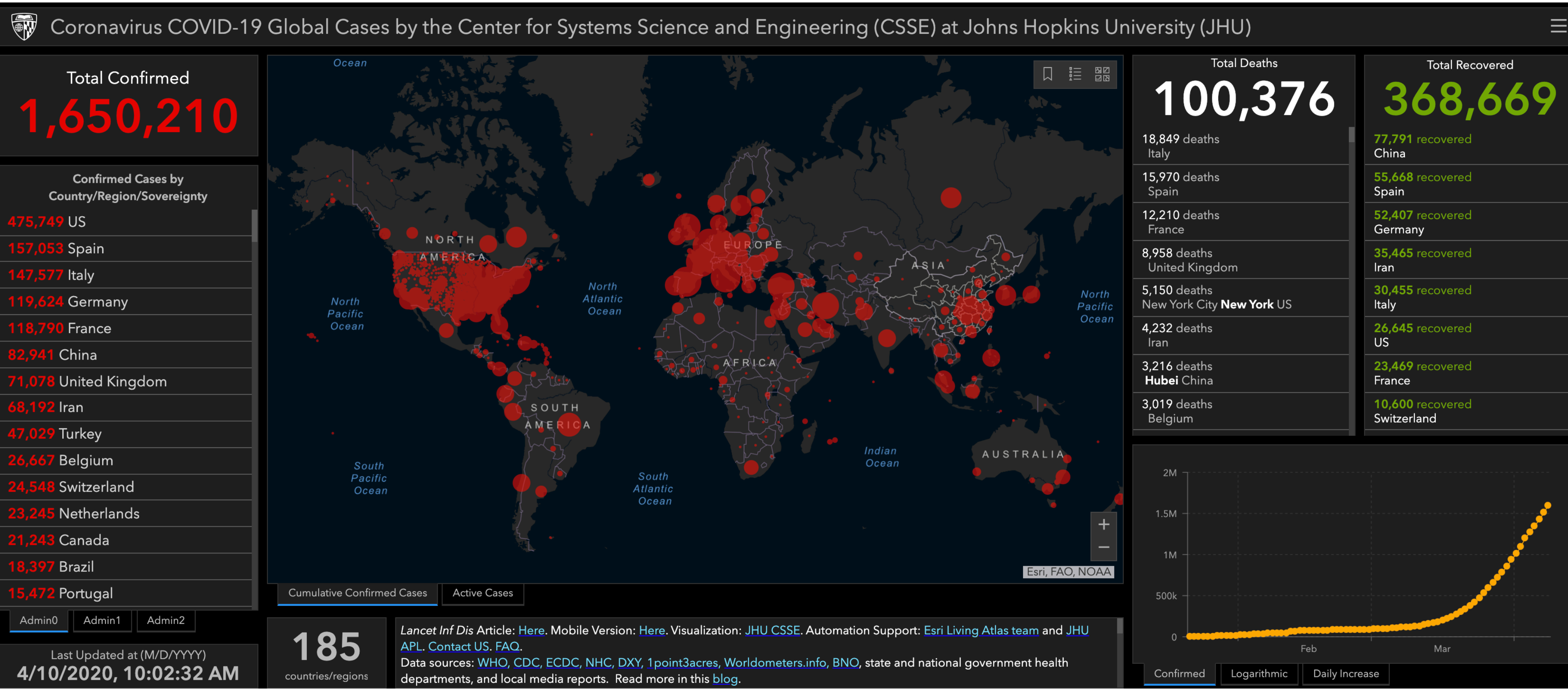
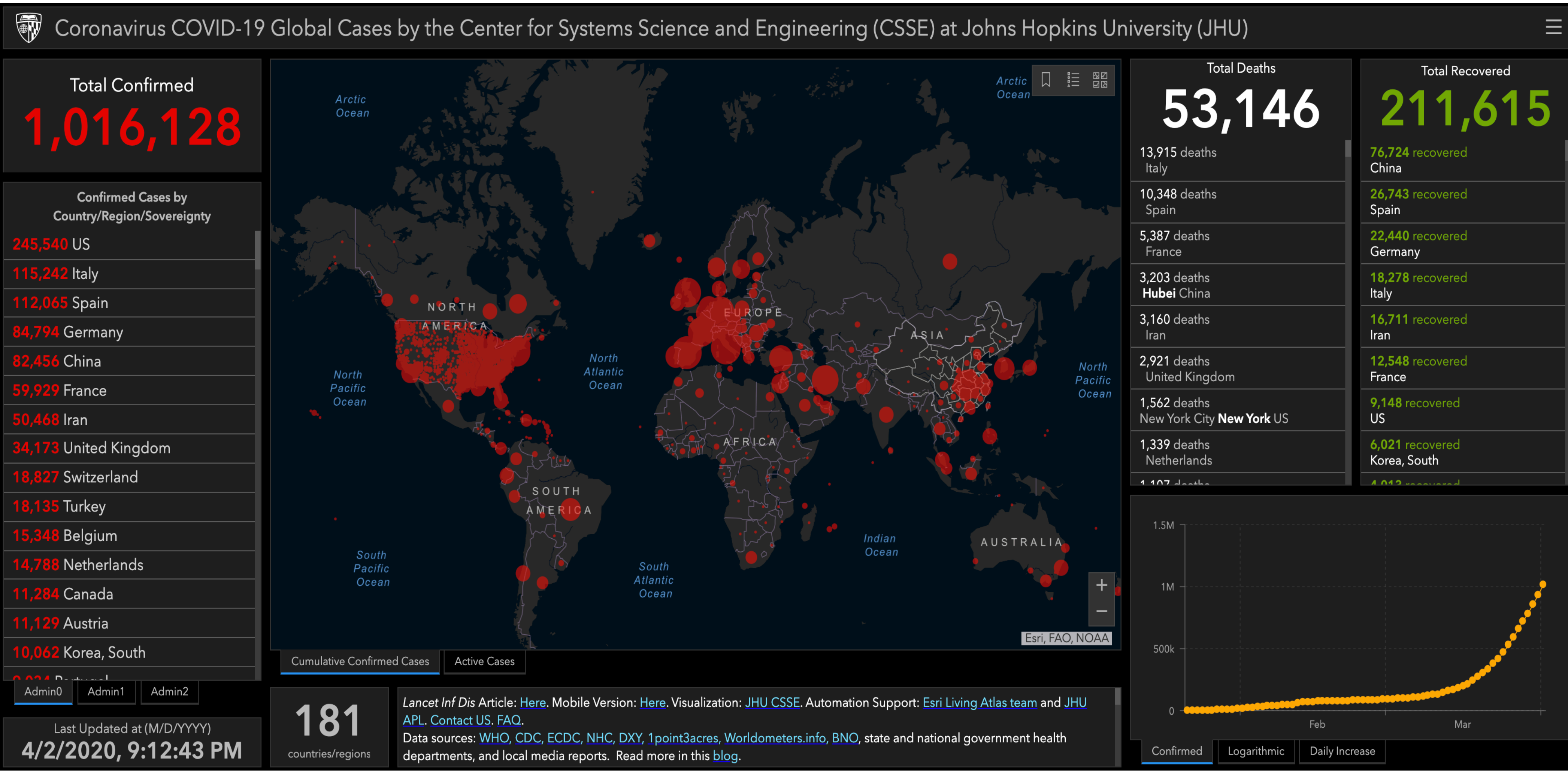
Math 189Z

Lecture 2: NLP Overview

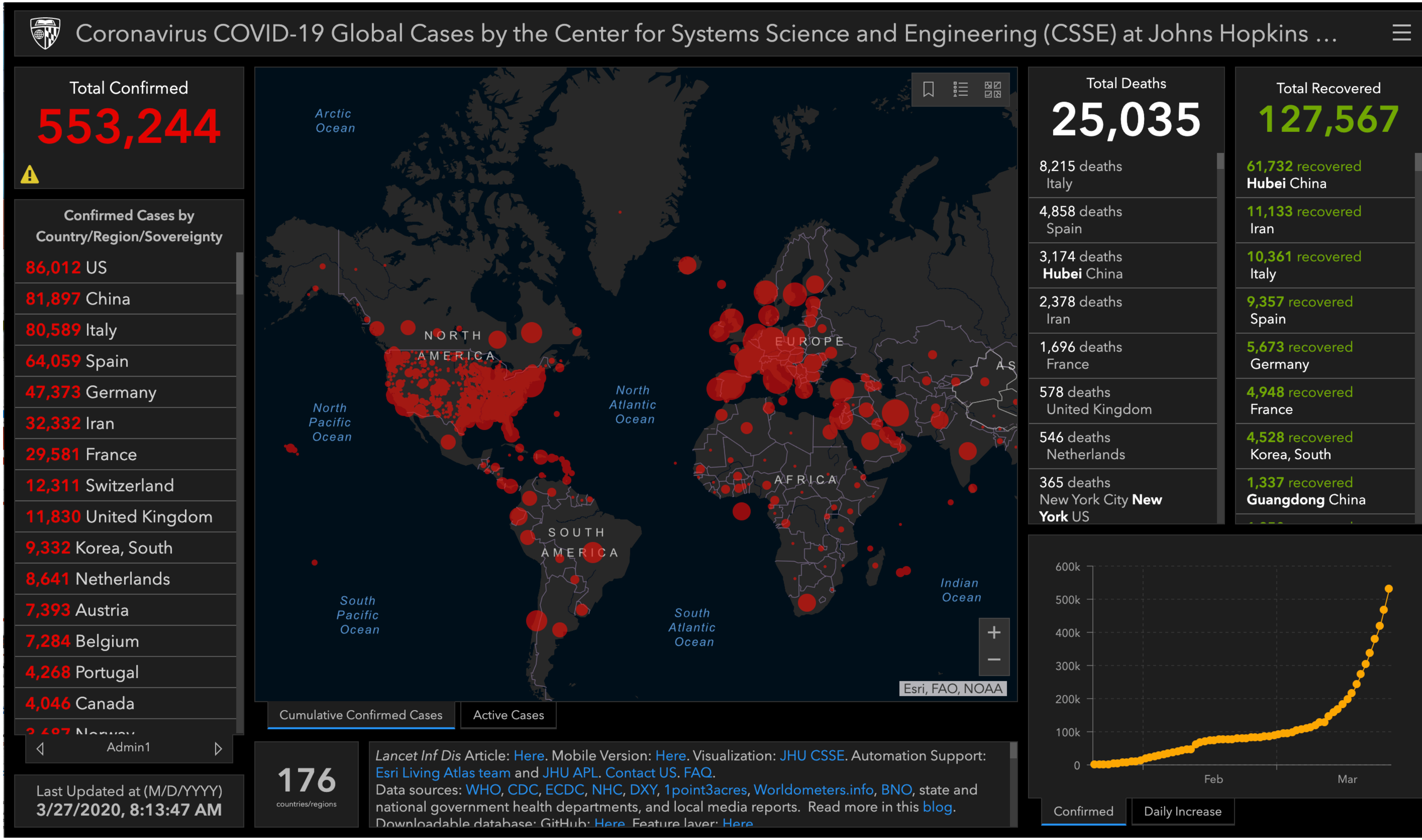
Topic Models: NMF & LSA

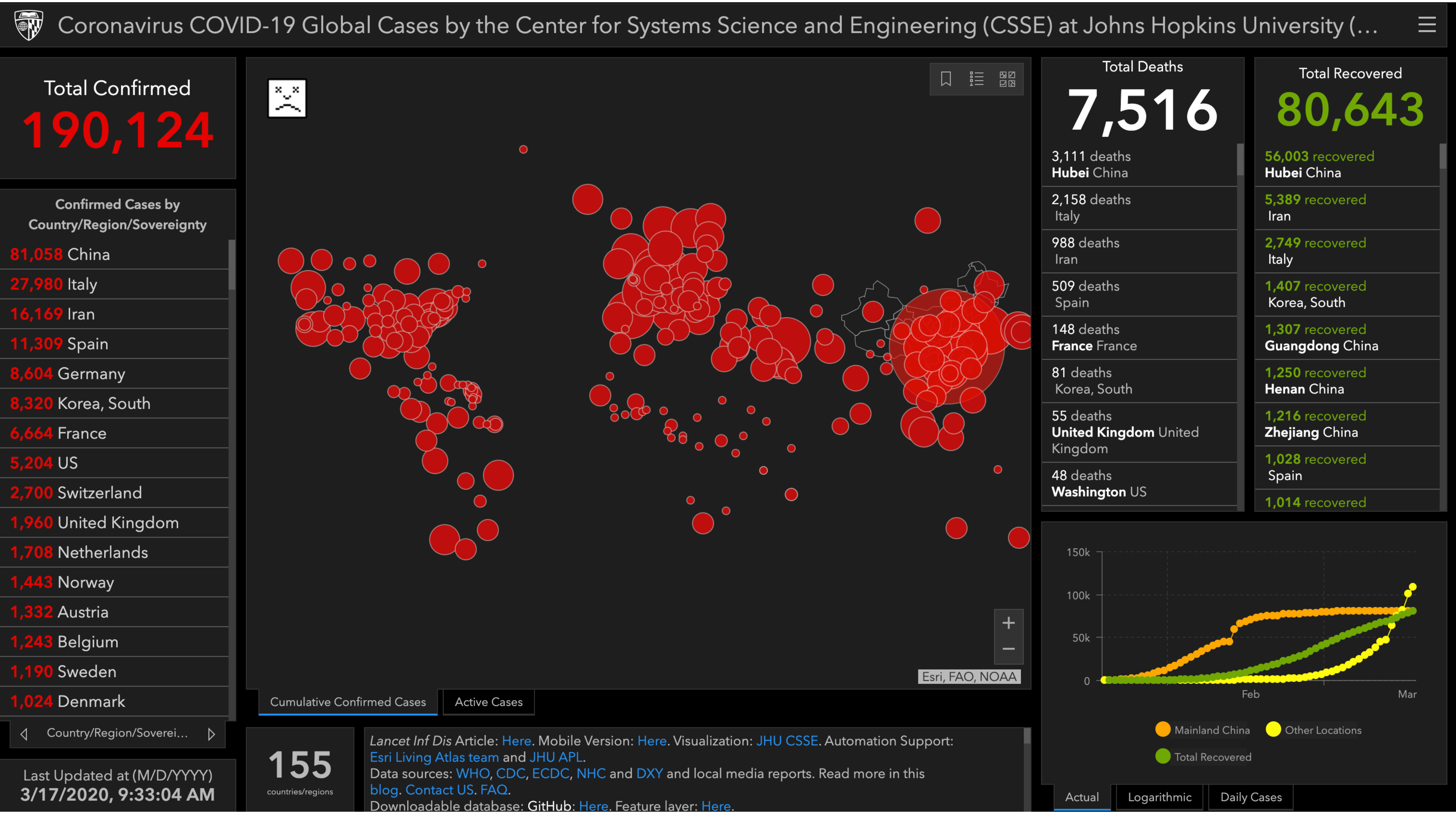
 https://math189covid19.github.io/


Coronavirus COVID-19 Global Cases by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University (JHU)
Total Confirmed
1,650,210
Confirmed Cases by Country/Region/Sovereignty
475,749 US
157,053 Spain
147,577 Italy
119,624 Germany
118,790 France
82,941 China
71,078 United Kingdom
68,192 Iran
47,029 Turkey
26,667 Belgium
24,548 Switzerland
23,245 Netherlands
21,243 Canada
18,397 Brazil
15,472 Portugal
Admin0
Admin1
Admin2
Last Updated at (M/D/YYYY)
4/10/2020, 10:02:32 AM
Ocean
NORTH AMERICA
EUROPE
ASIA
AFRICA
SOUTH AMERICA
AUSTRALIA
North Pacific Ocean
North Atlantic Ocean
North Pacific Ocean
South Pacific Ocean
South Atlantic Ocean
Indian Ocean
Esri, FAO, NOAA
Cumulative Confirmed Cases
Active Cases
185
countries/regions
Lancet Inf Dis Article: Here. Mobile Version: Here. Visualization: JHU CSSE. Automation Support: Esri Living Atlas team and JHU APL. Contact US. FAQ.
Data sources: WHO, CDC, ECDC, NHC, DXY, 1point3acres, Worldometers.info, BNO, state and national government health departments, and local media reports. Read more in this blog.
Total Deaths
100,376
18,849 deaths Italy
15,970 deaths Spain
12,210 deaths France
8,958 deaths United Kingdom
5,150 deaths New York City New York US
4,232 deaths Iran
3,216 deaths Hubei China
3,019 deaths Belgium
Total Recovered
368,669
77,791 recovered China
55,668 recovered Spain
52,407 recovered Germany
35,465 recovered Iran
30,455 recovered Italy
26,645 recovered US
23,469 recovered France
10,600 recovered Switzerland
2M
1.5M
1M
500k
0
Feb
Mar
Confirmed
Logarithmic
Daily Increase

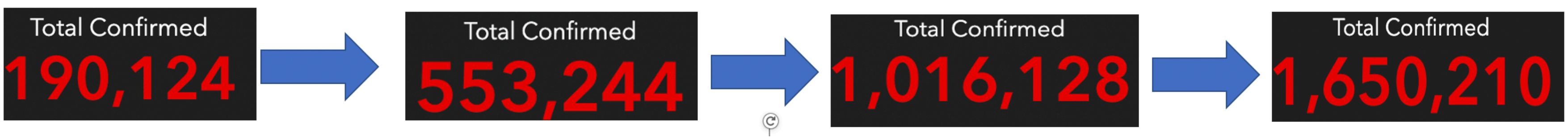


Coronavirus COVID-19 Global Cases by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University (JHU)
Total Confirmed
1,016,128
Confirmed Cases by Country/Region/Sovereignty
245,540 US
115,242 Italy
112,065 Spain
84,794 Germany
82,456 China
59,929 France
50,468 Iran
34,173 United Kingdom
18,827 Switzerland
18,135 Turkey
15,348 Belgium
14,788 Netherlands
11,284 Canada
11,129 Austria
10,062 Korea, South
Admin0
Admin1
Admin2
Last Updated at (M/D/YYYY)
4/2/2020, 9:12:43 PM
Arctic Ocean
NORTH AMERICA
North Pacific Ocean
North Atlantic Ocean
EUROPE
ASIA
AFRICA
SOUTH AMERICA
South Pacific Ocean
South Atlantic Ocean
Indian Ocean
AUSTRALIA
Esri, FAO, NOAA
Cumulative Confirmed Cases
Active Cases
181
countries/regions
Lancet Inf Dis Article: Here. Mobile Version: Here. Visualization: JHU CSSE. Automation Support: Esri Living Atlas team and JHU APL. Contact US. FAQ.
Data sources: WHO, CDC, ECDC, NHC, DXY, 1point3acres, Worldometers.info, BNO, state and national government health departments, and local media reports. Read more in this blog.
Total Deaths
53,146
13,915 deaths Italy
10,348 deaths Spain
5,387 deaths France
3,203 deaths Hubei China
3,160 deaths Iran
2,921 deaths United Kingdom
1,562 deaths New York City New York US
1,339 deaths Netherlands
Total Recovered
211,615
76,724 recovered China
26,743 recovered Spain
22,440 recovered Germany
18,278 recovered Italy
16,711 recovered Iran
12,548 recovered France
9,148 recovered US
6,021 recovered Korea, South
1.5M
1M
500k
0
Feb
Mar
Confirmed
Logarithmic
Daily Increase


Coronavirus COVID-19 Global Cases by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins …
Total Confirmed
553,244
Confirmed Cases by Country/Region/Sovereignty
86,012 US
81,897 China
80,589 Italy
64,059 Spain
47,373 Germany
32,332 Iran
29,581 France
12,311 Switzerland
11,830 United Kingdom
9,332 Korea, South
8,641 Netherlands
7,393 Austria
7,284 Belgium
4,268 Portugal
4,046 Canada
Admin1
Last Updated at (M/D/YYYY)
3/27/2020, 8:13:47 AM
Arctic Ocean
NORTH AMERICA
North Pacific Ocean
North Atlantic Ocean
EUROPE
AFRICA
SOUTH AMERICA
South Pacific Ocean
South Atlantic Ocean
Indian Ocean
Esri, FAO, NOAA
Cumulative Confirmed Cases
Active Cases
176
countries/regions
Lancet Inf Dis Article: Here. Mobile Version: Here. Visualization: JHU CSSE. Automation Support: Esri Living Atlas team and JHU APL. Contact US. FAQ.
Data sources: WHO, CDC, ECDC, NHC, DXY, 1point3acres, Worldometers.info, BNO, state and national government health departments, and local media reports. Read more in this blog.
Total Deaths
25,035
8,215 deaths Italy
4,858 deaths Spain
3,174 deaths Hubei China
2,378 deaths Iran
1,696 deaths France
578 deaths United Kingdom
546 deaths Netherlands
365 deaths New York City New York US
Total Recovered
127,567
61,732 recovered Hubei China
11,133 recovered Iran
10,361 recovered Italy
9,357 recovered Spain
5,673 recovered Germany
4,948 recovered France
4,528 recovered Korea, South
1,337 recovered Guangdong China
600k
500k
400k
300k
200k
100k
0
Feb
Mar
Confirmed
Daily Increase


Coronavirus COVID-19 Global Cases by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University (…
Total Confirmed
190,124
Confirmed Cases by Country/Region/Sovereignty
81,058 China
27,980 Italy
16,169 Iran
11,309 Spain
8,604 Germany
8,320 Korea, South
6,664 France
5,204 US
2,700 Switzerland
1,960 United Kingdom
1,708 Netherlands
1,443 Norway
1,332 Austria
1,243 Belgium
1,190 Sweden
1,024 Denmark
Country/Region/Sovereи…
Last Updated at (M/D/YYYY)
3/17/2020, 9:33:04 AM
Esri, FAO, NOAA
Cumulative Confirmed Cases
Active Cases
155
countries/regions
Lancet Inf Dis Article: Here. Mobile Version: Here. Visualization: JHU CSSE. Automation Support: Esri Living Atlas team and JHU APL.
Data sources: WHO, CDC, ECDC, NHC and DXY and local media reports. Read more in this blog. Contact US. FAQ.
Downloadable database: GitHub: Here. Feature layer: Here.
Total Deaths
7,516
3,111 deaths Hubei China
2,158 deaths Italy
988 deaths Iran
509 deaths Spain
148 deaths France France
81 deaths Korea, South
55 deaths United Kingdom United Kingdom
48 deaths Washington US
Total Recovered
80,643
56,003 recovered Hubei China
5,389 recovered Iran
2,749 recovered Italy
1,407 recovered Korea, South
1,307 recovered Guangdong China
1,250 recovered Henan China
1,216 recovered Zhejiang China
1,028 recovered Spain
1,014 recovered
150k
100k
50k
0
Feb
Mar
Mainland China
Other Locations
Total Recovered
Actual
Logarithmic
Daily Cases

- COVID-19 confirmed cases have been increased but not doubled since last Monday.



In the case of Italy:

$R = (147-115)/(115-80) = 0.91 < 1$

Overview

- What is NLP? Give an overview
- Today: Focus on Topic Modeling, especially
  1. **NMF (Non-negative Matrix Factorization)**
  2. **LSA (Latent Semantic Analysis)**

- https://math189covid19.github.io/

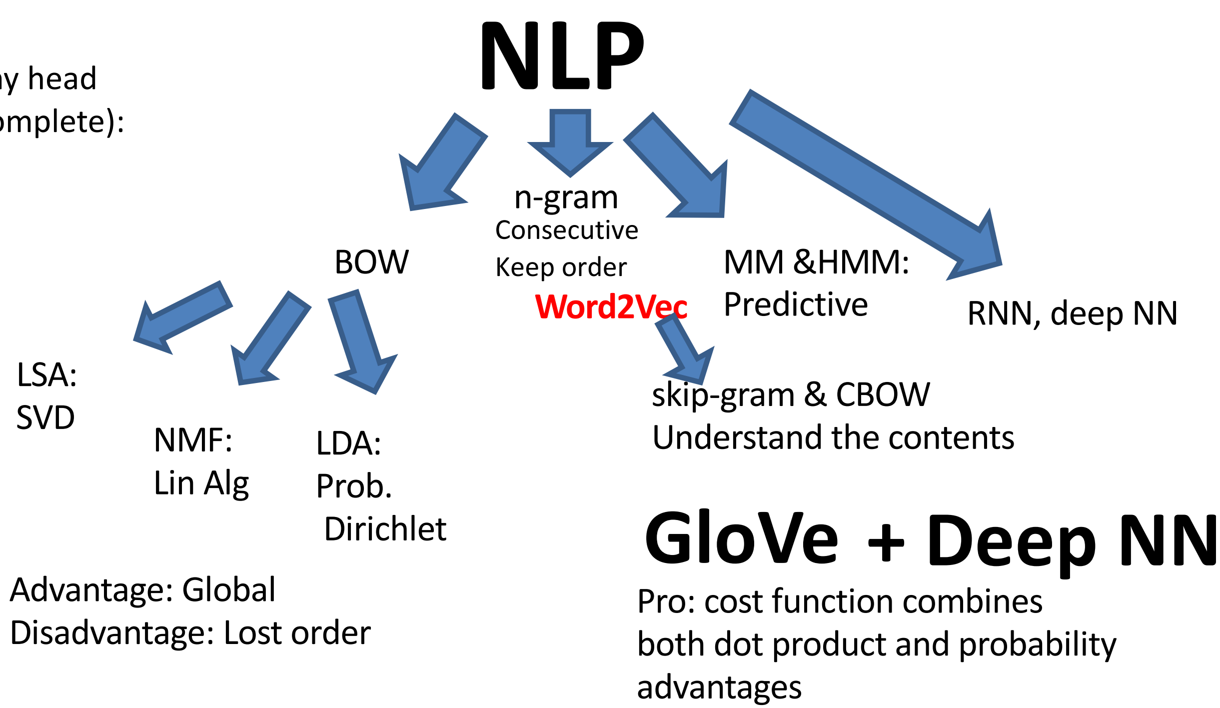
What is NLP?

- The field of study that focuses on the interactions between human language and computers is called Natural Language Processing, or NLP for short. It sits at the intersection of computer science, artificial intelligence, and computational linguistics.
- **Goal**: analyzing large pools of document sets (e.g. legislation docs), at-tempting to discover patterns and insights.
- **Key tasks**: organize and structure knowledge to perform tasks such as
  - automatic summarization,
  - translation,
  - named entity recognition,
  - relationship extraction,
  - sentiment analysis,
  - speech recognition, and
  - topic segmentation.

Overview: NLP

Most current powerful models in NLP

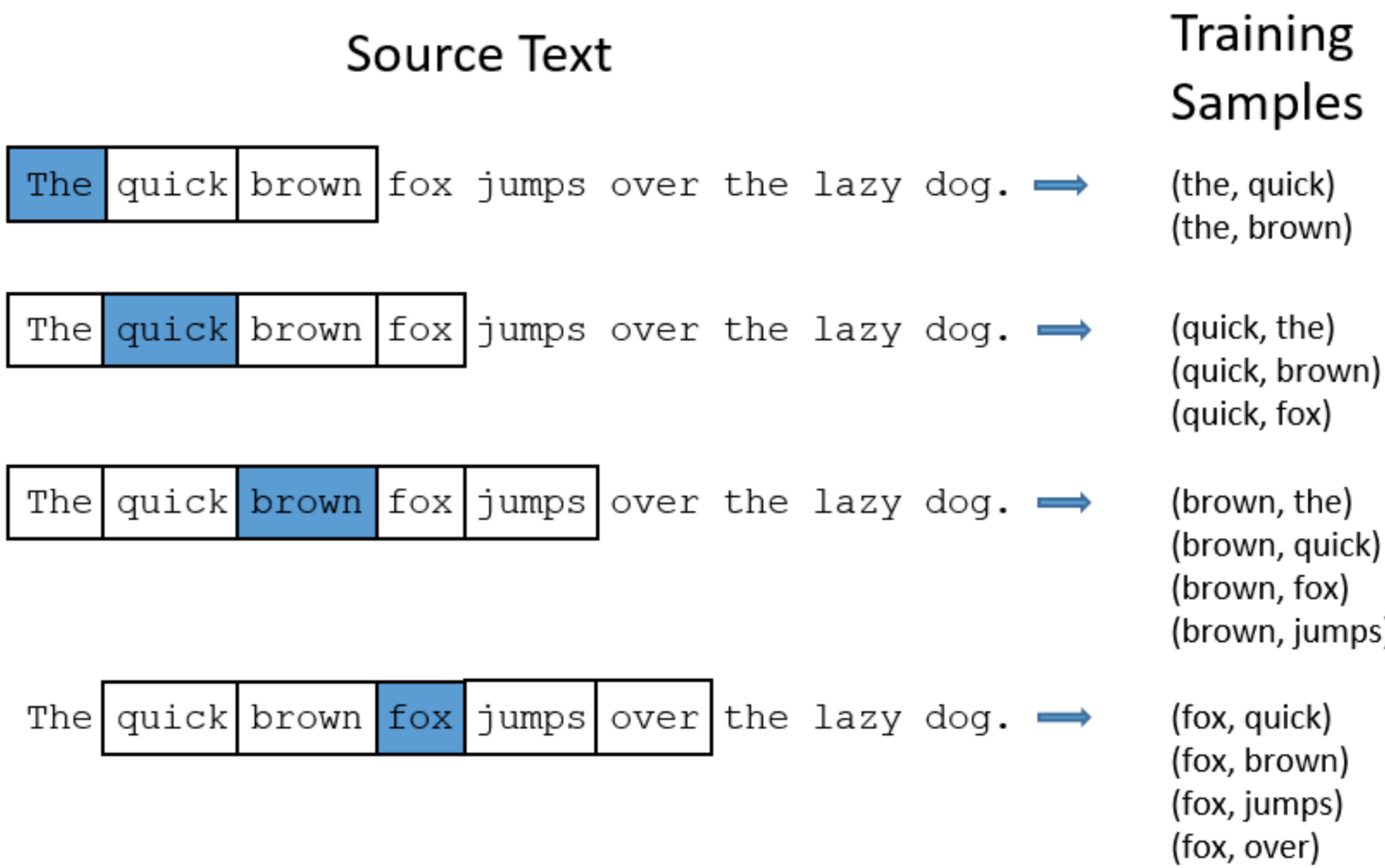
- Just off my head (may not be complete):



Natural Language Processing

- Overview: Working out details with students

Bag of Words (BOW) lost order.
How to keep contents around a word or
from a center word to understand content around it?
→Word2Vec



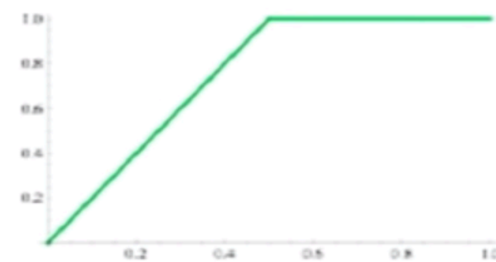
How to take both advantages of both BOW and Word2Vec?

Combining the best of both worlds: GloVe

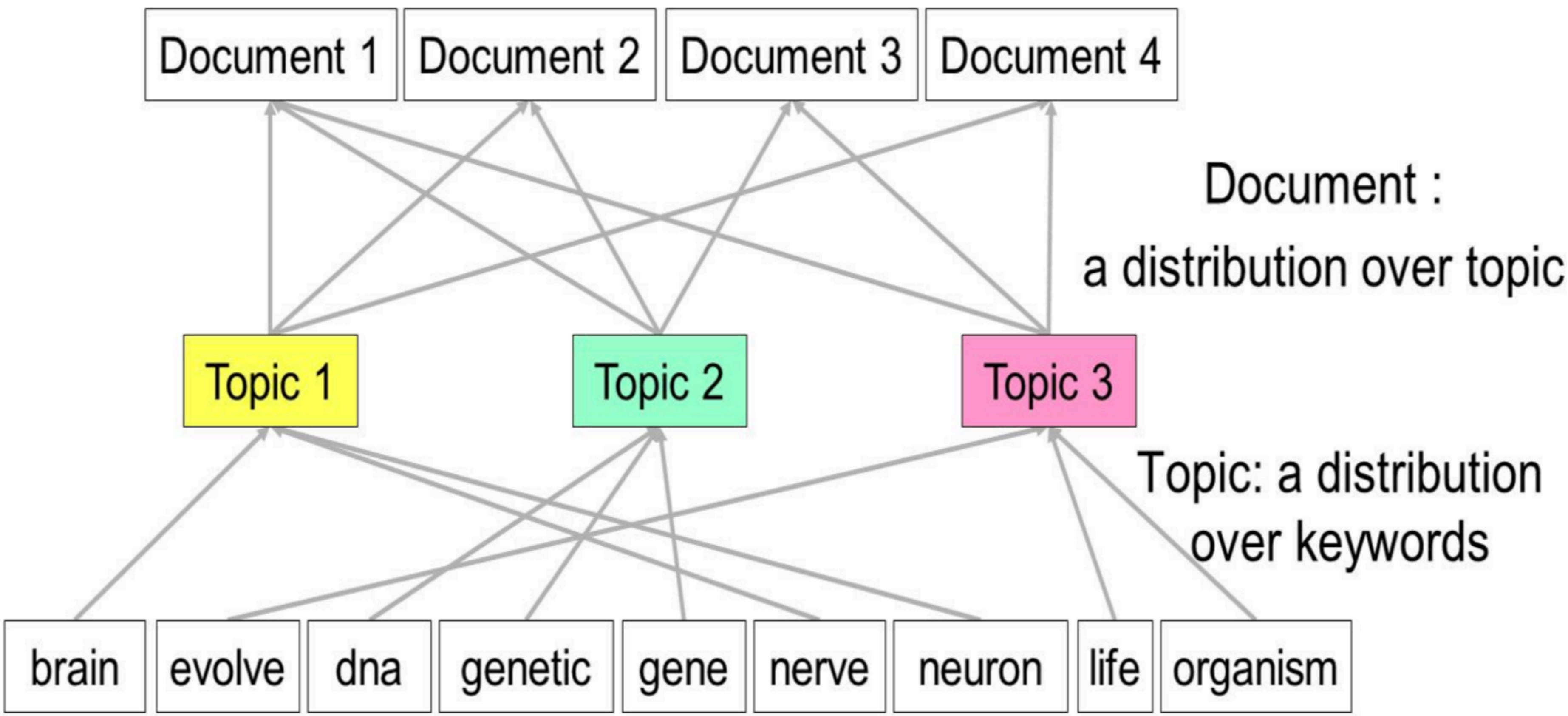
$$J(\theta) = \frac{1}{2} \sum_{i,j=1}^{W} f(P_{ij})(u_i^T v_j - \log P_{ij})^2$$

- Fast training
- Scalable to huge corpora
- Good performance even with small corpus, and small vectors
- By Pennington, Socher, Manning (2014)

$f \sim$

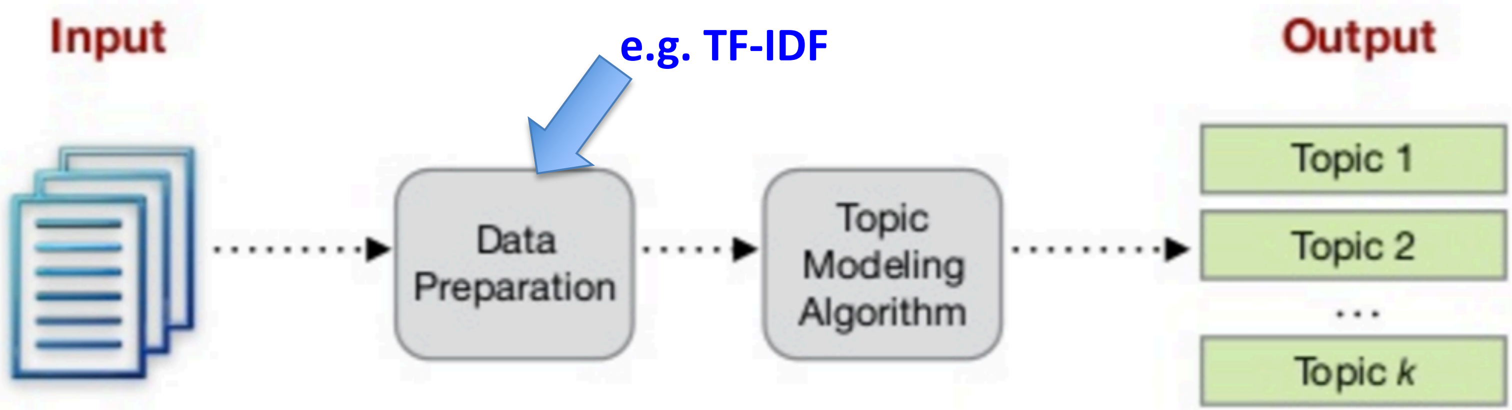


Intro: Topic Modeling



What is the Goal of Topic Modeling?

- **Goal:** Discover hidden thematic structure in a corpus of text (e.g. tweets, Facebook posts, news articles, political speeches).
- Unsupervised approach, no prior annotation required.



- Output of topic modeling is a set of *k* topics. Each topic has:
  1. A descriptor, based on highest-ranked terms for the topic.
  2. Membership weights for all documents relative to the topic.

What is the TF-IDF normalization?

tf–idf = term frequency–inverse document frequency

- TF-IDF is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.
- Mathematically, TF-IDF is the product of two statistics, term frequency and inverse document frequency.

Different ways to define Term Frequency $f_{t,d}$

- Raw frequency of a term in a document: *the number of times that term t occurs in document d, denoted by $f_{t.d}$.*
- Boolean "frequencies" defined as "= 1 if t occurs in d and 0 otherwise".
- logarithmically scaled frequency: 1 + log $f_{t.d}$, or zero if $f_{t.d}$ is zero.

Variants of TF weight

weighting scheme	TF weight
binary	$0, 1$
raw frequency	$f_{t,d}$
log normalization	$1 + \log(f_{t,d})$
double normalization 0.5	$0.5 + 0.5 \cdot \frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$
double normalization K	$K + (1 - K) \frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$

Inverse document frequency

- The inverse document frequency is a measure of how much information the word provides, that is, whether the term is common or rare across all documents.

$$\mathrm{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

- N: total number of documents in the corpus $N = |D|$
- $|\{d \in D : t \in d\}|$: number of documents where the term t appears (i.e., $\mathrm{tf}(t, d) \neq 0$). If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to adjust the denominator to $1 + |\{d \in D : t \in d\}|$.

Note: IDF then is a cross-document normalization, that puts less weight on common terms, and more weight on rare terms.

Different way to define
Inverse document frequency

Variants of IDF weight

weighting scheme	IDF weight ($n_t = \|\{d \in D : t \in d\}\|$)
unary	1
inverse document frequency	$\log \frac{N}{n_t}$
inverse document frequency smooth	$\log(1 + \frac{N}{n_t})$
inverse document frequency max	$\log\left(1 + \frac{\max_{\{t' \in d\}} n_{t'}}{n_t}\right)$
probabilistic inverse document frequency	$\log \frac{N - n_t}{n_t}$

How to calculate tf-idf?

Then tf–idf is calculated as

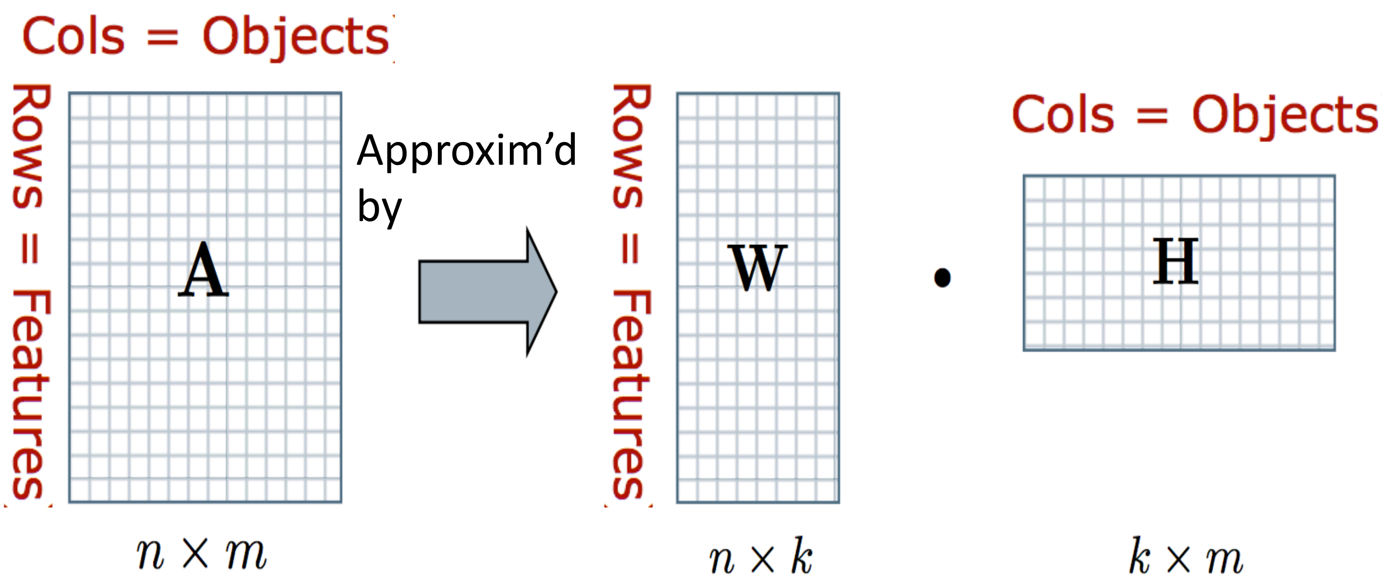
$$\mathrm{tfidf}(t, d, D) = \mathrm{tf}(t, d) \cdot \mathrm{idf}(t, D)$$

Recommended TF-IDF weighting schemes

weighting scheme	document term weight	query term weight
1	$f_{t,d} \cdot \log \frac{N}{n_t}$	$\left(0.5 + 0.5 \frac{f_{t,q}}{\max_t f_{t,q}}\right) \cdot \log \frac{N}{n_t}$
2	$1 + \log f_{t,d}$	$\log(1 + \frac{N}{n_t})$
3	$(1 + \log f_{t,d}) \cdot \log \frac{N}{n_t}$	$(1 + \log f_{t,q}) \cdot \log \frac{N}{n_t}$

What is Non-negative Matrix Factorization?

- Given a non-negative data matrix A.



Means each Element of W ≥0

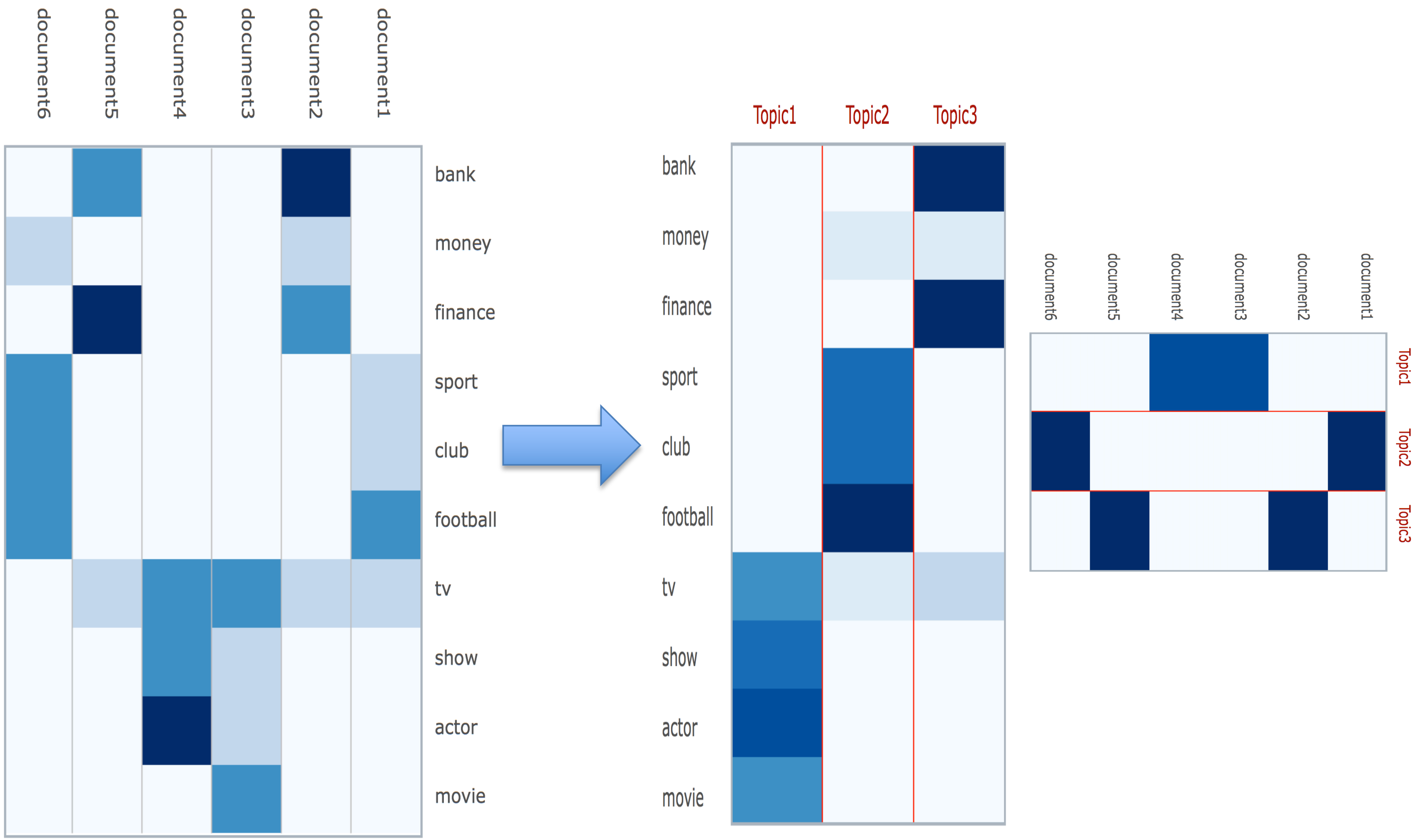
$$\mathbf{W} \geq 0\ ,\ \mathbf{H} \geq 0$$

Data matrix

Base vectors

Coefficient matrix

- W and H are called non-negative factors .



document6
document5
document4
document3
document2
document1
bank
money
finance
sport
club
football
tv
show
actor
movie
Topic1
Topic2
Topic3
bank
money
finance
sport
club
football
tv
show
actor
movie
document6
document5
document4
document3
document2
document1
Topic1
Topic2
Topic3

Goal: Minimizing the error between A and the approximation WH

$$\frac{1}{2}||\mathbf{A}-\mathbf{WH}||_{\mathsf{F}}^{2}=\sum_{i=1}^{n}\sum_{j=1}^{m}(A_{ij}-(WH)_{ij})^{2}$$

- Use EM optimization to refine W and H in order to minimize the objective function.

Non-negative Matrix Factorization Algorithm

- **Input:** Non-negative data matrix (**A**), number of basis vectors (*k*), initial values for factors **W** and **H** (e.g. random matrices).
- **Objective Function:** Some measure of reconstruction error between **A** and the approximation **WH**.

Euclidean Distance (Lee & Seung, 1999)

$$\frac{1}{2}||\mathbf{A} - \mathbf{WH}||_{\mathsf{F}}^2 = \sum_{i=1}^{n}\sum_{j=1}^{m}(A_{ij} - (WH)_{ij})^2$$

- **Optimisation Process:** Local EM-style optimisation to refine **W** and **H** in order to minimise the objective function.
- Common approach is to iterate between two multiplicative update rules until convergence (Lee & Seung, 1999).

1. Update **H**

$$H_{cj} \leftarrow H_{cj}\frac{(W\mathbf{A})_{cj}}{(W\mathbf{WH})_{cj}}$$

2. Update **W**

$$W_{ic} \leftarrow W_{ic}\frac{(\mathbf{A}H)_{ic}}{(\mathbf{WH}H)_{ic}}$$

So What?

- NMF: an unsupervised family of algorithms that simultaneously perform dimension reduction and clustering.

- NMF produces a “parts-based” decomposition of the hidden (or latent) relationships in a data matrix.

Applications of Non-negative Matrix Factorization

- Also known as positive matrix factorization (PMF) and nonnegative matrix approximation (NNMA).
- No strong statistical justification or grounding.
- But has been successfully applied in a range of areas:

  - *Bioinformatics (e.g. clustering gene expression networks).*
  - *Image processing (e.g. face detection).*
  - *Audio processing (e.g. source separation).*
  - *Text analysis (e.g. document clustering).*

How to select k?

- As with LDA, the selection of number of topics k is often
- performed manually. No definitive model selection strategy.
- • Various alternatives comparing different models:
- - Compare reconstruction errors for different parameters.
- Natural bias towards larger value of k.
- - Build a “consensus matrix” from multiple runs for each k, assess presence of block structure (Brunet et al, 2004).
- - Examine the stability (i.e. agreement between results) from multiple randomly initialized runs for each value of k.

Variants of Non-negative Matrix Factorization

Different objective functions:

- KL divergence (Sra & Dhillon, 2005).

More efficient optimization:

- Alternating least squares with projected gradient method for sub-problems (Lin, 2007).

Constraints:

- Enforcing sparseness in outputs (e.g. Liu et al, 2003).
- Incorporation of background information (Semi-NMF)

Different inputs:

- Symmetric matrices - e.g. document-document cosine similarity matrix (Ding & He, 2005).

- NMF is only one of Topic Modeling algorithms
  - **Key: low rank matrix factorization**
- There is another classic method, called Latent Semantic Analysis (LSA)
  - **Key: use spectral (eigenvalues-eigenvector) analysis**

NLP contains many algorithms

Classic: Latent Semantic Analysis (LSA)

- Latent Semantic Analysis (**LSA**)
- Recurrent Neural Network (**RNN**)
- **Word2Vec (includes n-gram)**
- Latent Dirichlet Allocation **(LDA)**
- **HMM**
- **GloVe**
- **Combinations of above (say GloVe + deep NN)**
- …

- There are many GitHub with code you may need to talk a close look at them. For example:
- https://github.com/vikparuchuri/vikparuchuri.com

Latent Semantic Analysis (LSA)

- Latent Semantic Analysis (LSA) comprises of certain mathematical operation to get insight on a document.
- This algorithm forms the basis of *Topic Modeling*.
- The core idea is to take a matrix of what we have — documents and terms — and decompose it into a separate document-topic matrix and a topic-term matrix.

Set up: Matrix representation of documents: ***changing to numbers as before***

Let X be a matrix where element (i, j) describes the occurrence of term i in document j (this can be, for example, the frequency). X will look like this:

$$
\begin{array}{cc}
 & \begin{array}{ccccc} & & \mathbf{d}_j & & \\ & & \downarrow & & \end{array} \\
\mathbf{t}_i^T \rightarrow & \begin{bmatrix}
x_{1,1} & \dots & x_{1,j} & \dots & x_{1,n} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
x_{i,1} & \dots & x_{i,j} & \dots & x_{i,n} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
x_{m,1} & \dots & x_{m,j} & \dots & x_{m,n}
\end{bmatrix}
\end{array}
$$

Each column is a document
What is each row?

Now a row in this matrix will be a vector corresponding to a term, giving its relation to each document:

$$\mathbf{t}_i^T = \begin{bmatrix} x_{i,1} & \dots & x_{i,j} & \dots & x_{i,n} \end{bmatrix}$$

Likewise, a column in this matrix will be a vector corresponding to a document, giving its relation to each term:

$$\mathbf{d}_j = \begin{bmatrix} x_{1,j} \\ \vdots \\ x_{i,j} \\ \vdots \\ x_{m,j} \end{bmatrix}$$

The dot product of two term vectors has meaning

Now the dot product $\mathbf{t}_i^T \mathbf{t}_p$ between two term vectors gives the correlation between the terms over the set of documents. The matrix product XX^T contains all these dot products. Element (i, p) (which is equal to element (p, i)) contains the dot product $\mathbf{t}_i^T \mathbf{t}_p$ ($= \mathbf{t}_p^T \mathbf{t}_i$). Likewise, the matrix $X^T X$ contains the dot products between all the document vectors, giving their correlation over the terms: $\mathbf{d}_j^T \mathbf{d}_q = \mathbf{d}_q^T \mathbf{d}_j$.

Recall: A gram matrix:

$$G(x_1, \ldots, x_n) = \begin{vmatrix} \langle x_1, x_1 \rangle & \langle x_1, x_2 \rangle & \ldots & \langle x_1, x_n \rangle \\ \langle x_2, x_1 \rangle & \langle x_2, x_2 \rangle & \ldots & \langle x_2, x_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle x_n, x_1 \rangle & \langle x_n, x_2 \rangle & \ldots & \langle x_n, x_n \rangle \end{vmatrix}.$$

Key: Use Singular Value Decomposition

Working out details with students on iPad.

Now, from the theory of linear algebra, there exists a decomposition of X such that U and V are orthogonal matrices and Σ is a diagonal matrix. This is called a singular value decomposition (SVD):

$$X = U\Sigma V^T$$

The matrix products giving us the term and document correlations then become

$$\begin{aligned} XX^T &= (U\Sigma V^T)(U\Sigma V^T)^T = (U\Sigma V^T)(V^{T^T}\Sigma^T U^T) = U\Sigma V^T V\Sigma^T U^T = U\Sigma\Sigma^T U^T \\ X^T X &= (U\Sigma V^T)^T (U\Sigma V^T) = (V^{T^T}\Sigma^T U^T)(U\Sigma V^T) = V\Sigma^T U^T U\Sigma V^T = V\Sigma^T \Sigma V^T \end{aligned}$$

Since $\Sigma\Sigma^T$ and $\Sigma^T\Sigma$ are diagonal we see that U must contain the eigenvectors of XX^T, while V must be the eigenvectors of $X^T X$. Both products have the same non-zero eigenvalues, given by the non-zero entries of $\Sigma\Sigma^T$, or equally, by the non-zero entries of $\Sigma^T\Sigma$. Now the decomposition looks like this:

$$\begin{array}{cccccc} & & & X & & \\ & & & (\mathbf{d}_j) & & \\ & & & \downarrow & & \\ (\mathbf{t}_i^T) \rightarrow & \left[\begin{matrix} x_{1,1} \\ \vdots \\ x_{i,1} \\ \vdots \\ x_{m,1} \end{matrix}\right. & \begin{matrix} \dots \\ \ddots \\ \dots \\ \ddots \\ \dots \end{matrix} & \begin{matrix} x_{1,j} \\ \vdots \\ x_{i,j} \\ \vdots \\ x_{m,j} \end{matrix} & \begin{matrix} \dots \\ \ddots \\ \dots \\ \ddots \\ \dots \end{matrix} & \left.\begin{matrix} x_{1,n} \\ \vdots \\ x_{i,n} \\ \vdots \\ x_{m,n} \end{matrix}\right] \end{array} = \quad (\hat{\mathbf{t}}_i^T) \rightarrow$$

$$\begin{array}{ccccc} U & & \Sigma & & \begin{array}{c} V^T \\ (\hat{\mathbf{d}}_j) \\ \downarrow \end{array} \end{array}$$

$$\left[\begin{bmatrix} \\ \mathbf{u}_1 \\ \\ \end{bmatrix} \dots \begin{bmatrix} \\ \mathbf{u}_l \\ \\ \end{bmatrix} \right] \cdot \begin{bmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_l \end{bmatrix} \cdot \begin{bmatrix} [& \mathbf{v}_1 &] \\ & \vdots & \\ [& \mathbf{v}_l &] \end{bmatrix}$$

The values $\sigma_1, \dots, \sigma_l$ are called the singular values, and $u_1, \dots, u_l$ and $v_1, \dots, v_l$ the left and right singular vectors. Notice the only part of U that contributes to $\mathbf{t}_i$ is the $i\text{'th}$ row. Let this row vector be called $\hat{\mathrm{t}}_i^T$. Likewise, the only part of V^T that contributes to $\mathbf{d}_j$ is the $j\text{'th}$ column, $\hat{\mathrm{d}}_j$. These are *not* the eigenvectors, but *depend* on *all* the eigenvectors.

It turns out that when you select the k largest singular values, and their corresponding singular vectors from U and V, you get the rank k approximation to X with the smallest error (Frobenius norm). This approximation has a minimal error. But more importantly we can now treat the term and document vectors as a "semantic space". The row "term" vector $\hat{\mathbf{t}}_i^T$ then has k entries mapping it to a lower-dimensional space dimensions. These new dimensions do not relate to any comprehensible concepts. They are a lower-dimensional approximation of the higher-dimensional space. Likewise, the "document" vector $\hat{\mathbf{d}}_j$ is an approximation in this lower-dimensional space. We write this approximation as

$$X_k = U_k \Sigma_k V_k^T$$

You can now do the following:

- See how related documents j and q are in the low-dimensional space by comparing the vectors $\Sigma_k \hat{\mathbf{d}}_j$ and $\Sigma_k \hat{\mathbf{d}}_q$ (typically by cosine similarity).
- Comparing terms i and p by comparing the vectors $\Sigma_k \hat{\mathbf{t}}_i$ and $\Sigma_k \hat{\mathbf{t}}_p$. Note that $\hat{\mathbf{t}}$ is now a column vector.
- Documents and term vector representations can be clustered using traditional clustering algorithms like k-means using similarity measures like cosine.
- Given a query, view this as a mini document, and compare it to your documents in the low-dimensional space.

To do the latter, you must first translate your query into the low-dimensional space. It is then intuitive that you must use the same transformation that you use on your documents:

$$\hat{\mathbf{d}}_j = \Sigma_k^{-1} U_k^T \mathbf{d}_j$$

Note here that the inverse of the diagonal matrix Σ_k may be found by inverting each nonzero value within the matrix.

This means that if you have a query vector q, you must do the translation $\hat{\mathbf{q}} = \Sigma_k^{-1} U_k^T \mathbf{q}$ before you compare it with the document vectors in the low-dimensional space. You can do the same for pseudo term vectors:

$$\mathbf{t}_i^T = \hat{\mathbf{t}}_i^T \Sigma_k V_k^T$$

$$\hat{\mathbf{t}}_i^T = \mathbf{t}_i^T V_k^{-T} \Sigma_k^{-1} = \mathbf{t}_i^T V_k \Sigma_k^{-1}$$

$$\hat{\mathbf{t}}_i = \Sigma_k^{-1} V_k^T \mathbf{t}_i$$