" in a grammatical relationship. The nominal subject is automatically extracted using the Python library spaCy

| | Grammatical subject in tweet | | | | | | |
|---|---|---|---|---|---|---|---|
| | America | Democrats | Dems | GOP | Obama | Republicans | Trump |
| Self-reported ideology in survey | | | | | | | |
| Democrat | 0.31 | 0.13 | 0.14 | 0.78 | 0.25 | 0.71 | 0.51 |
| Republican | 0.69 | 0.87 | 0.86 | 0.22 | 0.75 | 0.29 | 0.49 |

### *Appendix 1.6.6 Democrats and Republicans: Political hate*

The last set of results for the demonstration of the super-unsupervised approach's contextual usefulness concerns a reproduction of the correlation table above for the two groups. First of all, we note that our "political hate" word vector is related to the same constructs as before, i.e. political knowledge and hostility. This suggests that although we have established that there are important differences above, many of the same self-reported measures we use work quite well across these two defining contexts. Political knowledge seems to be slightly more strongly related to "political hate" for Democrats, however.

As a robustness check we have also included "affect" towards Democrats/Republicans across the two contexts. If the "political hate" vector is accurately depicting the two contexts, we would expect the "political hate" word vector to be more negatively correlated with the out-group, i.e. Democrats should have a higher negative correlation with "Affect towards Republicans" compared to their correlation with "Affect towards Democrats." This is also borne out: In both contexts, the correlations with "political hate" are larger for the out-group. Interestingly, however, the self-identified Republicans also have a negative correlation with "Affect towards Republicans." This suggests that "political hate" is more directed towards elites and authorities for Republicans compared to Democrats. Again, this is much in line with scholarly and journalistic work emphasizing how Republicans are increasingly embracing anti-elitist and populist beliefs (Groshek and Koc-Michalska 2017; Oliver and Rahn 2016).

**Table 12.** Correlation table for self-identified Republicans and Democrats.Democrats are the results above the diagonal and Republicans are below the diagonal

| | Political hate vector | Political knowledge | Hostility | Affect towards Democrats | Affect towards Republicans |
|---|---|---|---|---|---|
| Political hate vector | 1.00 | 0.48 | 0.30 | 0.03 | -0.17 |
| Political knowledge | 0.33 | 1.00 | 0.17 | 0.02 | -0.12 |
| Hostility | 0.27 | 0.19 | 1.00 | -0.02 | -0.13 |
| Affect towards Democrats | -0.15 | -0.13 | 0.04 | 1.00 | -0.11 |
| Affect towards Republicans | -0.10 | -0.23 | -0.11 | -0.31 | 1.00 |

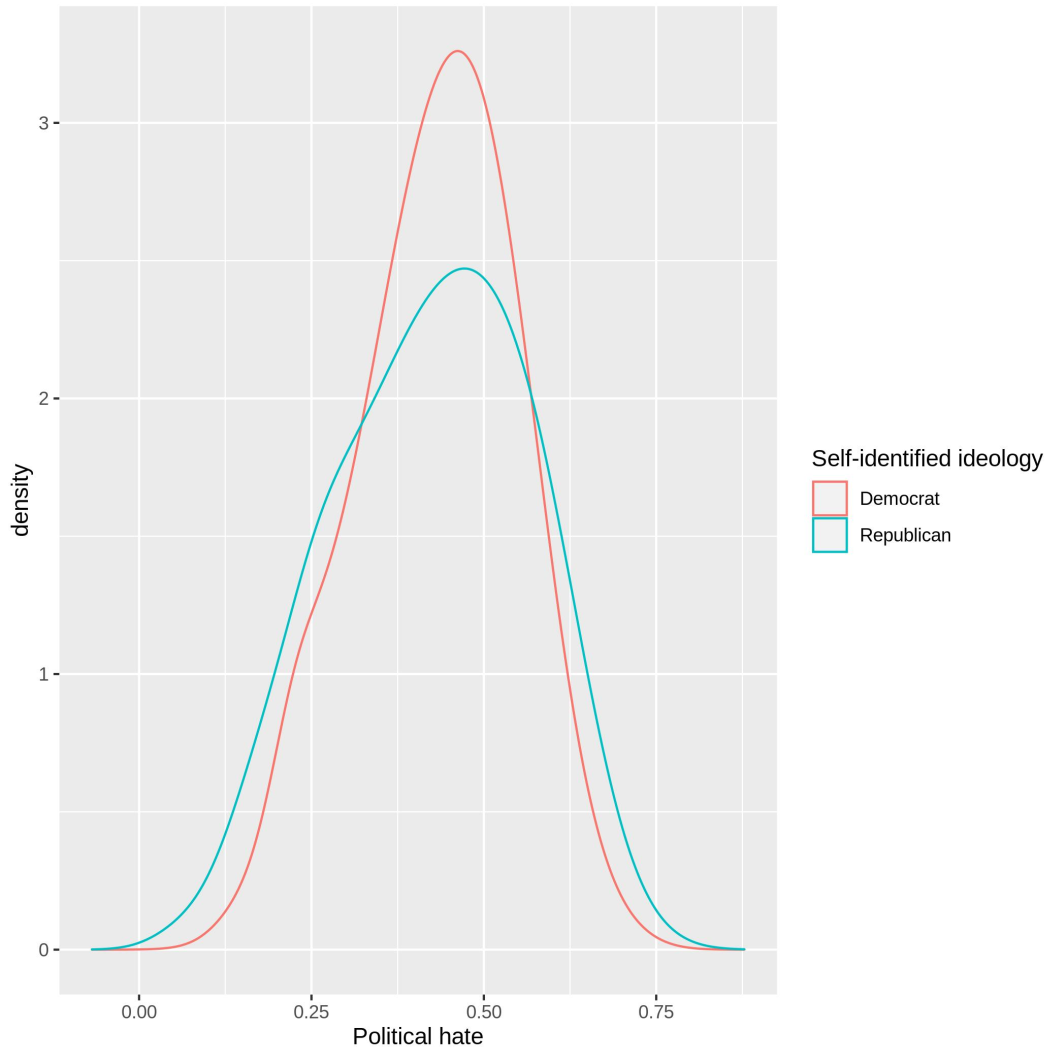

**Figure 4.** This figure shows the distributions for Political Hate scores for Republicans and Democrats in the full (i.e. not grouped) dataset

## Appendix 2. Is the approach measuring actual hate or talk about hate?

First, if the approach measures talk about hate or politics, and not hate or politics per se, we would not expect the measure of political hostility to correlate systematically with neither self-reported measures of hostility and political interest, nor external measures of hostility such as sentiment scores and toxicity scores. Yet, empirically, this is exactly what we observe.

Second we are able to validate our approach against a hand-labeled dataset of 2500 tweets using expert raters. At least, then, external raters are making the same mistake as the super-unsupervised approach.

Third, attack is an important part of hateful tweets. We demonstrate in SM Table 11 that for respondents who self-identify as Democrats, the “attacker” in the 20,000 most politically hateful tweets are Republican actors such as "Republicans" and "GOP" in the vast majority of cases. The reverse is true for self-identified Republicans where the persons doing the attack in the 20,000 most politically hostile tweets are Democratic actors such as "Democrats" and "Dems". If the tweets we term "politically hostile" were mostly measuring talk about hate, these findings are quite counter-intuitive.

Fourth and finally, for the political vector we see no difference between actually being about politics and talking about politics, so the issue is mostly of concern for the "political hate" vector or the "hate" vector. Taking all the validity tests into account, we find it highly unlikely that the majority of politically hostile tweets are talking about hate rather than being about hate, although we cannot rule out a small proportion of examples of this. No current classifiers of political hate that we are aware of can bolster an accuracy of a hundred percent either though.

## Appendix 3. Additional robustness tests for the measurement of Political Hostility

We have also investigated the required sample size for observed correlations to be of roughly the same magnitude to those observed in 4 in the main text. We did this using a bootstrap approach, where we drew 1000 bootstrap samples at a given sample size and then trained the word vector model for each sample and then calculated the correlations for self-reported hostility, political knowledge and the toxicity score. Our results suggest that a sample size of roughly 100,000 tweets is needed. This is illustrated in Figure 1.

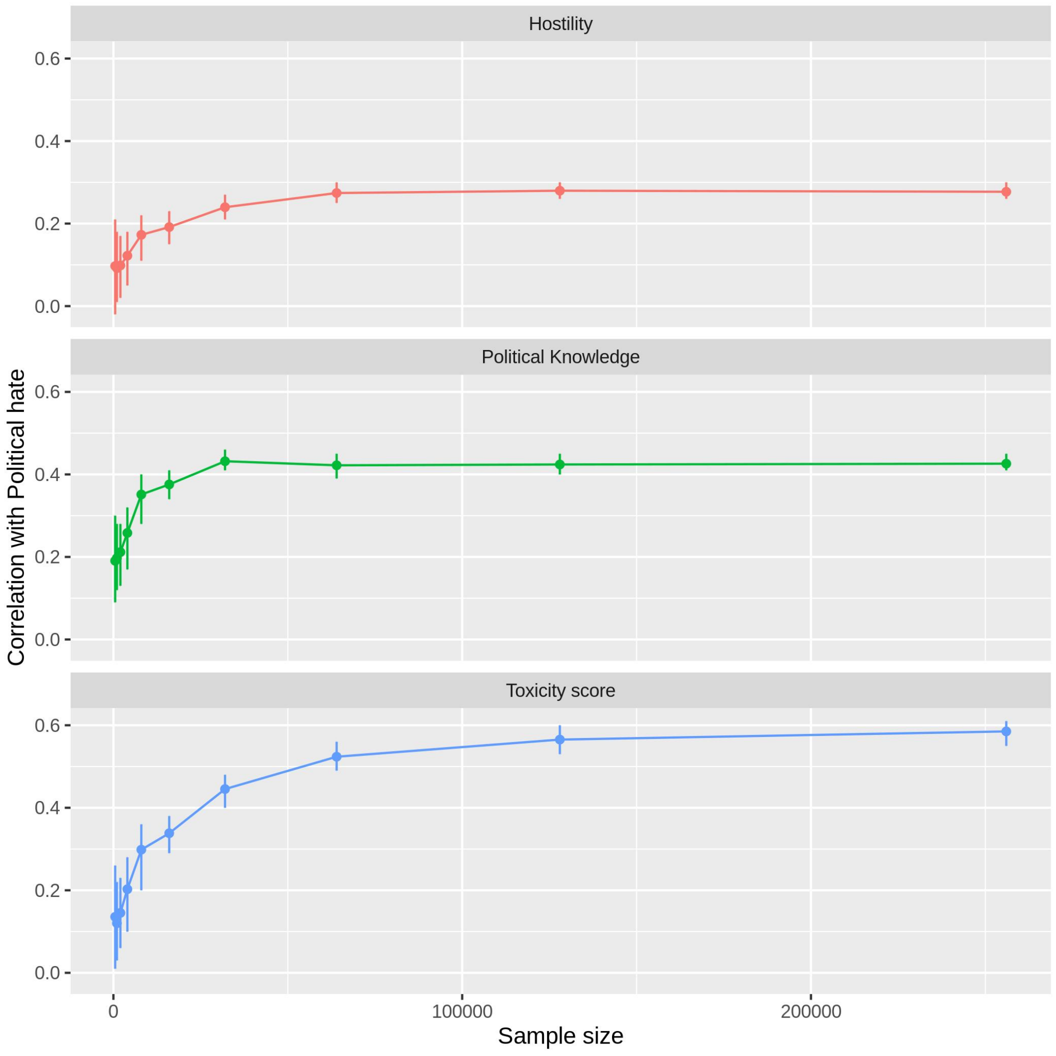

**Figure 1.** Effect of sample size on correlations.Illustration of the correlations between Political hate and the validation measures i.e. Toxicity Scores, Hostility and Political Knowledge as a function of the sample size. 1000 bootstrap samples are drawn where we first train a word embedding model for each sample and then use the model to create the Political Hate vector and then finally calculate the correlation between this vector and the validation measures. The error bars refer to the 97.5 and 2.25 percentiles of the correlations across the bootstrap samples

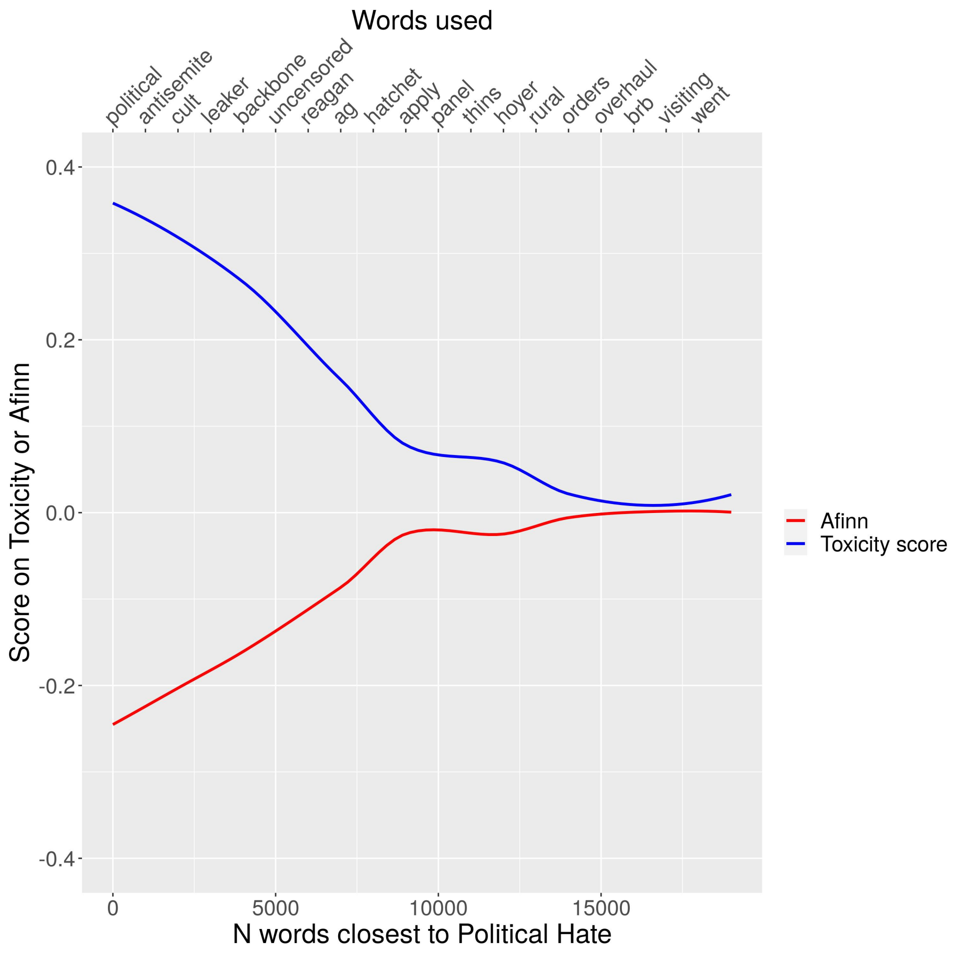

**Figure 2.** Correlations between Afinn/Toxicity scores and words farther and farther from political hate.This figure illustrated how the correlations between Political Hate on the one hand and Toxicity/Afinn scores on the other vary as we choose words that are farther and farther from the Political Hate vector. The first word "Political" is the closest word to Political Hate in the main analysis. We then calculate for each text the distance from this new word vector to the tweet. These distances are then used to calculate a new correlation between these newly calculated distances and Toxicity and Afinn scores for the full (i.e. not grouped) dataset. The second word is the 1000 closest word and so on. The reason we use Afinn and Toxicity here and not self-reported Hostility and Political Knowledge is because these are not measured at an individual tweet level but at the level of the respondents.

### *Appendix 3.1 Incivility and online hostility*

We investigate the relationship between incivility and online hostility in two different ways. First we investigate whether different aspects of Muddiman, McGregor, and Stroud (2019) incivility dictionary measure online hostility (using the super-unsupervised approach), sentiment (using Afinn scores), and/or Toxicity scores. This is illustrated in figure 3. We do the same as in figure 2 where the entire corpus is scored had we used each of the words the incivility dictionary as the target word (i.e. instead of "Political"+"Hate" using the "super-unsupervised" approach. Then we simply calculate the correlation between each of these measures and then our three comparison measures. The correlations between each of the incivility measures and these three other measures are quite inconsistent - they range from correlations between .6 to something around .0. It seems that incivility, using Muddiman, McGregor, and Stroud (2019) is capturing something different than either sentiment, toxicity or online hostility.

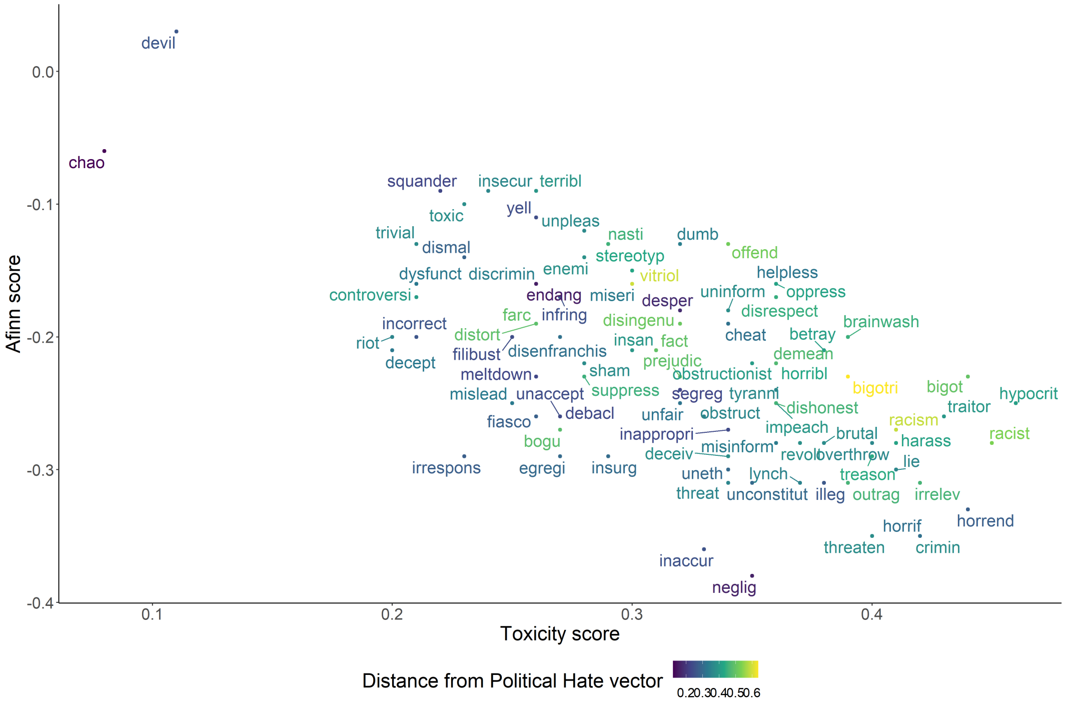

**Figure 3.** Correlations between Afinn/Toxicity scores and the Political Hate vector. The color indicates the distance from the Political Hate vector to the chosen word

In order to investigate further what might be causing these differences we investigate whether the subcategories of incivility outlined in (Muddiman, McGregor, and Stroud 2019) are in fact, in the eyes of the users on Twitter, internally coherent and mutually exclusive. This is illustrated in figure 4. This figure illustrates the first two principal components, from a PCA on the embeddings for each of the words in the incivility dictionary, using our word2vec model trained on the Twitter corpus. None of the categories are both internally coherent and mutually exclusive – as seen from the eyes of the users of Twitter. If they were in fact internally coherent and mutually exclusive each category, and the words it contains, should be close in PCA space to each other, and not overlap with the other catgories. Some are more internally coherent than others, the "Threat" category and the "Discrimination", category, seem to be somewhat internally consistent, but they are not mutually exclusive when compared to the other categories. These results do not mean that the incivility dictionary should not or can not be used, simply that, from the point of view of the super-unsupervised approach, the dictionary is not how users perceive the categories.

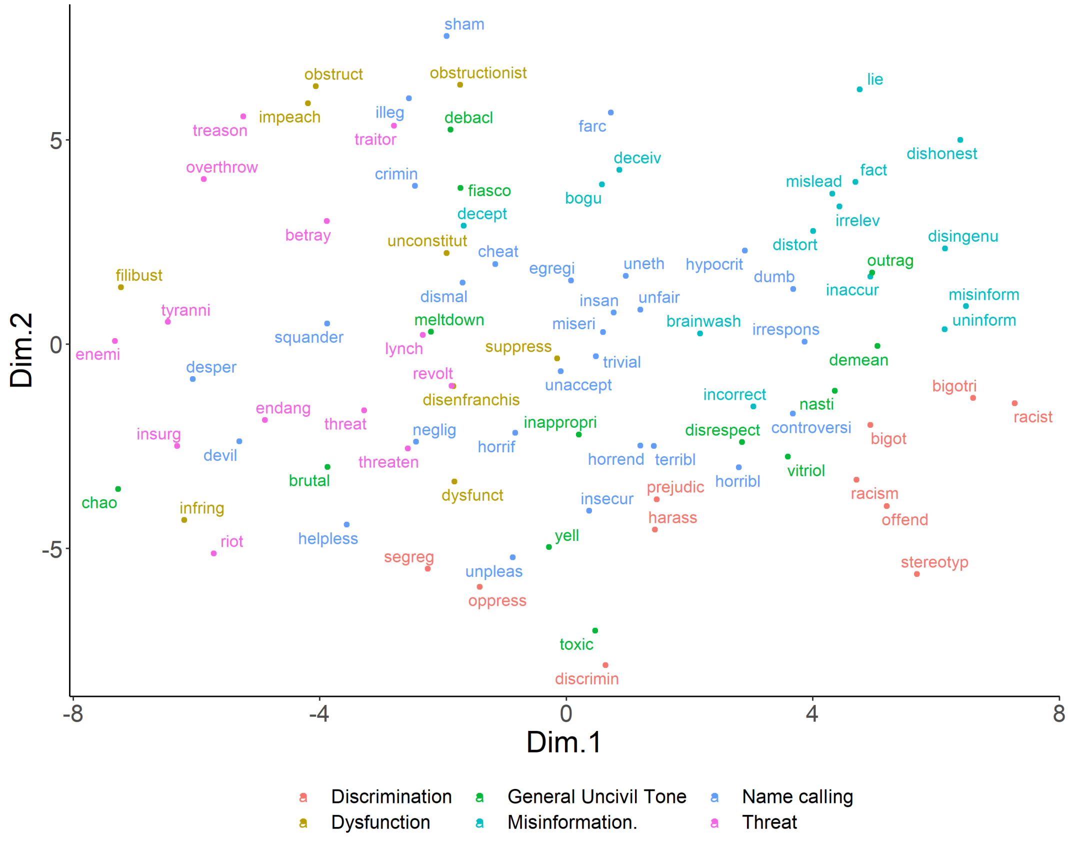

**Figure 4.** This figure illustrates how distinguishable the categories used in Muddimans incivility dictionary from the point of the users. For each word in the dictionary we calculate the cosine-similarity distance to each other word, using our word2vec model used in the rest of the paper, which leaves us with a symmetrical matrix where each cell represents the distance between a pair of words, much as a correlation matrix. We then use PCA on this distance matrix and extract the first two principal components of this PCA. As can be seen the theoretical categories do not easily correspond to the usage of these categories i.e. "language in use", with a partial exception of (some of) the words used to define "Discrimination"

## Appendix 4. Illustrations of the generality of the super-unsupervised technique using alternative constructs

This appendix outlines alternative usages of the SU method. We have chosen constructs and usage cases that demonstrate the *breadth* of the method as well as chosen constructs that demonstrate that it can also be applied to "traditional" sources of text in political science.

### *Appendix 4.1 Breadth of the SU approach*

The first example illustrates how the SU approach can be used to measure "fake news" domains. While the SU approach could also be used to measure what people denounce as "fake", we here use the approach to tackle a more difficult challenge: identifying domains that are, in general, not trustworthy. Current research typically identifies misinformation on the domain level (e.g. Lazer et al. 2018). That is, if someone shares a link from a website with a verified history of sharing dubious stories, it counts as misinformation. Conversely, links that come from news media with stringent editorial standards (e.g. "nytimes.com") are considered "real news" stories. This approach has a number of shortcomings. Most importantly, it makes a crude, binary distinction, effectively isolating the poorest quality media sources. Yet, these sources tend to have relatively little reach and are often short lived. Meanwhile, the news domains not on the "blacklist" range from some of the most rigorous outlets in the US (e.g. "washingtonpost.com") to a set of hyperpartisan news sites (e.g. InfoWars) that distribute normatively concerning content. Accordingly, recent scholarship developed measures of news source quality (Pennycook and Rand 2019), demonstrating that professional fact checkers and crowds of regular citizens have a high agreement on the *degree* of trustworthiness of a news site. While Pennycook and Rand (2019) share trustworthiness ratings for 60 news sources, there are literally thousands of domains that are being shared every day on social media. Ideally, we would want a continuous trustworthiness score for each of them. Here, the SU approach can potentially help.

For the panelists used in the main text, we have also collected a subset of the tweets in their networks (roughly 2,000,000,000 tweets). For each tweet, we extracted the URLs shared in the tweet, if any, and expanded all short URLs using the urlExpander package (Yin 2018). The final preprocessing step was simply extracting the domain from the URL also using the urlExpander package. After this we grouped the domains by year, month, and user. In this way, we get an overall impression of the typical sharing behavior of a user and thus also which types of domains tend to be shared "together", just replacing sentences with user-months. We did *not* use subwords for this application since domains are not closer together simply based on spelling. We then used PCA to reduce the dimensions and have used the three first principal components to illustrate that "fake" and real news seem to occupy different vector spaces using the super-unsupervised approach. This is illustrated in Figure 1.

As can be seen, domains that experts identify as "fake news" sources (in green) occupy a distinct region in vector space, which suggests that *users* are, in fact, aware that "true news" and "fake news" are relatively distinct entities (although they may not use these labels to describe the two entities). To further validate the approach, we simply used the SU approach on the trained corpus. We first calculate a combined vector consisting of "newyorktimes.com" + "washingtonpost.com" since these are the highest quality news sources according to expert raters (with a trustworthiness score of 0.91 on a 0-1 range) (Pennycook and Rand 2019). Next, as for online political hostility, we calculate the distance from this "target construct" for all 49 domains that are both rated by the experts and found in our data. For convenience, we transform this distance measure to refer to news quality with higher scores indicating a smaller distance from the NY Times and Washington Post.

Admittedly, these scores are useful as measures of news quality insofar as they correlate with a trusted benchmark – in our case quality measures based on expert ratings. Indeed, prior to calculating the final super-unsupervised scores, we pre-registered the prediction that these scores correlate highly (Spearman's rho > 0.4) with expert ratings on the domain level as well as among our sample of 590 Twitter users who shared at least one news story. Our pre-registration is available at https://aspredicted.org/blind.php?x=H23_1TM. While (average) expert ratings and super-unsupervised quality scores can be directly correlated, we need to calculate the average user score for both on the second test. In essence, the latter test weights the domains according to their prevalence in our data, meaning that a (dis)agreement on more popular domains counts more than (dis)agreement on rare domains.

Reassuringly, both predictions find support in the data. The original expert ratings and the SU-based scores are highly correlated both at the domain level (Spearman's rho = 0.84, see also Figure 2) and when calculating average scores for users (Spearman's rho = 0.90). Since the distance from the combined "newyorktimes.com" + "washingtonpost.com" has such a high correlation with expert trust ratings, we can now use the SU method to score *all* news domains in the corpus simply using the logic of the SU approach. This vastly increases the statistical power to study misinformation sharing since instead of the original 60 (49) domains, we now know news source quality for all 859 news domains in our data.

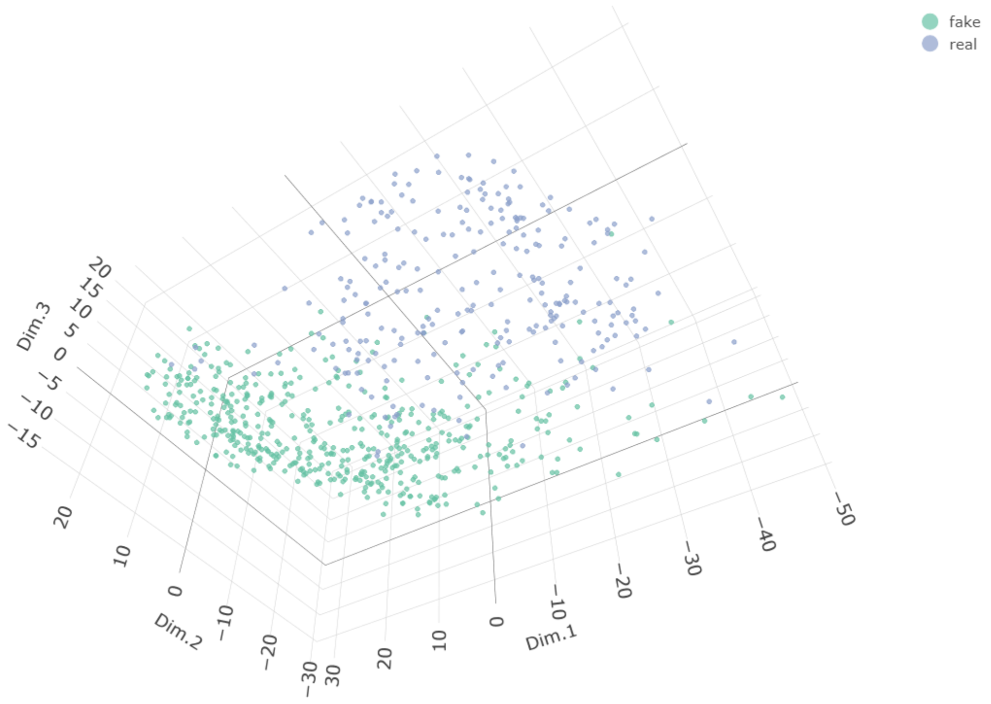

**Figure 1.** This figure illustrates the distinctiveness of users' media sharing behavior using the super-unsupervised approach.

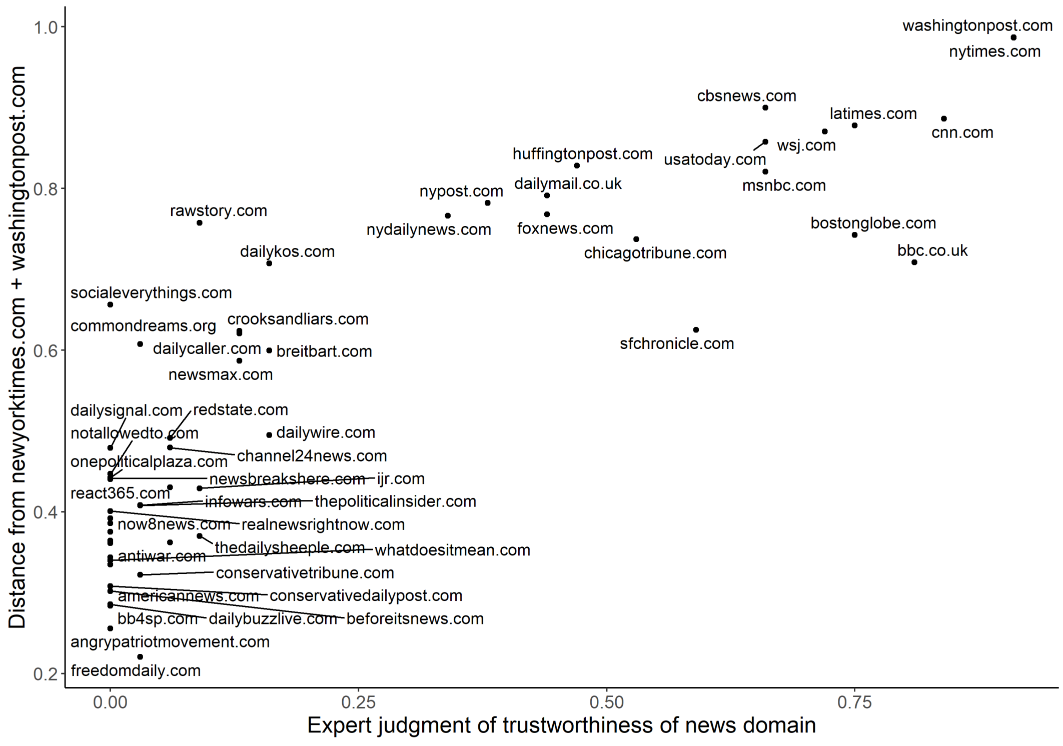

**Figure 2.** This figure illustrates the relationship between the ratings expert fact checkers have given the different domains and the distance score calculated using the super-unsupervised approach.

### *The importance of context - subgroups and time*

#### *Appendix 4.1.1 Subgroups*

We have already used the SU approach to study subgroups, i.e., how political hostility differs across Republicans and Democrats in the main text. It is, however, worth using the approach for a more illustrative example of the importance of how the conceptualization of constructs can vary quite dramatically across subgroups. We therefore used the SU approach to conceptualize how Republicans and Democrats conceptualize "abortion". Specifically, we used the CADE approach to run separate models for these two groups and calculate the 20 closest neighbours to the word abortion. This is illustrated in Figure 3.

As can clearly be seen, the models are measuring quite different things. Simplifying somewhat, Republicans take their point of departure in the view of the (unborn) child, "pro-life", whereas Democrats take their point of departure in the woman's viewpoint, "pro-choice". Had we *not* used the SU approach but, for example, Watanabe's LSS approach and conceptualized abortion in advance, we could clearly miss out on important conceptual differences between these two groups.

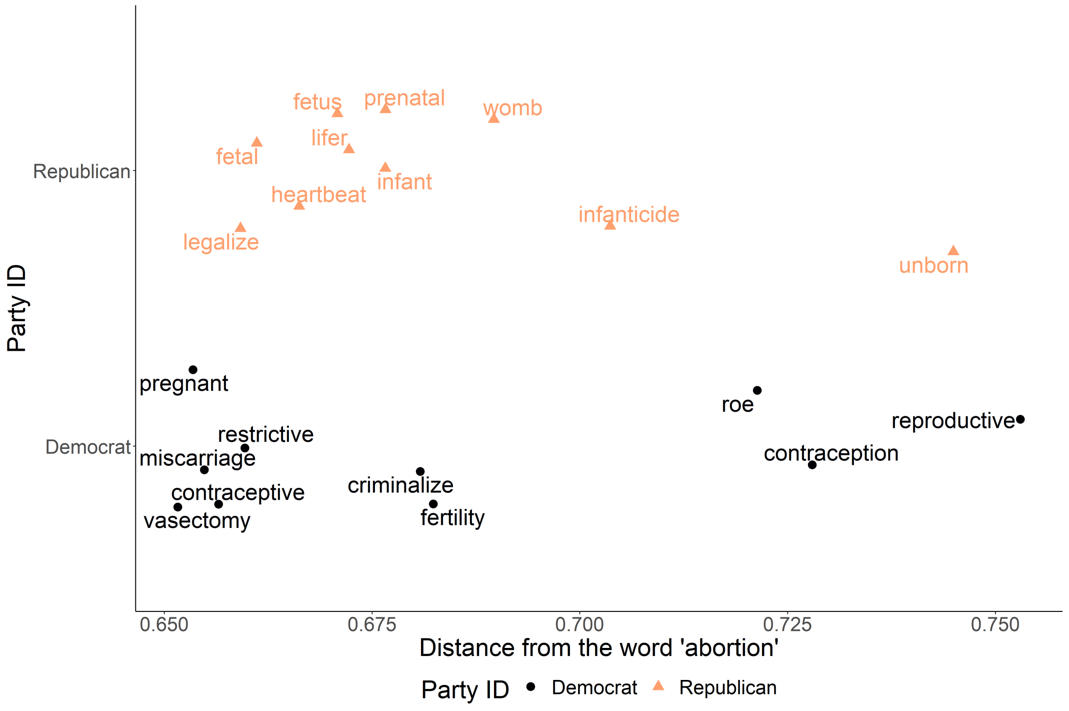

**Figure 3.** This figure illustrates the top 20 closest words not shared for the word embedding model trained for Democrats and Republicans using the word "abortion". We have removed common words to make the differences stand out clearer.

### Appendix 4.1.2 Time

In the main text, we have not yet discussed an example of how the SU approach can be used to study how constructs change meaning over time. In many cases, this could be essential. The logic of the CADE has been outlined above, so we simply use the SU approach to illustrate how it can be used to study the changes in conceptualizations of vaccines over time. We have run the model separately, over time, for Republicans and Democrats to illustrate the flexibility of the approach.

The results are shown in Figure 4. In 2019, i.e., before the advent of the coronavirus pandemic, vaccines were mainly associated with well-known communicable diseases. In 2020, we see that almost all words most closely associated with "vaccine" were related to the *development* of the corona vaccines and other treatments, i.e., words such as "pfizer", "moderna", "remdesivir", and so on. In 2021, as vaccines were rolled out, the words most closely associated with the word "vaccine" concern the *usage* and consequences of the vaccine.

Again, there is both a measurement and conceptualization aspect to these analyses. First of all, we can clearly see from these analyses that the content of what people associated with vaccines differs substantially over time. Had we simply used a unidimensional and never-changing conceptualization of this word, we would have missed out on what is actually going on. Importantly, we would also be missing out measurement-wise since the closest words to this construct change over time. If we wanted to score the closest text to a construct but stick to a general model, some years would probably be way off in terms of the scoring. This problem will become worse as we investigate longer and longer time periods, but here, even a shift in one year makes a huge difference.

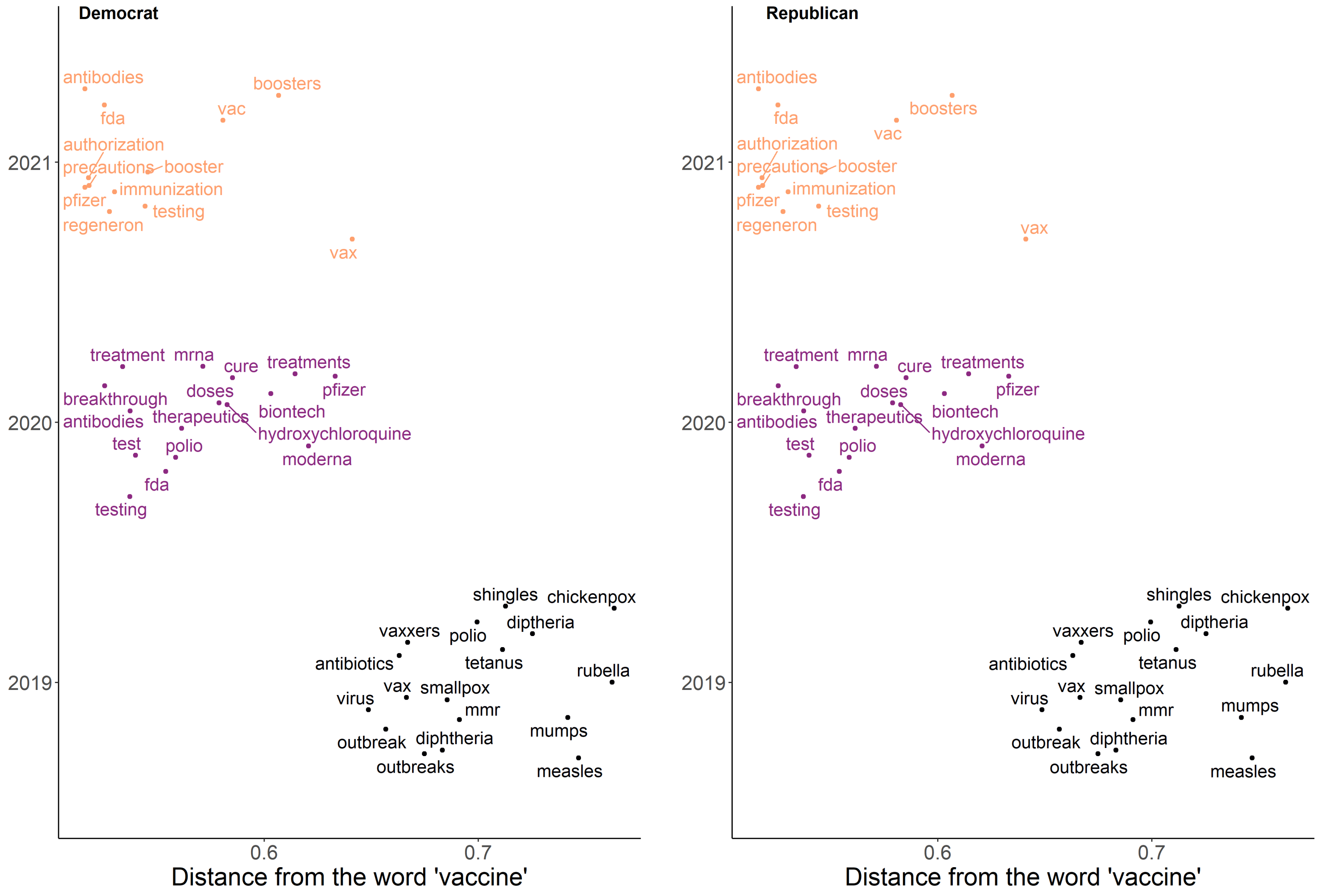

**Figure 4.** This figure illustrates the 20 closest neighbours for the word "vaccine" separately for Democrats and Republicans. Words starting with "vacc". are removed

### *Appendix 4.2 The SU applied to traditional political science sources of quantitative text*

The last example of the bredth of the SU approach focuses on how it can be used to score "traditional" quantitative text used in political science. One of the perhaps most used datasets is data from the Manifesto Project https://manifesto-project.wzb.eu/. We used the R package manifestoR (Lewandowski, Merz, and Regel 2020) to extract all Danish data. This allows us to investigate three things: (1) that the SU approach works on classic political science texts, (2) that it can also be used for scoring long texts, and (3) that pre-trained embeddings can sometimes help increase accuracy - but at the cost of interpretability.

We used the SU in two different ways on the Danish manifesto data. First, we trained a model on all the Danish manifesto data, and second, we used a pre-trained Danish model on Danish Twitter tweets which we collected for a different purpose. This dataset consisted of millions of Twitter tweets, which is quite a bit bigger than the 17,973 texts in the Danish manifesto project with data on the "cmp_code" (although these texts are obviously longer than single tweets).

Each sentence in the Danish manifesto dataset gets one of 112 categories. To illustrate the approach, we have chosen two fairly distinct categories, namely "military" and "education", where we expect the SU approach to work well since they are relatively distinct constructs, and finally a broader category, namely "democracy".[17] We then, following the SU approach, simply score each sentence according to its distance to the chosen word and sort by the distance. Since the SU approach does not use distinct and unique categories, such as the manifesto categorization, but rather provides a continuous score, we have simply taken the top N categories as our point of reference, where N corresponds to the number of sentences that the manifesto data has chosen belongs to this category. This is illustrated in Figure 5.

Figure 5 illustrates for the "domain trained" model that out of the 1207 sentences categorized as "education", the SU model is able to correctly capture 0.56 of these by taking the top 1207 texts closest to the "education" vector. This is remarkably good given that the manifesto data has used a very specific set of theoretical coding instructions, and the fact that there is a total of 14,979 texts/sentences with no missing data.

Another way to look at it is in terms of beating random guessing. If we randomly sampled 1207 sentences over and over again, roughly 97 of these (i.e., $\frac{1207}{14979} \cdot 1207 = 97$) would be correct. The SU model thus does roughly 12 times better than random guessing (i.e., $\frac{1207}{97} = 12.4$). And as we can see for both "military" and "democracy", we do a lot better than random guessing, i.e., roughly 66 and 50 times better than random guessing. Although the "per cent correct" is smaller than for education, we are thus beating random guessing to a very large extent since these are rarer categories. The purple dots, which are called "baseline", reflect how common the category is out of the total number of texts. For instance, we can see that the military categories account for roughly 0.01 of the total texts whereas "education" accounts for roughly 0.07 of the total texts, i.e., a lot more texts contain "education" compared to "military".

Neither the pre-trained model nor the domain-trained model does particularly well for the category "democracy", although they do a lot better than random guessing. This is to be expected as the category is much broader and also more heavily contested.

Figure 5 illustrates the usage of the SU model using the pre-trained Danish model. "Education" does slightly worse whereas the per cent correct for "military" increases from 0.13 to roughly 0.42. Thus, if we are only interested in using the SU approach as a first step in a subsequent labelling exercise, using pre-trained embeddings is something worth trying out. In contrast, if we wish to consider language-in-use, this is likely a bad idea, cf. the discussion in the conclusion in the main text.

17. The specific categories are 104 and 105 for "military", 506 and 507 for education, and 202, 201.1, 201.2, 202.3, and 202.4 for "democracy".

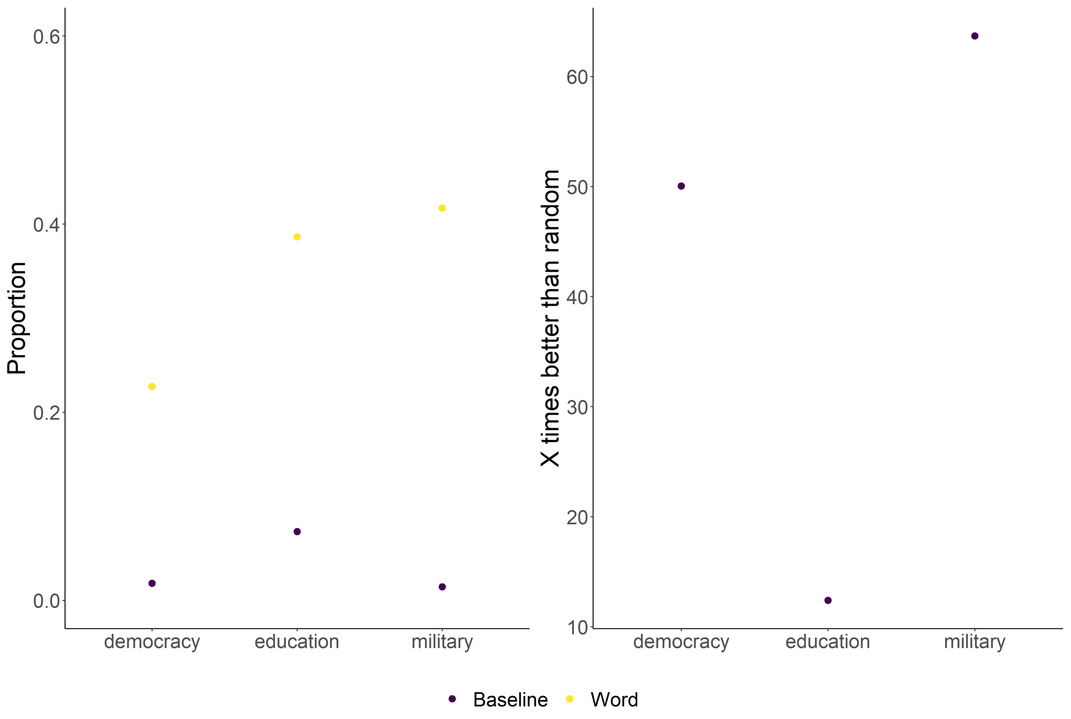

**Figure 5.** This figure illustrates how well the SU approach works for a "domain trained" model on the Danish manifesto dataset. The figure to the left illustrates two things. "Baseline" refers to how common the category, e.g. "military", is out of the total number of texts. "Word" refers to how many times, out of the total number of texts categorized belonging to this category, were also successfully captured by the SU approach. The figure to the right demonstrates how much better than random the SU approach is in terms of accurately choosing the correct category.

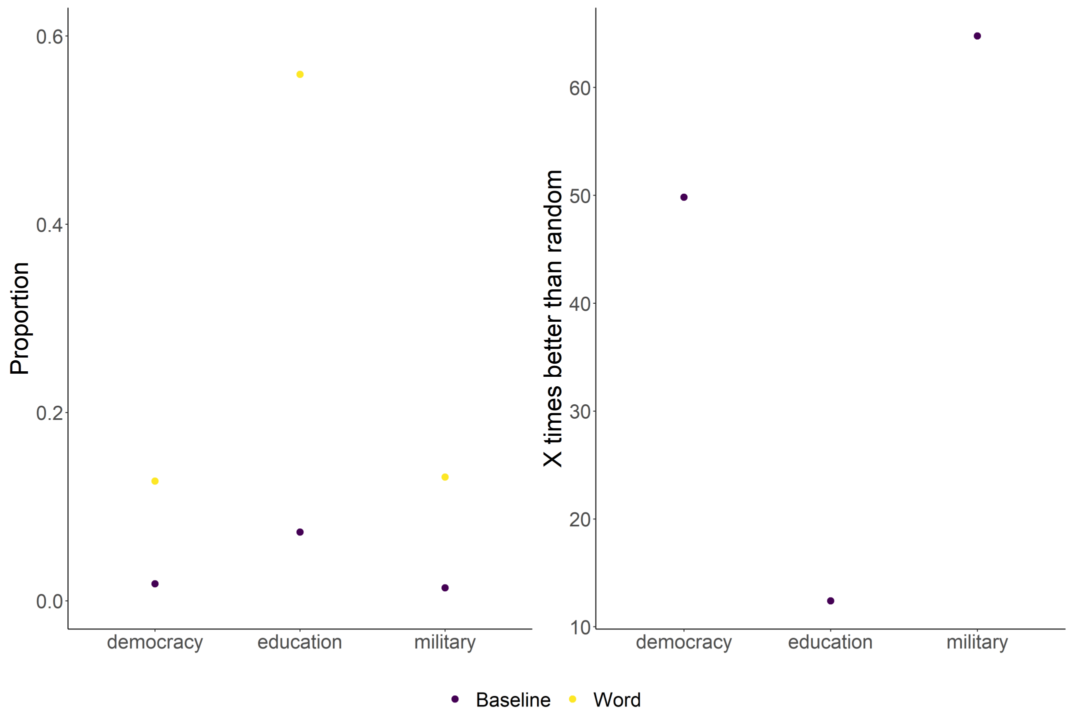

**Figure 6.** This figure illustrates how well the SU approach works using a "pre-trained" model on the Danish manifesto dataset. The figure to the left illustrates two things. "Baseline" refers to how common the category, e.g. "military", is out of the total number of texts. "Word" refers to how many times, out of the total number of texts categorized belonging to this category, were also successfully captured by the SU approach. The figure to the right demonstrates how much better than random the SU approach is in terms of accurately choosing the correct category.

## Appendix 5. "How-to" - applying the super-unsupervised approach for new constructs

Although we argue that the SU approach is ideally suited for studying "political hostility", we would not argue that the SU approach is *always* the best approach. Several other relevant approaches exist and they obtain measures of constructs in quite different ways. The SU approach is *only* relevant if we want to use the users' conceptualizations of constructs. If we, for example, want to measure a legal definition of hate speech, the SU approach is not the right approach. Figure 1 outlines three broad approaches to measurement and how the SU fits in.

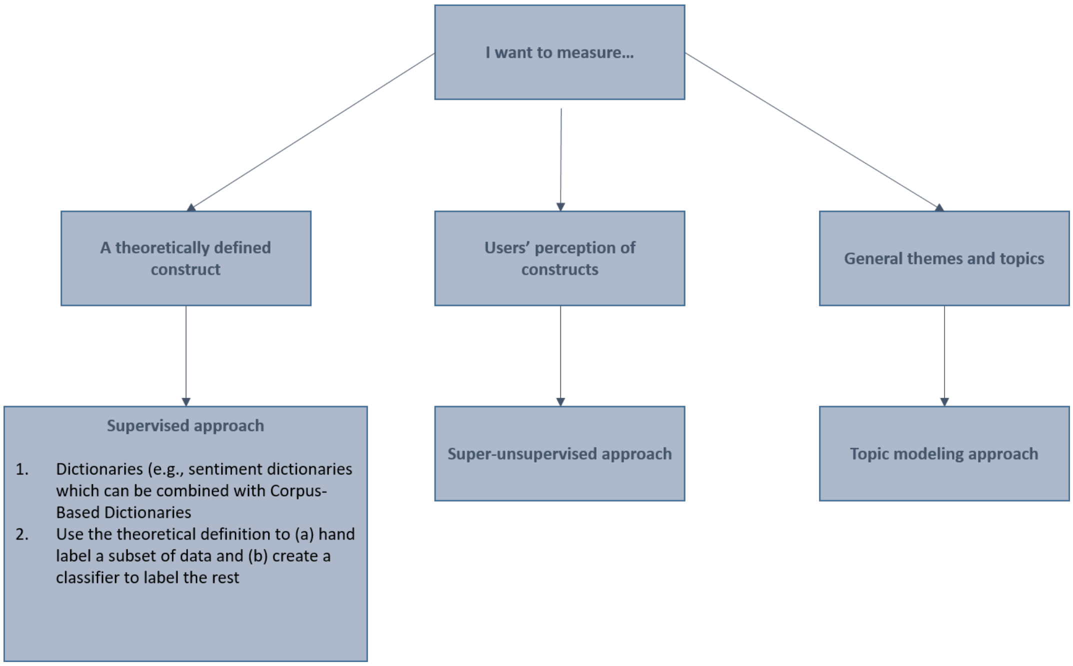
I want to measure...
A theoretically defined construct
Users' perception of constructs
General themes and topics
Supervised approach
1. Dictionaries (e.g., sentiment dictionaries which can be combined with Corpus-Based Dictionaries
2. Use the theoretical definition to (a) hand label a subset of data and (b) create a classifier to label the rest
Super-unsupervised approach
Topic modeling approach

**Figure 1.** Choosing the right quantitative text approach based on your desired measurement strategy

If researchers choose to use the SU because it is relevant for their research question, then Figure 2 outlines how the SU approach can be used for new constructs not related to "political hostility". We will discuss each in turn. Before doing so, an important caveat is in order. The SU approach requires at least some familiarity with the corpus and the users under study. Since the SU uses "language-in-use" to measure and define constructs, it cannot be used in an "automatic" fashion, as should also be clear in the discussion below; substantive knowledge of the corpus and language use of the users is needed.

The first step simply consists of choosing a construct to measure after the preprocessing and actual training of word vectors is done, which could be abortion attitudes, immigration attitudes, "fake news", politics, or literally anything you can come up with. The important thing to remember here is that we are *neither* defining a construct to measure *nor* conceptualizing it. We need to choose a construct that is reflective of language-in-use. If we, for example, tried to use Twitter data to measure how Twitter users conceptualize "ideal forms" as conceptualized by Plato, we would probably come up short. Any construct that is reflective of language-in-use can be measured using the SU approach; much like the measurement of personality traits originally developed out of an idea to use natural language as a way of describing people's personalities. If the construct a researcher chooses is not being used in the corpus under investigation, the SU approach cannot be used.

Once this construct is chosen, a formal first test consists of investigating the "nearest neighbours" of the construct. If the closest neighbours are related conceptually, this is a first important step in the SU approach. If, on the other hand, the nearest neighbours are mostly random words with no common denominator, something has gone wrong. There are multiple reasons this could happen, but in our experience with the SU approach, the most common problem is simply one of *word frequencies*. If the construct under investigation is not used frequently enough, the word2vec method will not be able to tie it to what the users find are its logical neighbours. It is difficult to determine in advance what the "minimum frequency" would be since this is corpus specific, i.e., how many heterogeneous groups there are in the corpus, how distinct the construct you wish to measure is, and so on. Maybe the construct can be captured by increasing the corpus size, if this is a possibility, or by doing more extensive preprocessing such as removing rare words and removing more stopwords. For "political hostility", we found that preprocessing made little difference, but here, the data was also plentiful, so this might make a difference in a smaller corpus. If neither of these options work out, the SU approach cannot be used to measure the construct and the investigation should stop.

At a measurement level, this is a first "face validity" test, but, at a conceptual level, this first face validity test is actually also a first step in conceptualizing the construct. As we saw for Republicans and Democrats in Appendix 4, abortion is conceptualized very differently. Or, more specifically, a word derives meaning from its context and, hence, the other words surrounding it (Jurafski and Martin 2019). The conceptualization of a construct is thus partly explained by doing this first validation step.

Steps 5-6 have already been discussed in the main text so we will not repeat them here but have simply included them for the sake of completeness. After these steps have been completed, the researcher should perform step 7, i.e., a second face validity test. Here, we recommend sampling 5000 texts of the top 10 per cent of the texts and manually inspecting them to investigate what the users are actually writing. Does this correspond to the information contained in the first face validity test, and does this second validity test add something to this first test? Actually, reading the texts might also help put the conceptualization into more focus since we can see the actual texts being used. Maybe this step reveals a disconnect between these texts and the first validity test, or maybe the actual usage in the example texts reveal something completely different from what was initially revealed in the first validity test. Perhaps this requires using a different construct than that originally chosen, or perhaps this requires that we combine several constructs into one such as our example with "political" + "hate" in the main text.

Once this step is finished we can make the final validity test. This consists of calculating a simple marginal correlation matrix with the construct (or constructs if there are more) and external variables that the researcher has information on. Perhaps there is plenty of data, such as in the main text where we have survey data to do additional validity tests, but sometimes, the data is much less plentiful. The better the external data, the better the possibilities for validation of course.

Without these external data, it is difficult to examine what we in Figure 2 have called "Distinctiveness of constructs". Perhaps the researcher is interested in using the SU approach to measure two somewhat related constructs, e.g., "hate" and "incivility", but is unsure whether these two constructs, in the eyes of the users, are distinct enough to treat them as separate constructs. A first obvious test would be to investigate the cosine similarity distance measure. If the cosine similarity is close to zero, they are distinct. If, however, they are moderately related, then we can perform steps 2 and 4 again in order to investigate whether users conceptualize them differently. There is, however, an additional option. This requires the use of external variables as we have done in Appendix 3 in Figure 2. Here, we we have moved farther and farther away from the words closest to "political hate" to see when the correlation with Toxicity Scores and AFINN scores completely disappears. A similar approach could also be done in general if we wanted to investigate how closely related two words are: (1) pick a target word, (2) pick as many words as desired until you get to the "somewhat related" second word, and (3) calculate correlations with relevant external variables and see if they systematically increase or decrease. If nothing happens, they are not distinct, but if they systematically increase/decrease, they are distinct.

We have also included two additional analyses that might be relevant in a particular investigation but are not needed in general if one wants to use the SU approach; specifically, that constructs can change conceptualizations over time (such as for the word "vaccine" and across groups such as the word "abortion"). We have already discussed these two examples in Appendix 4 above.

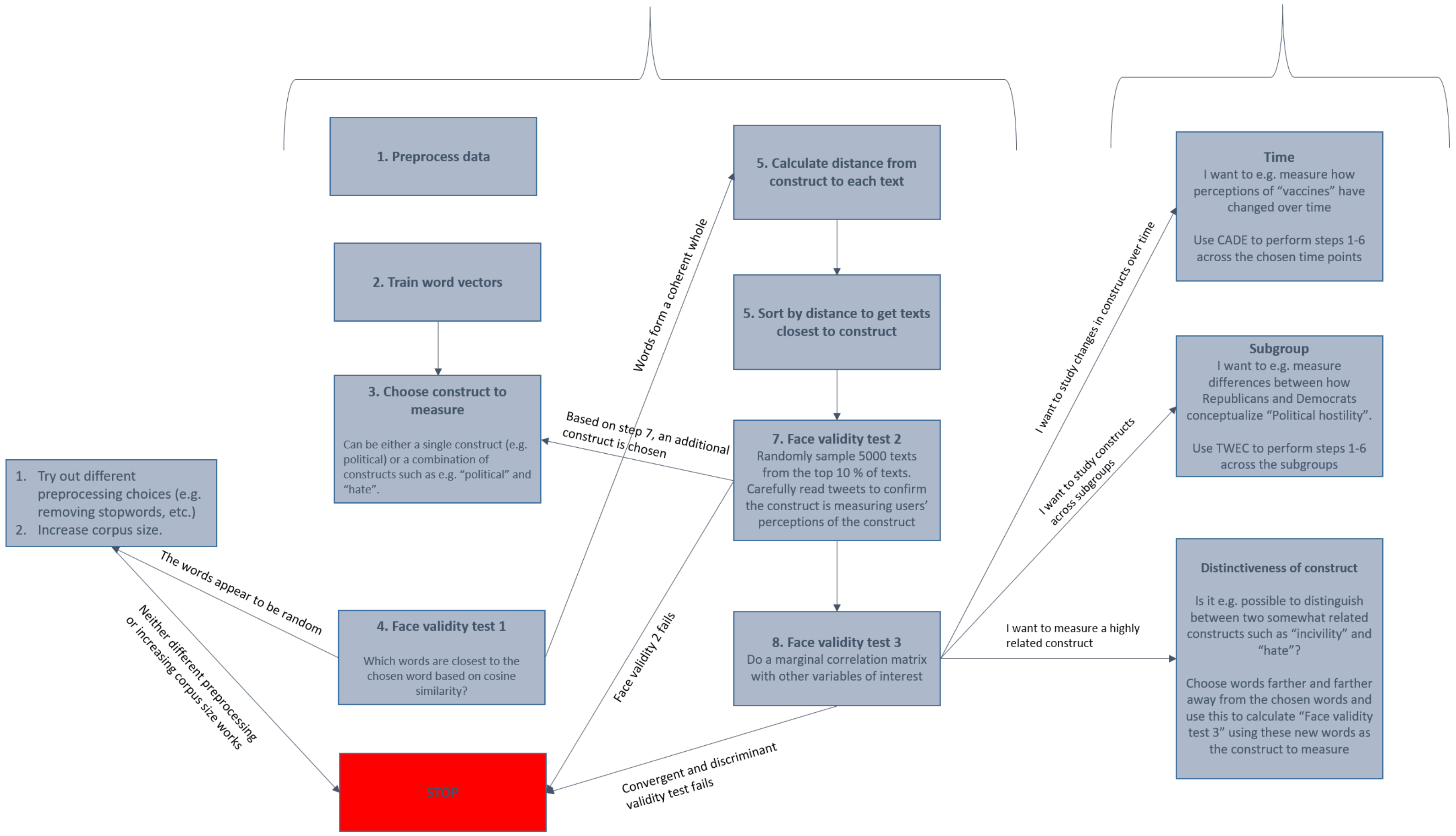
Basic approach
Additional potential analyses
1. Preprocess data
2. Train word vectors
3. Choose construct to measure
Can be either a single construct (e.g. political) or a combination of constructs such as e.g. "political" and "hate".
4. Face validity test 1
Which words are closest to the chosen word based on cosine similarity?
STOP
1. Try out different preprocessing choices (e.g. removing stopwords, etc.)
2. Increase corpus size.
The words appear to be random
Neither different preprocessing or increasing corpus size works
Words form a coherent whole
5. Calculate distance from construct to each text
5. Sort by distance to get texts closest to construct
7. Face validity test 2
Randomly sample 5000 texts from the top 10 % of texts. Carefully read tweets to confirm the construct is measuring users' perceptions of the construct
Based on step 7, an additional construct is chosen
Face validity 2 fails
8. Face validity test 3
Do a marginal correlation matrix with other variables of interest
Convergent and discriminant validity test fails
I want to study changes in constructs over time
I want to study constructs across subgroups
I want to measure a highly related construct
Time
I want to e.g. measure how perceptions of "vaccines" have changed over time
Use CADE to perform steps 1-6 across the chosen time points
Subgroup
I want to e.g. measure differences between how Republicans and Democrats conceptualize "Political hostility".
Use TWEC to perform steps 1-6 across the subgroups
Distinctiveness of construct
Is it e.g. possible to distinguish between two somewhat related constructs such as "incivility" and "hate"?
Choose words farther and farther away from the chosen words and use this to calculate "Face validity test 3" using these new words as the construct to measure

**Figure 2.** Process describing how to apply the super-unsupervised approach on new constructs

## References

Bianchi, Federico, Valerio Di Carlo, Paolo Nicoli, and Matteo Palmonari. 2020. Compass-aligned distributional embeddings for studying semantic differences across corpora. *arXiv preprint arXiv:2004.06519.*

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5:135–146.

Bor, Alexander, and Michael Bang Petersen. 2019. The psychology of online political hostility: a comprehensive, cross-national test of the mismatch hypothesis.

Cronbach, Lee J, and Paul E Meehl. 1955. Construct validity in psychological tests. *Psychological bulletin* 52 (4): 281.

Davidson, Thomas, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th international aaai conference on web and social media,* 512–515. ICWSM '17. Montreal, Canada.

Denny, Matthew J., and Arthur Spirling. 2018. Text preprocessing for unsupervised learning: why it matters, when it misleads, and what to do about it. *Political Analysis* 26 (2): 168–189.

Di Carlo, Valerio, Federico Bianchi, and Matteo Palmonari. 2019. Training temporal word embeddings with a compass. In *Proceedings of the aaai conference on artificial intelligence,* 33:6326–6334. 01.

Gröndahl, Tommi, Luca Pajola, Mika Juuti, Mauro Conti, and N Asokan. 2018. All you need is" love" evading hate speech detection. In *Proceedings of the 11th acm workshop on artificial intelligence and security,* 2–12.

Groshek, Jacob, and Karolina Koc-Michalska. 2017. Helping populism win? social media use, filter bubbles, and support for populist presidential candidates in the 2016 us election campaign. *Information, Communication & Society* 20 (9): 1389–1407.

Hamilton, William L, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic word embeddings reveal statistical laws of semantic change. *arXiv preprint arXiv:1605.09096.*

Honnibal, Matthew, and Ines Montani. 2017. spaCy 2: natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Hosseini, Hossein, Sreeram Kannan, Baosen Zhang, and Radha Poovendran. 2017. Deceiving google's perspective api built for detecting toxic comments. *arXiv preprint arXiv:1702.08138.*

Jurafski, D., and J.H. Martin. 2019. *Speech and language processing.* https://web.stanford.edu/~jurafsky/slp3/.

Kuhn, Max, Kjell Johnson, et al. 2013. *Applied predictive modeling.* Vol. 26. Springer.

Lazer, David MJ, Matthew A Baum, Yochai Benkler, Adam J Berinsky, Kelly M Greenhill, Filippo Menczer, Miriam J Metzger, Brendan Nyhan, Gordon Pennycook, David Rothschild, et al. 2018. The science of fake news. *Science* 359 (6380): 1094–1096.

Le, Quoc, and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning,* 1188–1196.

Lewandowski, Jirka, Nicolas Merz, and Sven Regel. 2020. *Manifestor: access and process data and documents of the manifesto project.* R package version 1.5.0. https://CRAN.R-project.org/package=manifestoR.

Malik, Momin, Hemank Lamba, Constantine Nakos, and Jürgen Pfeffer. 2015. Population bias in geotagged tweets. *Proceedings of the International AAAI Conference on Web and Social Media* 9 (44): 18–27. ISSN: 2334-0770.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781.*

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems,* 3111–3119.

Muddiman, Ashley, Shannon C McGregor, and Natalie Jomini Stroud. 2019. (re) claiming our expertise: parsing large text corpora with manually validated and organic dictionaries. *Political Communication* 36 (2): 214–226.

Nakayama, Hiroki, Takahiro Kubo, Junya Kamura, Yasufumi Taniguchi, and Xu Liang. 2018. *doccano: text annotation tool for human.* Software available from https://github.com/doccano/doccano. https://github.com/doccano/doccano.

Nielsen, Finn Årup. 2011. A new anew: evaluation of a word list for sentiment analysis in microblogs. *arXiv preprint arXiv:1103.2903.*

Oliver, J. Eric, and Wendy M. Rahn. 2016. Rise of the trumpenvolk. *The ANNALS of the American Academy of Political and Social Science* 667, no. 1 (August): 189–206.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al. 2011. Scikit-learn: machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Pennington, Jeffrey, Richard Socher, and Christopher D Manning. 2014. Glove: global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp),* 1532–1543.

Pennycook, Gordon, and David G Rand. 2019. Fighting misinformation on social media using crowdsourced judgments of news source quality. *Proceedings of the National Academy of Sciences* 116 (7): 2521–2526.

R Core Team. 2019. *R: a language and environment for statistical computing.* Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Řehůřek, Radim, and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora [in English]. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks,* 45–50. Http://is.muni.cz/publication/884893/en. Valletta, Malta: ELRA, May.

Reimers, Nils, and Iryna Gurevych. 2019. Sentence-bert: sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing.* Association for Computational Linguistics, November. https://arxiv.org/abs/1908.10084.

Revelle, William. 2019. *Psych: procedures for psychological, psychometric, and personality research.* R package version 1.9.12. Evanston, Illinois: Northwestern University. https://CRAN.R-project.org/package=psych.

Rheault, Ludovic, and Christopher Cochrane. 2020. Word embeddings for the analysis of ideological placement in parliamentary corpora. *Political Analysis* 28 (1): 112–133.

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

Salminen, Joni, Hind Almerekhi, Milica Milenkovic, Soon-Gyu JUNG, Haewoon KWAK, and Bernard J JANSEN. 2018. Anatomy of online hate: developing a taxonomy and machine learning models for identifying and classifying hate in online news media.

Søgaard, Anders, Ivan Vulić, Sebastian Ruder, and Manaal Faruqui. 2019. Cross-lingual word embeddings. *Synthesis Lectures on Human Language Technologies* 12 (2): 1–132.

Vannoni, Matia, Elliott Ash, and Massimo Morelli. 2020. Measuring discretion and delegation in legislative texts: methods and application to us states. *Political Analysis,* 1–15. https://doi.org/10.1017/pan.2020.9.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems,* 5998–6008.

Yin, Leon. 2018. *Smappnyu/urlexpander: initial release,* August. https://doi.org/10.5281/zenodo.1345144. https://doi.org/10.5281/zenodo.1345144.

Zhang, Aston, Zachary C. Lipton, Mu Li, and Alexander J. Smola. 2020. *Dive into deep learning.* Https://d2l.ai.