Sequencing Legal DNA

NLP for Law and Political Economy

9. Document Embeddings

Objectives

1. **What is the research question?**

Objectives

1. **What is the research question?**
2. Corpus and Data:
   - obtain, clean, preprocess, and link.
   - Produce descriptive visuals and statistics on the text and metadata

Objectives

1. **What is the research question?**
2. Corpus and Data:
   - obtain, clean, preprocess, and link.
   - Produce descriptive visuals and statistics on the text and metadata
3. Embedding phrases and documents:
   - **What are we trying to measure?**

Objectives

1. **What is the research question?**
2. Corpus and Data:
   - obtain, clean, preprocess, and link.
   - Produce descriptive visuals and statistics on the text and metadata
3. Embedding phrases and documents:
   - **What are we trying to measure?**
   - Select a model and train it.
   - Probe sensitivity to hyperparameters.
   - Validate that the model is measuring what we want.

Objectives

1. **What is the research question?**
2. Corpus and Data:
   - obtain, clean, preprocess, and link.
   - Produce descriptive visuals and statistics on the text and metadata
3. Embedding phrases and documents:
   - **What are we trying to measure?**
   - Select a model and train it.
   - Probe sensitivity to hyperparameters.
   - Validate that the model is measuring what we want.
4. Empirical analysis
   - Produce statistics or predictions with the trained model.
   - **Answer the research question.**

Vectorizing Documents

- Quantitative analysis of language requires that documents be transformed to numbers – that is, vectors.
- We started with the baseline approach: documents become sparse vectors of token counts/frequencies.

Vectorizing Documents

- Quantitative analysis of language requires that documents be transformed to numbers – that is, vectors.
- We started with the baseline approach: documents become sparse vectors of token counts/frequencies.
  - high-dimensionality can cause issues, but sparsity mitigates.
  - can use documents of arbitrary length
  - can capture local word order with n-grams, but long-run word order is lost.

Embedding layers

- Previously, we introduced embedding layers:
  - take the whole document as input, pad documents to the same length, and represent the document as a flattened series of embedding vectors.

Embedding layers

- Previously, we introduced embedding layers:
  - take the whole document as input, pad documents to the same length, and represent the document as a flattened series of embedding vectors.
  - potentially captures information on long-range ordering of features in documents
  - DNNs work better with dense vectors
  - computationally demanding
  - only works with short documents

Outline

Outline

From Word Vectors to Document Vectors

$$\vec{D} = \sum_{w \in D} a_w \vec{w}$$

- The "continuous bag of words" representation for document D is the sum, or the average (potentially weighted by a_w), of the vectors $\vec{w}$ for each word w in the document.
  - word vectors $\vec{w}$ constructed using Word2Vec or GloVe (pre-trained or trained on the corpus).
  - "Document" could be sentence, paragraph, section, etc.
- Wieting et al (2016) find that this simple representation often out-performs complex recurrent architectures, for example on sentence entailment.

$$\vec{D} = \sum_{w \in D} a_w \vec{w}$$

- Can filter tokens – drop stopwords or filter on parts of speech (e.g., keep only nouns, adjectives, and verbs)

$$\vec{D} = \sum_{w \in D} a_w \vec{w}$$

- Can filter tokens – drop stopwords or filter on parts of speech (e.g., keep only nouns, adjectives, and verbs)
- Token weighting:
  - set a_w to weight words by inverse term frequency or inverse document frequency (that is, up-weight rare/informative words)

$$\vec{D} = \sum_{w \in D} a_w \vec{w}$$

- Can filter tokens – drop stopwords or filter on parts of speech (e.g., keep only nouns, adjectives, and verbs)
- Token weighting:
  - set a_w to weight words by inverse term frequency or inverse document frequency (that is, up-weight rare/informative words)
  - **Arora, Liang, and Ma (2017)** provide a "tough to beat baseline", the SIF-weighted ("smoothed inverse frequency") average of the vectors:

    $$a_w = \frac{\alpha}{\alpha + p_w}$$

    where p_w is the probability (frequency) of the word and $\alpha = .001$ is a smoothing parameter.
  - they also take out the first principal component of the matrix of document embeddings.

$$\vec{D} = \sum_{w \in D} a_w \vec{w}$$

- Can filter tokens – drop stopwords or filter on parts of speech (e.g., keep only nouns, adjectives, and verbs)
- Token weighting:
  - set a_w to weight words by inverse term frequency or inverse document frequency (that is, up-weight rare/informative words)
  - **Arora, Liang, and Ma (2017)** provide a "tough to beat baseline", the SIF-weighted ("smoothed inverse frequency") average of the vectors:

    $$a_w = \frac{\alpha}{\alpha + p_w}$$

    where p_w is the probability (frequency) of the word and $\alpha = .001$ is a smoothing parameter.
  - they also take out the first principal component of the matrix of document embeddings.
- Can normalize weights to sum to one:

$$\vec{D} = \frac{1}{\sum_{w \in D} a_w} \sum_{w \in D} a_w \vec{w}$$

- many other options/possibilities – e.g. weighting by location in document

Outline

- Rhetoric plays central role in political theory of representative and deliberative democracy (Dryzek, 2010)
- But rhetoric can also serve politicians' goals to maximize votes by persuading or mobilizing voters (Riker 1986)

- Rhetoric plays central role in political theory of representative and deliberative democracy (Dryzek, 2010)
- But rhetoric can also serve politicians' goals to maximize votes by persuading or mobilizing voters (Riker 1986)
- More specifically: an important dimension in political communication is that between **reason** and **emotion**.
  - Empathy influences politics (Gault & Sabini, 2000)
  - Individuals vary in response to political arguments phrased in emotional language (Loewen et al., 2017)

- Rhetoric plays central role in political theory of representative and deliberative democracy (Dryzek, 2010)
- But rhetoric can also serve politicians' goals to maximize votes by persuading or mobilizing voters (Riker 1986)

- More specifically: an important dimension in political communication is that between **reason** and **emotion**.
  - Empathy influences politics (Gault & Sabini, 2000)
  - Individuals vary in response to political arguments phrased in emotional language (Loewen et al., 2017)

- This paper:
  - Build new measure of cognitive/emotive valence in text.
  - Apply to speeches of U.S. Congress members.
  - Analyze behavioral/institutational determinants of emotive rhetoric.

Corpus: *U.S. Congressional Record*

- Universe of floor speeches in U.S. Congress (House and Senate), 1858-2014
  - Exclude non-speech content such as roll calls, bill sponsorships, and legislation.
  - $N = 9,799,375$ speeches.

Corpus: *U.S. Congressional Record*

- Universe of floor speeches in U.S. Congress (House and Senate), 1858-2014
  - Exclude non-speech content such as roll calls, bill sponsorships, and legislation.
  - $N = 9,799,375$ speeches.
- Pre-Processing:
  - Tag parts of speech; keep nouns, adjectives, and verbs.
  - Drop punctuation, capitalization, numbers, stopwords (including names of states, cities, politicians), and word endings (snowball stemmer).

Corpus: *U.S. Congressional Record*

- Universe of floor speeches in U.S. Congress (House and Senate), 1858-2014
  - Exclude non-speech content such as roll calls, bill sponsorships, and legislation.
  - $N = 9,799,375$ speeches.
- Pre-Processing:
  - Tag parts of speech; keep nouns, adjectives, and verbs.
  - Drop punctuation, capitalization, numbers, stopwords (including names of states, cities, politicians), and word endings (snowball stemmer).
  - Drop tokens occurring in less than 10 speeches.
  - Final vocabulary: 63,334 words

Lexicons for Cognition and Emotion

- Starting point: Linguistic Inquiry and Word Count (LIWC) (Tauszcik and Pennebaker 2010).
  - provides coherent sets of words, word stems, and idiomatic expressions.
  - variety of structural, cognitive, and emotional components of text.

Lexicons for Cognition and Emotion

- Starting point: Linguistic Inquiry and Word Count (LIWC) (Tauszcik and Pennebaker 2010).
  - provides coherent sets of words, word stems, and idiomatic expressions.
  - variety of structural, cognitive, and emotional components of text.
- "Cognitive Processing" ("reason"):
  - 799 words, phrases, and wildcard expressions.
  - insight, causation, discrepancy, tentativeness, certainty, inhibition, inclusion, and exclusion

Lexicons for Cognition and Emotion

- Starting point: Linguistic Inquiry and Word Count (LIWC) (Tauszcik and Pennebaker 2010).
  - provides coherent sets of words, word stems, and idiomatic expressions.
  - variety of structural, cognitive, and emotional components of text.
- "Cognitive Processing" ("reason"):
  - 799 words, phrases, and wildcard expressions.
  - insight, causation, discrepancy, tentativeness, certainty, inhibition, inclusion, and exclusion
- "Affective Processing" ("emotion")
  - 1,445 words, phrases, and wildcard expressions.
  - positive and negative emotions, pleasure, pain, happiness, anxiety, anger, and sadness.

Lexicon Processing

- Lexicon Pre-Processing Steps:
  1. Exclude non-verbal expressions (e.g. emojis), punctuation, digits, and multi-word expressions.

Lexicon Processing

- Lexicon Pre-Processing Steps:
  1. Exclude non-verbal expressions (e.g. emojis), punctuation, digits, and multi-word expressions.
  2. Expand wildcards using WordNet.
     - exclude false positives through manual checks (e.g., "admir*" matching to "admiral").

Lexicon Processing

- Lexicon Pre-Processing Steps:
  1. Exclude non-verbal expressions (e.g. emojis), punctuation, digits, and multi-word expressions.
  2. Expand wildcards using WordNet.
     - exclude false positives through manual checks (e.g., "admir*" matching to "admiral").
  3. Exclude words that are not domain-appropriate.
     - Partially automate by identifying outlier words in embedding space.
     - 279 cognitive words dropped, 536 words affective words dropped.

Lexicon Processing

- Lexicon Pre-Processing Steps:
  1. Exclude non-verbal expressions (e.g. emojis), punctuation, digits, and multi-word expressions.
  2. Expand wildcards using WordNet.
     - exclude false positives through manual checks (e.g., "admir*" matching to "admiral").
  3. Exclude words that are not domain-appropriate.
     - Partially automate by identifying outlier words in embedding space.
     - 279 cognitive words dropped, 536 words affective words dropped.
  4. Filter by part of speech (noun/adjective/verb) and stem.

Lexicon Processing

- Lexicon Pre-Processing Steps:
  1. Exclude non-verbal expressions (e.g. emojis), punctuation, digits, and multi-word expressions.
  2. Expand wildcards using WordNet.
     - exclude false positives through manual checks (e.g., "admir*" matching to "admiral").
  3. Exclude words that are not domain-appropriate.
     - Partially automate by identifying outlier words in embedding space.
     - 279 cognitive words dropped, 536 words affective words dropped.
  4. Filter by part of speech (noun/adjective/verb) and stem.
- Resulting lexicons:
  - 359 cognition tokens, 848 emotion tokens

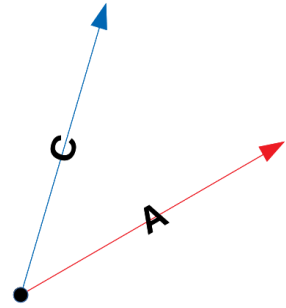
Word Embeddings for Congressional Speeches

- We train Word2Vec on all congressional speeches:
  - 300 dimensions, eight-word context window, 10 epochs.

Word Embeddings for Congressional Speeches

- We train Word2Vec on all congressional speeches:
  - 300 dimensions, eight-word context window, 10 epochs.
- For each of the lexicons (cognitive and affective), we form the centroid (average) vector:

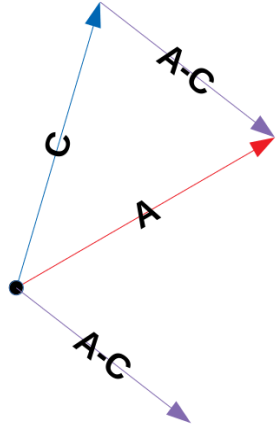
- $\vec{A}$ = affective centroid
- $\vec{C}$ = cognitive centroid



Word Embeddings for Congressional Speeches

- We train Word2Vec on all congressional speeches:
  - 300 dimensions, eight-word context window, 10 epochs.
- For each of the lexicons (cognitive and affective), we form the centroid (average) vector:

- $\vec{A}$ = affective centroid
- $\vec{C}$ = cognitive centroid
- $\vec{A} - \vec{C}$ = emotion-cognition dimension (Kozlowski, Evans, and Taddy 2018)



Document Vectors

- A speech i is a list of words indexed by w with corresponding vectors $\vec{w}$.

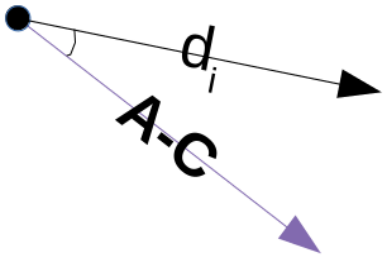
Document Vectors

- A speech i is a list of words indexed by w with corresponding vectors $\vec{w}$.
- Construct document vector for speech i as the centroid (average) of the word vectors, weighted by smoothed inverse frequency (SIF):

$$\vec{d}_i = \frac{1}{|i|} \sum_{w \in i} \underbrace{\frac{\alpha}{f(w) + \alpha}}_{\text{SIF}} \vec{w} \tag{1}$$

  - $|i|$, number of tokens in speech i
  - $f(w)$, relative frequency of word w in corpus
  - $\alpha = 0.001$, smoothing parameter (Arora, Liang, and Ma 2016)
- up-weights relatively rare (distinctive) words.

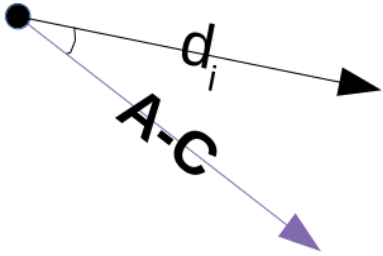
Emotionality Metric



- Relative emotionality of i is **cosine similarity** to the emotion-cognition dimension:

$$Y_i = \frac{\vec{d}_i \cdot (\vec{A} - \vec{C})}{\left\|\vec{d}_i\right\| \left\|\vec{A} - \vec{C}\right\|}$$

Emotionality Metric

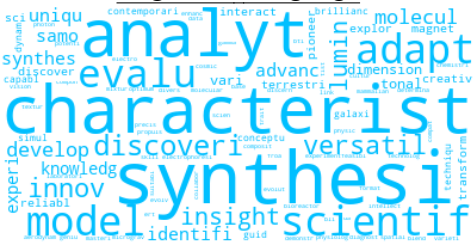


- Relative emotionality of i is **cosine similarity** to the emotion-cognition dimension:

$$Y_i = \frac{\vec{d}_i \cdot (\vec{A} - \vec{C})}{\left\|\vec{d}_i\right\| \left\|\vec{A} - \vec{C}\right\|}$$

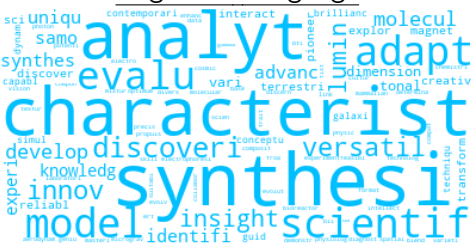
Increase in $Y_i \leftrightarrow$ shift towards emotion pole and away from cognition pole.

Cognition Language



- "In my judgment, neither is true in the case of this amendment."
- "Is that correct?"
- "R. 15 contains a provision that is similar but, in fact, broader in scope."

Cognition Language

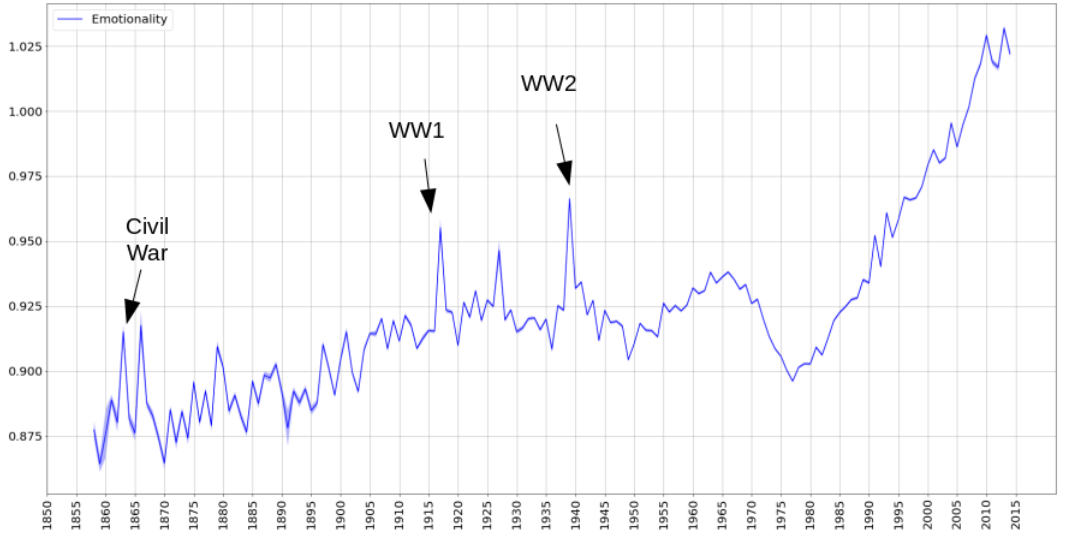


- "In my judgment, neither is true in the case of this amendment."
- "Is that correct?"
- "R. 15 contains a provision that is similar but, in fact, broader in scope."

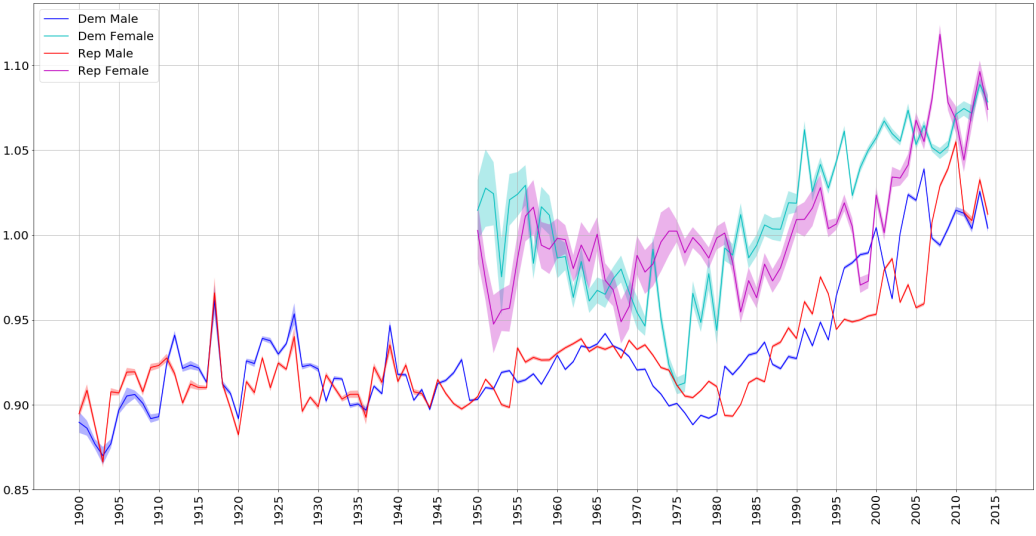
Emotion Language

- "With joy in his heart and a smile on his face he graced practically every social occasion with a song."
- "We Democrats may disagree, but we love our fellow men and we never hate them."

Emotion Language in Congress, 1958-2014



Emotion in Congress, by Party and Gender



Democrat Male (blue), Republican Male (red),
Democrat Female (teal), Republican Female (magenta).

Congressmen Rankings, 2007-2014

- Top 5 emotional senators:
  - **Joe Biden (D-DE)**
  - Jeffrey Chiesa (R-NJ)
  - **Hillary Clinton (D-NY)**
  - Paul Kirk (D-MA)
  - Ken Salazar (D-CO)
- Top 5 emotional house members:
  - Arthur Davis (D-AL)
  - **Nancy Pelosi (D-CA)**
  - Walter Jones (R-NC)
  - Roger Williams (R-TX)
  - Joyce Beatty (D-OH)
  - Robin Kelly (D-IL).
- Top 5 cognitive senators:
  - Carte Goodwin (D-WV)
  - Daniel Inouye (D-HI)
  - Mark Pryor (D-AR)
  - **Harry Reid (D-NV)**
  - Jeff Bingaman (D-NM)
- Top 5 cognitive house members:
  - Robert Aderholt (R-AL)
  - Justin Amash (R-MI)
  - Edward Whitfield (R-KY)
  - Peter Visclosky (D-IN)
  - Rodney Frelinghuysen (R-NJ).
- Top Emotion States: Rhode Island, Ohio, South Carolina, Illinois and Vermont
- Top Cognition States: Nevada, Montana, Idaho, West Virginia and New Mexico

Relation to Congressman Characteristics

- Emotionality Y_{ijt} of speech i by politician j at year t:

$$Y_{ijt} = \alpha_{ijt} + X'_{jt}\beta + \epsilon_{ijt} \tag{2}$$

  - α_{jt}: fixed effects
    - chamber-year
    - topic
  - X_{jt}: covariates of interest
    - gender
    - party
    - race
    - religion
  - Standard errors clustered by speaker.

	(1)	(2)	(3)	(4)	(5)
	Estimated Effect on Emotionality Score				
Female	0.0516**		0.0475**	0.0489**	0.0301**
	(0.00651)		(0.00752)	(0.00645)	(0.00285)
Democrat		0.00638*	0.00502*	0.00315	0.00409**
		(0.00254)	(0.00252)	(0.00250)	(0.00136)
Female × Democrat			0.00405		
			(0.0110)		
Black				0.0282*	0.0208**
				(0.0117)	(0.00645)
Hispanic				0.0149	0.0133*
				(0.0113)	(0.00613)
Catholic				0.00953*	0.00567*
				(0.00442)	(0.00235)
Jewish				0.0109	0.00272
				(0.00780)	(0.00356)
Chamber-Year FE	X	X	X	X	X
Topic FE					X
N	5869780	5869780	5869780	5869780	5839095
adj. R^2	0.062	0.060	0.062	0.063	0.479

Std err. in parens, clustered by speaker. + p < .1, * p < .05, ** p<0.01.

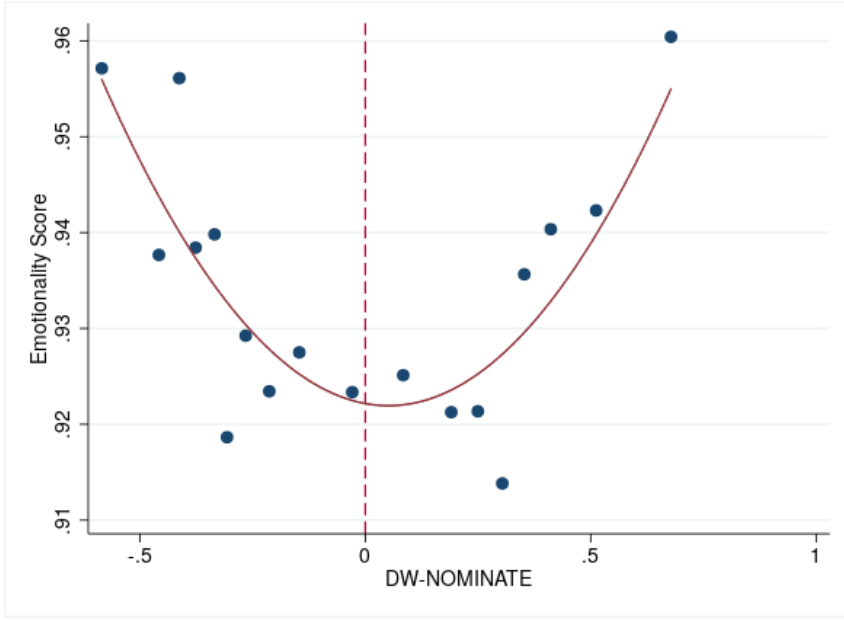
Effect of Party Structure

Effect of Party Structure

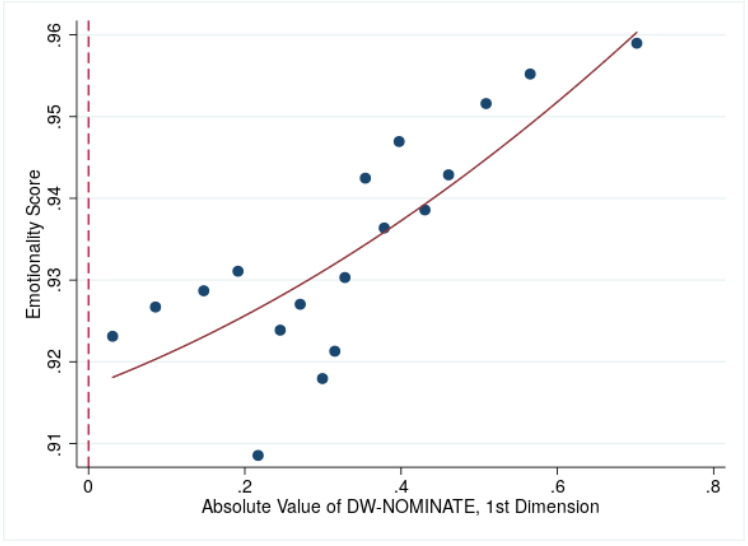
	(1)	(2)	(3)	(4)
	Estimated Effect on Emotionality Score			
Minority Party		0.0128**	0.0150**	0.00375**
		(0.00190)	(0.00151)	(0.000621)
Divided Govt	0.000850			
	(0.00172)			
Chamber-Year FE		X	X	X
Speaker FE	X		X	X
Topic FE				X
N	5869730	5869780	5869730	5839045
adj. R^2	0.111	0.061	0.112	0.493

Std err. in parens, clustered by speaker. + p $<$ .1, * p $<$ .05, ** p$<$0.01.

Emotionality and Ideology



Ideologically Extreme Politicians are More Emotive



Relationship is driven by topic selection.

Outline

Analyzing polarization in social media: Method and application to tweets on 21 mass shootings

Demszky, Garg, Voigt, Zou, Gentzkow, Shapiro, and Jurafsky

- Research Object:
  - use NLP to understand four dimensions of social media polarization: topic choice, framing, affect, modality.

Analyzing polarization in social media: Method and application to tweets on 21 mass shootings

Demszky, Garg, Voigt, Zou, Gentzkow, Shapiro, and Jurafsky

- Research Object:
  - use NLP to understand four dimensions of social media polarization: topic choice, framing, affect, modality.
- Context:
  - tweets in response to mass shooting events.

Analyzing polarization in social media: Method and application to tweets on 21 mass shootings

Demszky, Garg, Voigt, Zou, Gentzkow, Shapiro, and Jurafsky

- Research Object:
  - use NLP to understand four dimensions of social media polarization: topic choice, framing, affect, modality.
- Context:
  - tweets in response to mass shooting events.
- Research question:
  - does political partisanship manifest in polarized responses to violent/polarizing events?

Dataset

- 21 mass shooting events, 2015-2018, from Gun Violence Archive

Dataset

- 21 mass shooting events, 2015-2018, from Gun Violence Archive
- tweets about those events, identified by:
  - location keywords (e.g. chattanooga, roseburg, san bernardino, fresno, etc.)
  - event keywords (lemmas): shoot, gun, kill, attack, massacre, victim

Dataset

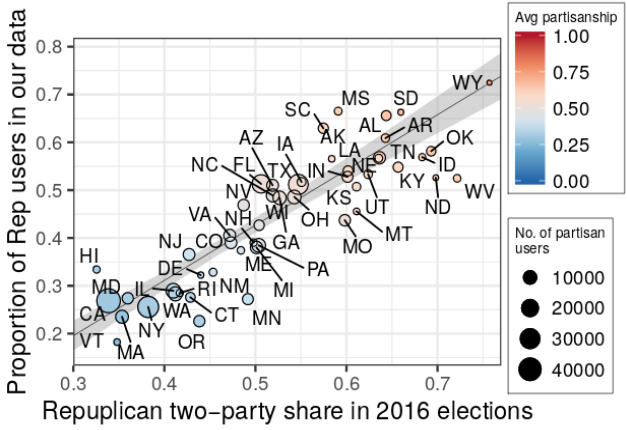
- 21 mass shooting events, 2015-2018, from Gun Violence Archive
- tweets about those events, identified by:
  - location keywords (e.g. chattanooga, roseburg, san bernardino, fresno, etc.)
  - event keywords (lemmas): shoot, gun, kill, attack, massacre, victim
  - filter out retweets and tweets from deactivated accounts
  - $N = 10,000$ (out of 4.4 million tweets from the firehose archive).

Identifying party affiliation of Twitter users

- Party affiliation identified off of whether you follow more Democrats or Republicans, from a list of Twitter accounts associated with legislators, presidential candidates, and party organizations (Volkova et al 2014).
  - at least 51% of tweets for each event can be assigned partisanship this way.

Identifying party affiliation of Twitter users

- Party affiliation identified off of whether you follow more Democrats or Republicans, from a list of Twitter accounts associated with legislators, presidential candidates, and party organizations (Volkova et al 2014).
  - at least 51% of tweets for each event can be assigned partisanship this way.
- For geolocated users this matches up pretty well with party vote shares by state ($R^2 = .82$):



Measuring Partisanship: Pre-processing

- Stemming and stopword removal.
- Event-specific vocabulary:
  - unigrams and bigrams
  - occur in event's tweets at least 50 times
  - must be used by at least two tweeters.

Partisanship metric

- Leave-one-out estimator from Gentzkow et al (2019), applied to each shooting event:

$$\pi = \frac{1}{2}\left(\frac{1}{|D|}\sum_{i \in D} \hat{\boldsymbol{q}}_i \cdot \hat{\boldsymbol{\rho}}_{-i} + \frac{1}{|R|}\sum_{i \in R} \hat{\boldsymbol{q}}_i \cdot (1 - \hat{\boldsymbol{\rho}}_{-i})\right)$$

  - $\hat{\boldsymbol{q}}_i$ = token frequencies for user i, drawn from set of democrats D and set of republicans R
  - $\hat{\boldsymbol{\rho}}_{-i}$ has elements

$$\rho_{-i} = \frac{q_i^D}{q_i^D + q_i^R}$$

    empirical posterior probabilities computed from all other users.

Partisanship metric

- Leave-one-out estimator from Gentzkow et al (2019), applied to each shooting event:

$$\pi = \frac{1}{2}\left(\frac{1}{|D|}\sum_{i\in D}\hat{\boldsymbol{q}}_i \cdot \hat{\boldsymbol{\rho}}_{-i} + \frac{1}{|R|}\sum_{i\in R}\hat{\boldsymbol{q}}_i \cdot (1-\hat{\boldsymbol{\rho}}_{-i})\right)$$

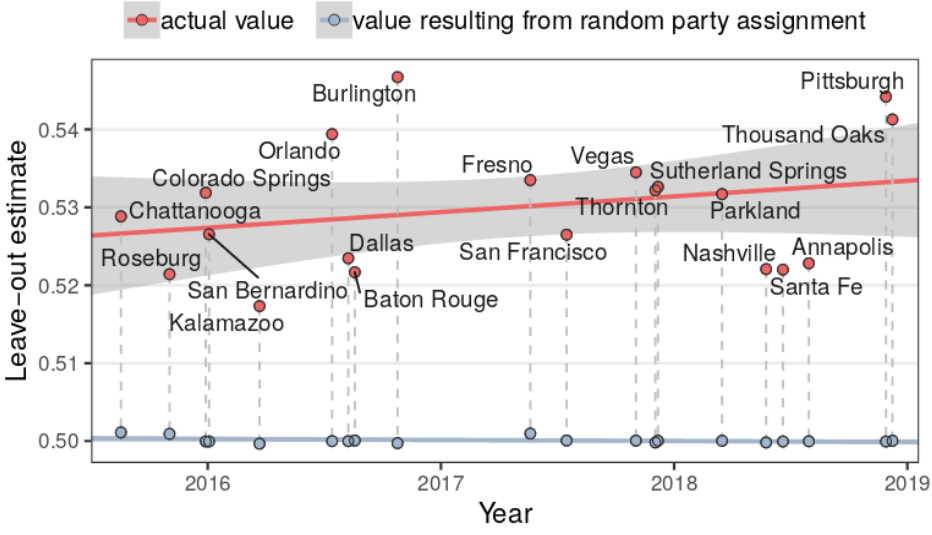
  - $\hat{\boldsymbol{q}}_i$ = token frequencies for user i, drawn from set of democrats D and set of republicans R
  - $\hat{\boldsymbol{\rho}}_{-i}$ has elements

$$\rho_{-i} = \frac{q_i^D}{q_i^D + q_i^R}$$

    empirical posterior probabilities computed from all other users.

- π is an estimate for expected posterior probability that a Bayesian observer would correctly predict party after observing one randomly sampled token.
  - consistency assumes tokens are drawn from multinomial logit.

Tweet texts about mass shootings are predictive of party



- comparable to $\pi = .53$ in Congressional speeches (GST 2019).
- The increase in polarization over time is not statistically significant.

Questions/Issues with this Analysis

- How polarized are tweets about other topics (not mass shootings)?
  - why not use a tweeter fixed effect and compare to their other tweets?
  - why not show pre-trends in polarization?

Questions/Issues with this Analysis

- How polarized are tweets about other topics (not mass shootings)?
  - why not use a tweeter fixed effect and compare to their other tweets?
  - why not show pre-trends in polarization?
- Can show polarization separately by party?

Questions/Issues with this Analysis

- How polarized are tweets about other topics (not mass shootings)?
  - why not use a tweeter fixed effect and compare to their other tweets?
  - why not show pre-trends in polarization?
- Can show polarization separately by party?
- Validating π:
  - How accurate is π at the individual level?
  - Where is the binscatter of π versus actual party affiliation?

Sentence Embeddings for Topic Assignment

1. Make a new vocabulary:
   1.1 Sample 10,000 tweets from each event
   1.2 vocabulary of stemmed words occuring at least ten times in at least three events ($N = 2000$)

Sentence Embeddings for Topic Assignment

1. Make a new vocabulary:
   1.1 Sample 10,000 tweets from each event
   1.2 vocabulary of stemmed words occuring at least ten times in at least three events ($N = 2000$)
2. Train GloVe embeddings on random samples of tweets from each event (samples were different sizes, this is not explained)

Sentence Embeddings for Topic Assignment

1. Make a new vocabulary:
   1.1 Sample 10,000 tweets from each event
   1.2 vocabulary of stemmed words occuring at least ten times in at least three events ($N = 2000$)
2. Train GloVe embeddings on random samples of tweets from each event (samples were different sizes, this is not explained)
3. Create Arora et al (2017) embeddings:
   3.1 for each tweet t, compute weighted average vectors v_t for each word, weighted by inverse frequency.
   3.2 take out first principal component of matrix whose rows are v_t

Topics = Embedding Clusters

1. Cluster the embeddings using k-means

Topics = Embedding Clusters

1. Cluster the embeddings using k-means
2. Identify and drop hard-to-classify tweets:
   2.1 compute ratio of distance to closest topic and distance to second-closest topic.
   2.2 drop tweets above the 75th percentile.

Topics = Embedding Clusters

1. Cluster the embeddings using k-means
2. Identify and drop hard-to-classify tweets:
   2.1 compute ratio of distance to closest topic and distance to second-closest topic.
   2.2 drop tweets above the 75th percentile.

- Validation using Amazon Mechanical Turk to choose number of clusters:
  - Identify word intruder: five from one cluster, one from another cluster.
  - Identify tweet intruder: three from one cluster, and one from another cluster.

Topic Content

Topic	10 Nearest Stems
news (19%)	break, custodi, #breakingnew, #updat, confirm, fatal, multipl, updat, unconfirm, sever
investigation (9%)	suspect, arrest, alleg, apprehend, custodi, charg, accus, prosecutor, #break, ap
shooter's identity & ideology (11%)	extremist, radic, racist, ideolog, label, rhetor, wing, blm, islamist, christian
victims & location (4%)	bar, thousand, california, calif, among, los, southern, veteran, angel, via
laws & policy (14%)	sensibl, regul, requir, access, abid, #gunreformnow, legisl, argument, allow, #guncontolnow
solidarity (13%)	affect, senseless, ach, heart, heartbroken, sadden, faculti, pray, #prayer, deepest
remembrance (6%)	honor, memori, tuesday, candlelight, flown, vigil, gather, observ, honour, capitol
other (23%)	dude, yeah, eat, huh, gonna, ain, shit, ass, damn, guess

- The embedding method resulted in more coherent topics (better MTurk validation for words and tweets) than a topic model. $k = 8$ got best coherence.
  - Appendix reports samples of tweets for each topic (but does not say how samples were selected).

Between-topic vs within-topic polarization

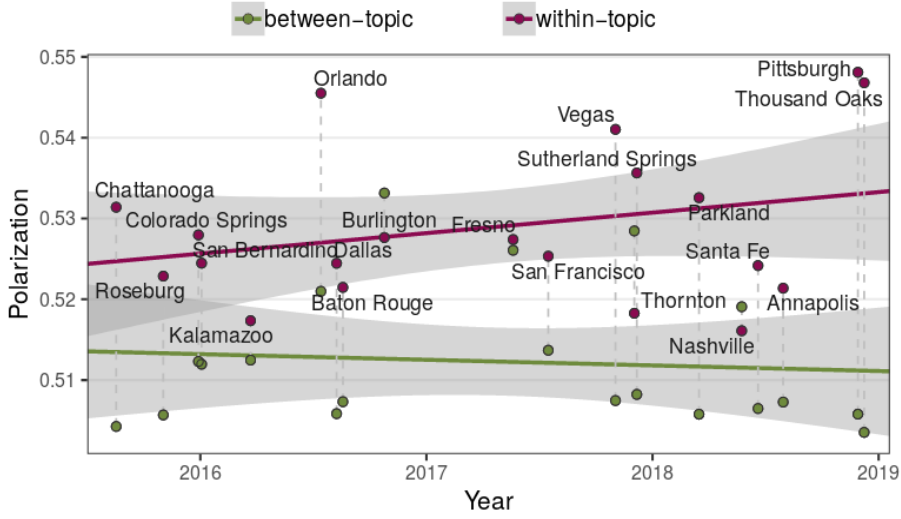
- Within-topic polarization: compute π separately by the tweet clusters.

Between-topic vs within-topic polarization

- Within-topic polarization: compute π separately by the tweet clusters.
- Between-topic polarization: Compute π using cluster counts, rather than token counts.

Between-topic vs within-topic polarization

- Within-topic polarization: compute π separately by the tweet clusters.
- Between-topic polarization: Compute π using cluster counts, rather than token counts.



Trends in within-topic polarization

- Most polarized topics: shooter's identity & ideology (.55), laws & policy (.54)

Figure 6: Las Vegas within-topic polarization in the days after the event. The bar charts show the proportion of each topic in the data at a given time.

- "measuring polarization of topics for other events over time is noisy".

Partisanship of Topics, by Race of Shooter

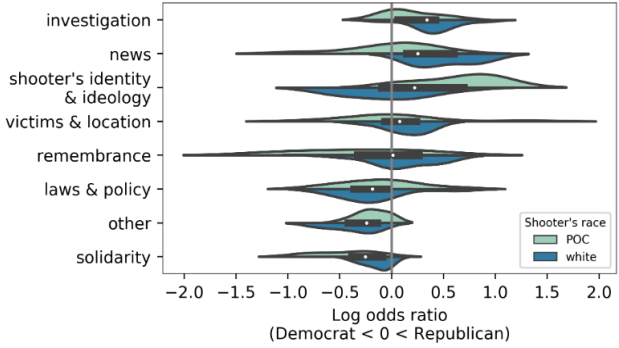


Figure 7: The plot shows the kernel density of the partisan log odds ratios of each topic (one observation per event). The white points show the median and the black rectangles the interquartile range across events.

Partisan Framing Devices: Words

- Partisanship of phrases from supervised model:

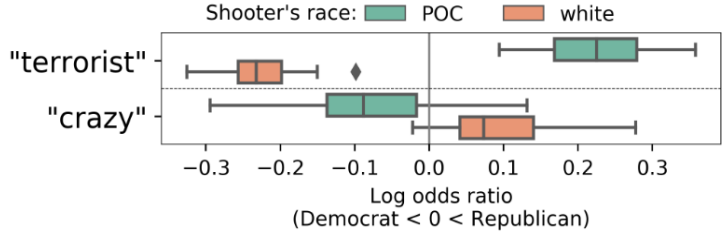
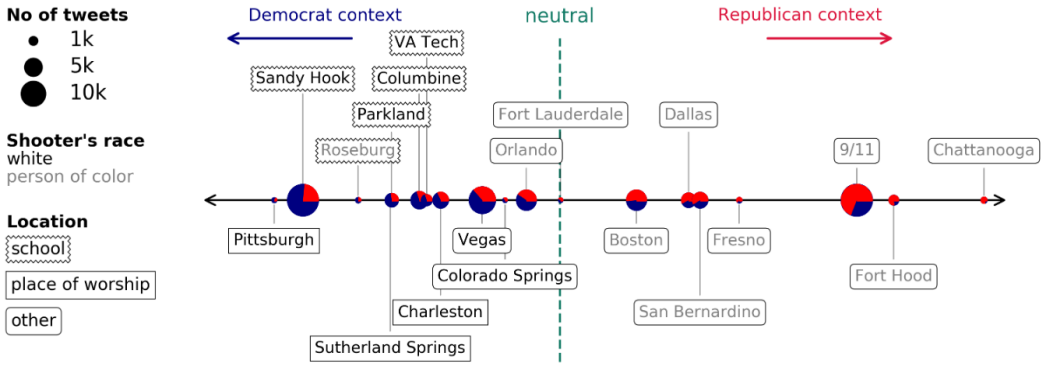


Figure 8: The log odds ratios of "terrorist" and "crazy" across events, grouped by the shooter's race. The boxes show the interquartile range and the diamond an outlier.

- Partisan valence of "terrorist" and "crazy" flip depending on race of shooter (these words have the largest racial difference in the joint vocabulary).

Partisan Framing Devices: Events

- Partisanship of keywords for previous events:



- Democrats invoke white shooters, Republicans invoke POC shooters.

Affect (Emotions)

- Starting point: Emotion lexicon from Mohammad and Turney (2013), available at `saifmohammad.com`.
  - 14,182 words assigned to sentiment (positive/negative) and emotions (anger, anticipation, disgust, fear, joy, sadness, surprise, trust).

Affect (Emotions)

- Starting point: Emotion lexicon from Mohammad and Turney (2013), available at `saifmohammad.com`.
  - 14,182 words assigned to sentiment (positive/negative) and emotions (anger, anticipation, disgust, fear, joy, sadness, surprise, trust).
- Domain propagation (Hamilton et al 2018):
  - pick 5-11 representative words per emotion category (Appendix E)
  - for each word in vocabulary, compute average distance to each member of each category. take 30 closest words as lexicon.

Affect (Emotions)

- Starting point: Emotion lexicon from Mohammad and Turney (2013), available at `saifmohammad.com`.
  - 14,182 words assigned to sentiment (positive/negative) and emotions (anger, anticipation, disgust, fear, joy, sadness, surprise, trust).
- Domain propagation (Hamilton et al 2018):
  - pick 5-11 representative words per emotion category (Appendix E)
  - for each word in vocabulary, compute average distance to each member of each category. take 30 closest words as lexicon.

sadness senseless, loss, tragedi, lost, devast, sad, love, griev, horrif, terribl, pain, violenc, condol, broken, hurt, feel, victim, mourn, horrifi, will, grief, ach, suffer, sick, kill, aw, sicken, evil, massacr, mad

disgust disgust, sick, shame, ignor, wrong, blame, hell, ridicul, idiot, murder, evil, coward, sicken, feel, disgrac, slaughter, action, bad, insan, attack, pathet, outrag, polit, terrorist, mad, damn, lose, shit, lie, asshol

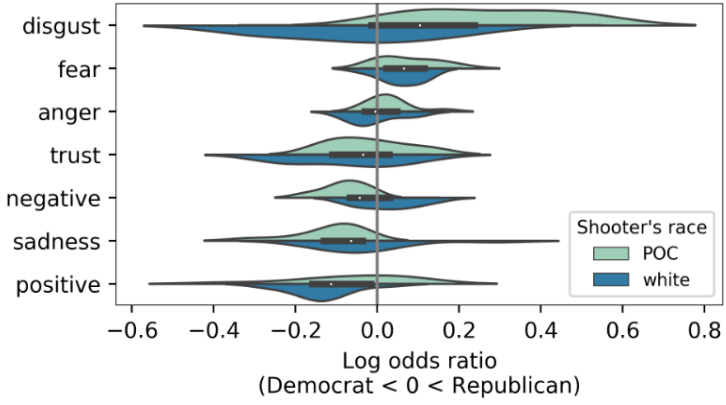
anger gun, will, murder, kill, violenc, wrong, shoot, bad, death, attack, feel, shot, action, arm, idiot, crazi, crimin, terrorist, mad, hell, crime, blame, fight, ridicul, insan, shit, die, threat, terror, hate

fear danger, threat, fear, arm, gun, still, shooter, attack, feel, fight, hide, murder, shot, shoot, bad, kill, chang, serious, violenc, forc, risk, defend, warn, govern, concern, fail, polic, wrong, case, terrorist

trust school, like, good, real, secur, show, nation, don, protect, call, teacher, help, law, great, save, true, wonder, respons, sad, answer, person, feel, safe, thought, continu, love, guard, church, fact, support

Partisanship of Affect Categories

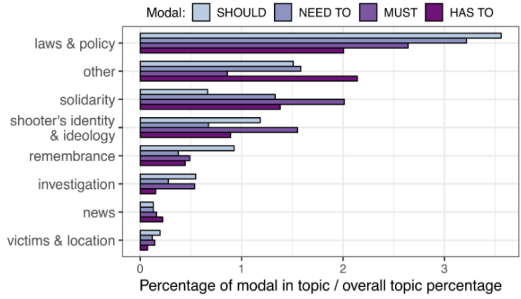
- Compute partisanship scores using affect-category counts:



- Disgust affect flips along partisan lines depending on race of shooter.

Modality

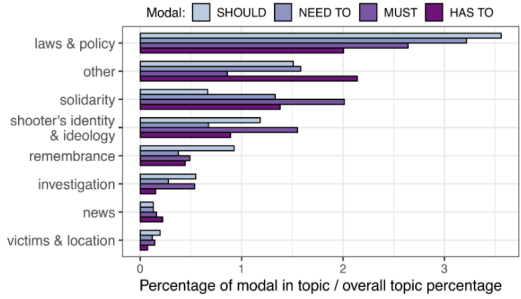
This roller coaster debate **MUST** STOP! Sensible gun ownership is one thing but assault weapons massacre innocent lives. The savagery of gore at #Parkland was beyond belief & **must** be the last.
In times of tragedy **shouldn't** we all come together?! Prayers for those harmed in the #PlannedParenthood shooting.
Communities **need to** step up and address white on white crime like the Las Vegas massacre. White men are out of control.
he BLM protest shooting, planned parenthood, now cali... domestic terrorism will crumble this country, SANE PPL **HAVE TO** FIGHT BACK
Shooting cops is horrible, cannot be condoned. But **must be** understood these incidents are outgrowth of decades of police abuses. #BatonRouge
1. Islamic terrorists are at war with us 2. Gun free zones = kill zones 3. Americans **should be** allowed to defend themselves #Chattanooga
Las Vegas shooting Walmart shooting and now 25 people killed in Texas over 90 people killed Mexico **should** build that wall to keep the US out
CNN reporting 20 dead, 42 injured in Orlando night club shooting. Just awful. The US **must** act to control guns or this carnage will continue.



- Count the four most frequent necessity modals in the data: should, must, have to, need to.
  - in this context, they are used as calls to action.

Modality

This roller coaster debate **MUST** STOP! Sensible gun ownership is one thing but assault weapons massacre innocent lives. The savagery of gore at #Parkland was beyond belief & **must** be the last.
In times of tragedy **shouldn't** we all come together?! Prayers for those harmed in the #PlannedParenthood shooting.
Communities **need to** step up and address white on white crime like the Las Vegas massacre. White men are out of control.
he BLM protest shooting, planned parenthood, now cali... domestic terrorism will crumble this country, SANE PPL **HAVE TO** FIGHT BACK
Shooting cops is horrible, cannot be condoned. But **must be** understood these incidents are outgrowth of decades of police abuses. #BatonRouge
1. Islamic terrorists are at war with us 2. Gun free zones = kill zones 3. Americans **should be** allowed to defend themselves #Chattanooga
Las Vegas shooting Walmart shooting and now 25 people killed in Texas over 90 people killed Mexico **should** build that wall to keep the US out
CNN reporting 20 dead, 42 injured in Orlando night club shooting. Just awful. The US **must** act to control guns or this carnage will continue.



- Count the four most frequent necessity modals in the data: should, must, have to, need to.
  - in this context, they are used as calls to action.
- Democrats use modals more than Republicans; Republicans seem more fatalistic.

Comments

- This is an impressive array of NLP tools aimed at the same research question.
  - could be moving toward a standard for analyzing interpretable dimension in language.

Comments

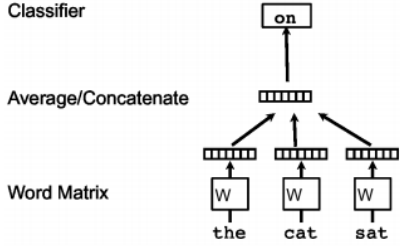
- This is an impressive array of NLP tools aimed at the same research question.
  - could be moving toward a standard for analyzing interpretable dimension in language.
- For all outcomes, would help to compare to other types of events, and to show pre-trends.
  - there is no baseline for polarization for comparison.
  - they do not distinguish whether outcomes are driven by different people selecting into tweeting, vs within-user changes.

Outline

Outline

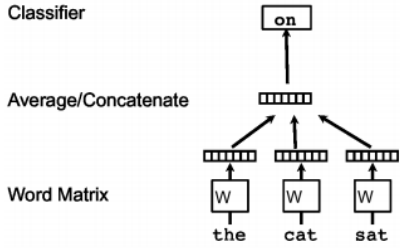
Doc2Vec (Le and Mikolov)

- Recall that Word2Vec trains word embeddings to predict a word given neighboring context words:

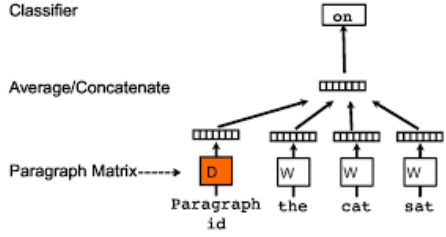


Doc2Vec (Le and Mikolov)

- Recall that Word2Vec trains word embeddings to predict a word given neighboring context words:

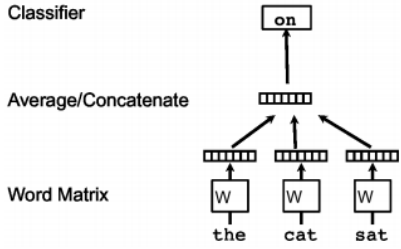


- Doc2Vec augments Word2Vec with a categorical embedding for the document (e.g. paragraph):

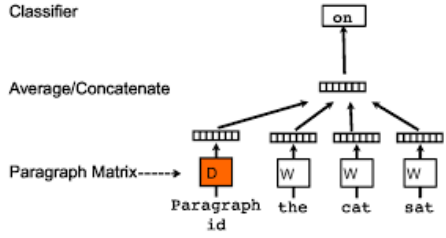


Doc2Vec (Le and Mikolov)

- Recall that Word2Vec trains word embeddings to predict a word given neighboring context words:

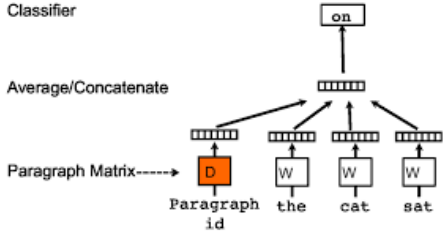


- Doc2Vec augments Word2Vec with a categorical embedding for the document (e.g. paragraph):



- Note that there are two layers:
  - an input embedding layer, embedding both the words in the context, and the document ID.
  - an output layer, which is softmax across the vocabulary – this produces a separate word embedding.

Vectorizing New Documents



- A new document that wasn't in training does not have a vector.
- Document inference step:
  - freeze word embeddings in input layer and in output layer.
  - train embedding for new document to predict randomly sampled words in new document until convergence.

Document Embeddings Geometry

- Just as directions in word embedding space encode semantic information about the words, directions in document embedding space encode topical information about the documents.

Document Embeddings Geometry

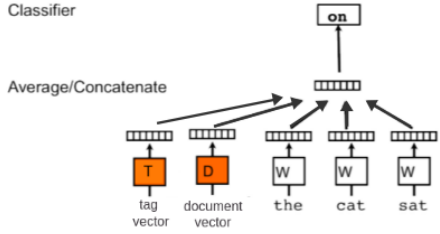
- Just as directions in word embedding space encode semantic information about the words, directions in document embedding space encode topical information about the documents.
- In topic models, each dimension has a topical interpretation; in document embeddings, a direction (might) have a topical interpretation.

Doc2Vec in gensim

- can train both document vectors and word vectors
- can get similarity between documents, and use clustering to get groups of related documents.

Doc2Vec in gensim

- can train both document vectors and word vectors
- can get similarity between documents, and use clustering to get groups of related documents.
- Can add additional non-unique document "tags"; these will be embedded separately from the unique doc ID:




```
In [168]: tagged_docs[3]

Out[168]: TaggedDocument(words=['aftershere', 'project', 'finishing', 'stages', 'home', 'decor', 'kitchen', 'design', 'beforeher
          e', 'project', 'finishing', 'stages', 'home', 'decor', 'kitchen', 'design', 'afterhere', 'project', 'finishing', 'stag
          es', 'home', 'decor', 'kitchen', 'design', 'beforehere', 'project', 'finishing', 'stages', 'home', 'decor', 'kitchen',
           'design'], tags=['Remodeling & Renovating', 'SENT_3'])
```

- will improve performance if using the embeddings to classify the tag.

Outline

Doc2Vec on Wikipedia

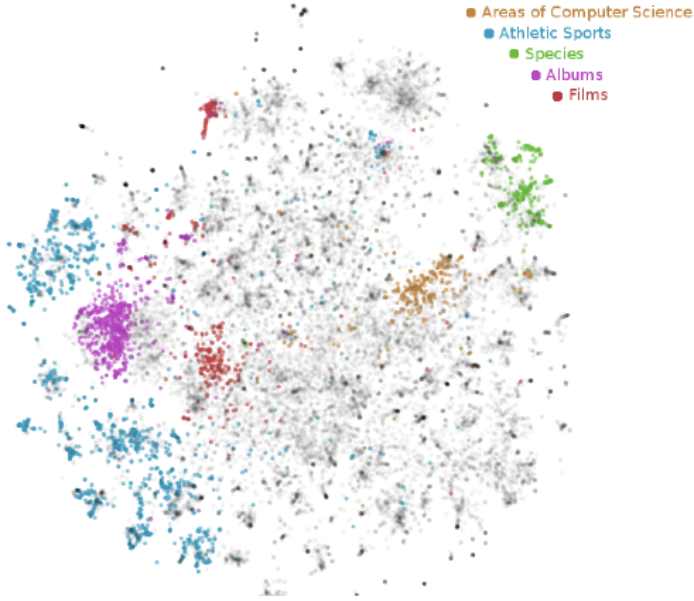


Figure 3: Visualization of Wikipedia paragraph vectors using t-SNE.

Table 1: Nearest neighbours to "Machine learning." Bold face texts are articles we found unrelated to "Machine learning." We use Hellinger distance for LDA and cosine distance for Paragraph Vectors as they work the best for each model.

LDA	**Paragraph Vectors**
Artificial neural network	Artificial neural network
Predictive analytics	Types of artificial neural networks
Structured prediction	Unsupervised learning
Mathematical geophysics	Feature learning
Supervised learning	Predictive analytics
Constrained conditional model	Pattern recognition
Sensitivity analysis	Statistical classification
SXML	Structured prediction
Feature scaling	Training set
Boosting (machine learning)	Meta learning (computer science)
Prior probability	Kernel method
Curse of dimensionality	Supervised learning
Scientific evidence	Generalization error
Online machine learning	Overfitting
N-gram	Multi-task learning
Cluster analysis	Generative model
Dimensionality reduction	Computational learning theory
Functional decomposition	Inductive bias
Bayesian network	Semi-supervised learning

Table 5: arXiv nearest neighbours to "Distributed Representations of Sentences and Documents" using Paragraph Vectors.

Title	Cosine Similarity
Evaluating Neural Word Representations in Tensor-Based Compositional Settings	0.771
Polyglot: Distributed Word Representations for Multilingual NLP	0.764
Lexicon Infused Phrase Embeddings for Named Entity Resolution	0.757
A Convolutional Neural Network for Modelling Sentences	0.747
Distributed Representations of Words and Phrases and their Compositionality	0.740
Convolutional Neural Networks for Sentence Classification	0.735
SimLex-999: Evaluating Semantic Models With (Genuine) Similarity Estimation	0.735
Exploiting Similarities among Languages for Machine Translation	0.731
Efficient Estimation of Word Representations in Vector Space	0.727
Multilingual Distributed Representations without Word Alignment	0.721

Table 2: Wikipedia nearest neighbours

(a) Wikipedia nearest neighbours to "Lady Gaga" using Paragraph Vectors. All articles are relevant.

Article	Cosine Similarity
Christina Aguilera	0.674
Beyonce	0.645
Madonna (entertainer)	0.643
Artpop	0.640
Britney Spears	0.640
Cyndi Lauper	0.632
Rihanna	0.631
Pink (singer)	0.628
Born This Way	0.627
The Monster Ball Tour	0.620

(b) Wikipedia nearest neighbours to "Lady Gaga" - "American" + "Japanese" using Paragraph Vectors. Note that Ayumi Hamasaki is one of the most famous singers, and one of the best selling artists in Japan. She also has an album called "Poker Face" in 1998.

Article	Cosine Similarity
Ayumi Hamasaki	0.539
Shoko Nakagawa	0.531
Izumi Sakai	0.512
Urbangarde	0.505
Ringo Sheena	0.503
Toshiaki Kasuga	0.492
Chihiro Onitsuka	0.487
Namie Amuro	0.485
Yakuza (video game)	0.485
Nozomi Sasaki (model)	0.485

Table 7: arXiv nearest neighbours to "Distributed Representations of Sentences and Documents" - "neural" + "Bayesian". I.e., the Bayesian equivalence of the Paragraph Vector paper.

Title	Cosine Similarity
Content Modeling Using Latent Permutations	0.629
SimLex-999: Evaluating Semantic Models With (Genuine) Similarity Estimation	0.611
Probabilistic Topic and Syntax Modeling with Part-of-Speech LDA	0.579
Evaluating Neural Word Representations in Tensor-Based Compositional Settings	0.572
Syntactic Topic Models	0.548
Training Restricted Boltzmann Machines on Word Observations	0.548
Discrete Component Analysis	0.547
Resolving Lexical Ambiguity in Tensor Regression Models of Meaning	0.546
Measuring political sentiment on Twitter: factor-optimal design for multinomial inverse regression	0.544
Scalable Probabilistic Entity-Topic Modeling	0.541

Outline

Document Vectors for Judicial Opinions

- Ash and Chen (2018) produce document vectors for each case to understand differences between judges and courts.
  - Corpus: 300,000 cases from U.S. Circuit Courts, 1870-2010.

Document Vectors for Judicial Opinions

- Ash and Chen (2018) produce document vectors for each case to understand differences between judges and courts.
  - Corpus: 300,000 cases from U.S. Circuit Courts, 1870-2010.
- We de-mean vectors by group (court, topic, or year) to extract relevant information:
  - de-mean by topic-year to distinguish courts.
  - de-mean by court-topic to distingush years.
  - de-mean by court-year to distinguish topics.

Figure 1: Centered by Topic-Year, Averaged by Judge, Labeled by Court

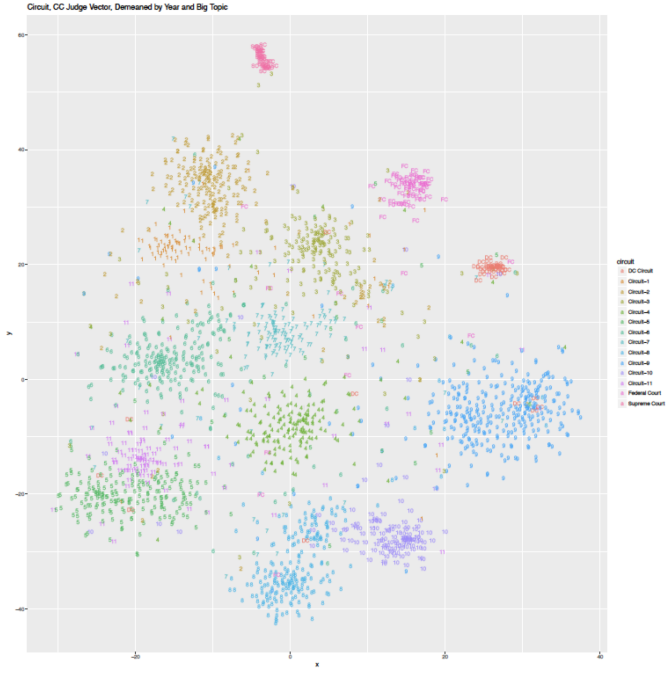
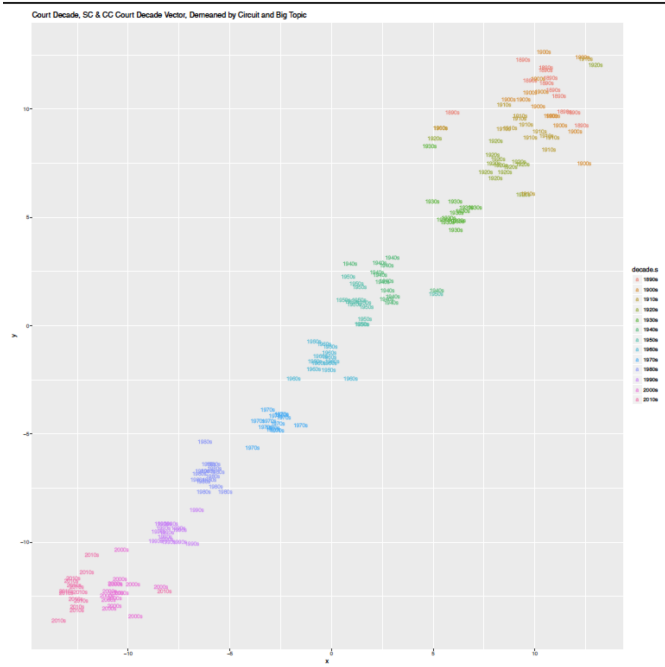


Figure 2: Centered by Court-Topic, Averaged by Court-Year, Labeled by Decade



tsne plot

Figure 3: Centered by Judge-Year, Averaged by Topic-Year, Labeled by Topic

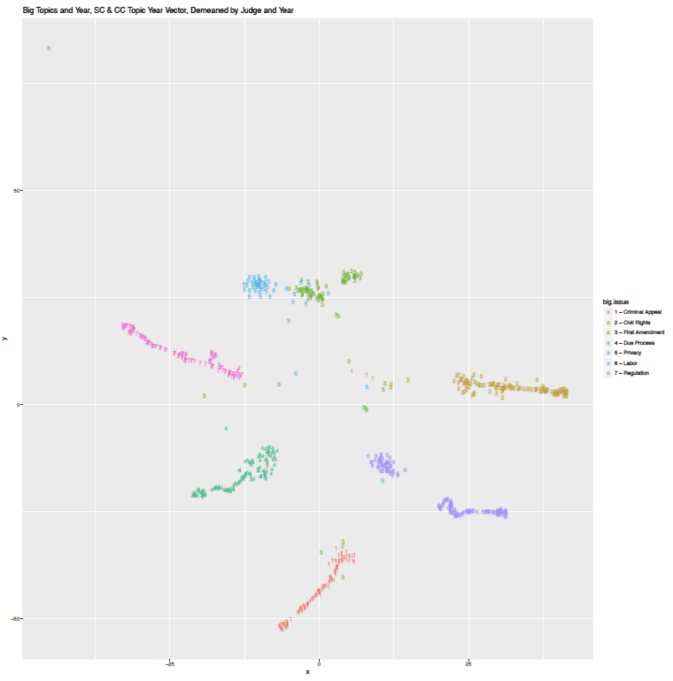
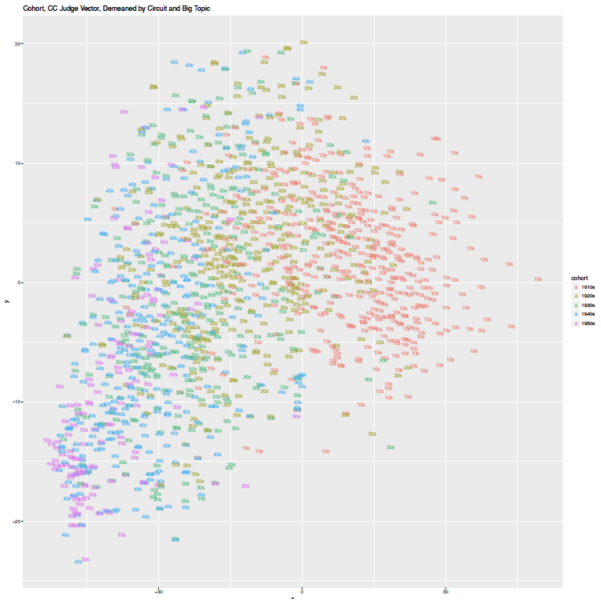
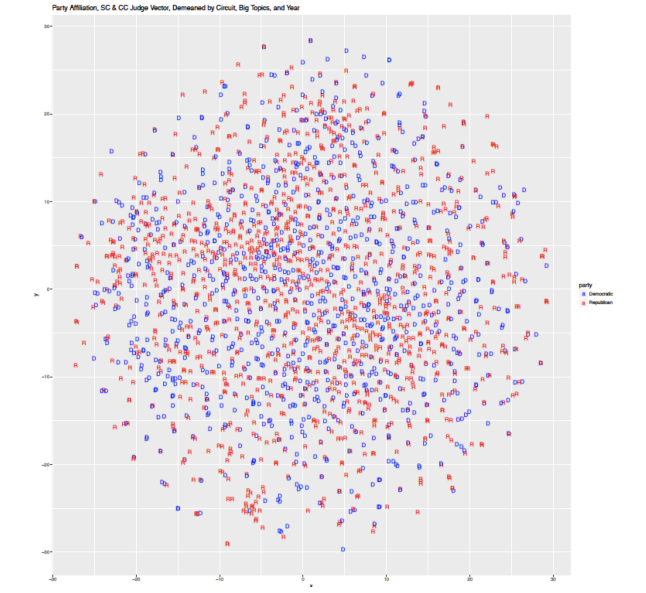


Figure 5: Centered by Court-Topic-Year, Averaged by Judge, Labeled by Judge Birth Cohort



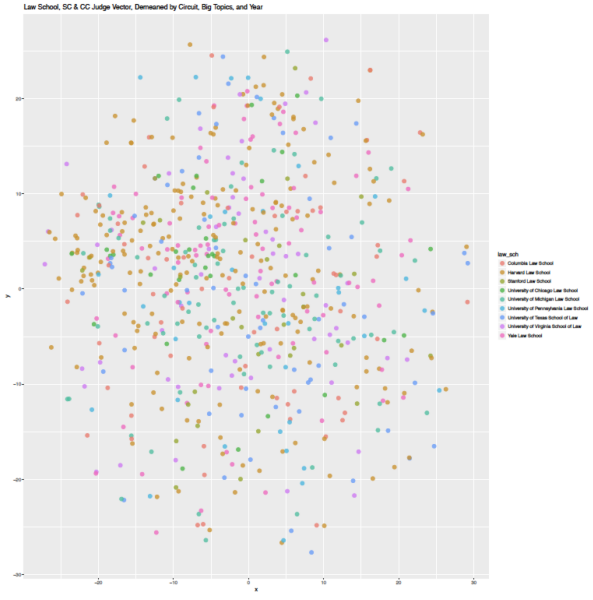
tsne plot

Figure 4: Centered by Court-Topic-Year, Averaged by Judge, Labeled by Political Party



tsne plot

Figure 6: Centered by Court-Topic-Year, Averaged by Judge, Labeled by Law School Attended



Relatedness between judges (e.g. Richard Posner)

Circuit Judge Name	Similarity	Rank	Circuit Judge Name	Similarity	Rank
POSNER, RICHARD A.	1.000	1	TONE, PHILIP W.	0.459	16
EASTERBROOK, FRANK H.	0.663	2	SIBLEY, SAMUEL	0.459	17
SUTTON, JEFFREY S.	0.620	3	SCALIA, ANTONIN	0.456	18
NOONAN, JOHN T.	0.596	4	COLLOTON, STEVEN M.	0.445	19
NELSON, DAVID A.	0.592	5	DUNIWAY, BENJAMIN	0.438	20
CARNES, EDWARD E.	0.567	6	GIBBONS, JOHN J.	0.422	21
FRIENDLY, HENRY	0.566	7	BOGGS, DANNY J.	0.420	22
KOZINSKI, ALEX	0.563	8	BREYER, STEPHEN G.	0.414	23
GORSUCH, NEIL M.	0.559	9	GOODRICH, HERBERT	0.412	24
CHAMBERS, RICHARD H.	0.546	10	LOKEN, JAMES B.	0.410	25
FERNANDEZ, FERDINAND F.	0.503	11	WEIS, JOSEPH F.	0.408	26
EDMONDSON, JAMES L.	0.501	12	SCALIA, ANTONIN (SCOTUS)	0.406	27
KLEINFELD, ANDREW J.	0.491	13	BOUDIN, MICHAEL	0.403	28
WILLIAMS, STEPHEN F.	0.481	14	RANDOLPH, A. RAYMOND	0.397	29
KETHLEDGE, RAYMOND M.	0.459	15	MCCONNELL, MICHAEL W.	0.390	30

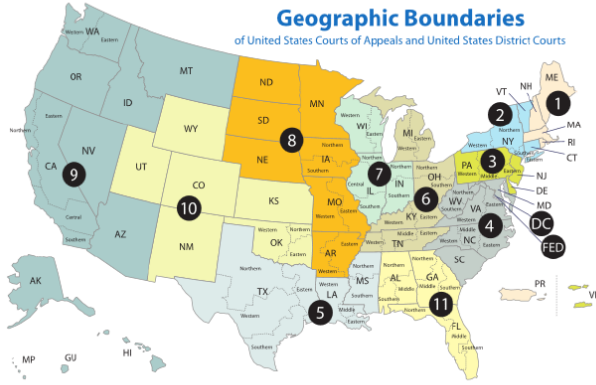
Document vectors demeaned by court, year, and topic, then aggregated by judge.

Outline

How does judge writing influence attitudes?

- We treat the circuit courts as a set of natural-experiment laboratories:

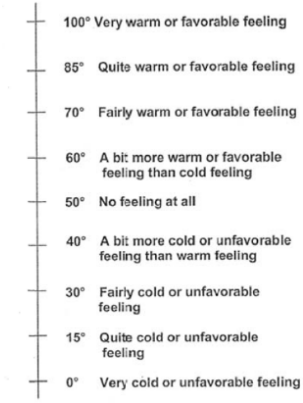
Figure 1: U.S. Courts of Appeals



- We ask: Does the variation in what judges write about different groups influence how people in that region feel about those groups?

Social Attitudes: ANES Feeling Thermometer

Figure A.1: Example Thermometer Question - ANES 2012



Targets

Black: blacks, black, african, african-american, african-americans, negro, negroes

Business: business, businesses, corporation, corporations, factory, firm, market, organization, partnership, shop, store, venture

Catholic: catholics, catholic

Congress: congress, parliament, legislature, senate, house, representative, senators, representatives

Conservative: conservatives, conservative

Democrat: democrat, democrats

Elderly: elderly, aged, old

Federal government: federal, government, executive

Illegal: illegal, immigrants, undocumented

Labor unions: labor, unions, union, trade-union

Liberal: liberals, liberal

Military: military, army

Police: policemen, police, policeman

Protestant: protestant, protestants

Republican: republican, republicans

Supreme Court: supreme, court

White: whites, white, caucasian, caucasians

Woman: woman, women

Young: youngster, youth, budding, adolescent

Sentence Vectors

- We used Doc2Vec to produce vectors for each sentence in the circuit court corpus for the years 1964 through 2008.
  - Each sentence gets a similarity metric to each target from the American National Election Study Feeling Thermometer.

Targets

Black: blacks, black, african, african-american, african-americans, negro, negroes
Business: business, businesses, corporation, corporations, factory, firm, market, organization, partnership, shop, store, venture
Catholic: catholics, catholic
Congress: congress, parliament, legislature, senate, house, representative, senators, representatives
Conservative: conservatives, conservative
Democrat: democrat, democrats
Elderly: elderly, aged, old
Federal government: federal, government, executive
Illegal: illegal, immigrants, undocumented
Labor unions: labor, unions, union, trade-union
Liberal: liberals, liberal
Military: military, army
Police: policemen, police, policeman
Protestant: protestant, protestants
Republican: republican, republicans
Supreme Court: supreme, court

White: whites, white, caucasian, caucasians
Woman: woman, women
Young: youngster, youth, budding, adolescent

Judicial Sentiment

- Each sentence also gets a sentiment level based on the cosine similarity of the document vector to the positive-negative dimension constructed from:

Attributes

Negative: cold, unfavorable, bad, adverse, antagonistic, calamitous, damaging, destructive, disadvantageous, hostile, negative, objectionable, ominous, troublesome, unfriendly, contrary, discommodious, ill, ill-advised, improper, inadvisable, inauspicious, inconvenient, inexpedient, infelicitous, inimical, inopportune, late, low, malapropos, opposed, poor, regrettable, tardy, threatening, unfit, unfortunate, unlucky, unpromising, unpropitious, unseasonable, unseemly, unsuited, untimely, untoward, wrong.

Positive: warm, favorable, good, agreeable, benign, encouraging, positive, supportive, sympathetic, acclamatory, affirmative, amicable, approbative, approbatory, assenting, benevolent, benignant, commending, complimentary, enthusiastic, inclined, kind, kindly, laudatory, okay, praiseful, predisposed, reassuring, recommendatory, understanding, welcoming, well-disposed, well-intentioned.

Figure 3: Positive and Negative Sentiment Language

Notes: Most similar words in the embedding space to the average vector for the lexicon of positive words (left) and negative words (right). See text for details.

- Let $\mathbf{W}_i^k$ = vector of sentence similarities to target k in case i
- Let $\mathbf{S}_i$ = vector of sentence sentiments in case i.
- Construct case-level sentiment towards target k as $S_i^k = \mathbf{S}_i \cdot \mathbf{W}_i^k$.

Empirical Analysis

- Let C_{ct} be the set of cases filed in circuit c during year t. Average S_i^k across all cases i for each target k. The main treatment regressor is, therefore,

$$S_{ckt} = \frac{1}{|C_{ct}|} \sum_{i \in C_{ct}} S_i^k, \tag{3}$$

the average case-level sentiment toward k for each case in circuit-year ct.

Empirical Analysis

- Let C_{ct} be the set of cases filed in circuit c during year t. Average S_i^k across all cases i for each target k. The main treatment regressor is, therefore,

$$S_{ckt} = \frac{1}{|C_{ct}|} \sum_{i \in C_{ct}} S_i^k, \tag{3}$$

  the average case-level sentiment toward k for each case in circuit-year ct.

- We would like to estimate β (effect of judge writing sentiment on individuals' attitudes) using

$$Y_{ckt} = \alpha_k + \alpha_{ct} + \beta S_{ckt} + \epsilon_{ckt} \tag{4}$$

  - Y_{ckt} = thermometer response from ANES in circuit c toward taret k at t.
  - γ_{ck} = dummy variables (fixed effects) for each circuit-year
  - γ_k = dummy variables (fixed effects) for each target.

Identification

- we want to estimate causal effect of judge sentiments on citizen attitudes.
- but they could be correlated without indicating causation:
  - citizen attitudes could influence judges
  - or there could be a third unobserved factor.

Identification

- we want to estimate causal effect of judge sentiments on citizen attitudes.
- but they could be correlated without indicating causation:
  - citizen attitudes could influence judges
  - or there could be a third unobserved factor.
- Solution: instrumental variables using random assignment of judges
  - But we need machine learning to extract variation from many weak instruments.
  - Will revisit in Week 12 Lecture.

Outline

Outline

Sentence Mover Distance

Clark et al (2019) generalize the idea of word mover distance to sentences.

- find minimal cost of moving a set of sentence embeddings in document A to co-locate wth a set of sentence embeddings in document B.
- sentence embeddings weighted by length.

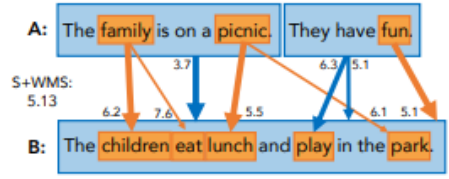


Figure 1: An illustration of S+WMS (a sentence mover similarity metric that uses both word and sentence embeddings) between two documents. This metric finds the minimal cost of "moving" both the word embeddings (orange) and the sentence embeddings (blue) in Document A to those in Document B. An arrow's width is the proportion of the embedding's weight being moved, and its label is the Euclidean distance. Here we show only the highest weighted connections.

Clustered Sentences as Features

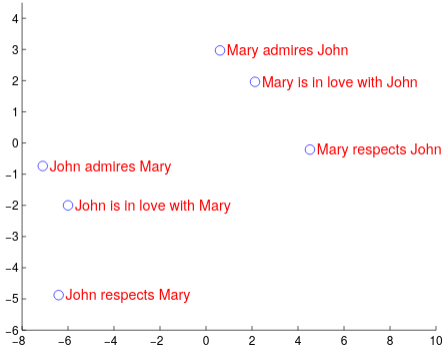
- After doing k-means clustering (or other clustering) on the sentences, can represent documents as counts/frequencies over the sentence clusters.

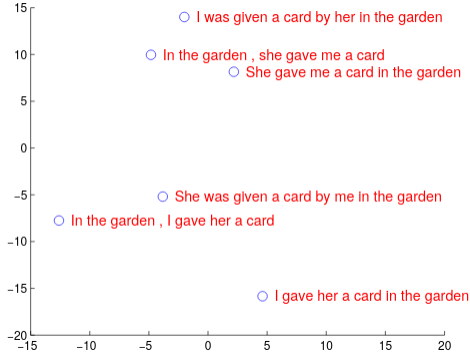
Autoencoder Encodings

- A recurrent autoencoder compresses a document (e.g. a sentence) into a vector to be reconstructed.
  - Can use the compressed representation as a document embedding.
- But these embeddings don't tend to work well for sentence similarity tasks because autoencoders try to reproduce the specific wording (reconstruction objective), rather than the conceptual meaning.
- We will discuss variational autoencoders for text generation in the Week 10 Lecture.

Machine Translation Embeddings

- Machine translators produce a sentence vector that must be decoded into another language.
- if the vector produces a good translation, it must contain the important information in the sentence.

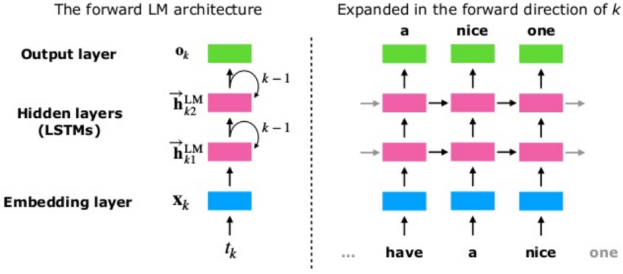




ELMo

- ELMo is more designed as a word embedding rather than document embedding method:

With long short term memory (LSTM) network, predicting the next words in both directions to build biLMs

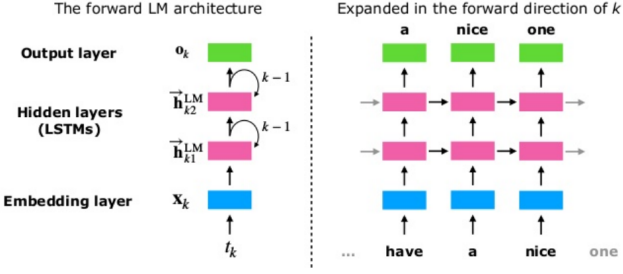


- The task:
  - predict previous and next words in a sentence using a birectional LSTM.

ELMo

- ELMo is more designed as a word embedding rather than document embedding method:

With long short term memory (LSTM) network, predicting the next words in both directions to build biLMs



- The task:
  - predict previous and next words in a sentence using a birectional LSTM.
- embeddings go through two hidden layers before the softmax output:
  - first layer learns syntax
  - second layer learns semantics

	Source	Nearest Neighbors
GloVe	play	playing, game, games, played, players, plays, player, Play, football, multiplayer
biLM	Chico Ruiz made a spectacular play on Alusik 's grounder {...}	Kieffer , the only junior in the group , was commended for his ability to hit in the clutch , as well as his all-round excellent play .
	Olivia De Havilland signed to do a Broadway play for Garson {...}	{...} they were actors who had been handed fat roles in a successful play , and had talent enough to fill the roles competently , with nice understatement .

Table 4: Nearest neighbors to "play" using GloVe and the context embeddings from a biLM.

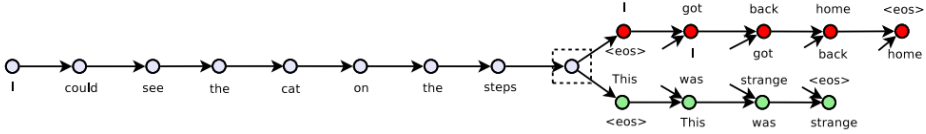
GloVe mostly learns *sport*-related context

ELMo can distinguish the word sense based on the context

- Pre-trained ELMo models are available from AllenNLP (allennlp.org/elmo)

Skip-Thought Embeddings

- Kiros et al (2015), drawing on the intuition of skip-gram embeddings in word2vec, produce sentence embeddings for a sentence prediction task.
  - GRU model: encoder vectorizes a sentence, and the decoder tries to reproduce the next sentence.
  - uses negative sampling: produce embeddings to guess whether two sentences are in the same paragraph.



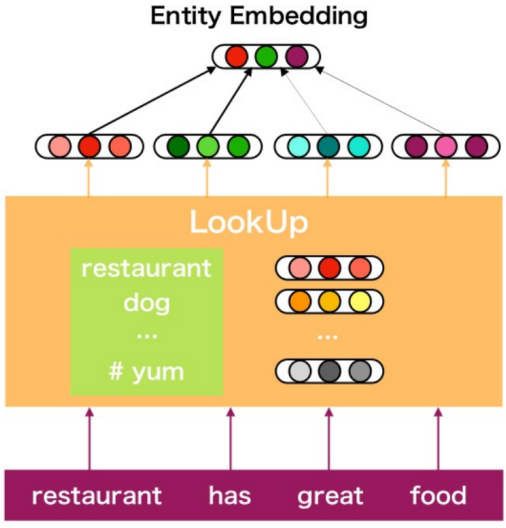
Query and nearest sentence
he ran his hand inside his coat , double-checking that the unopened letter was still there . he slipped his hand between his coat and his shirt , where the folded copies lay in a brown envelope .
im sure youll have a glamorous evening , she said , giving an exaggerated wink . im really glad you came to the party tonight , he said , turning to her .
although she could tell he had n't been too invested in any of their other chitchat , he seemed genuinely curious about this . although he had n't been following her career with a microscope , he 'd definitely taken notice of her appearances .
an annoying buzz started to ring in my ears , becoming louder and louder as my vision began to swim . a weighty pressure landed on my lungs and my vision blurred at the edges , threatening my consciousness altogether .
if he had a weapon , he could maybe take out their last imp , and then beat up errol and vanessa . if he could ram them from behind , send them sailing over the far side of the levee , he had a chance of stopping them .
then , with a stroke of luck , they saw the pair head together towards the portaloos . then , from out back of the house , they heard a horse scream probably in answer to a pair of sharp spurs digging deep into its flanks .
" i 'll take care of it , " goodman said , taking the phonebook . " i 'll do that , " julia said , coming in .
he finished rolling up scrolls and , placing them to one side , began the more urgent task of finding ale and tankards . he righted the table , set the candle on a piece of broken plate , and reached for his flint , steel , and tinder .

Table 2: In each example, the first sentence is a query while the second sentence is its nearest neighbour. Nearest neighbours were scored by cosine similarity from a random sample of 500,000 sentences from our corpus.

StarSpace

- StarSpace provides a general neural embedding system for putting even different types of entities into the same space.
  - generalizes negative sampling to any setting with entities and features
  - useful for most any ML task, including NLP.

StarSpace



Sum each feature embeddings

$$\sum_{i \in a} F_i$$

dictionary F, (D x d) matrix

D: # of features
d: # of embed dims

features Fi

Embed for a i-th feature

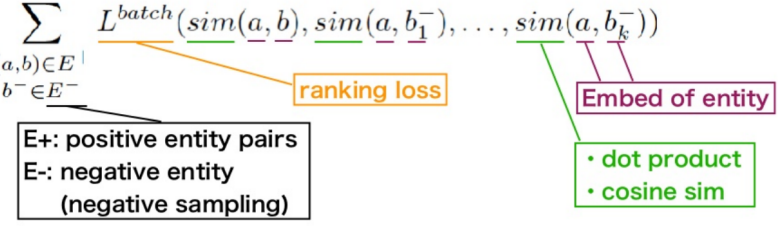
entity a

represented as bag-of-features

Learning to embed in the same space

- such that comparisons are meaningful
- free to compare entities of **different kinds**
  e.g. **user entity** and **item entity** (recommendation)
  document entity and **label entities** (classification)

Minimize Following Loss Function using SGD



How can we provide Entity pairs (a, b) to the system?

Multiclass Classification (e.g. TextClassification)

sports

business

technology

a: documents (bag-of-words)

b: labels (singleton features)

b-: sampled from set of possible labels

How can we provide Entity pairs (a, b) to the system?

Multiclass Classification (e.g. TextClassification)

sports

business

technology

a: documents (bag-of-words)
b: labels (singleton features)
b-: sampled from set of possible labels

Learning (unsupervised) Sentence Embeddings

Directly/Optimally learn sentence embed

Select a pair of sents (s1, s2) from the same doc:

a: s1
b: s2
b-: sampled from sents coming from other docs

FastText Embeddings

- **Joulin et al (2016), "Bag of Tricks for Efficient Text Classification"**
  - document vector = averaged hashed n-gram embeddings
  - extremely fast, scalable, strong baselines for text classification.

FastText Embeddings

- **Joulin et al (2016), "Bag of Tricks for Efficient Text Classification"**
  - document vector = averaged hashed n-gram embeddings
  - extremely fast, scalable, strong baselines for text classification.
- **Bojanowski et al (2017), "Enriching word vectors with subword information"**
  - each word is represented as a bag of character n-grams. (e.g., spicy = (spi, pic, icy)).
  - learn embeddings for the character segments, word embedding = sum over these segment embeddings
  - character n-gram embeddings are hashed.
  - competitive with word2vec in standard tasks; better in some languages.
  - produces good embeddings for unseen words.

InferSent

- Another step forward for sentence representations.
- They train a bidirectional LSTM on Stanford Natural Language Inference task:
  - classifying 570K sentence pairs by entailment, contradiction, and neutral.
- The resulting embeddings do better than skip-thought vectors on transfer learning tasks.

Outline

Universal Sentence Encoder

```
import tensorflow_hub as hub

embed = hub.Module("https://tfhub.dev/google/"
    "universal-sentence-encoder/1")

embedding = embed([
    "The quick brown fox jumps over the lazy dog."])
```

Listing 1: Python example code for using the universal sentence encoder.

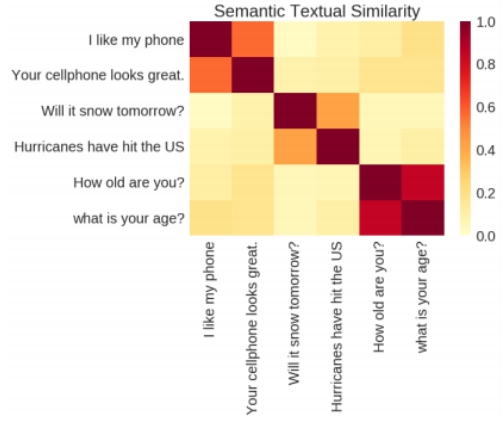


Figure 1: Sentence similarity scores using embeddings from the universal sentence encoder.

- Universal Sentence Encoder (USE) uses a trained neural net's encoding subgraph to embed a sentence or short paragraph (actually, any string) to a 512-dimensional vector.

USE: Architecture and Tasks

- Input embeddings for words and bigrams are averaged, passed through an MLP to produce sentence embeddings (Iyyer et al 2015), to be fed to the downstream tasks.
  - USE gives this sentence embedding vector.

USE: Architecture and Tasks

- Input embeddings for words and bigrams are averaged, passed through an MLP to produce sentence embeddings (Iyyer et al 2015), to be fed to the downstream tasks.
  - USE gives this sentence embedding vector.
- USE is trained on multiple tasks to make embeddings more robust:
  - Identifying co-occuring sentences (as in skip thought vectors, Kiros et al 2015)
  - Identifying message-response pairs (Henderson et al 2017)
  - Some supervised learning tasks (see Cer et al 2018).

USE: Architecture and Tasks

- Input embeddings for words and bigrams are averaged, passed through an MLP to produce sentence embeddings (Iyyer et al 2015), to be fed to the downstream tasks.
  - USE gives this sentence embedding vector.
- USE is trained on multiple tasks to make embeddings more robust:
  - Identifying co-occuring sentences (as in skip thought vectors, Kiros et al 2015)
  - Identifying message-response pairs (Henderson et al 2017)
  - Some supervised learning tasks (see Cer et al 2018).
- Cer et al (2018) show that USE embeddings have less bias than word2vec or GloVe, following the WEAT score approach from Caliskan et al (2017).

MUSE

- The multilingual sentence encoder (**MUSE**) expands the USE model to sixteen languages, in a single embedding model!
- Trained on a similar array of tasks in all languages, so that it can be used out-of-the-box.

Languages	Family
Arabic (ar)	**Semitic**
Chinese (PRC) (zh) Chinese (Taiwan) (zh-tw)	**Sino-Tibetan**
Dutch(nl) English(en) German (de)	**Germanic**
French (fr) Italian (it) Portuguese (pt) Spanish (es)	**Latin**
Japanese (ja)	**Japonic**
Korean (ko)	**Koreanic**
Russian (ru) Polish (pl)	**Slavic**
Thai (th)	**Kra–Dai**
Turkish (tr)	**Turkic**

Table 1: Supported languages (ISO 639-1).

```
import tensorflow_hub as hub

module = hub.Module("https://tfhub.dev/google/"
    "universal-sentence-encoder-multilingual/1")

multilingual_embeddings = module([
 "Hola Mundo!", "Bonjour le monde!", "Ciao mondo!"
 "Hello World!", "Hallo Welt!", "Hallo Wereld!",
 "你好世界!", "Привет, мир!", "!مرحبا بالعالم"])
```

Listing 1: Encoding for STS/Bitext retrieval.

```
module = hub.Module("https://tfhub.dev/google/"
    "universal-sentence-encoder-multilingual-qa/1")

query_embeddings = module(
    dict(text=["What is your age?"]),
    signature="question_encoder", as_dict=True)

candidate_embeddings = module(
    dict(text=["I am 20 years old."],
         context=["I will be 21 next year."]),
    signature="response_encoder", as_dict=True)
```

Listing 2: Encoding for QA retrieval.

Measuring Innovation in Journalism

Ash, Cage, Guillot, and Herve (2020):

- compute MUSE embeddings on 20 million French-language news articles.
- compute backward and forward text similarity measures as in Kelly et al (2018)
  - limit to seven-day window of articles before and after.

Measuring Innovation in Journalism

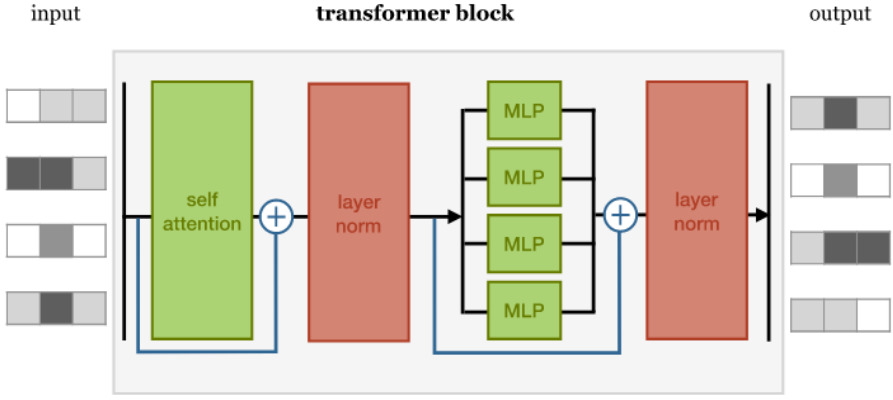
Ash, Cage, Guillot, and Herve (2020):

- compute MUSE embeddings on 20 million French-language news articles.
- compute backward and forward text similarity measures as in Kelly et al (2018)
  - limit to seven-day window of articles before and after.
- Descriptive results:
  - Journalists with more innovative articles have higher wages.

Outline

BERT

- BERT (Bidirectional Encoder Representations from Transformers) consists of a stack of transformer blocks (encoders) like this one:



- The largest model has 24 blocks, embedding dimension of 1024, and 16 attention heads.
  - = 340M parameters to learn.

BERT Pre-Training Corpus and Tasks

- Corpus:
  - 800M words from English books (modern work, from unpublished authors)
  - 2.5B words of text from English Wikipedia articles (without markup).

BERT Pre-Training Corpus and Tasks

- Corpus:
  - 800M words from English books (modern work, from unpublished authors)
  - 2.5B words of text from English Wikipedia articles (without markup).
- Masked language modeling:
  - 15% of words masked
  - if masked: replace with [MASK] 80% of the time, a random token 10% of the time, and left unchanged 10% of the time.
  - model has to predict the original word.

BERT Pre-Training Corpus and Tasks

- Corpus:
  - 800M words from English books (modern work, from unpublished authors)
  - 2.5B words of text from English Wikipedia articles (without markup).
- Masked language modeling:
  - 15% of words masked
  - if masked: replace with [MASK] 80% of the time, a random token 10% of the time, and left unchanged 10% of the time.
  - model has to predict the original word.
- Next sentence prediction:
  - two sentence are sampled; they are either sequential or randomly paired.
  - BERT predicts whether it is a true pair or a negative sample.

BERT input representation

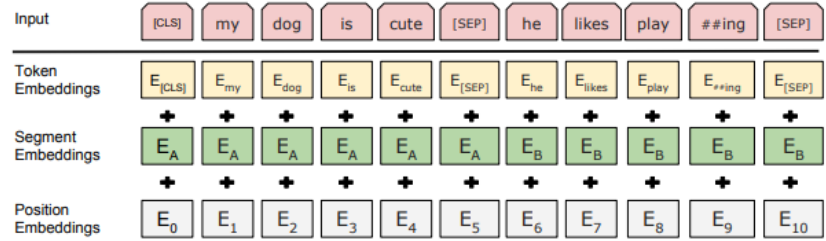


Figure 2: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

- "<cls>" indicates start of sequence.

BERT input representation

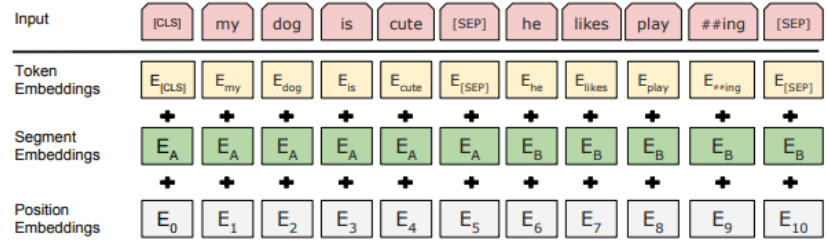


Figure 2: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

- "<cls>" indicates start of sequence.
- SentencePiece tokenization:
  - character-level byte-pair encoder, learns character n-grams to breaks words like "playing" into "play" and "##ing".
  - have to fix a vocabulary size: here, 30K.

BERT input representation

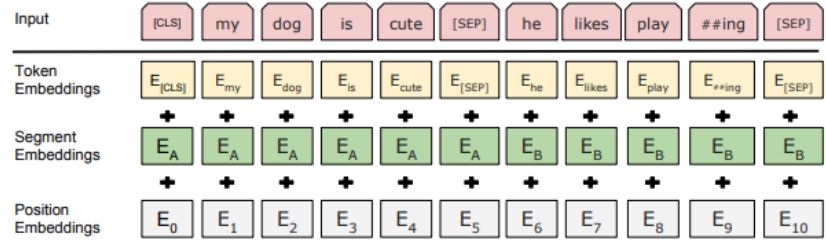


Figure 2: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

- "<cls>" indicates start of sequence.
- SentencePiece tokenization:
  - character-level byte-pair encoder, learns character n-grams to breaks words like "playing" into "play" and "##ing".
  - have to fix a vocabulary size: here, 30K.
- Segment embeddings: dummy variable for first or second sentence in a pair.
- Position embeddings: categoricals for location in sequence.

Using BERT

- The pre-training provides a trained transformer model.
- It provides, for example, an embedding for a given document which can be used for any task.
  - this is very slow, and they are not very good embeddings for semantic similarity.
- The model can also be fine-tuned as needed.

Using BERT

- The pre-training provides a trained transformer model.
- It provides, for example, an embedding for a given document which can be used for any task.
  - this is very slow, and they are not very good embeddings for semantic similarity.
- The model can also be fine-tuned as needed.
- BERT is an NLP breakthrough: it obtains state-of-the-art results on many many tasks (see Devlin et al 2019).

RoBERTa

- modify some key hyperparameters
- focus on a modified version of the masked language modeling task
  - (drop sentence pair task)
- tends to improve downstream performance

Sentence-BERT

Reimers and Gurevych (2019):

- Using BERT for producing document embeddings is slow and does not get very good results, in terms of ranking sentences by semantic similarity.
  - RoBERTa and XLNet are also not that good for this task.

Sentence-BERT

Reimers and Gurevych (2019):

- Using BERT for producing document embeddings is slow and does not get very good results, in terms of ranking sentences by semantic similarity.
  - RoBERTa and XLNet are also not that good for this task.
- S-BERT:
  - optimize the encoding task with a Siamese neural net (shared weights) architecture.
  - Decreases encoding time (e.g.) from 65 hours (BERT) to 5 seconds (SBERT).

Extracting morality dimension from BERT

- Schramowski et al (2019) project BERT embeddings onto a "moral subspace", analogous to what Bolukbasi et al (2016) do for a gender subspace:

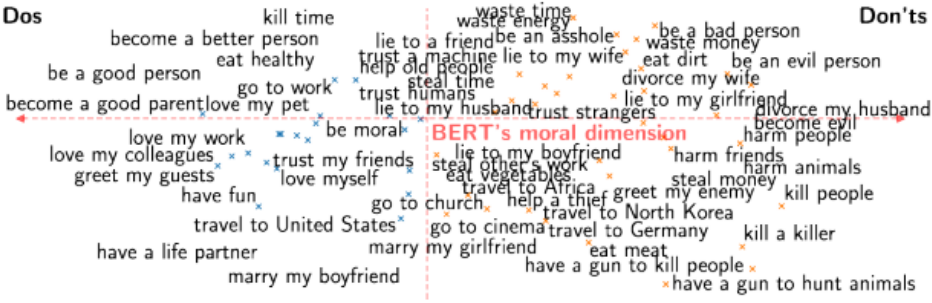


Figure 4: Context-based actions projected —based on PCA computed by selected atomic actions— along two axes: x (top PC) defines the moral direction m (Left: *Dos* and right: *Don'ts*). Compare Tab. 9(Appendix) for detailed moral bias scores.

- tricky to distinguish morality from sentiment, though.