Text and Images as Data in Social Science

AI and Economics Summer School

Andrea Ciccarone

ETH Zurich

August 31, 2026

Why Do Text and Images Need a Representation?

To study text, images, or video, we first need to represent the raw information numerically:

$$r_i \xrightarrow{\text{representation } f} x_i \in \mathbb{R}^d \xrightarrow{\text{statistical model } g} \widehat{y}_i.$$

- Text had natural units, such as words and phrases, that could be counted in a vocabulary
- Images and video were harder to represent because pixels have no immediate social-science meaning
- Modern embeddings learn numerical representations of each modality and can place them in a common semantic space

We begin with the traditional text-as-data workflow, then introduce learned representations.

Outline

1. Text as Data
2. Image as Data
3. Application: Measuring Partisanship in Images and Text
4. Beyond CLIP: Multimodal Representations

A Standard Text Analysis Workflow

Most text-as-data applications follow three steps:

1. Represent raw documents $d \in \mathcal{D}$ as a numerical array X
2. Map X into a predicted or latent measure $\widehat{Y}$
3. Use $\widehat{Y}$ in descriptive or causal analysis

- The collection of documents $\mathcal{D}$ is the **corpus**
- For example, newspaper articles can be represented numerically, classified by sentiment, and then used to study changes in political tone

The first problem is turning the corpus into a useful X.

Text as Data: Building the Representation

Each document contains tokens drawn from a vocabulary $\mathcal{V}$:

$$d_i \longrightarrow (w_{i1}, \ldots, w_{iT_i}), \qquad \mathcal{V} = \{v_1, \ldots, v_V\}.$$

- **Tokenize**: choose words, phrases, n-grams, or subword pieces as the units of analysis
- **Normalize**: decide how to treat capitalization, punctuation, numbers, and spelling
- **Reduce features**: remove stopwords, stem words, or retain them when the research question requires them
- **Construct** X: represent every document using the same vocabulary

These choices determine which textual distinctions the statistical model can use.

Bag of Words: Counting Words

Choose a vocabulary and count how often each term appears in each document.

Documents

d_1 "tax cuts create jobs"

d_2 "jobs create growth"

d_3 "jobs jobs jobs"

$\Longrightarrow$

Document–term matrix X

	tax	cuts	create	jobs	growth
d_1	1	1	1	1	0
d_2	0	0	1	1	1
d_3	0	0	0	**3**	0

$$x_{iv} = \text{count of term } v \text{ in document } i, \qquad X \in \mathbb{R}^{N \times V}.$$

The vector records word frequencies. It contains no information about word order, syntax, or context.

TF–IDF: Reweighting Word Counts

For term v in document i:

$$\text{tfidf}_{iv} = \underbrace{\frac{\text{count}_{iv}}{\sum_{v'} \text{count}_{iv'}}}_{\text{term frequency}} \times \underbrace{\log\left(\frac{N}{\#\{i' : x_{i'v} > 0\}}\right)}_{\text{inverse document frequency}}.$$

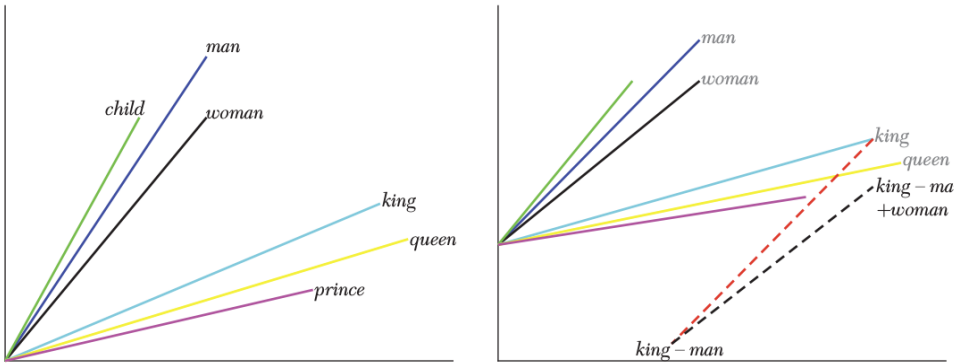
- Words with high frequency in one document and low frequency in the corpus receive more weight
- Words appearing everywhere receive little weight
- Cosine similarity between tf–idf vectors is a strong retrieval baseline
- TF–IDF changes the weights. The representation remains nonsemantic

What Do Count Representations Miss?

- **Synonymy**: "job security" and "employment stability" share meaning, not words
- **Polysemy**: "bank" means something different beside "river" and "loan"
- **Negation**: "not guilty" contains the word "guilty"
- **Composition**: "hardly effective" is not the sum of "hardly" and "effective"
- **Long-range context**: meaning can depend on words many tokens away

These failures motivate representations that use context to encode meaning.

Word Embeddings: Meaning as Geometry



$$\text{vec}(\text{king}) - \text{vec}(\text{man}) + \text{vec}(\text{woman}) \approx \text{vec}(\text{queen}).$$

Words that share meaning or relationships occupy related positions in the embedding space.

Mikolov et al. (2013)

How Do We Read an Embedding?

For a raw object $r_i \in \mathcal{X}$, an encoder produces a fixed-length vector:

$$f_\theta : \mathcal{X} \to \mathbb{R}^d, \qquad z_i = f_\theta(r_i) \in \mathbb{R}^d.$$

After normalization, $\tilde{z}_i = z_i / \|z_i\|$, cosine similarity is a dot product:

$$\text{sim}(i,j) = \frac{z_i^\top z_j}{\|z_i\|\|z_j\|} = \tilde{z}_i^\top \tilde{z}_j.$$

- Nearby vectors represent semantically related objects
- Geometric directions can encode concepts and relationships
- A fixed-length z_i can enter prediction, clustering, retrieval, or measurement models
- Dense coordinates require additional interpretation strategies, which we return to in the application

Word2Vec: Learning from Context

Self-supervised task: use a target word to predict the words that appear nearby.

"the **quick brown** fox **jumps over** the lazy dog"

target: **fox** context: **quick, brown, jumps, over**

$$\max_{\{e_v, c_v\}} \sum_t \sum_{j \in \mathcal{C}(t)} \log p(w_j \mid w_t), \qquad p(w_j \mid w_t) \propto \exp\left(c_{w_j}^\top e_{w_t}\right).$$

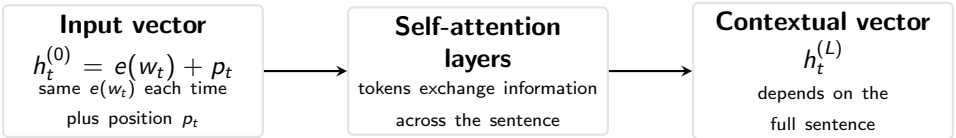
- Observed target–context pairs receive higher scores than unrelated pairs
- Words used in similar contexts acquire similar vectors
- The result is one learned vector e_v for each word v

e_{bank} is identical in "bank approved the loan" and "bank by the river."
This fixed representation is the key limitation that contextual embeddings address.

Mikolov et al. (2013)

BERT: Contextual Word Embeddings

BERT starts from a fixed token embedding and updates it using the surrounding words.



The input embedding e_{bank} is the same in both sentences. The final hidden state is not:

$$\text{“the } \textbf{bank} \text{ approved the loan”} \implies h_{\text{bank}}^{\text{finance}}$$

$$\text{“the } \textbf{bank} \text{ by the river”} \implies h_{\text{bank}}^{\text{geography}}$$

Devlin et al. (2019)

BERT Uses Self-Attention

At layer ℓ, token t has a current representation $h_t^{(\ell)}$.

BERT maps the current token representations into queries, keys, and values:

$$q_t = h_t^{(\ell)} W_Q, \qquad k_j = h_j^{(\ell)} W_K, \qquad v_j = h_j^{(\ell)} W_V.$$

- q_t: what information is token t looking for?
- k_j: what information does token j contain?
- v_j: what information can token j contribute?

BERT compares the query of token t with the key of every token j:

$$s_{tj} = \frac{q_t k_j'}{\sqrt{d_k}}.$$

A larger s_{tj} means token j is more relevant for updating token t.

Self-Attention Updates the Representation

The relevance scores become attention weights:

$$\alpha_{tj} = \text{softmax}_j\left(\frac{q_t k_j'}{\sqrt{d_k}}\right), \qquad \sum_j \alpha_{tj} = 1.$$

The target token combines information from all tokens:

$$\widetilde{h}_t = \sum_j \alpha_{tj} v_j.$$

"the **bank** approved the **loan**" $\Longrightarrow$ large $\alpha_{\text{bank,loan}}$

"the **bank** by the **river**" $\Longrightarrow$ large $\alpha_{\text{bank,river}}$

Attention lets the representation of "bank" incorporate the words that clarify its meaning.

BERT performs this operation with multiple attention heads and repeats it across many layers:

$$h_t^{(0)} \to h_t^{(1)} \to \cdots \to h_t^{(L)}.$$

How Is BERT Trained?

BERT learns its parameters through **masked language modeling**.

"the **[MASK]** approved the loan" $\Longrightarrow$ "**bank**"

BERT uses the left and right context to assign high probability to the hidden token:

$$p_\theta\,(\text{bank} \mid \text{the [MASK] approved the loan})\,.$$

Across all masked positions M, it minimizes

$$\mathcal{L}_{\text{MLM}}(\theta) = -\sum_{t \in M} \log p_\theta\left(w_t \mid w_{1:T \setminus M}\right).$$

The prediction errors update the token embeddings, W_Q, W_K, W_V, and the feed-forward layers.

Repeated prediction teaches BERT which context is informative.

Devlin et al. (2019)

Implementation: BERT in Python

```
from transformers import AutoTokenizer, AutoModel

tokenizer = AutoTokenizer.from_pretrained("bert-base-uncased")
bert = AutoModel.from_pretrained("bert-base-uncased")

batch = tokenizer(texts, padding=True, truncation=True,
                  return_tensors="pt")
H = bert(**batch).last_hidden_state # N x T x d

mask = batch["attention_mask"].unsqueeze(-1)
z = (H * mask).sum(1) / mask.sum(1) # N x d
```

$$H_i = (h_{i1}, \ldots, h_{iT_i}) \quad \longrightarrow \quad z_i = \frac{1}{T_i} \sum_{t=1}^{T_i} h_{it}.$$

Mean pooling converts the contextual token vectors in H into one document vector z_i.

A Common Application: Topic Models

Topic models are unsupervised: they find recurring themes in a collection of documents.

1. The model learns the topics

Jobs	wages, workers, employment
Immigration	border, asylum, visa

Each topic is a distribution over words that tend to appear together.

2. Each document mixes them

Jobs	**65%**
Immigration	**25%**
Other topics	**10%**

$$\theta_i = (0.65,\ 0.25,\ 0.10), \qquad \sum_k \theta_{ik} = 1.$$

Researchers use the estimated topic shares $\widehat{\theta}_i$ as variables for speeches, news, legislation, and reviews.

The Generative Process

Input: a bag-of-words representation of the documents and a chosen number of topics K.

1. Each topic β_k is a probability distribution over words
2. Each document draws topic shares

$$\theta_i \sim \text{Dirichlet}(\alpha), \qquad \sum_{k=1}^{K} \theta_{ik} = 1$$

3. Those shares define the document's distribution over words

$$q_i = \sum_{k=1}^{K} \theta_{ik} \beta_k$$

4. Given document length m_i, the observed word counts are generated as

$$c_i \sim \text{Multinomial}(m_i, q_i)$$

Estimation reverses the process: $c_i \longrightarrow \widehat{\beta}_k, \widehat{\theta}_i$. Blei, Ng, and Jordan (2003)

Modern Topic Models Can Use Richer Representations

Modern topic modeling involves two familiar objects:

$$d_i \xrightarrow{\text{representation } h_\phi} x_i \xrightarrow{\text{topic model } p_\theta} \widehat{\pi}_i.$$

Document representation

- Word or n-gram counts
- Contextual embeddings
- Sparse, labeled concept counts

Topic-model structure

- Each topic is a distribution over features
- Each document is a mixture of topics
- Neural inference for topic shares

A richer representation can enter the familiar topic-model structure.

Biderman et al. (2026)

Example: Replacing Words with Concepts

"Working night shift can be difficult especially long-term, adapting your social life to a nighttime world can be seriously difficult for some people."

Count	Labeled SAE concept
13	Job performance / employment
11	Sleep and nighttime activities
7	Scheduling and time management
6	Fatigue and exhaustion
5	Difficulty and challenges

Biderman et al. (2026); GemmaScope SAE

$$a_{ij} \xrightarrow{\text{SAE}} r_{ij}, \qquad c_{iv} = \sum_j \mathbf{1}\{r_{ij,v} > q_v\}.$$

- The language model and SAE are pretrained and frozen
- Each column of c_i counts a labeled semantic concept
- The topic model retains the familiar count-based structure with a richer vocabulary

Do the Estimated Topics Make Sense?

Four topics estimated from Glassdoor reviews using SAE concept counts:

Financial performance and strategy

Financial performance; business strategy; employee evaluation

Retail and service work

Food preparation; brands and stores; online shopping and customer service

Healthcare careers and growth

Health-profession requirements; mentorship; growth opportunities

Leave rights and scheduling

FMLA; schedules and time management; job responsibilities

The unsupervised model recovers recognizable dimensions of workplace experience.

Biderman et al. (2026)

Taking Stock: Text as Data

- **Counts and dictionaries** give us transparent, named features
- **Embeddings** capture semantic information in coordinates that require interpretation
- **Dimensionality reduction** turns a high-dimensional representation into a smaller set of useful variables
- **Pretrained representations** can enter familiar tools such as topic models

New representations can be mixed with traditional statistical machinery to match the research question.

Outline

1. Text as Data
2. Image as Data
3. Application: Measuring Partisanship in Images and Text
4. Beyond CLIP: Multimodal Representations

Why Study Images?

- Political communication, identity, emotion, and social behavior are often expressed visually
- Historically, researchers relied on manual coding, which limited scale and reproducibility
- Pretrained vision and multimodal models now turn images into reusable numerical representations
- These representations make large visual corpora available for prediction and measurement

The empirical task is to construct valid variables from visual evidence.

Supervised Tasks with Images

As before, we can write the prediction problem as $Y = f(X)$.

- **Image classification**: assign a label or score to the entire image
  - The label can describe an object, a scene, or a broader concept such as partisanship
- **Object detection**: locate candidate regions and classify what appears inside them
  - Examples include faces, people, political symbols, and text inside an image
- **Face and sentiment analysis**: predict identity, demographic traits, expressed emotion, or evoked emotion

The outcome must be defined before the model is trained, and subjective labels require particular care.

Example: Measuring Partisanship in Images



High predicted probability of Democratic content



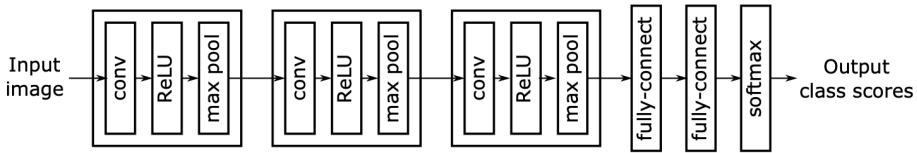
Low predicted probability of Democratic content

The classifier learns visual patterns associated with the label.

Convolutional Neural Networks

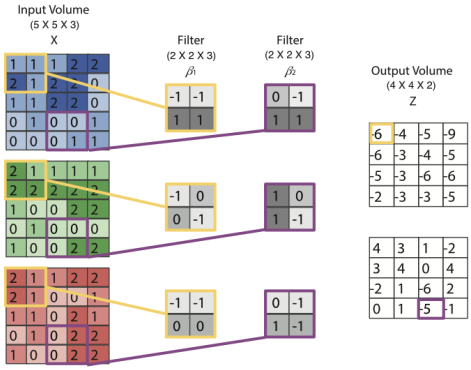
- Many image-analysis tasks traditionally rely on **Convolutional Neural Networks** (CNNs)
- CNNs are neural networks characterized by an operation called **convolution**
- A convolutional layer can reduce an image's dimensions while preserving important features
- A small array of weights, called a **kernel**, moves across the input image
- At each location, the kernel and the corresponding image values are multiplied element by element and summed to produce one output value

CNN Architecture: An Example



Source: Joo and Steinert-Threlkeld (2018)

Convolution Layer: An Example

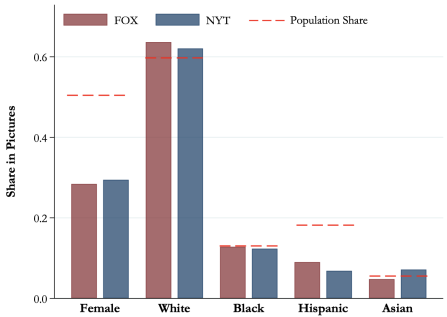


Source: Webb Williams, Casas and Wilkerson (2020)

How Do We Train a CNN?

- In general, we do not need to design a CNN architecture from scratch
- Training a CNN from scratch requires substantial data, computing power, and time
- Common alternatives include:
  - off-the-shelf computer-vision libraries and APIs
  - **fine-tuning** a pretrained CNN such as ResNet
- Fine-tuning adapts an existing network by updating its final layers for the task at hand
- Removing the final classification layer gives a feature vector that can be used as an **image representation**

Who Appears in News Images? Fox, NYT, and the U.S. Population



Notes: This figure shows the shares of different gender and ethnic groups in the pictures of Fox and NYT. The red lines indicate the share of each group in the U.S. population.

Source: Ash et al. (2022)

CLIP: Images and Text in a Shared Space

CLIP trains two encoders:

$$u_i = \frac{f_I(I_i)}{\|f_I(I_i)\|}, \qquad v_i = \frac{f_T(T_i)}{\|f_T(T_i)\|}, \qquad u_i, v_i \in \mathbb{R}^d.$$

- I_i is an image and T_i is its caption
- Both vectors occupy the same learned semantic geometry
- A text prompt can classify an image with no task-specific training
- A researcher can directly compare what an image shows with what a transcript says

Why it matters: CLIP lets us describe, compare, and classify images using language.

Radford et al. (2021)

How Is CLIP Trained?

In a batch of N matched image–text pairs, define every cross-modal similarity

$$s_{ij} = \frac{u_i^\top v_j}{\tau}.$$

CLIP solves the matching problem in both directions:

$$\mathcal{L}_{\text{CLIP}} = -\frac{1}{2N} \sum_{i=1}^{N} \left[\log \frac{e^{s_{ii}}}{\sum_j e^{s_{ij}}} + \log \frac{e^{s_{ii}}}{\sum_j e^{s_{ji}}} \right].$$

- i indexes the anchor image–text pair; j indexes every candidate match in the batch
- $j = i$ is the positive match; $j \neq i$ gives the in-batch negatives
- The loss teaches which image belongs with which text without hand-labeled categories
- $\tau > 0$ is the temperature: smaller values make the loss more sensitive to similarity differences

Describing Images with Text Concepts

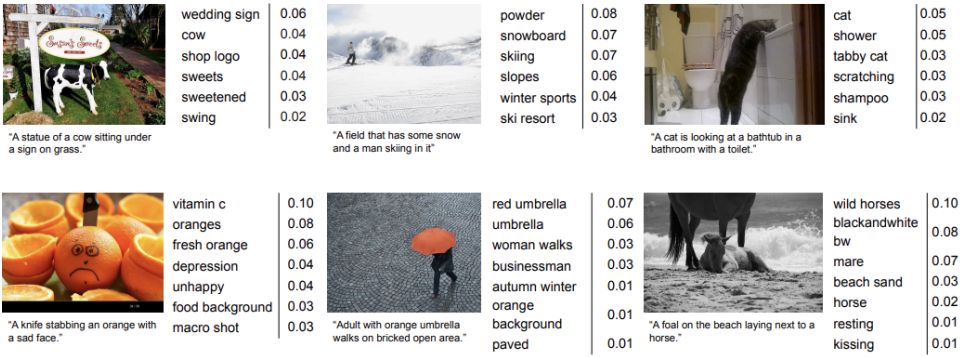
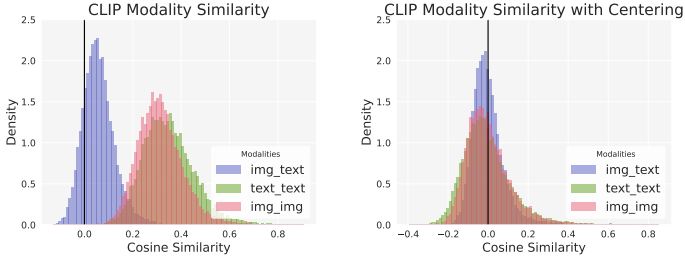


Figure 2: Example images from MSCOCO shown with their captions below and their concept decompositions on the right. We display the top seven concepts for visualization purposes, but images in the figure had decompositions with 7-20 concepts.

Bhalla et al. (2024), Figure 2

CLIP Still Has a Modality Gap



- Raw image and text embeddings share a geometry and occupy different regions
- SpLICE mean-centers each modality and renormalizes the vectors
- After centering, the distributions align, so one text-defined concept dictionary can describe images and text

Bhalla et al. (2024), Figure 7; MSCOCO images and captions

Taking Stock: Image as Data

- CNNs replaced engineered visual features with learned hierarchical features
- Pretraining and transfer learning made those features usable with modest labeled samples
- CLIP makes visual and textual meaning directly comparable

Image embeddings support comparison, decomposition, and downstream research designs.

Outline

1. Text as Data
2. Image as Data
3. Application: Measuring Partisanship in Images and Text
4. Beyond CLIP: Multimodal Representations

A Running Example: Partisanship in Images and Text

Research task: measure partisan content separately in the images and language of political videos.

- **Labeled corpus:** 15,427 Democratic and Republican television ads from the Wesleyan Media Project, 2016–2020
- **Shared representation:** CLIP maps video frames and transcript segments into the same 512-dimensional space
- **Supervised measure:** train separate image and text classifiers to predict the same sponsor-party label
- **Interpretation:** decompose the dense embeddings into named concepts and test which concepts drive each prediction

The application follows a reusable workflow: **represent** $\rightarrow$ **predict** $\rightarrow$ **interpret**.

Gentzkow and Shapiro (2010); Fowler et al. (2021)

Representing Political Ads in CLIP Space



$\downarrow$

$x^1_{img} \in \mathbb{R}^{512}$

...



$\downarrow$

$x^t_{img} \in \mathbb{R}^{512}$

...



$\downarrow$

$x^N_{img} \in \mathbb{R}^{512}$

"Dad helped build this business from scratch..."

$\downarrow$

$x^1_{txt} \in \mathbb{R}^{512}$

...

"...protect Medicare and Social Security, especially with COVID..."

$\downarrow$

$x^t_{txt} \in \mathbb{R}^{512}$

...

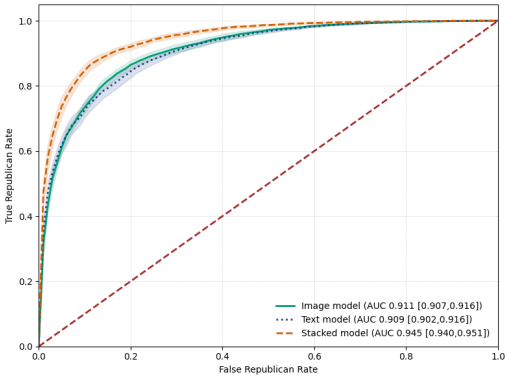
"My friends, my family, my community, that's why I'm running for Congress..."

$\downarrow$

$x^N_{txt} \in \mathbb{R}^{512}$

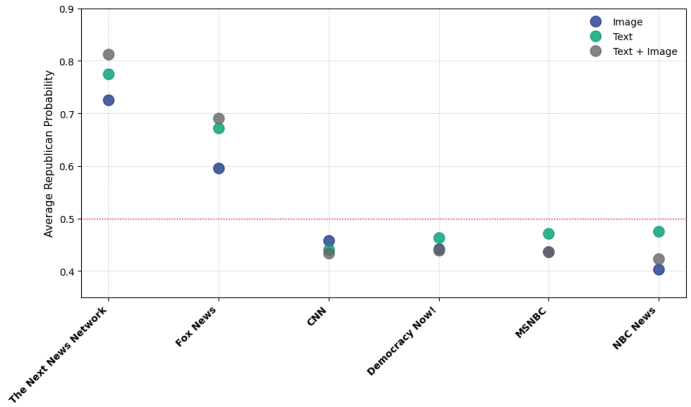
$$\text{Raw videos} \xrightarrow[\text{pre-trained parameters}]{\text{CLIP}} \{x^1_{mod}, \ldots, x^N_{mod}\} \xrightarrow{\text{aggregation}} \mathbf{v}_{mod} \xrightarrow[\text{trainable parameters}]{\text{neural network}} P(\text{Rep} \mid \mathbf{v}_{mod})$$

How Predictive Are Images and Text?



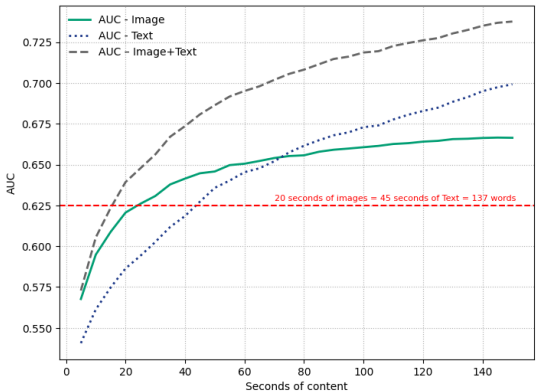
The same outcome and evaluation metric let us compare how much predictive information each modality carries.

Classifying Partisanship Across News Channels



The classifier was trained on political ads and then applied to news videos without retraining.

Images Convey Partisan Information More Quickly



At short exposure lengths, images carry more predictive signal than text.

Opening the Black Box: Emotions vs. Topics

- Images convey partisan information quickly. Do images and text rely on **different concepts**?



$$\mathbf{x}^{\text{img}} = \begin{pmatrix} x_1^{\text{img}} \\ \vdots \\ x_K^{\text{img}} \end{pmatrix}$$

Joint Vocabulary

{fear, anger..., immigration}

$$\mathbf{C} = \begin{bmatrix} c_1^1 & c_1^2 & \dots \\ \vdots & \ddots & \\ c_K^1 & & c_K^C \end{bmatrix}$$

Sparse Decomposition

$$\mathbf{w}^{\text{img}} = \begin{cases} \text{immigration} = 0.15 \\ \text{fear} = 0.08 \\ \vdots \\ \text{calm} = 0 \end{cases}$$

Intuition: represent videos as linear combinations of emotions and topics.

Bhalla et al. (2024)

Estimating the Concept Weights

For an image or text embedding $\widetilde{x}$, SpLICE estimates nonnegative concept weights:

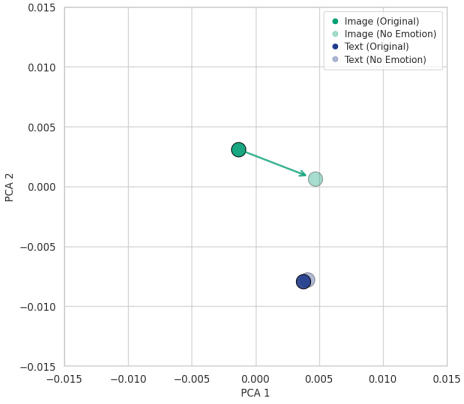
$$w^\star = \arg\min_{w \geq 0} \|Cw - \widetilde{x}\|_2^2 + 2\lambda_1 \|w\|_1, \qquad \widehat{x} = Cw^\star.$$

- $w_c^\star > 0$ means concept c contributes to the sparse reconstruction
- The ℓ_1 penalty limits the number of active concepts, making the representation readable
- The same emotion–topic dictionary is used for images and text
- Sparsity is tuned separately by modality to match reconstruction quality before comparing weights or downstream effects

SpLICE uses CLIP geometry to express an embedding through named semantic components.

Bhalla et al. (2024)

What Happens If We Remove a Concept?



Original reconstruction

$$\widehat{x}_i = Cw_i.$$

Remove concept group E

$$\widehat{x}_{i,-E} = Cw_{i,-E}.$$

The arrow shows the semantic shift

$$\widehat{x}_i \longrightarrow \widehat{x}_{i,-E}.$$

Pass both through the same predictor

$$\Delta_{i,E} = m(\widehat{x}_i) - m(\widehat{x}_{i,-E}).$$

A larger $|\Delta_{i,E}|$ means the prediction relies more on the removed concepts.

Using Concept Removal to Explain a Model

Once $\widetilde{x}$ is reconstructed as $\widehat{x}_i = Cw_i$, set one concept weight to zero and reconstruct again:

$$w_{i,-c} = w_i \text{ with } w_{ic} = 0, \qquad \widehat{x}_{i,-c} = Cw_{i,-c}.$$

Compare the model's prediction with and without concept c:

$$\Delta_c^{\text{mod}} = \frac{1}{|I_c|} \sum_{i \in I_c} \left[m^{\text{mod}}(\hat{\mathbf{x}}_i) - m^{\text{mod}}(\hat{\mathbf{x}}_{i,-c}) \right]$$

- $\Delta_c > 0$ means concept c raises the model's predicted outcome on average
- Comparing Δ_c across modalities reveals whether image and text models use the same semantic signal

What Can Researchers Do with These Methods?

- **Build comparable measures** from text and images using one semantic space
- **Compare information across modalities** under a common outcome and evaluation design
- **Recover named semantic components** from otherwise opaque embedding coordinates
- **Audit prediction models** through concept removal and cross-modal attribution

These operations apply to many multimodal research questions.

Outline

1. Text as Data
2. Image as Data
3. Application: Measuring Partisanship in Images and Text
4. Beyond CLIP: Multimodal Representations

Beyond CLIP: More Than Two Modalities

CLIP aligns image and text. New multimodal embedding models extend the same idea:

$$z = f_\theta(r), \qquad r \in \{\text{text, image, audio, video, document}\}, \qquad z \in \mathbb{R}^d.$$

- Google's **Gemini Embedding 2** maps all five input types into one unified space
- The same cosine similarity can compare a sentence with a frame, an audio clip, or a video segment
- For long videos, embed overlapping clips and aggregate or retain a time series of semantic vectors
- Researchers can study a multimedia object jointly across its text, image, audio, and video components

Google Gemini API, Gemini Embedding 2 documentation (2026)

Implementation: Gemini Embedding 2 in Python

```python
from google import genai
from google.genai import types

client = genai.Client()
video = open("campaign_clip.mp4", "rb").read()

result = client.models.embed_content(
    model="gemini-embedding-2",
    contents=[
        types.Content(parts=[types.Part.from_text(
            text="a political campaign rally")]),
        types.Content(parts=[types.Part.from_bytes(
            data=video, mime_type="video/mp4")]),
    ],
)

z_text, z_video = [e.values for e in result.embeddings]
```

Separate `Content` objects return comparable vectors for each modality.

Google AI for Developers, Embeddings guide (2026)

Recap

- Text was historically easier to represent because words provide a natural vocabulary
- Embeddings map information into learned semantic spaces
- BERT adds context; labeled concepts can make text representations easier to interpret
- CLIP and multimodal models extend this idea across text, images, audio, and video

A shared space makes different forms of information directly comparable.