Quantitative Text Analysis
Meeting 12: Embeddings

Petro Tolochko

Why Word2Vec?

Feedforward neural LMs learn embeddings indirectly.
Word2Vec (Mikolov et al., 2013):

- uses a tiny neural network,
- learns only word embeddings,
- trains very fast (billions of tokens),
- produces high-quality semantic vectors.

Two separate architectures:

- **CBOW**: context $\rightarrow$ target
- **Skip-gram**: target $\rightarrow$ context

Key question: Where do embeddings come from? *They are simply the learned weight matrices of this small network.*

CBOW: Context $\rightarrow$ Target

Given context words around w_t:

$$\hat{w}_t = \text{softmax}(W \cdot \frac{1}{|C|} \sum_{c \in C} E[c])$$

Intuition:

- Look up embeddings of surrounding words,
- Average them,
- Predict the center word.

Embeddings: E is the embedding matrix, **a trainable parameter**.

Skip-gram: Target → Context

Given a target word w_t, predict nearby context words:

$$P(c \mid w_t) \propto \exp(E[c]^\top E[w_t])$$

Intuition:

- Use embedding of w_t,
- Predict which words appear around it,
- Works well for rare words.

Where embeddings come from: Rows of E are updated so the model gets better at these predictions.

Negative Sampling

Full softmax is too expensive for large vocabularies.
Negative sampling trains a binary classifier:

- Real pair $(w, c) \rightarrow$ push $E[w]$ and $E[c]$ closer.
- Random pair $(w, n) \rightarrow$ push $E[w]$ and $E[n]$ apart.

Training objective (schematic):

$$\log \sigma(E[c]^\top E[w]) + \sum_{n \sim P_n} \log \sigma(-E[n]^\top E[w])$$

Result: Words that occur in similar contexts get similar vectors.

Where Do Word2Vec Embeddings Come From?

Word2Vec is a neural network with two trainable matrices:

$$E_{\text{in}} \quad \text{and} \quad E_{\text{out}}$$

During training:

- The model receives **positive pairs** (word, context),
- Receives **negative samples** (word, random word),
- Adjusts rows of E_{in} and E_{out} to classify them correctly.

The word embeddings ARE these weight matrices.
After training:

- Each word w is represented by its learned row $E[w]$,
- Semantic structure emerges from millions of tiny updates.

From Embeddings to Measurement

Static word embeddings (Word2Vec, GloVe) give each word:

$$w \longrightarrow \mathbf{v}_w \in \mathbb{R}^d$$

Key idea: If meaning is encoded geometrically, then *cultural or political dimensions* should correspond to *directions* in the space.

From Embeddings to Measurement

Static word embeddings (Word2Vec, GloVe) give each word:

$$w \longrightarrow \mathbf{v}_w \in \mathbb{R}^d$$

Key idea: If meaning is encoded geometrically, then *cultural or political dimensions* should correspond to *directions* in the space.
Two influential applications:

- **Latent Semantic Scaling (Watanabe, 2020):** measure ideology or sentiment by projecting onto a semantic direction.
- **The Geometry of Culture (Kozlowski et al., 2019):** uncover cultural schemas using geometric axes in embedding spaces.

Latent Semantic Scaling (LSS)

LSS uses static embeddings to place words, documents, or actors on a *latent semantic dimension*.

Core idea:

$$\text{score}(w) = \mathbf{v}_w \cdot (\mathbf{v}_+ - \mathbf{v}_-)$$

Where:

- $\mathbf{v}_+$ = average embedding of positive/pole words
- $\mathbf{v}_-$ = average embedding of negative/opposing words

This defines a semantic direction.

Semantic Directions in Embedding Space

Given seed words:

$$D = \mathbf{v}_+ - \mathbf{v}_-$$

Any word gets:

$$\text{LSS}(w) = \mathbf{v}_w \cdot D$$

Interpretation:

- Positive score $\rightarrow$ closer to the positive pole
- Negative score $\rightarrow$ closer to the negative pole
- Magnitude $\rightarrow$ strength of association

Examples of Latent Semantic Scaling

Ideology:

$$+\text{pole} : \{\text{freedom},\ \text{market}, \text{tradition}\}$$

$$-\text{pole} : \{\text{equality},\ \text{rights}, \text{redistribution}\}$$

Sentiment:

$$+\text{pole} : \{\text{good}, \text{love}, \text{happy}\}$$

$$-\text{pole} : \{\text{bad}, \text{hate}, \text{sad}\}$$

Documents scored by averaging their words:

$$\text{LSS}(d) = \frac{1}{|d|} \sum_{w \in d} \text{LSS}(w)$$

Why LSS?

- Extremely fast, only requires dot products.
- Works with pretrained static embeddings.
- Uncovers *latent ideological or cultural bias*.
- Useful for political ideology estimation, framing, discourse analysis.

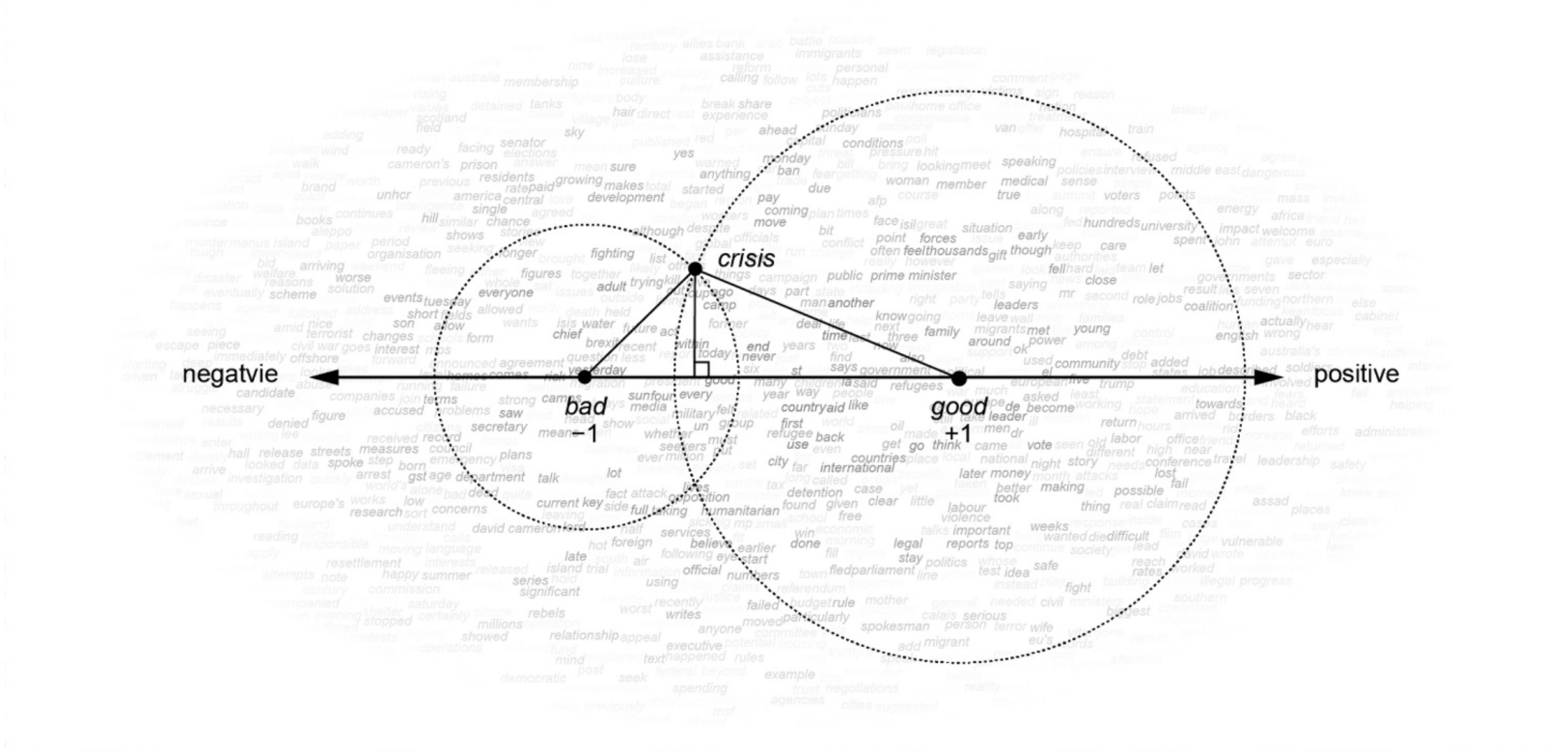


Figure 1. Conceptual illustration of word weighting by seed words in a latent semantic space. The arrow is the sentiment dimension in the semantic space and circles are proximities of positive seed ("good") words and negative seed words ("bad") to "crisis", which is projected on the sentiment dimension and receives a negative score.

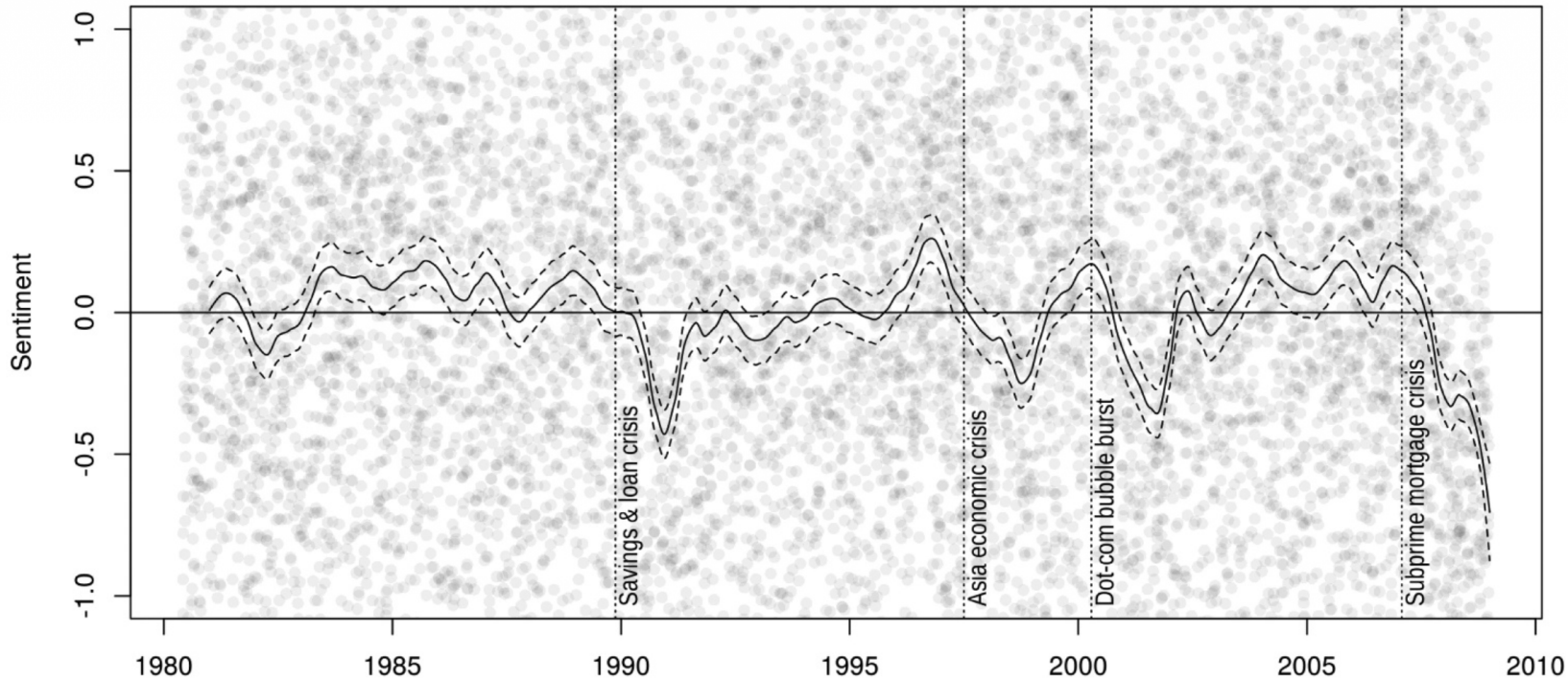


Figure 7. Longitudinal analysis of news articles on the economy (English) in the *New York Times* corpus by LSS. Curves are LOESS smoothed sentiment scores with 95% confidence intervals. Circles are individual sentiment scores of 10,000 news articles.

AMERICAN SOCIOLOGICAL ASSOCIATION

American Sociological Review
2019, Vol. 84(5) 905–949

DOI: 10.1177/0003122419877135
journals.sagepub.com/home/asr

SAGE

The Geometry of Culture: Analyzing the Meanings of Class through Word Embeddings

Austin C. Kozlowski,[a] Matt Taddy,[b] and James A. Evans[a,c]

The Geometry of Culture (Kozlowski et al., 2019)

Premise: Culture is encoded in geometric relationships in embedding space.

They model cultural schemas as:

- high-dimensional structures
- represented by semantic axes
- learned from large corpora (Google Books Ngrams)

The Geometry of Culture (Kozlowski et al., 2019)

Premise: Culture is encoded in geometric relationships in embedding space.

They model cultural schemas as:

- high-dimensional structures
- represented by semantic axes
- learned from large corpora (Google Books Ngrams)

Example schemas:

- gender
- class
- race
- modernity/tradition

Semantic Axes Represent Cultural Schemas

Given a binary concept:

$$\text{axis} = \mathbf{v}_A - \mathbf{v}_B$$

Example:

$$\text{gender axis} = \mathbf{v}_{\text{man}} - \mathbf{v}_{\text{woman}}$$

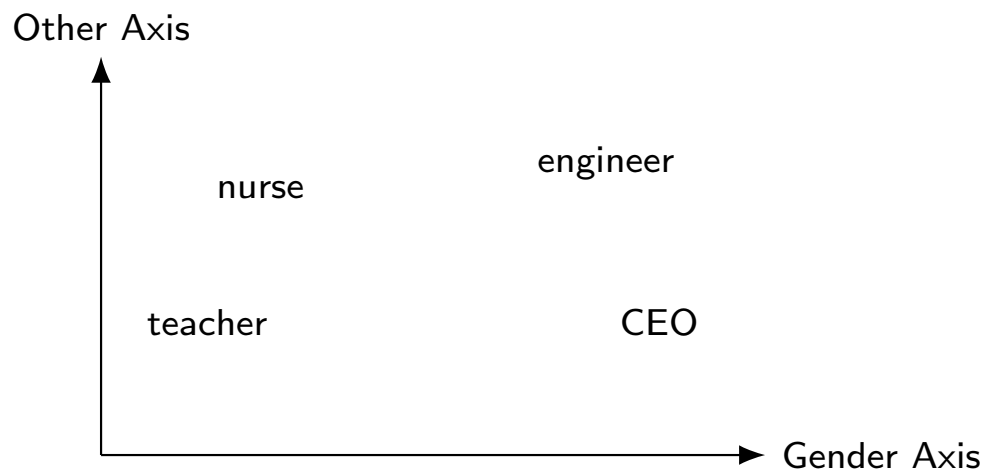
Then any concept c has projection:

$$\text{score}(c) = \mathbf{v}_c \cdot \text{axis}$$

This reveals cultural associations:

- occupations
- adjectives
- political concepts

Projections Reveal Cultural Structure



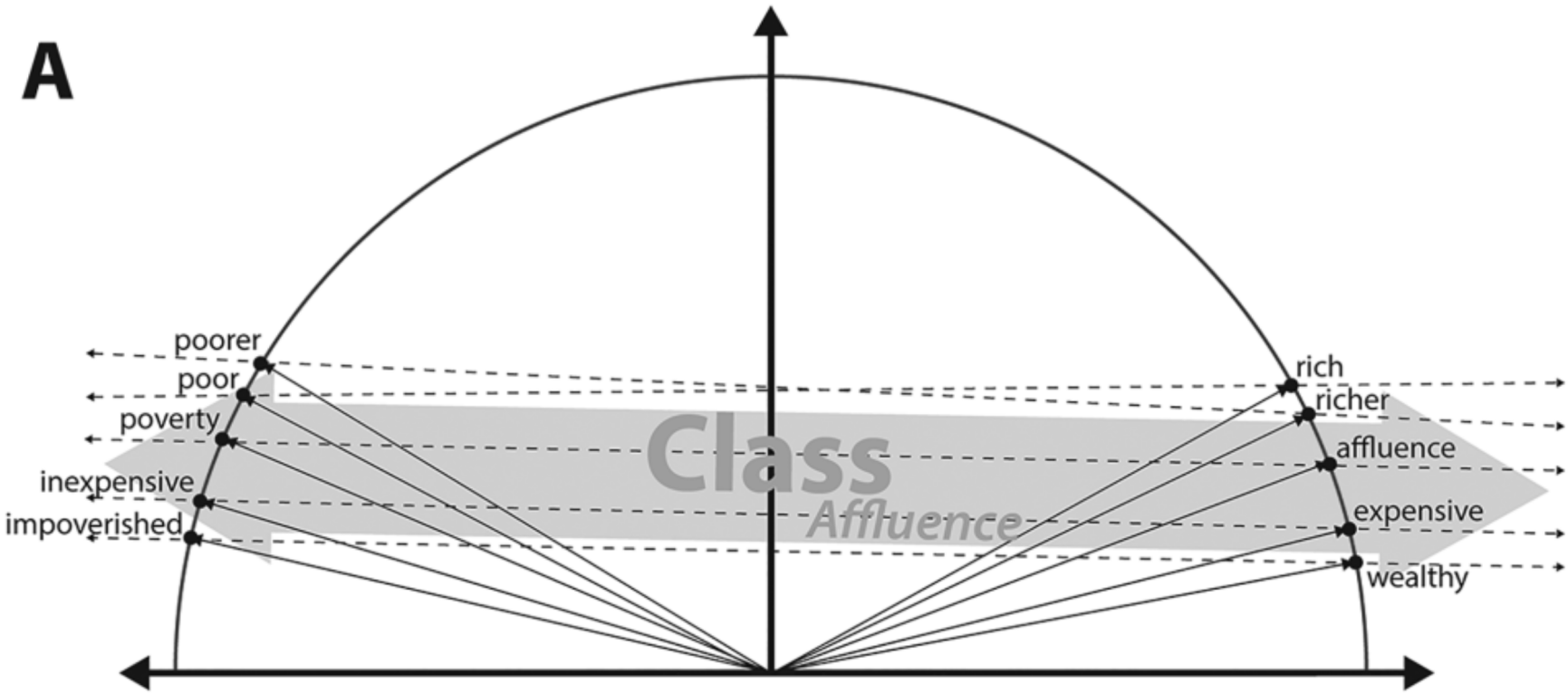
Cultural schemas emerge as geometric patterns.

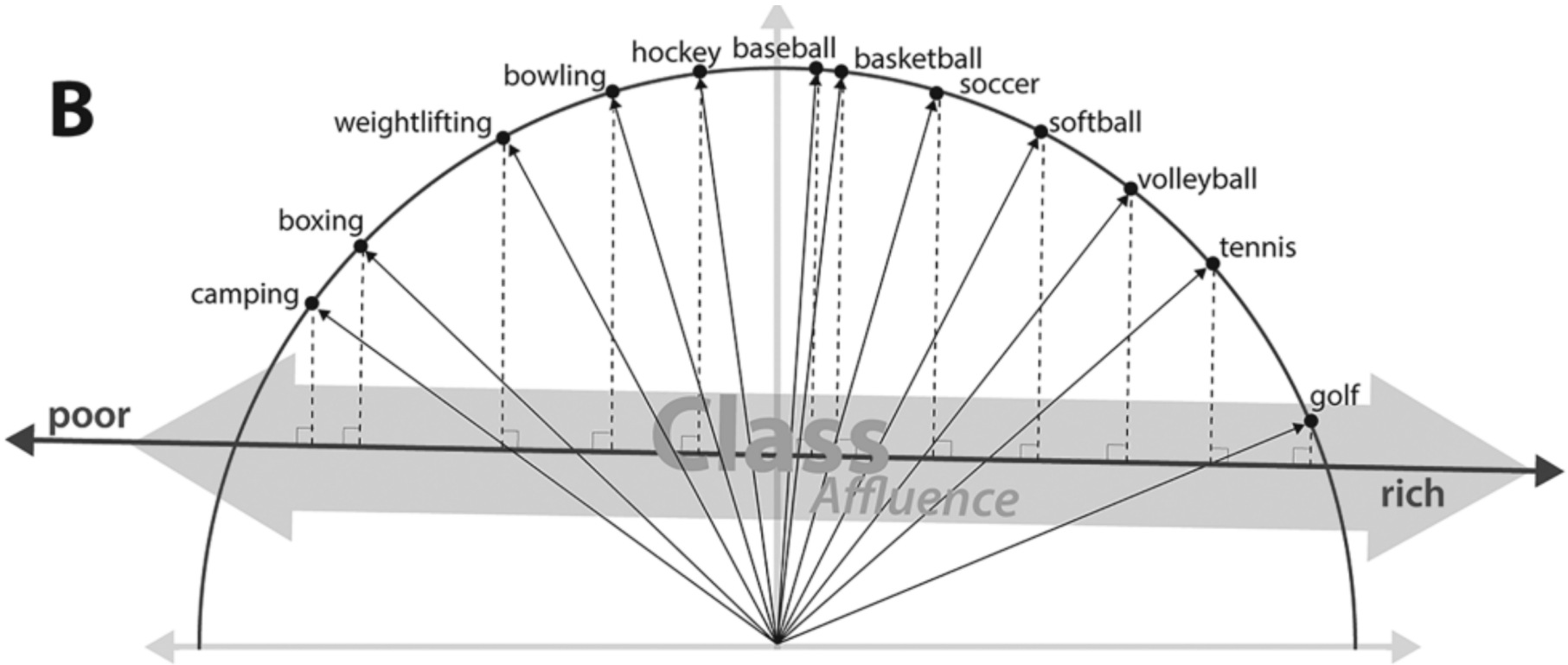
Why the Geometry of Culture Matters

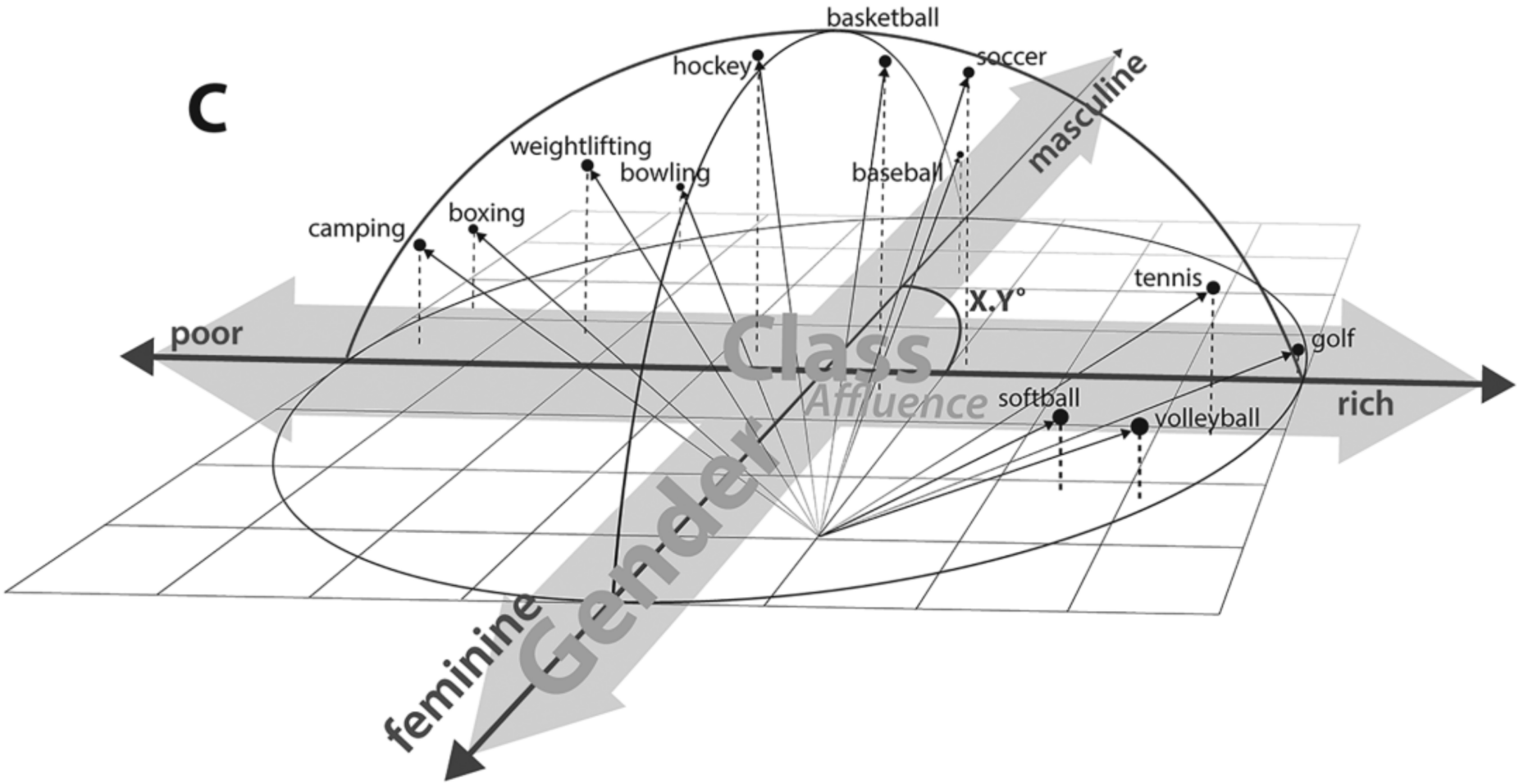
Implication: Cultural meaning can be measured empirically as *locations in vector space*.
Applications:

- cultural sociology
- ideology & belief systems
- bias in language
- representation of social categories

Key insight: Meaning is not discrete, it is geometric.


A
poorer
poor
poverty
inexpensive
impoverished
Class
Affluence
rich
richer
affluence
expensive
wealthy


B
hockey
baseball
basketball
bowling
soccer
weightlifting
softball
volleyball
boxing
tennis
camping
golf
poor
Class
Affluence
rich


C
basketball
hockey
soccer
weightlifting
bowling
baseball
masculine
camping
boxing
tennis
X.Y°
poor
Class
Affluence
golf
softball
rich
volleyball
Gender
feminine

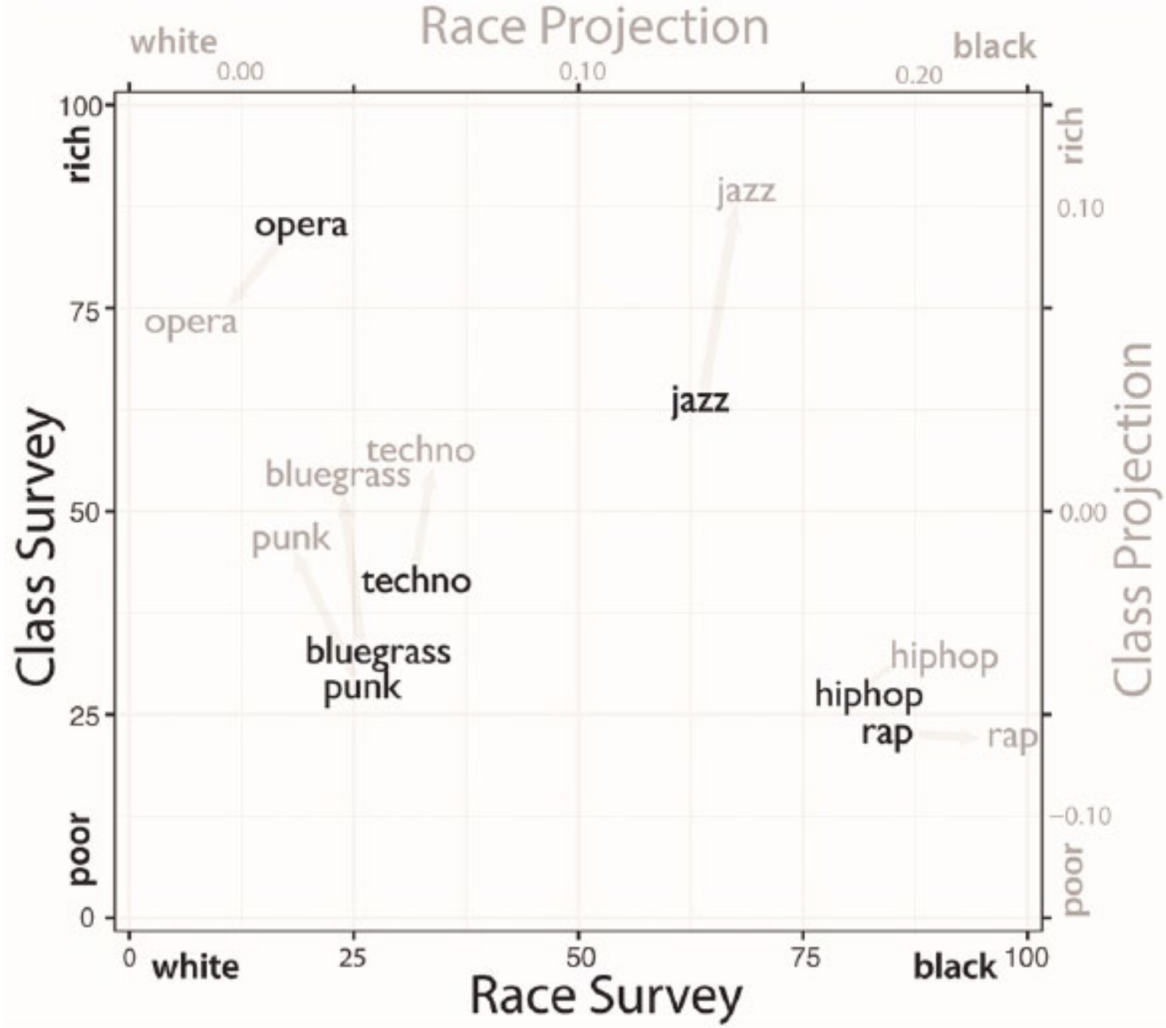


Figure 3. Projection of Music Genres onto Race and Class Dimensions of the Google News Word Embedding (Gray) and Average Survey Ratings for Race and Class Associations (Black)

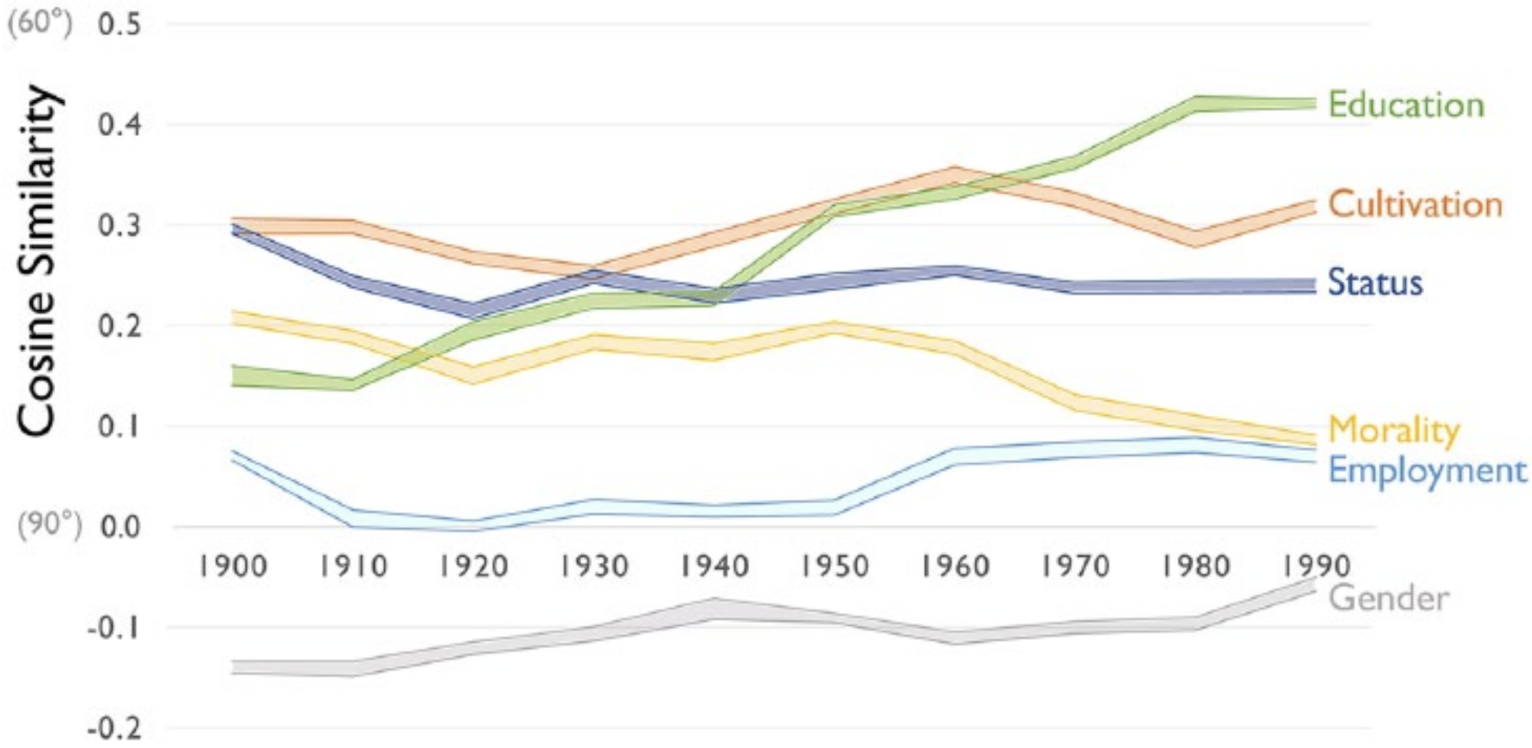
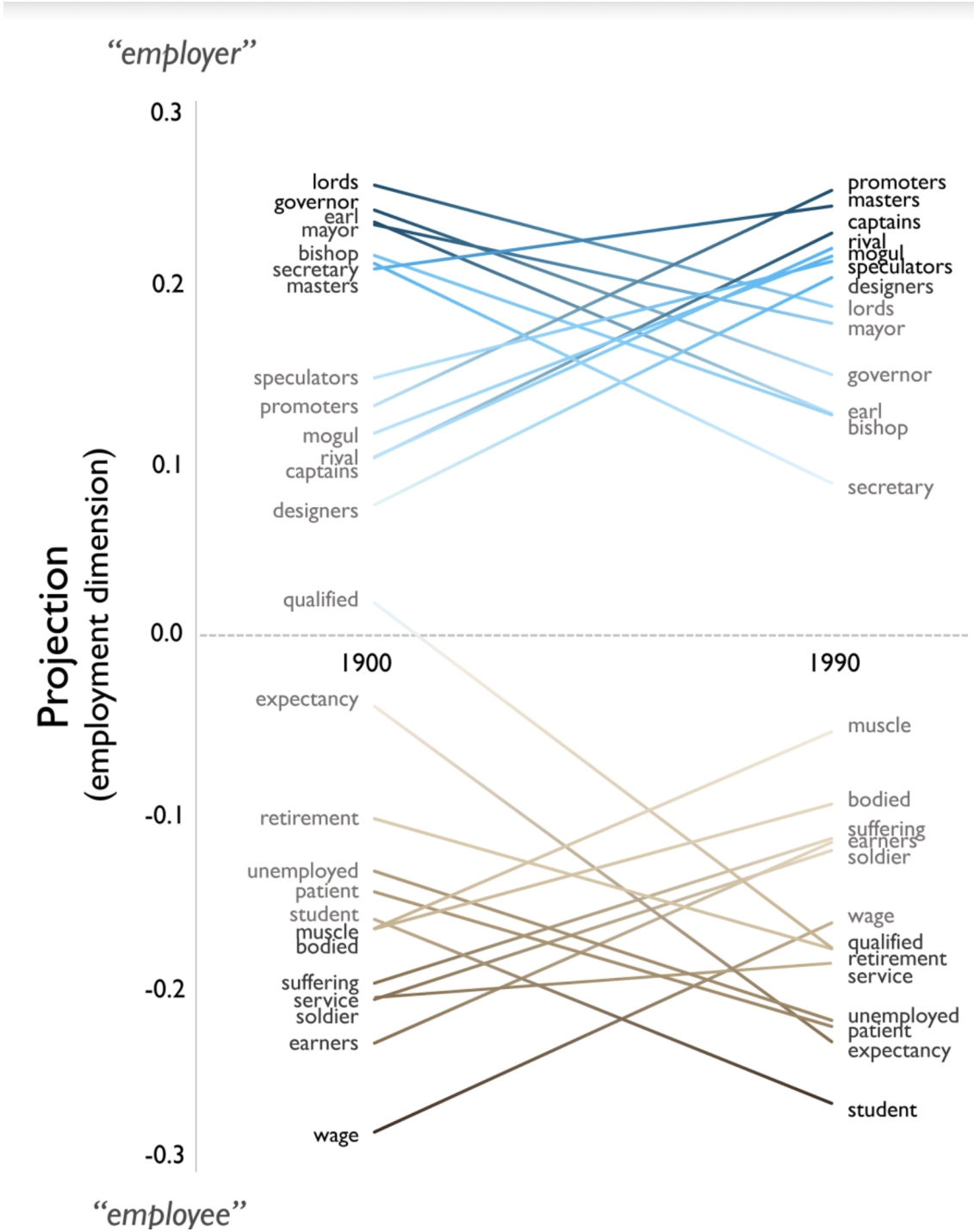


Figure 5. Cosine Similarity between the Affluence Dimension and Six Other Cultural Dimensions of Class by Decade; 1900 to 1999 Google Ngrams Corpus
Note: Bands represent 90 percent bootstrapped confidence intervals produced by subsampling.


"employer"
0.3
0.2
0.1
0.0
-0.1
-0.2
-0.3
Projection
(employment dimension)
lords
governor
earl
mayor
bishop
secretary
masters
speculators
promoters
mogul
rival
captains
designers
qualified
promoters
masters
captains
rival
mogul
speculators
designers
lords
mayor
governor
earl
bishop
secretary
1900
1990
expectancy
retirement
unemployed
patient
student
muscle
bodied
suffering
service
soldier
earners
wage
muscle
bodied
suffering
earners
soldier
wage
qualified
retirement
service
unemployed
patient
expectancy
student
"employee"

From Static to Contextual Meaning

Static embeddings assume:

$$\mathbf{v}_{\text{bank}} = \text{one vector}$$

But meaning is context-dependent:

- "river bank"
- "central bank"

From Static to Contextual Meaning

Static embeddings assume:

$$\mathbf{v}_{\text{bank}} = \text{one vector}$$

But meaning is context-dependent:

- "river bank"
- "central bank"

Next: Contextual embeddings (BERT, GPT) represent meaning as a function of surrounding words.

Why Transformers?

RNNs/LSTMs:

- process tokens sequentially,
- struggle with long-range dependencies,
- slow to train (cannot parallelize).

Transformers replace recurrence with *self-attention*:

- all tokens attend to all others in one step,
- capture long-range structure easily,
- highly parallel on GPUs.

Core Idea: Self-Attention

Each token computes:

$$\text{output}_i = \sum_j \alpha_{ij} \, V_j$$

- α_{ij} = how much token i attends to token j
- V_j = information carried by token j

Intuition: Each word extracts what it needs from every other word in the sentence.

Queries, Keys, Values

For token representation x:

$$Q = W_Q x, \quad K = W_K x, \quad V = W_V x$$

- Query = what this token is looking for.
- Key = how this token can be identified.
- Value = the information this token provides.

Scaled Dot-Product Attention

Weights:

$$A = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)$$

Output:

$$\tilde{H} = A\,V$$

Dot products give similarity; softmax converts it to attention weights. Scaling keeps gradients stable.

Multi-Head Attention

Run several attentions in parallel:

$$\text{MHA}(H) = \text{Concat}(\tilde{H}_1, \ldots, \tilde{H}_h)\, W_O$$

Why multiple heads?

- capture different relations (syntax, entities, topics),
- richer contextual representations.

What Do Positional Encodings Do?

Self-attention compares token embeddings only by content:

$$\text{Attention}(X) = \text{Attention}(\text{shuffle}(X))$$

It has **no sense of order**.

Transformers add a position vector p_t:

$$z_t = x_t + p_t$$

- x_t = token embedding,
- p_t = encoding of position t (sinusoidal or learned).

Now "dog" at position 1 and "dog" at position 5 have different internal representations, letting the model learn word order and syntax.

Transformer Encoder Block

Each layer:

1. Multi-head self-attention
2. Add & LayerNorm
3. Feed-forward network (MLP)
4. Add & LayerNorm

Residuals + LayerNorm enable deep, stable training.

Feed-Forward Network

Per token:

$$\text{FFN}(x) = W_2\,\sigma(W_1 x + b_1) + b_2$$

Adds nonlinear transformations and expands representational power.

GPT vs BERT

GPT (decoder-only):

- masked self-attention (autoregressive),
- predicts next token,
- used for generation.

BERT (encoder-only):

- bidirectional attention,
- masked language modeling,
- used for classification, embeddings, QA.

What Are Contextual Embeddings?

Static embeddings:

$$w \mapsto \mathbf{v}_w$$

Contextual embeddings:

$$w_t,$$

context $\mapsto \mathbf{h}_t$
Meaning depends on surrounding words:

- "bank" in *river bank*
- "bank" in *central bank*

Transformers compute these via repeated layers of attention + MLP.

How Contextual Embeddings Are Computed

For each token w_t:

1. Lookup embedding x_t and add position p_t.
2. Pass through L transformer layers:
   - self-attention mixes information across tokens,
   - feed-forward network transforms each position,
   - residuals + layernorm stabilize and refine.
3. Output $\mathbf{h}_t$ = representation of w_t *in context*.

These $\mathbf{h}_t$ are the **contextual embeddings**.

Why Contextual Embeddings for Social Science?

Contextual embeddings (BERT, GPT) produce:

$$w_t, \text{context} \mapsto \mathbf{h}_t$$

Meaning shifts with context (unlike Word2Vec).
This enables social scientists to:

- capture framing and semantic nuance,
- detect ideology or stance embedded in text,
- analyze discourse beyond surface word counts.

They allow us to represent *how language is used*, not just *which words appear*.

Framing Analysis with Contextual Embeddings

Same word, different meaning:

- "crisis" in *economic crisis*
- "crisis" in *migration crisis*
- "crisis" in *health crisis*

Contextual embeddings produce distinct vectors:

$$\mathbf{h}^{\text{econ}}_{\text{crisis}}, \mathbf{h}^{\text{mig}}_{\text{crisis}}, \mathbf{h}^{\text{health}}_{\text{crisis}}$$

Applications:

- mapping media framing across outlets,
- tracking how concepts shift over time,
- measuring competing narratives in political communication.

Measuring Ideology, Sentiment, and Stance

Contextual embeddings capture evaluative language:

- "immigrants are welcome" vs
- "immigrants are dangerous"

Embedding geometry encodes latent attitudes:

$$\text{ideology/stance} \approx \text{direction in embedding space}$$

Social science uses:

- estimating ideological positions of texts or actors,
- detecting political stance in social media,
- analyzing subtle cues (dogwhistles, implicit bias).

Improved Classification with Contextual Representations

Contextual embeddings improve:

- hate speech detection,
- misinformation classification,
- stance and sentiment analysis,
- event and protest detection,
- extremist content detection.

Why?

- handle negation: “not good” “good”
- understand sarcasm and irony,
- resolve references: “he lied again”

Models interpret *meaning*, not just word frequency.

Contextual Embeddings for Topic Discovery

BERT/GPT embeddings group texts by *semantic similarity*:

$$\mathbf{h}_{\text{doc}} = f(\mathbf{h}_1, \ldots, \mathbf{h}_T)$$

Benefits relative to bag-of-words models:

- identify subtle frames, not just topics,
- detect emergent narratives and events,
- cluster paraphrases and synonyms together,
- distinguish different senses of the same word.

Useful for:

- media ecosystem mapping,
- agenda-setting and framing studies,
- discourse analysis.

Analyzing Identity, Bias, and Social Meaning

Contextual embeddings capture how identity terms are used:

- "women voters" vs "women protesters"
- "migrants" in humanitarian vs threat frames
- "police" in reform vs crime contexts

These representations help uncover:

- stereotypes and implicit bias,
- in-group vs out-group framing,
- shifts in political rhetoric,
- patterns of marginalization or empowerment.

Key insight: embeddings represent *relational meaning* encoded in discourse.