

# TIMC Data Science Tools and Techniques

## Detecting Social Influence

“We can’t let up....

This is something we cannot be episodic about.

The defense of our Nation,  
the defense of our elections  
[will be part of my focus]  
every single day

for as long as I can see into the future.”

GEN P. M. Nakasone

DIRNSA, Commander US CYBERCOM

Reagan National Defense Forum, December 7, 2019.

# Tools and Techniques

# HDBSCAN:

Towards pushbutton  
density-based clustering

- HDBSCAN: Hierarchical Density-Based Spatial Clustering of Applications with Noise
  - Campello-Moulavi-Sander, Density-Based Clustering Based on Hierarchical Density Estimates, Pacific-Asia KDD 2013, 160-172.
  - Robust single linkage clustering with flat cluster extraction

<https://github.com/scikit-learn-contrib/hdbscan>

```
pip install hdbscan
```

We'd like an algorithm that:

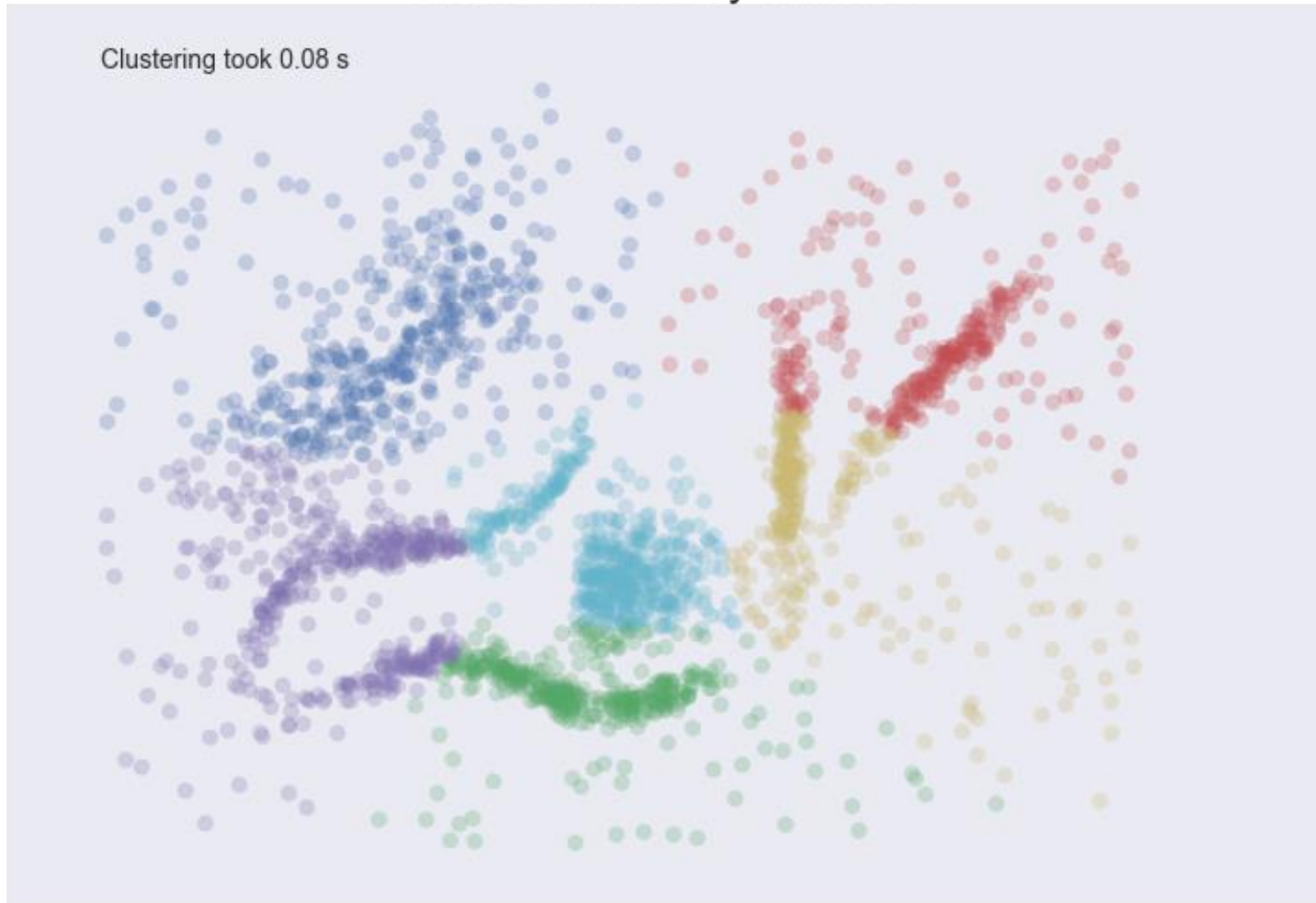
- Is density-based
- Is suitable for any metric (e.g., Euclidean, Hamming, Manhattan)
- Does not require a fixed number of clusters
- Is robust to noise and small perturbations in data
- Is parameter-free\*

1. Transform the space by changing the distance between points (**mutual reachability**)
2. Build the minimum spanning tree of the distance weighted graph of connections
3. Construct a **cluster hierarchy** of connected components
4. Condense the cluster hierarchy based on minimum cluster size
5. Extract **stable clusters** from the condensed tree
6. Use fast methods for nearest neighbors/spanning trees

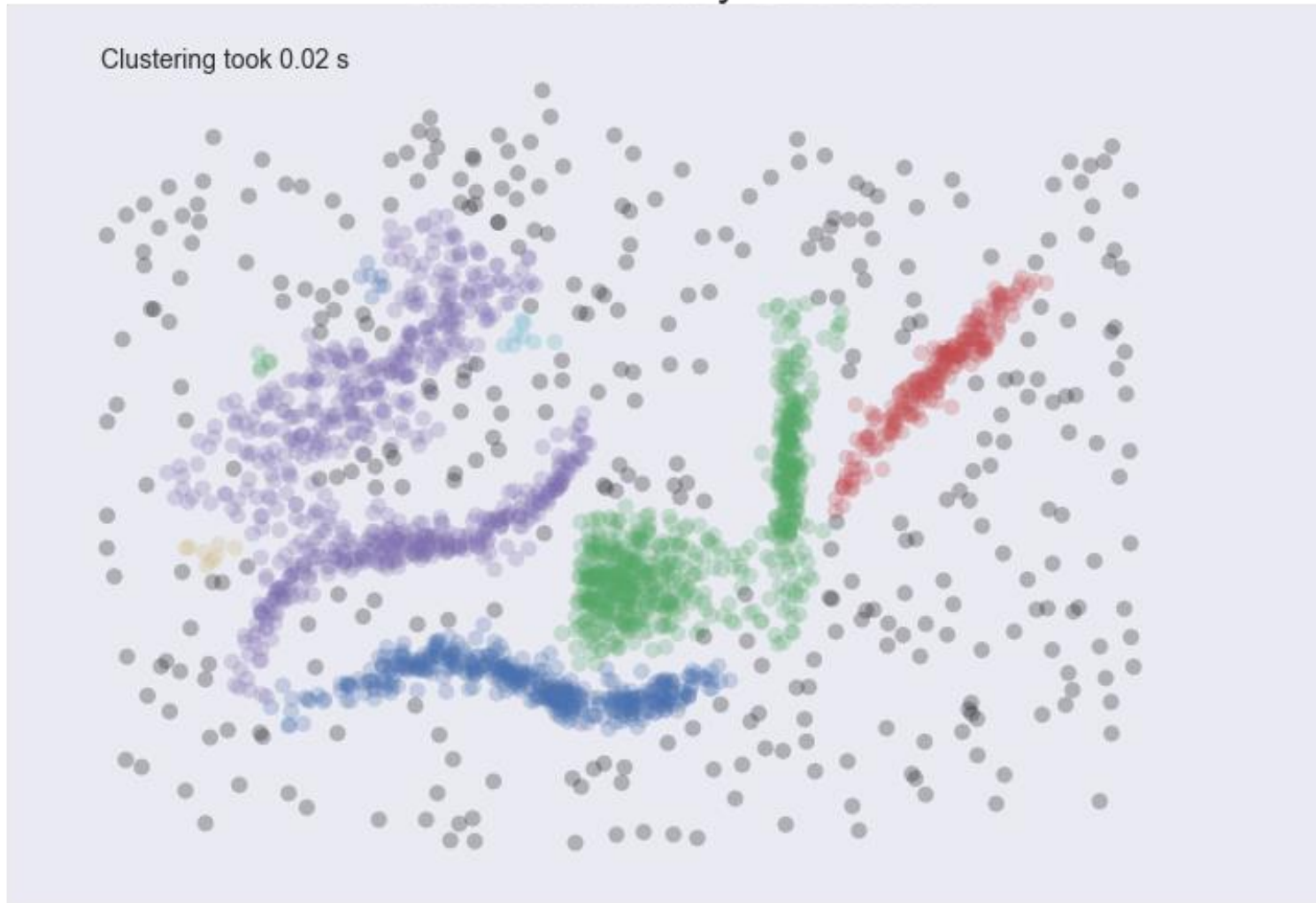




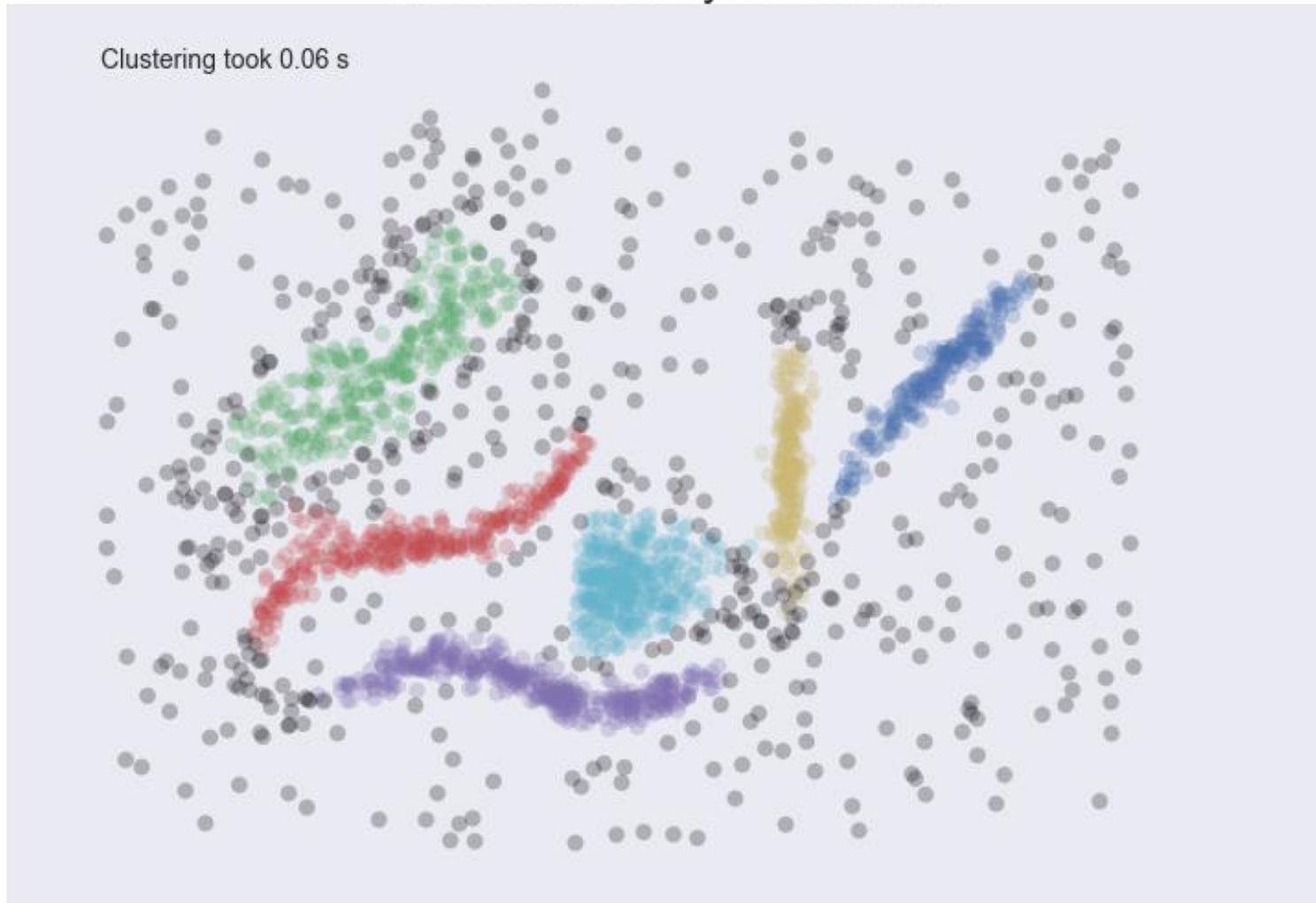
## Clusters found by KMeans



## Clusters found by DBSCAN



## Clusters found by HDBSCAN



	Number of points
Interactive	100,000
Over coffee	500,000
Over lunch	1,000,000
Over night	5,000,000

- Small-to-moderate dimension; precompiled distance

■ See also:

- Remember Leland McInnes' excellent NSC 2016 talk
- John Healy's PyData NYC 2018 talk <https://youtu.be/dGsxd67IFiU>
- Leland McInnes' SciPy 2016 talk <https://youtu.be/AgPQ76Rli6A>
- Read the Docs: [How HDBSCAN works](#)

UMAP:

Dimensionality reduction  
grounded in theory

- UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction
- Leland McInnes, John Healy, James Melville,  
<https://arxiv.org/abs/1802.03426>
  - Fuzzy topology-based low-dimension models of data

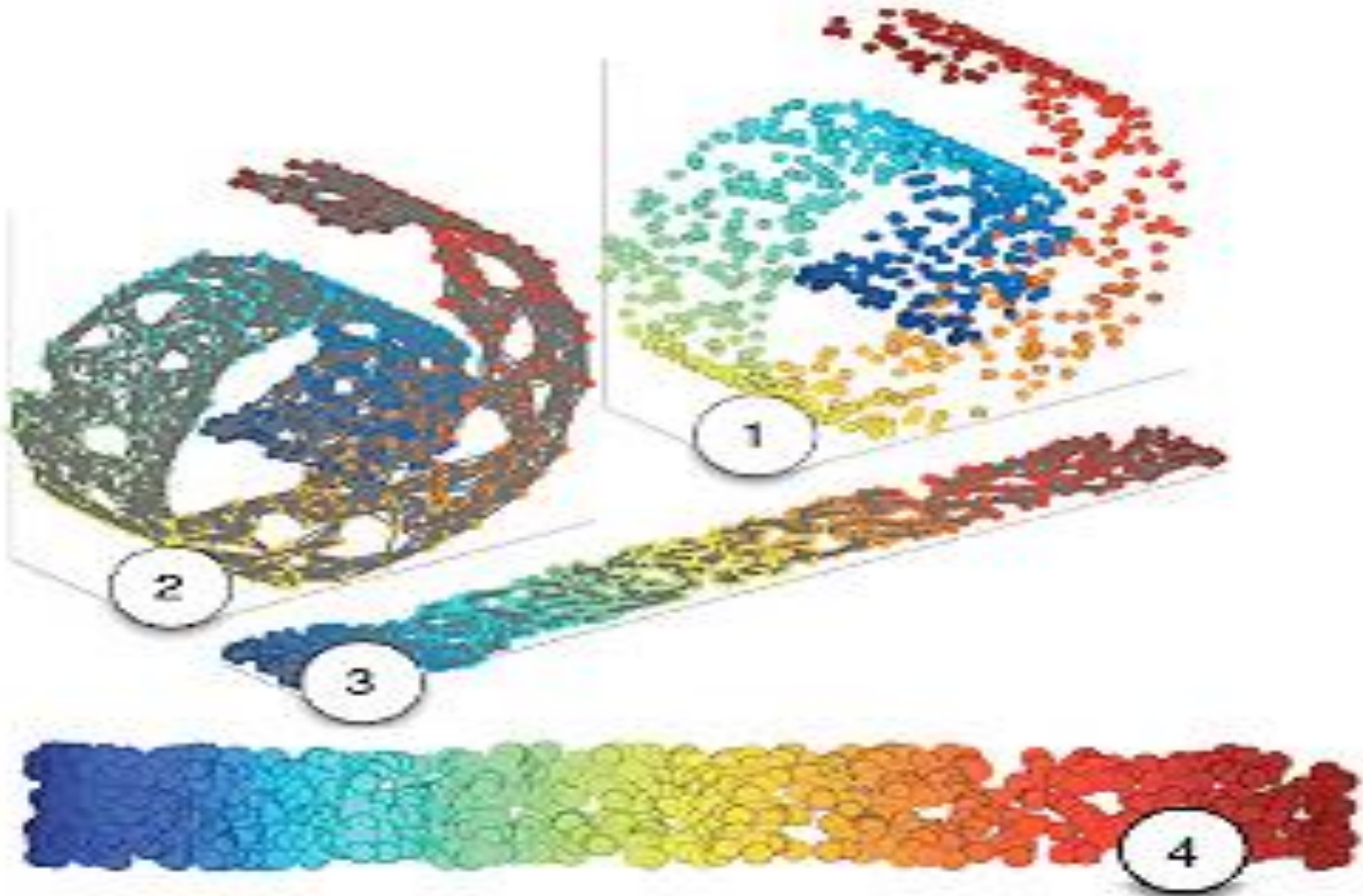


<https://github.com/Imcinnnes/umap>

`pip install umap-learn`



# What is manifold learning?



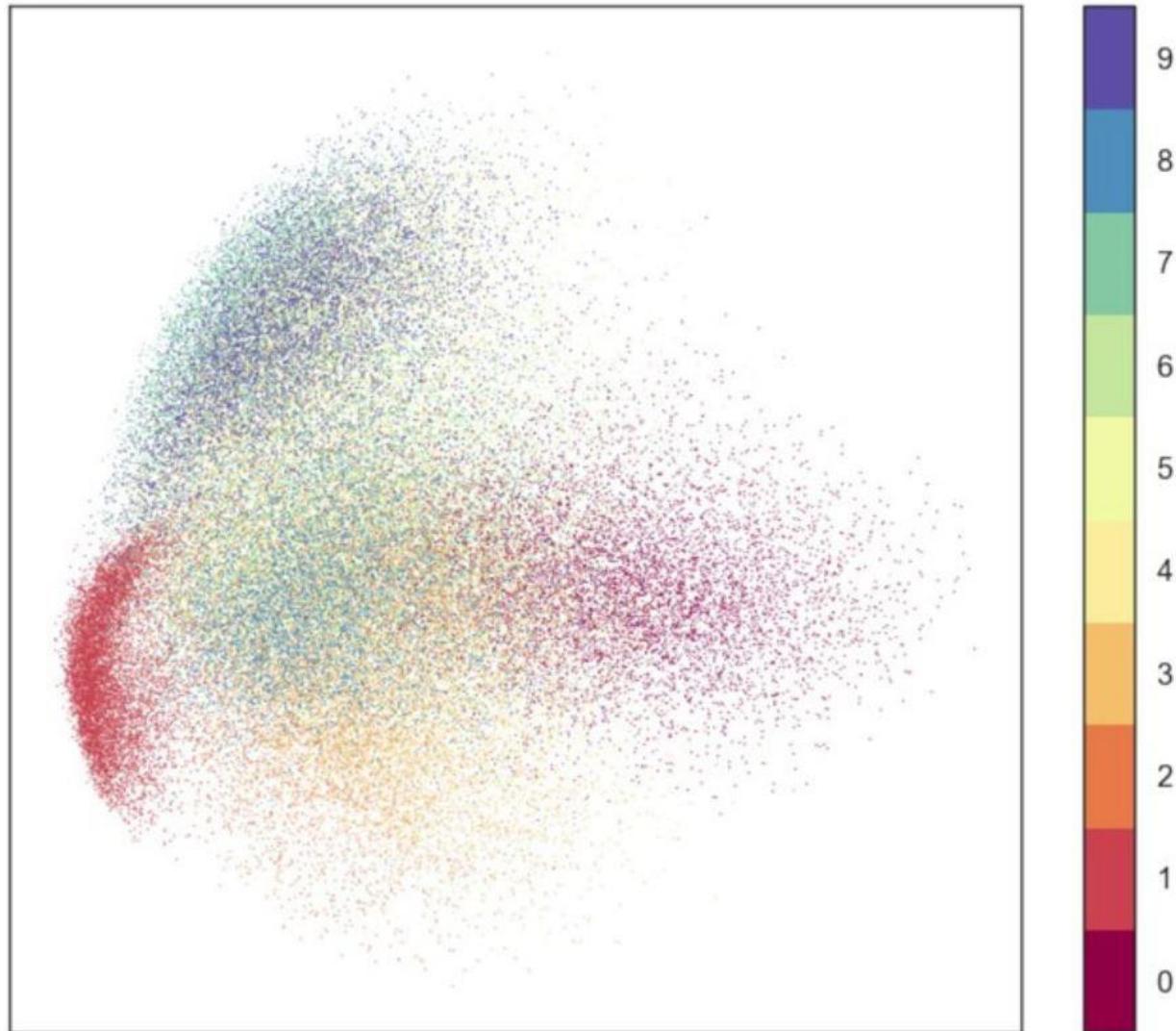
<https://prateekvjoshi.com/2014/06/21/what-is-manifold-learning/>

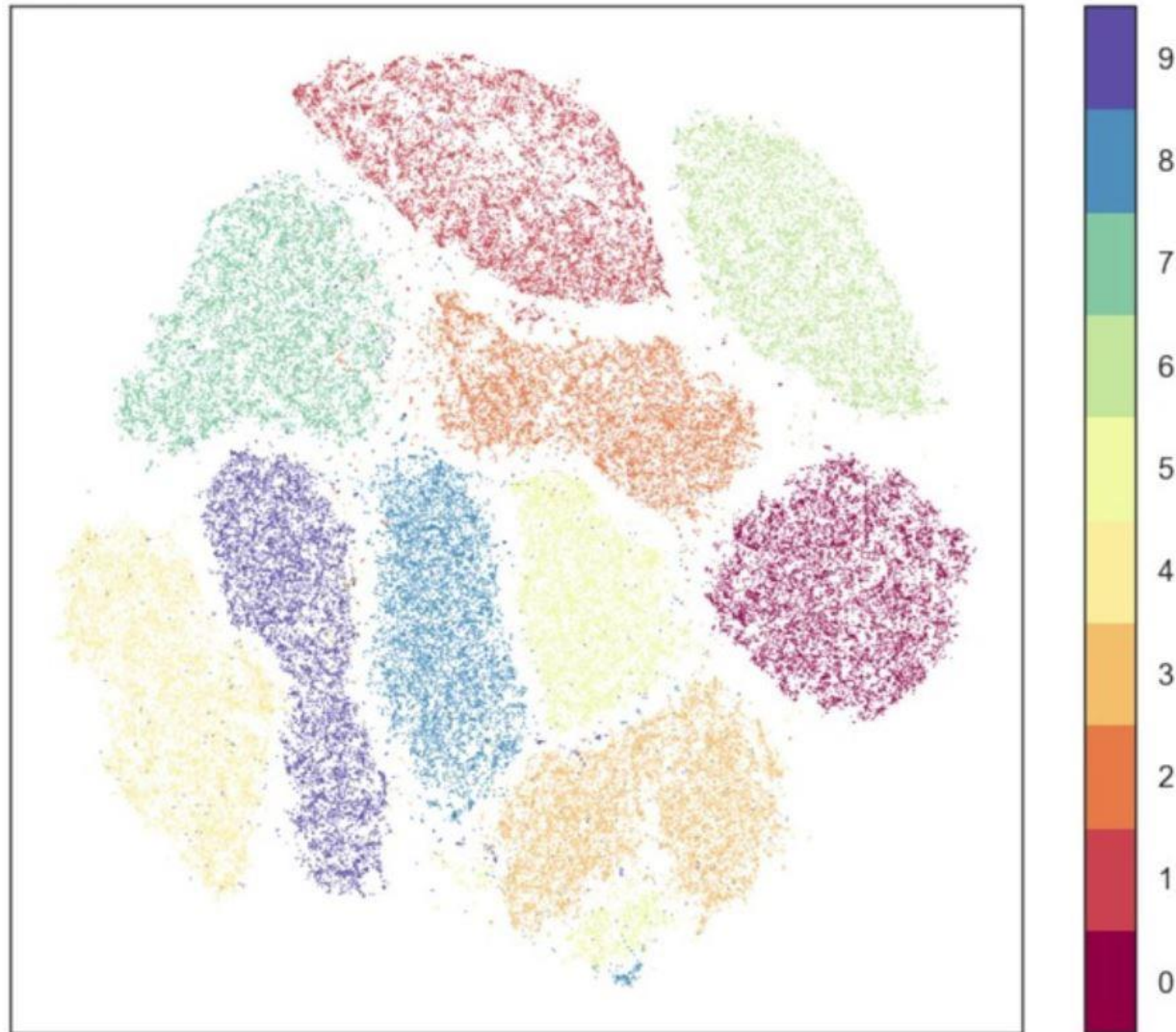


- **UMAP Assumption:** Data is uniformly distributed on a manifold
  - Possibly with non-uniform metric!
  - But the metric can be modelled as locally constant.
- 1. Learn the underlying topological structure
- 2. Iteratively solve an optimization of projections that respect this underlying structure



# PCA (linear; SVD-based) on MNIST





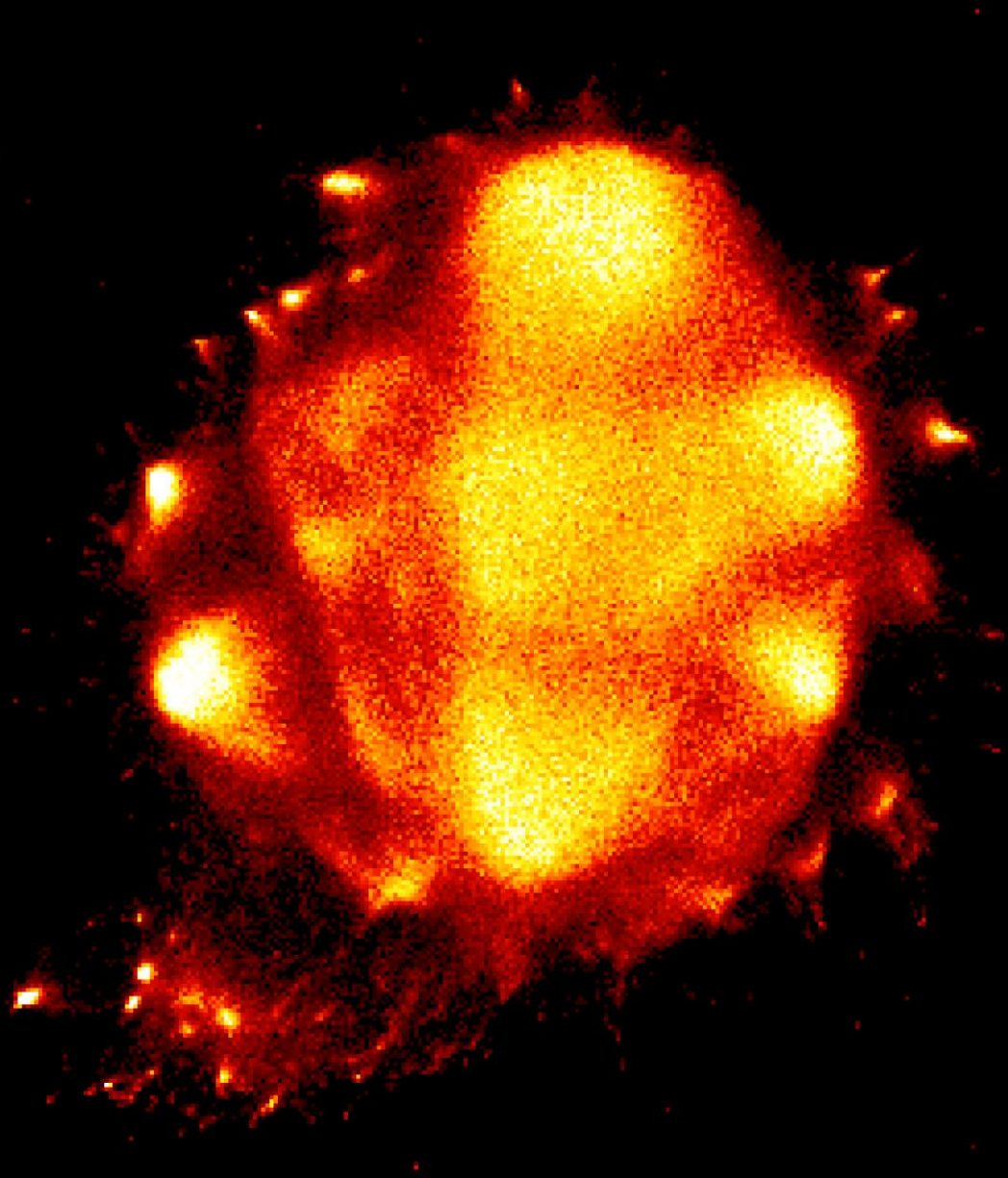


- Read the Docs: [How UMAP works](#)
- A really great Google PAIR article (/w interactive animations) <https://pair-code.github.io/understanding-umap>
- Leland McInnes YouTube talks (just a sample):
  - Topological Techniques for Unsupervised Learning, PyData LA 2019 talk <https://youtu.be/7pAVPjwBppo>
  - UMAP at SciPy 2018 <https://youtu.be/nq6iPZVUxZU>
  - PyData Ann Arbor: Modern Approaches to Dimension Reduction <https://youtu.be/YPJQydzTLwQ>

# Embeddings

An embedding is  
a numeric representation of  
your data  
along with a  
distance





# I need embeddings for

Clustering

Outlier Detection

Grouping

Anomaly Detection

## Unsupervised Learning

Exploratory Data Analysis

Visualization

# Document Embedding (DocMAP)

We'd like to:

Embed documents

Cluster documents

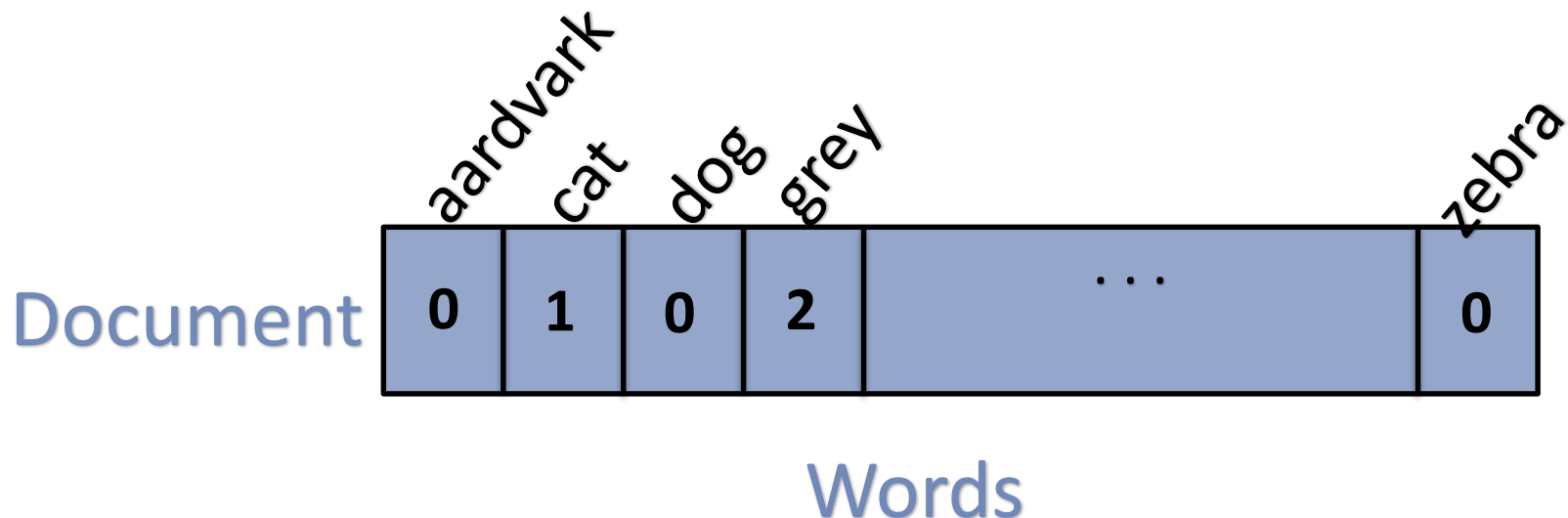
Find similar documents

Find strange or outlier documents

An embedding is  
a numeric representation of  
your data  
along with a  
distance

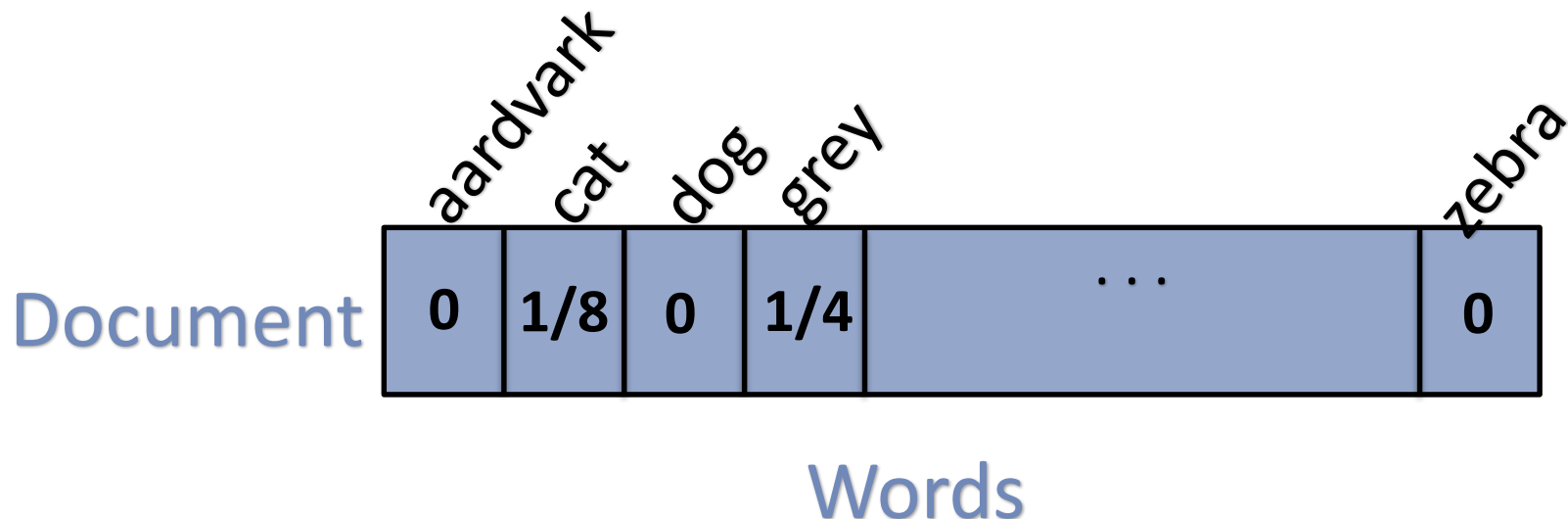
Document is a bag of words

“The grey cat sat on the grey rug”



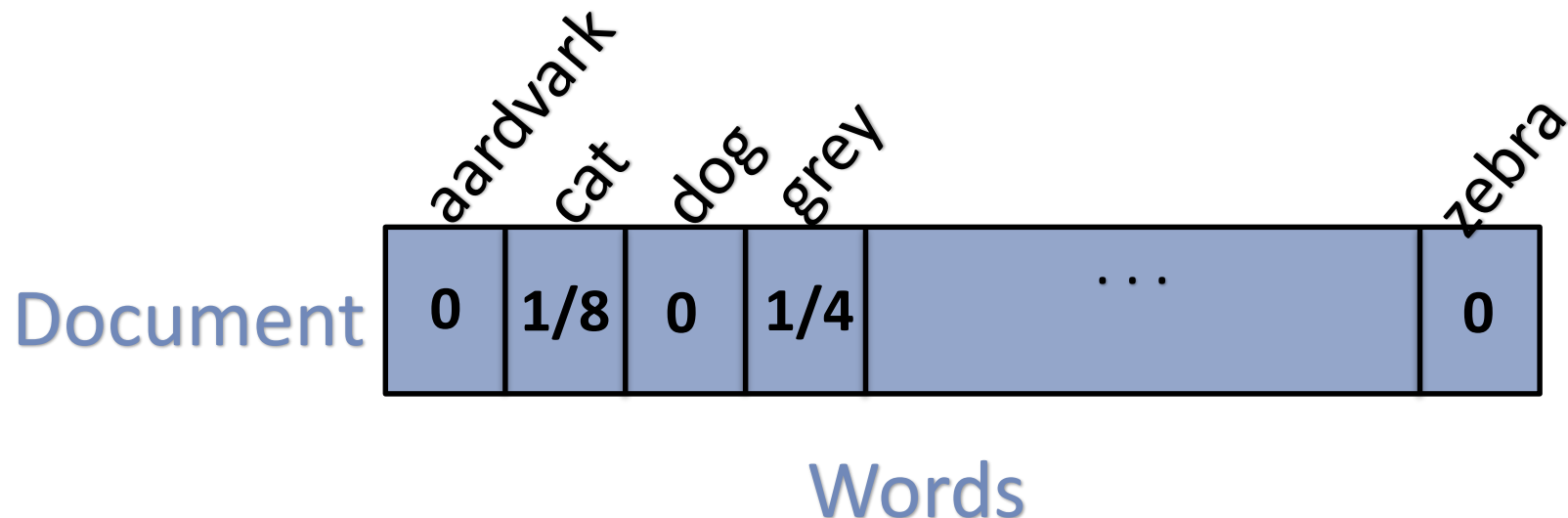
Document is a bag of ~~words~~  
probabilities

“The grey cat sat on the grey rug”



Document is a multinomial distribution  
across our vocabulary space

“The grey cat sat on the grey rug”





A corpus is a document by word matrix

Documents

	aardvark	cat	dog	grey		zebra
					⋮	
	0	$\frac{1}{8}$	0	$\frac{1}{4}$	...	0
	$\frac{1}{3}$	0	0	$\frac{1}{6}$	...	0
					⋮	

Vocabulary or words

\*We also remove expectation to mitigate the effect of Zipf's law on the column distribution.

An embedding is  
a numeric representation of  
your data  
along with a  
distance

# Theoretical Statistics to the rescue!

## Fisher Information Metric:

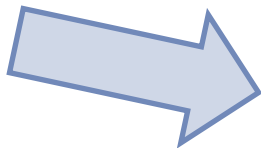
$$(\Delta_{n-1}, f) \rightarrow (S^n, \mu)$$

$$x_i \mapsto \sqrt{x_i}$$

$$d_a(w_a, w_b) = \arccos\left(\frac{\sum_{i=1}^n \sqrt{a_i b_i}}{\sqrt{|a|_1 |b|_1}}\right)$$

$$\arccos(\theta) \approx \sqrt{1 - \theta},$$

**Hellinger  
distance**



$$d_H(w_a, w_b) = \sqrt{1 - \frac{\sum \sqrt{a_i b_i}}{\sqrt{|a|_1 |b|_1}}}$$

Carter et al 2009

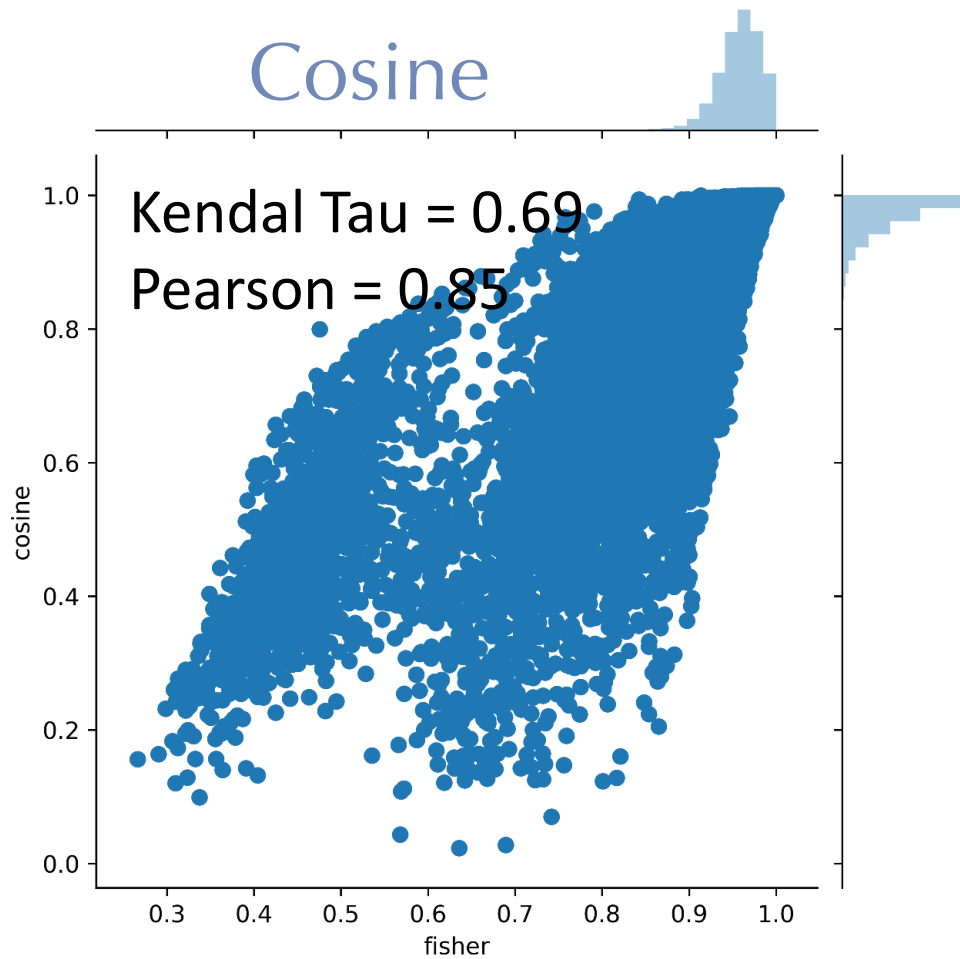
Amari, 2012

## Relationship with cosine distance

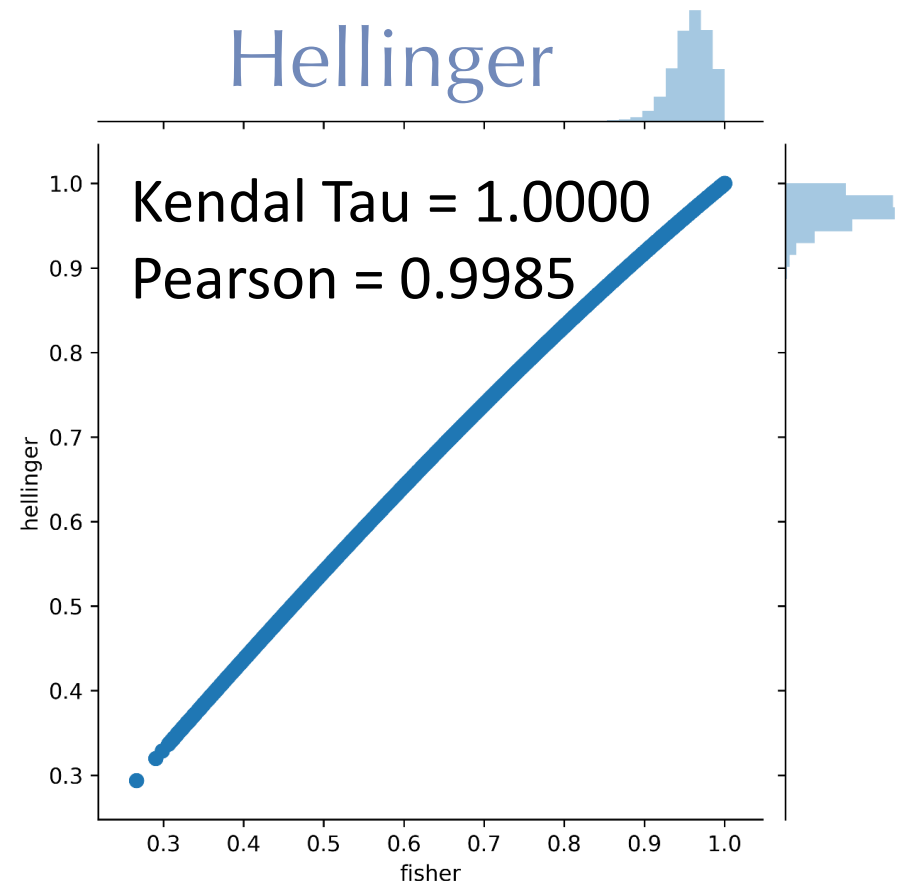
$$d_H(w_a, w_b) = \sqrt{1 - \frac{\sum \sqrt{a_i b_i}}{\sqrt{|a|_1 |b|_1}}}$$

$$d_{cos}(w_a, w_b) = 1 - \frac{\sum a_i b_i}{\sqrt{|a|_2 |b|_2}}$$

## Cosine

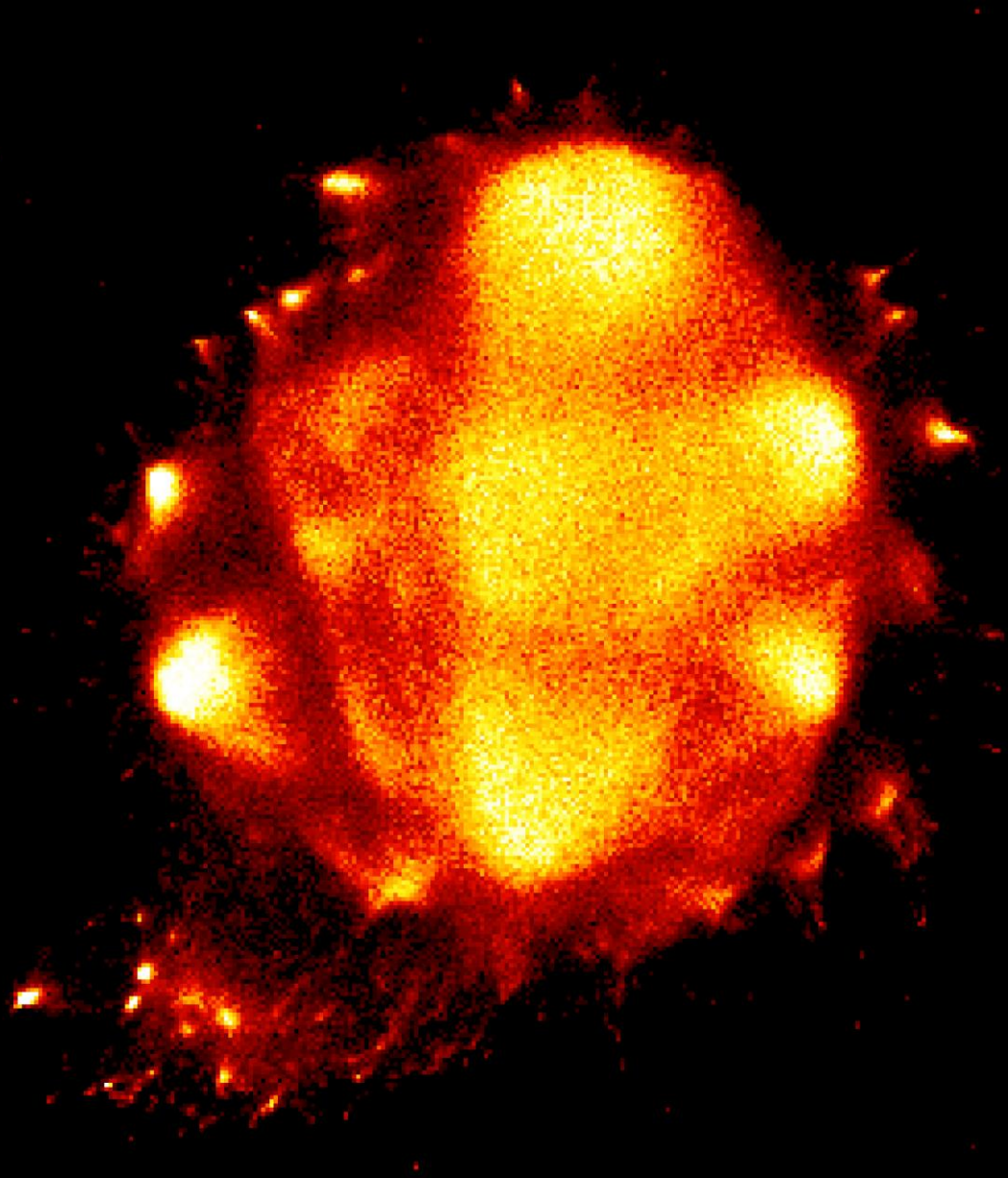


## Hellinger



An embedding is  
a numeric representation of  
your data  
along with a  
distance

# Documents are a bag of words



## We can:

- Embed documents into low dimensions and visualize
- Cluster those embeddings (e.g., with HDBSCAN)
- Find similar documents via nearest neighbor searches
- Find strange or outlier documents via anomaly detection



# Word Embedding (WordMAP)

We'd like to:

Understand a corpus

Embed short documents

Identify documents that are close to words

“You shall know a word by the company it keeps”

John Rupert Firth, 1957

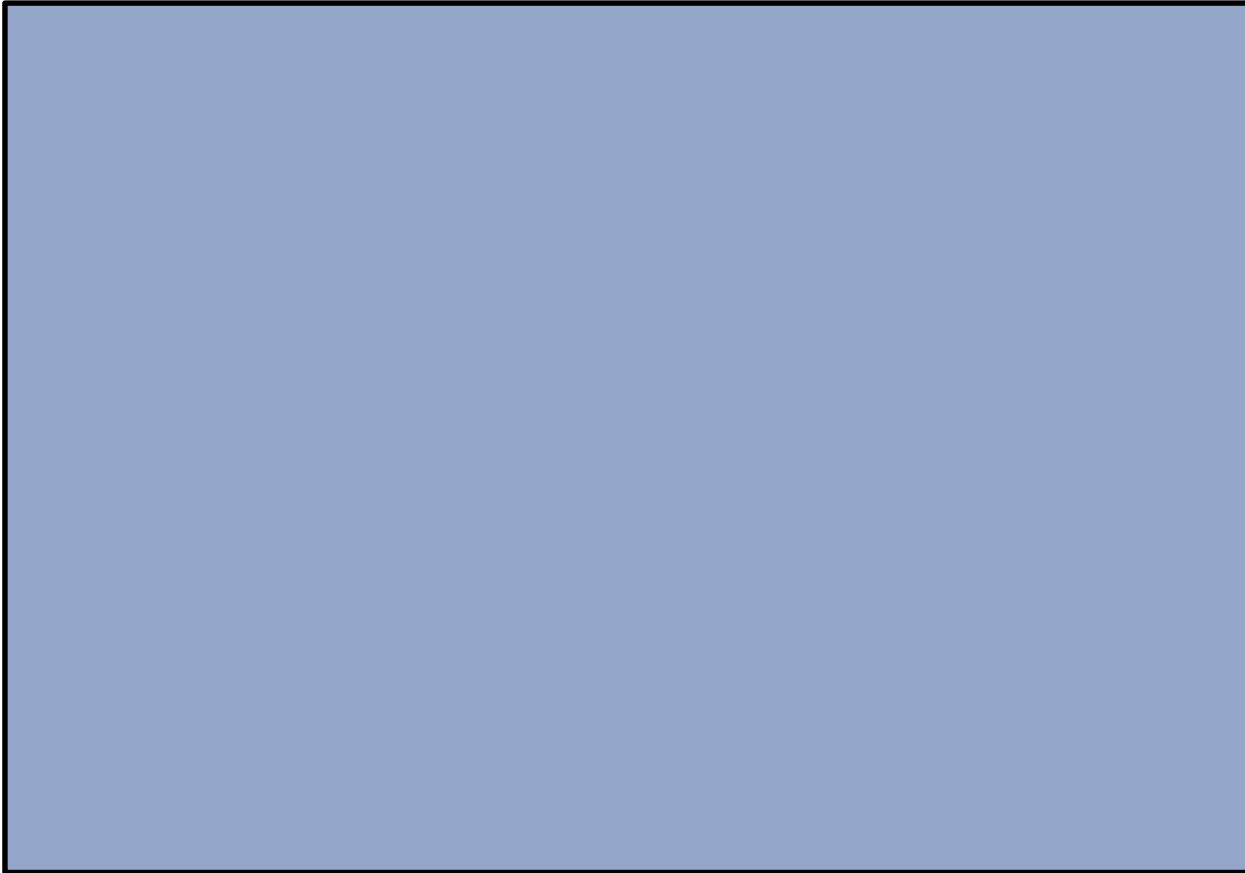
(a famous linguist)

A word is a document of all \_\_\_\_\_ containing it

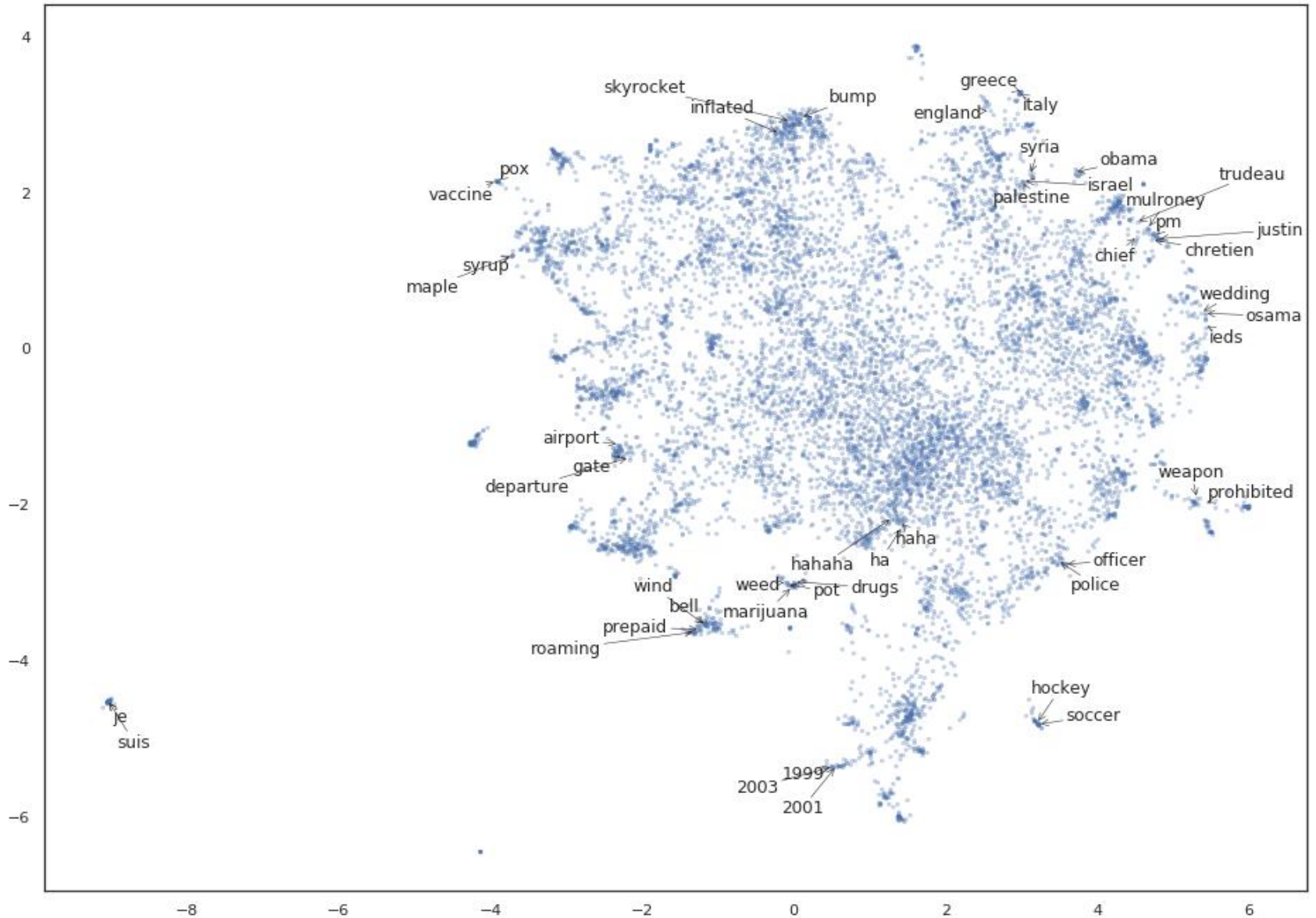
- sentences
- contexts
- windows

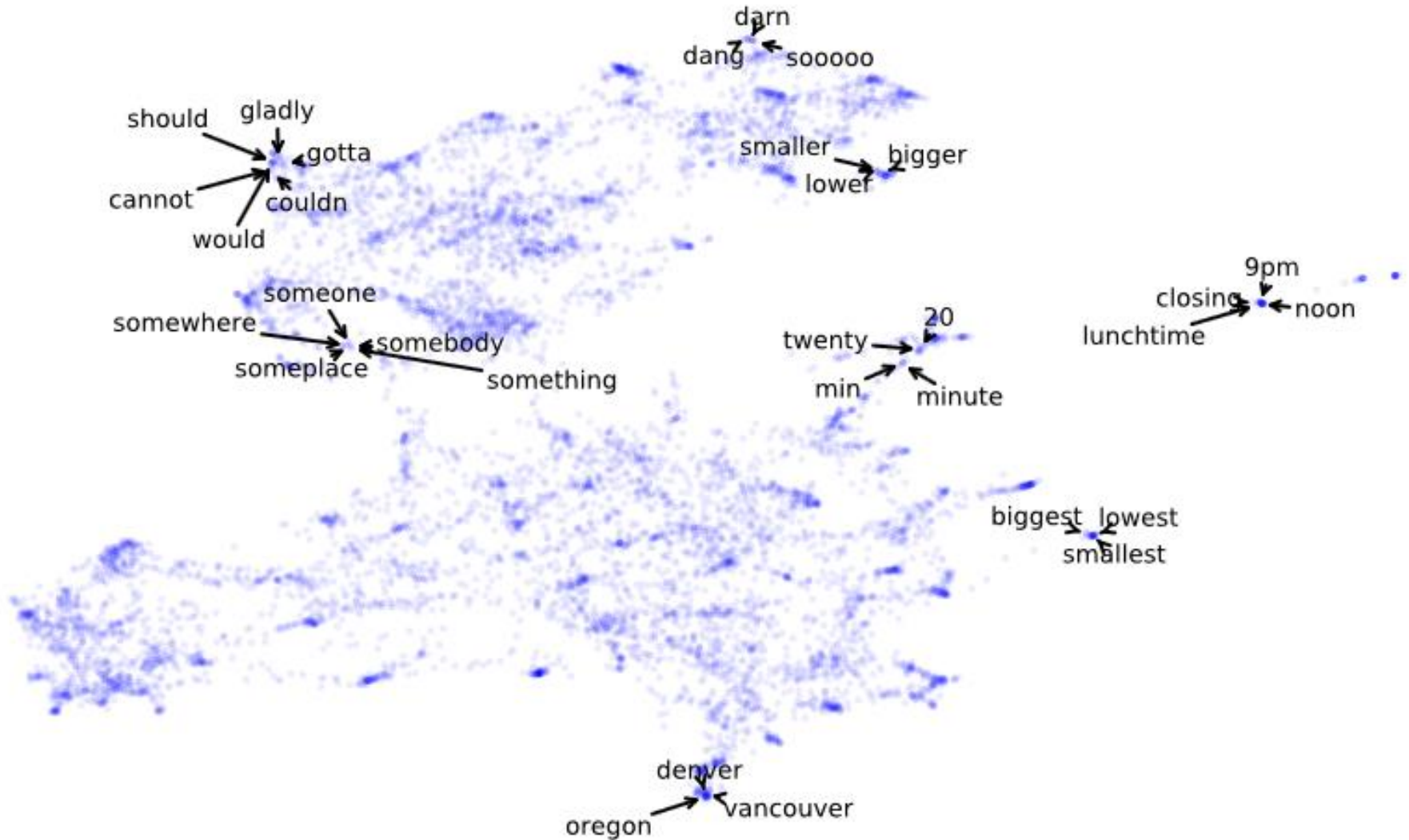
Word usage can be represented by a document by word matrix

Word “documents”



Vocabulary or words





## We can:

Understand a corpus through interactive visualization

Embed short documents via joint embeddings

Perform query expansion by via nearest neighbor search



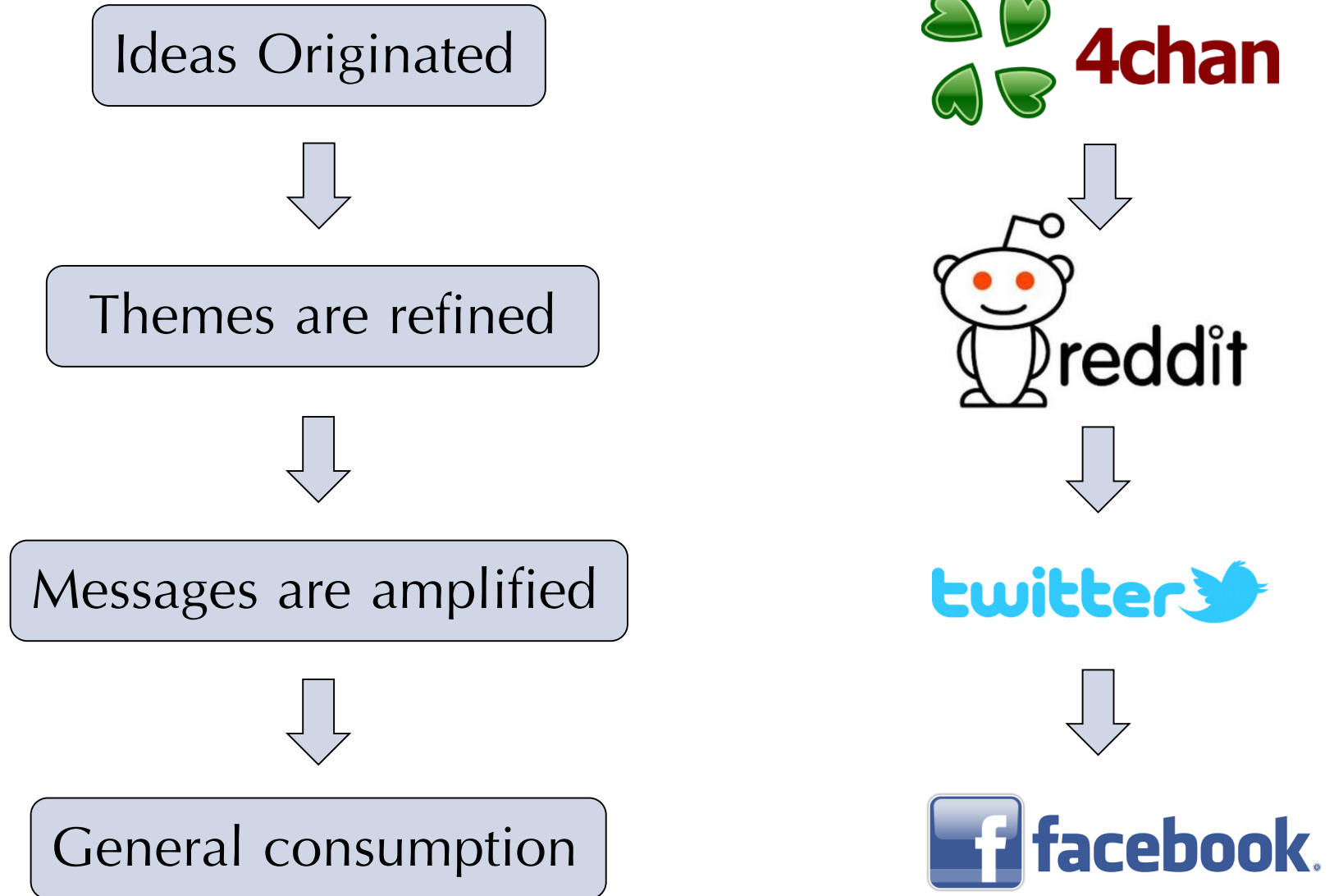
\_\_\_\_\_ is a bag of \_\_\_\_\_

A document is a bag of words

A word is a bag of co-occurring words

We're limited only by our creativity

# Detecting Influence and Effects



## ■ Reddit:

- “The Front Page of the Internet”
- Social news aggregation, web content rating, discussion board
- 9% of online Canadian adults have a Reddit account
- 5<sup>th</sup> most popular site in Canada (Google.com, Youtube.com, Facebook.com, Google.ca, Reddit.com)

# Topic Modeling

## (Top2Vec)

We'd like to:

Characterize a document by a short list of the  
topics contained within

Find documents that discuss similar topics

A **topic** is a set of words along with importance weights

co2, temperature, climate, warming → **global warming**

guns, firearms, owners, restricted, rcmp → **gun control**

## Finding topic words:

**Step 1:** Embed documents

**Step 2:** Find dense areas (clusters) of documents

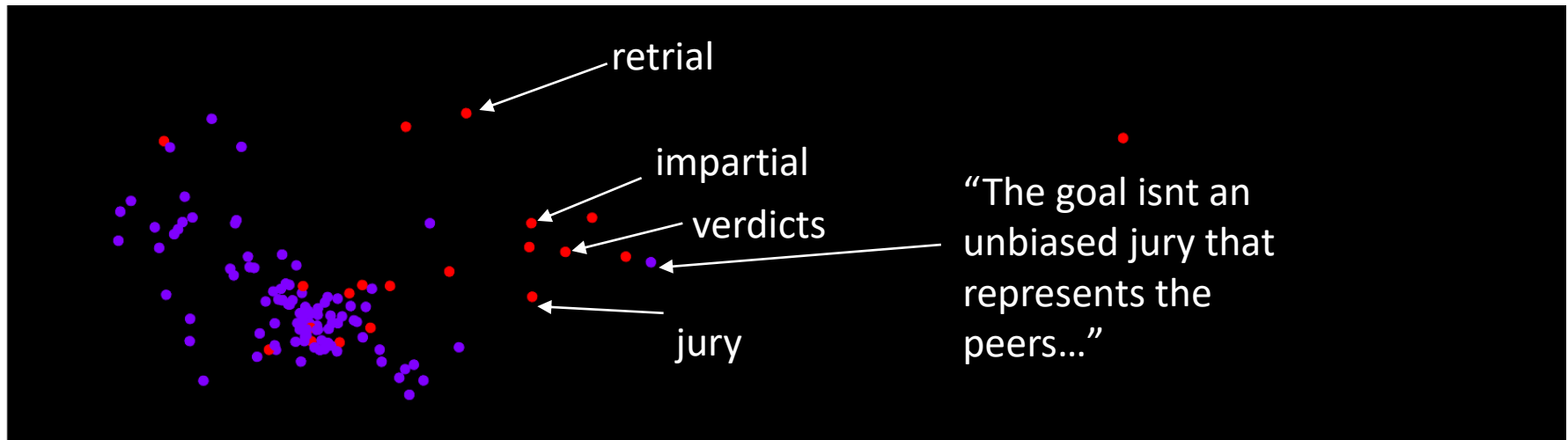
**Step 3:** Find topic vector within dense areas

**Step 4:** Use topic vector to find topic words



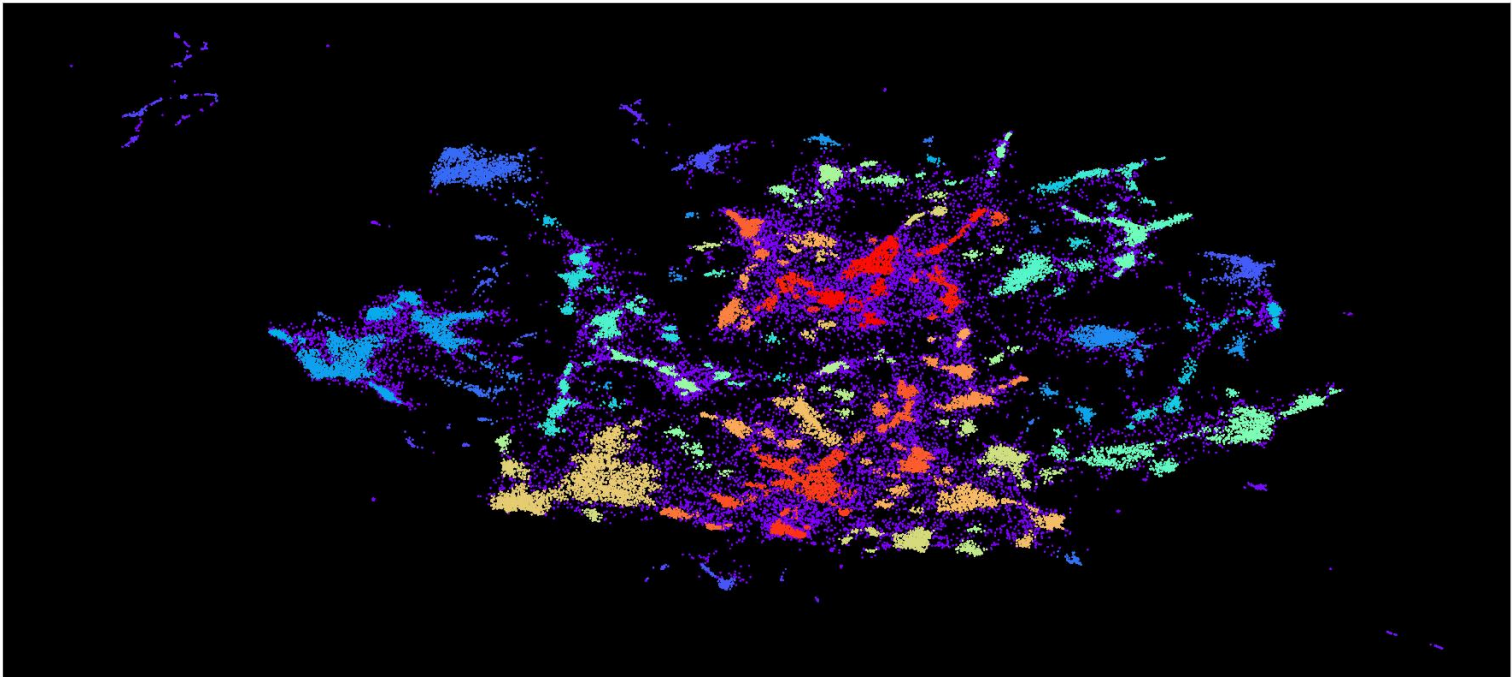
## Step 1: Use DocMAP/WordMAP

- Learn word embedding where similar words are close together
- Embed documents in word vector space, placing them close to words that most describe document



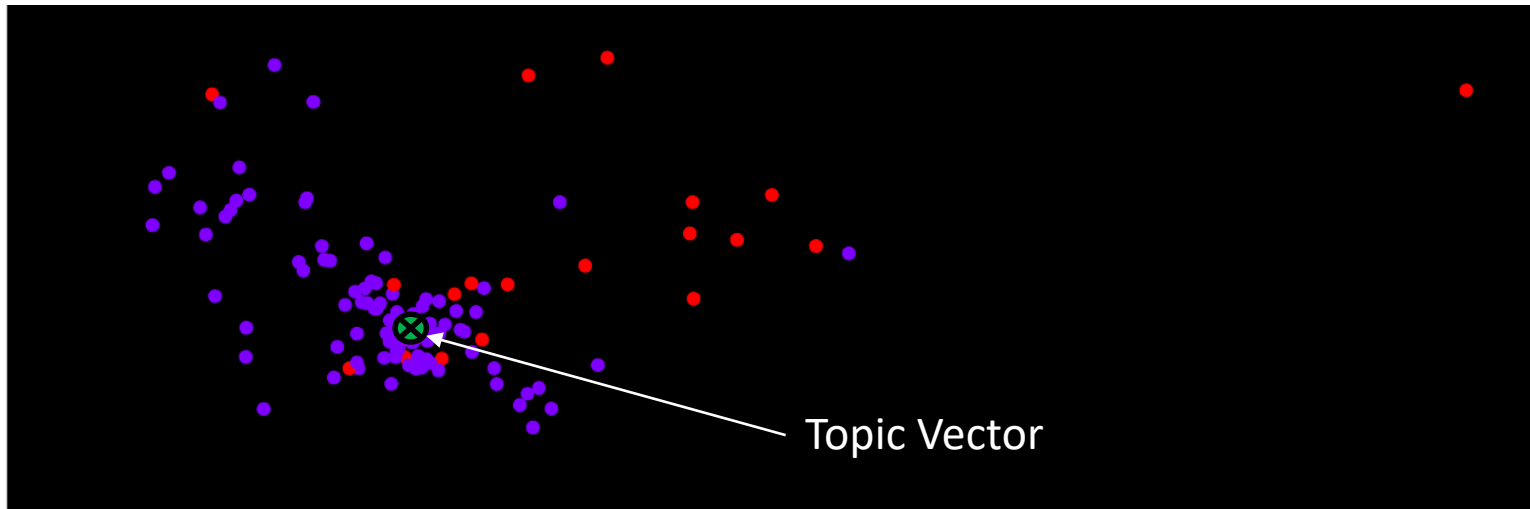
## Step 2: Find dense areas of documents

- **Assumption:** dense areas of documents represent common topic
- Use UMAP to project document vectors to lower dimension
- Use HDBSCAN to find dense clusters



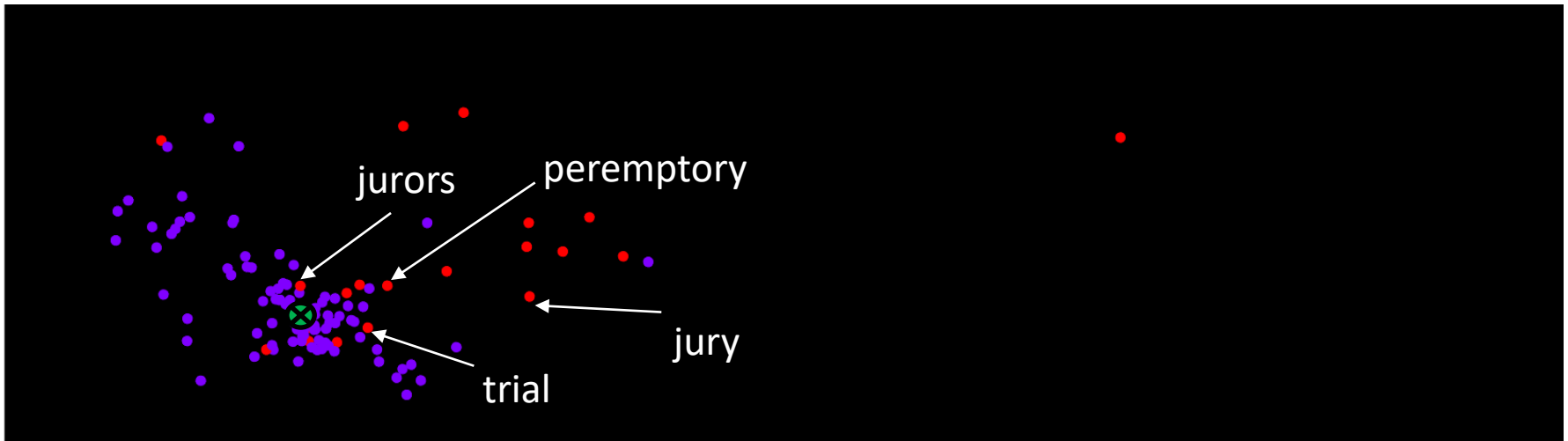
## Step 3: Use dense areas to find topic vector

- Find centroid of documents belonging to dense cluster in original space
- Centroid = Topic Vector



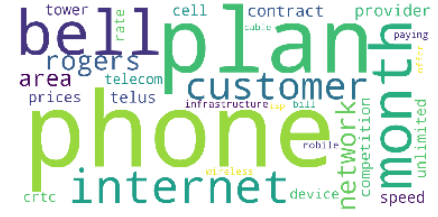
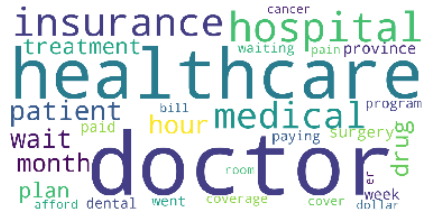
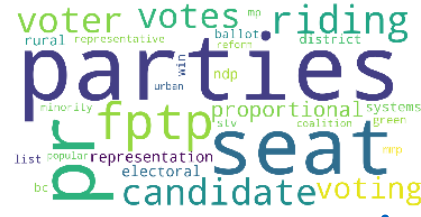
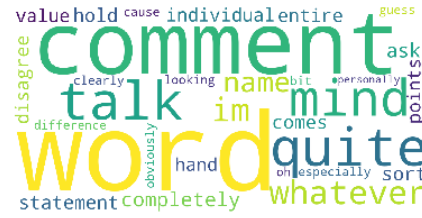
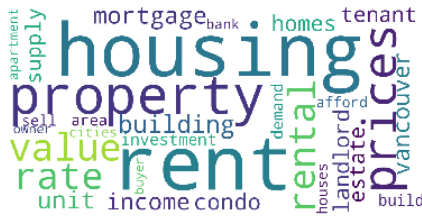
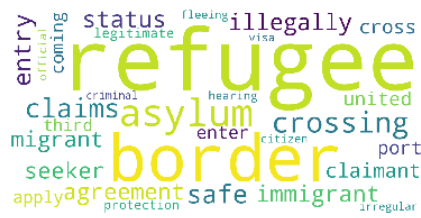
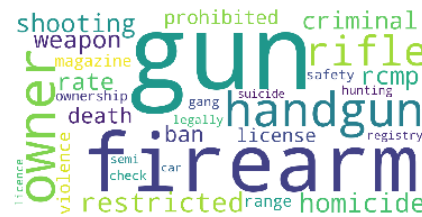
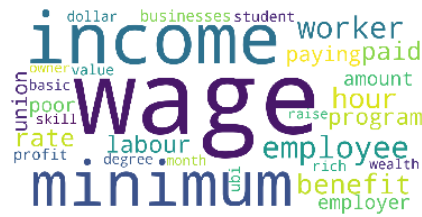
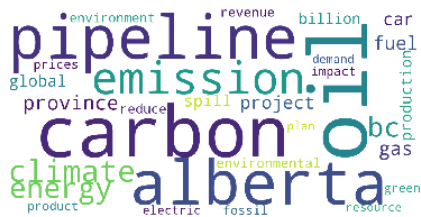
## Step 4: Use topic vector to find topic words

- **Assumption:** The closest words to the centroid will best represent our documents
- Topic =  $k$ -closest word vectors to topic vector



**E.g.:** jurors, jury, juries, peremptory, juror, verdict, trial

## Reddit politics 2018 summarization



Changes in time may indicate  
what bots are talking about

- No need to select number of topics in advance
- No need for stop words
- Jointly embeds documents and words
- Works with short text

- A user is:
  - The set of subreddits they post in, or
  - The words they type, or
  - The topics they talk about, or
  - The time of day they post, or
  - The posts they comment on, or...
- Explore the nearest neighbours/clusters of users
  - Doc2vec/top2vec -> hdbscan -> interact with output

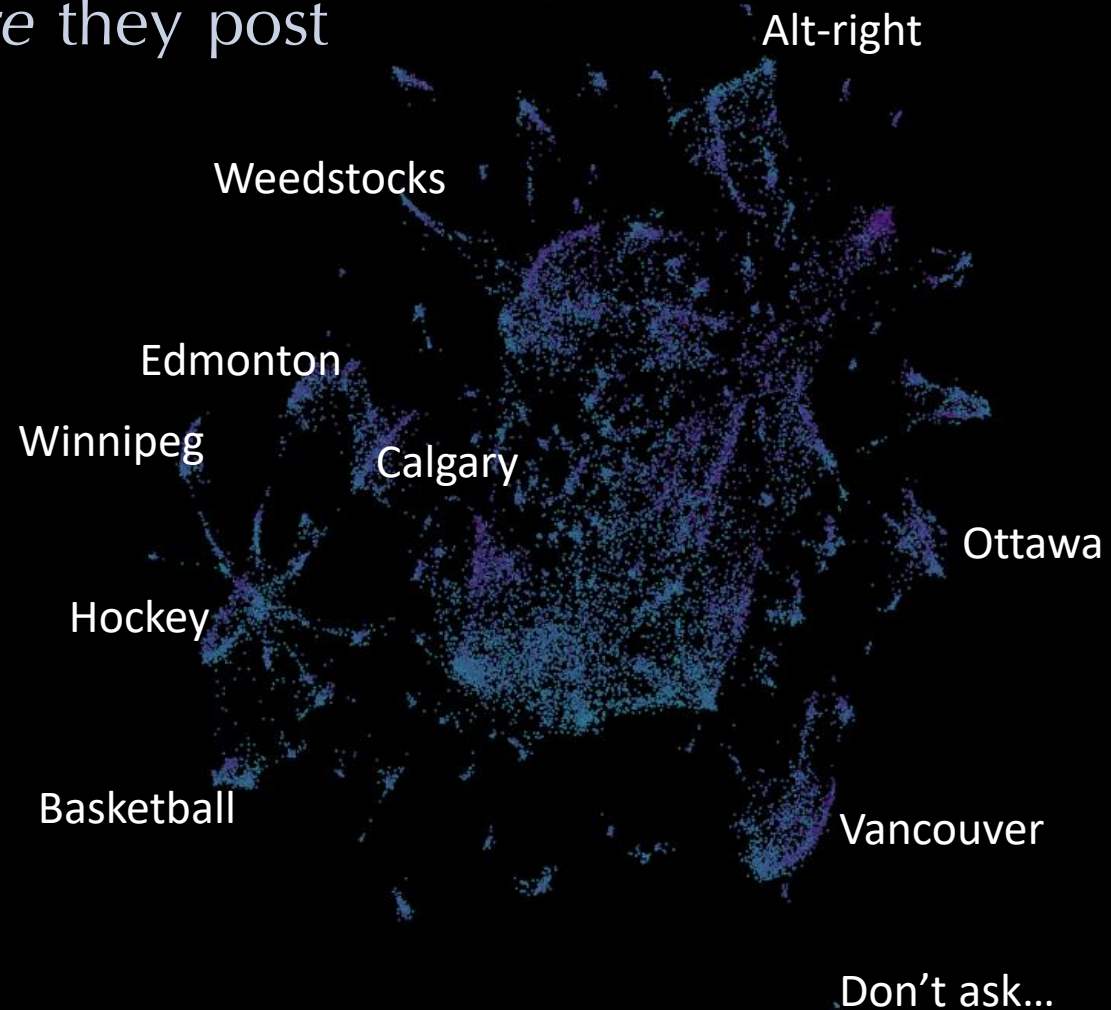
Authors are *the words* they use



UMAP: n\_neighbors=15, min\_dist=0.1

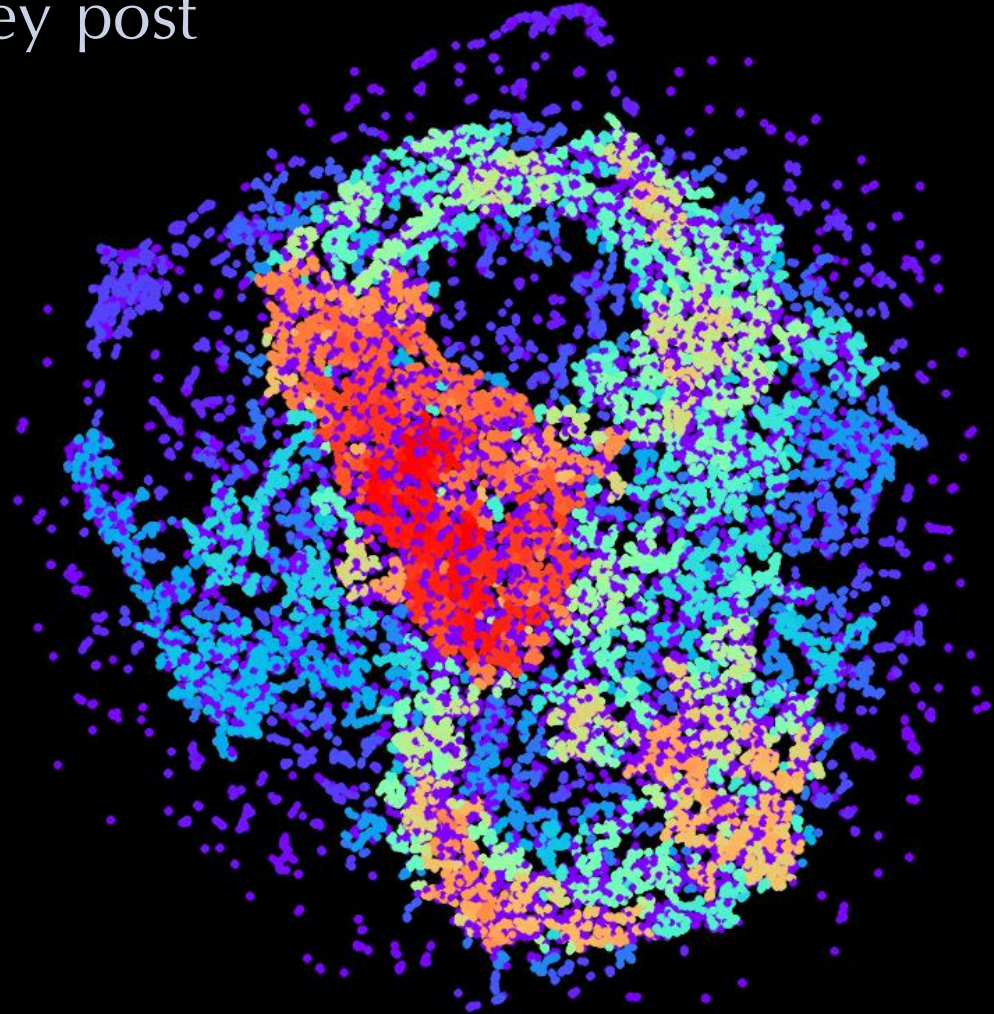


Authors are *where* they post



UMAP: n\_neighbors=15, min\_dist=0.1

Authors are *when* they post



\_\_\_\_\_ is a bag of \_\_\_\_\_

A document is a bag of words

A word is a bag of co-occurring words

A subreddit is a bag of users

A user is a bag of post statistics

Software is a bag of libraries it loads

## Now we have:

- A general technique for embedding “all the things”
  - See also: <https://github.com/jc-healy/EmbedAllTheThings>
- A series of techniques for summarizing authors, reddit forums, corpora of documents.
- Experience working with our partners to leverage these techniques to **empower analysts** to search for malicious foreign influence campaigns.

BEAT NAVY