UNIVERSITY of LOUISIANA LAFAYETTE

CSCE566-DATA MINING
WEEK 4

Text Data Mining

Min Shi
min.shi@louisiana.edu

Sep 17, 2024

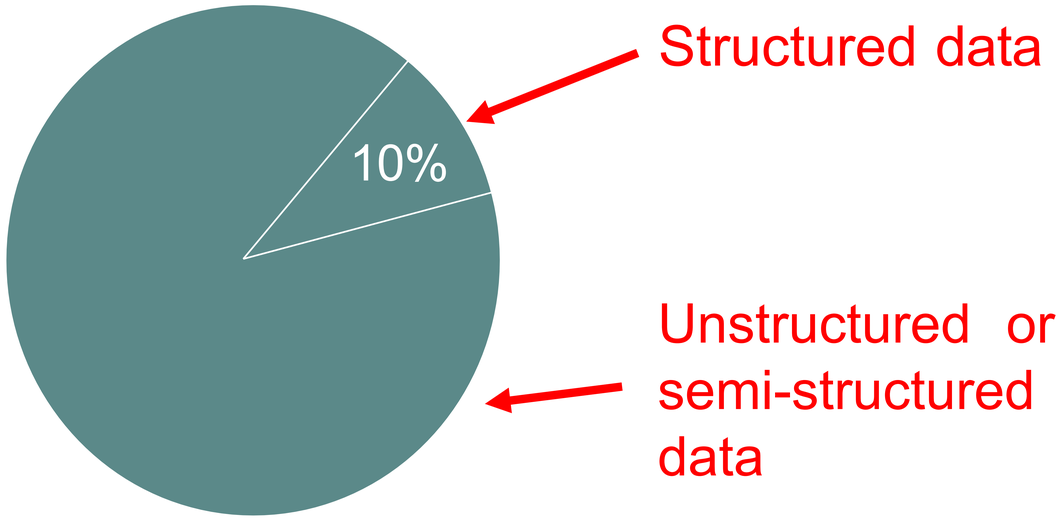
Outline: Text Data Mining

1. Introduction to Text Mining
2. Vector Space Model
3. Text Classification
4. Probabilistic Topic Models

Motivation for text mining

Approximately 90% of the world's data is held in unstructured formats.

Source: Oracle Corporation

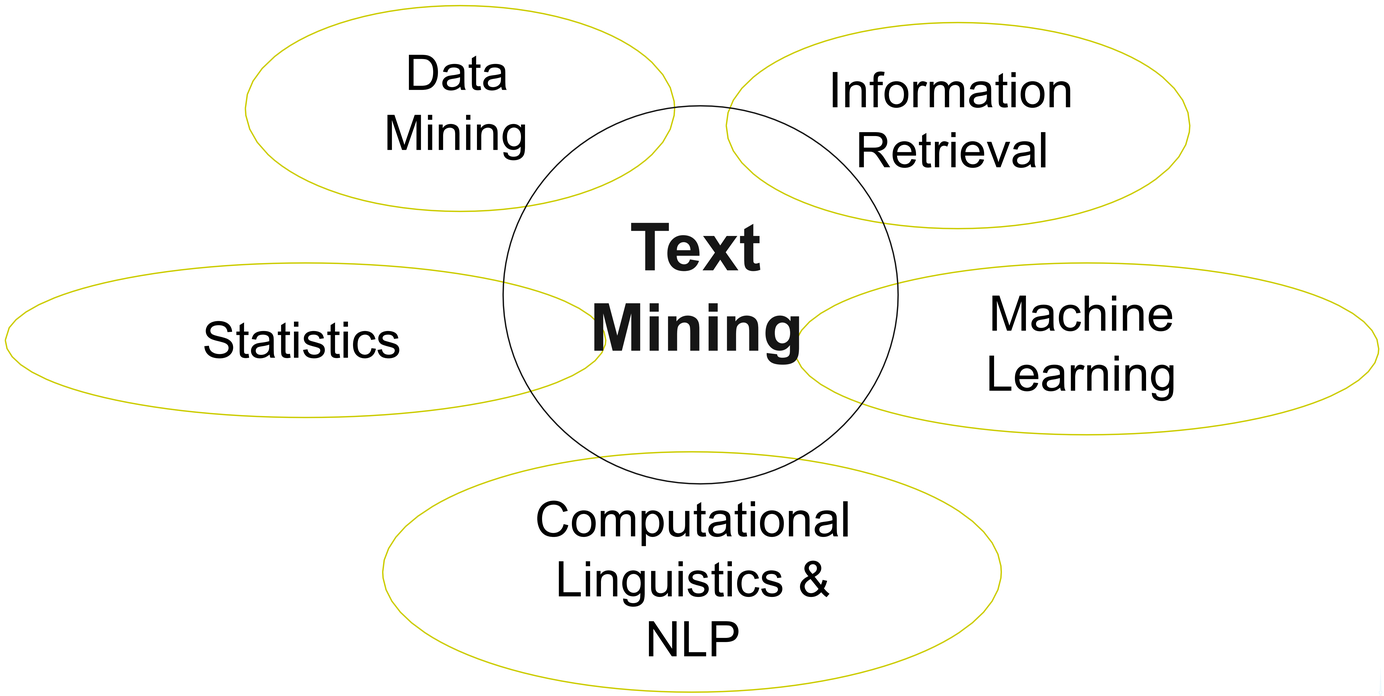


Examples:

- web pages
- emails
- customer complaint letters
- corporate documents
- scientific papers
- books in digital libraries

Text mining

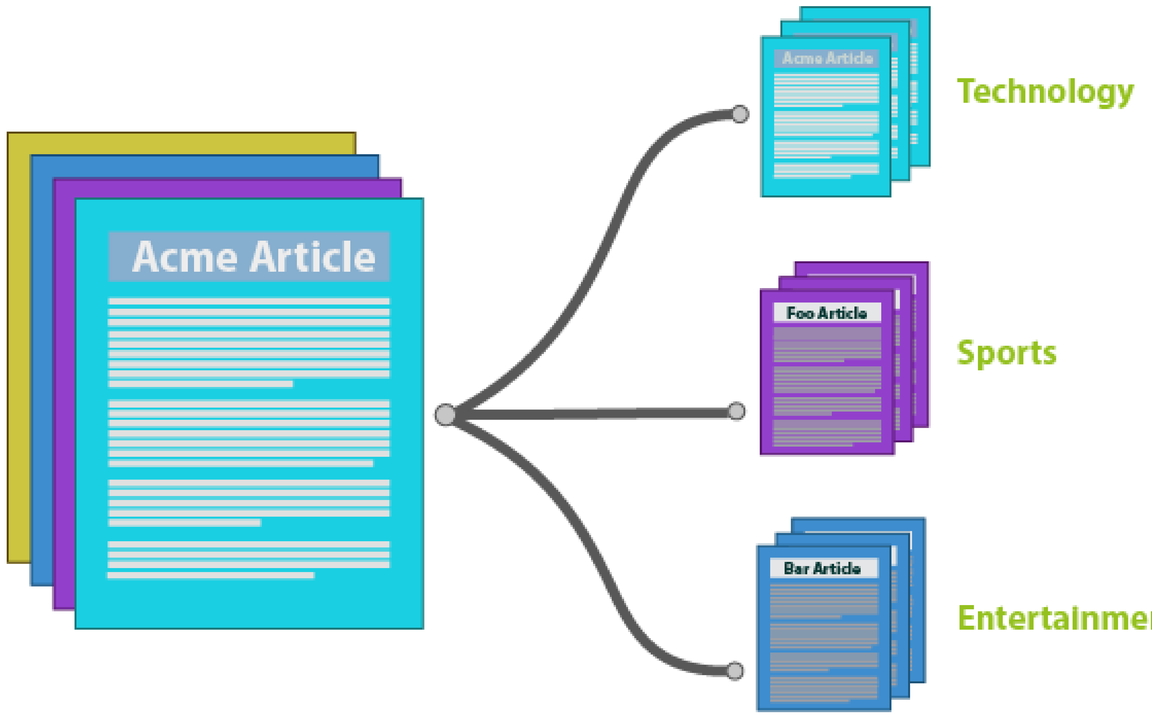
The extraction of implicit, previously unknown and potentially useful information from large amounts of textual resources.

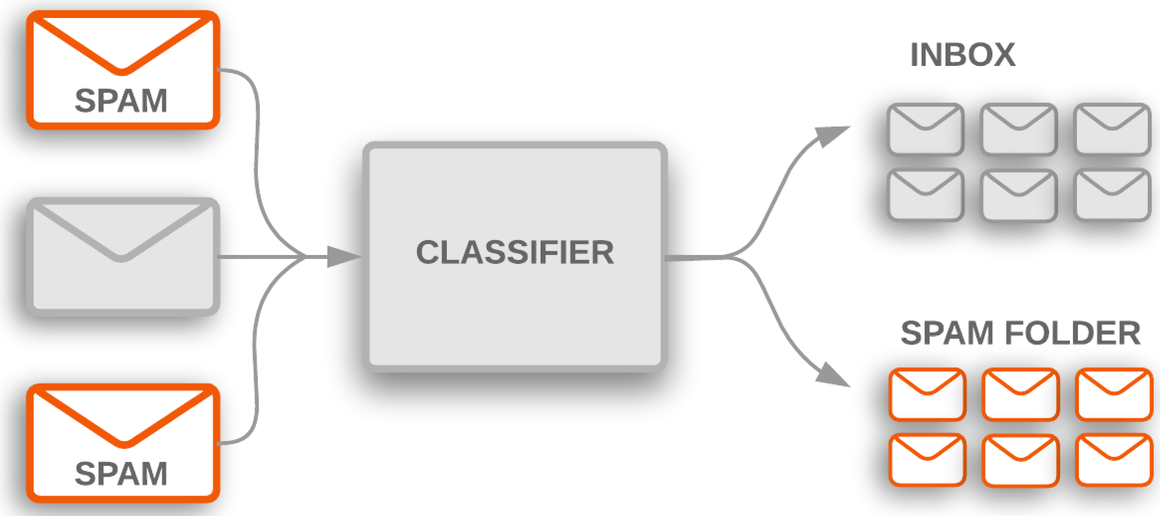


Text mining applications

1. Classification of news stories
2. Email and news filtering / SPAM detection
3. Sentiment analysis
4. Grouping of documents or web pages
5. Information extraction
6. Text summarization / Text generation
7. Question answering

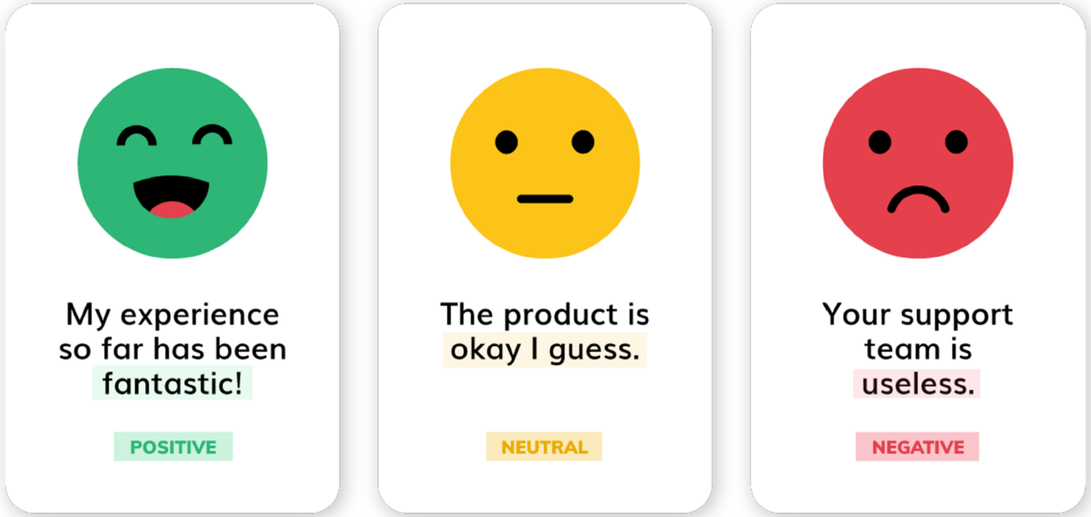
Document clustering and classification





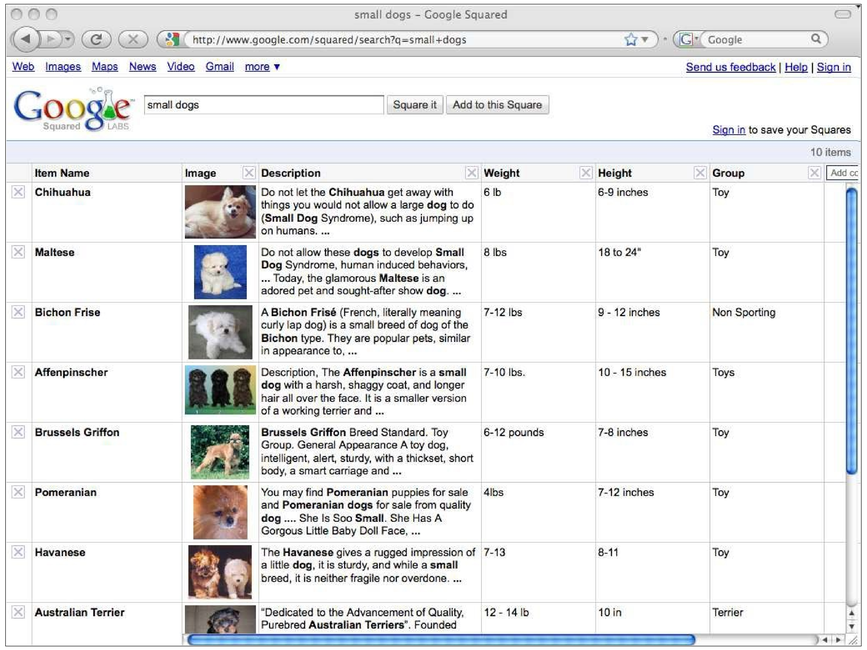
Sentiment analysis

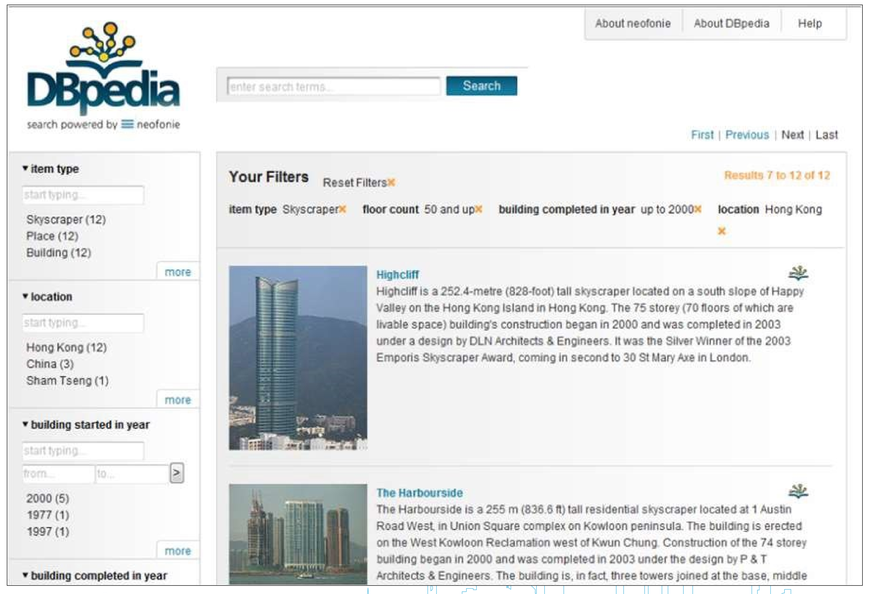
- The goal of sentiment analysis is to determine the polarity of a given text at the document, sentence, or feature/aspect level
- Polarity values
  - positive, neutral, negative
  - like scale (1 to 10)
- Application examples
  - Document level
    - analysis of tweets about politicians
  - Feature/aspect level
    - analysis of product reviews



Information extraction

- Information extraction is the task of automatically extracting structured information from unstructured or semi-structured documents.
- Subtasks
  1. Named Entity Recognition and Disambiguation
     - "The parliament in Berlin has decided …"
     - Which parliament? Which Berlin?
  2. Relationship Extraction
     - PERSON works for ORGANIZATION
     - PERSON located in LOCATION
  3. Fact Extraction
     - CITY has population NUMBER
     - COMPANY has turnover NUMBER [Unit]





Cross-language translation

Example: language translation

- **Input**: Length-variable sequence
- **Output**: Length-variable sequence

Figure: Translation from English to Chinese.

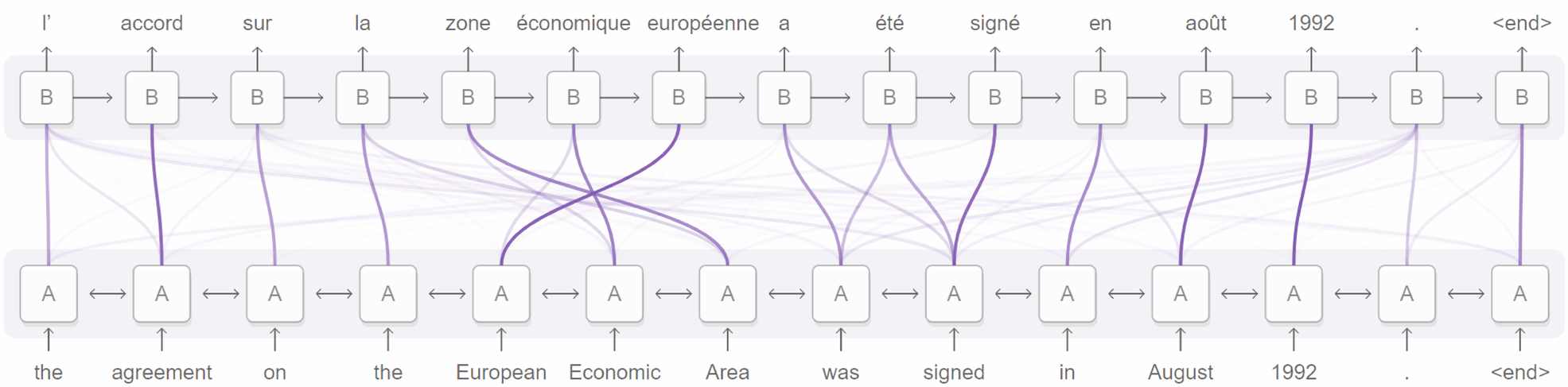


Diagram derived from Fig. 3 of Bahdanau, *et al.* 2014

Outline: Text Data Mining

1. Introduction to Text Mining
2. Vector Space Model
3. Text Classification
4. Probabilistic Topic Models
5. Language Models

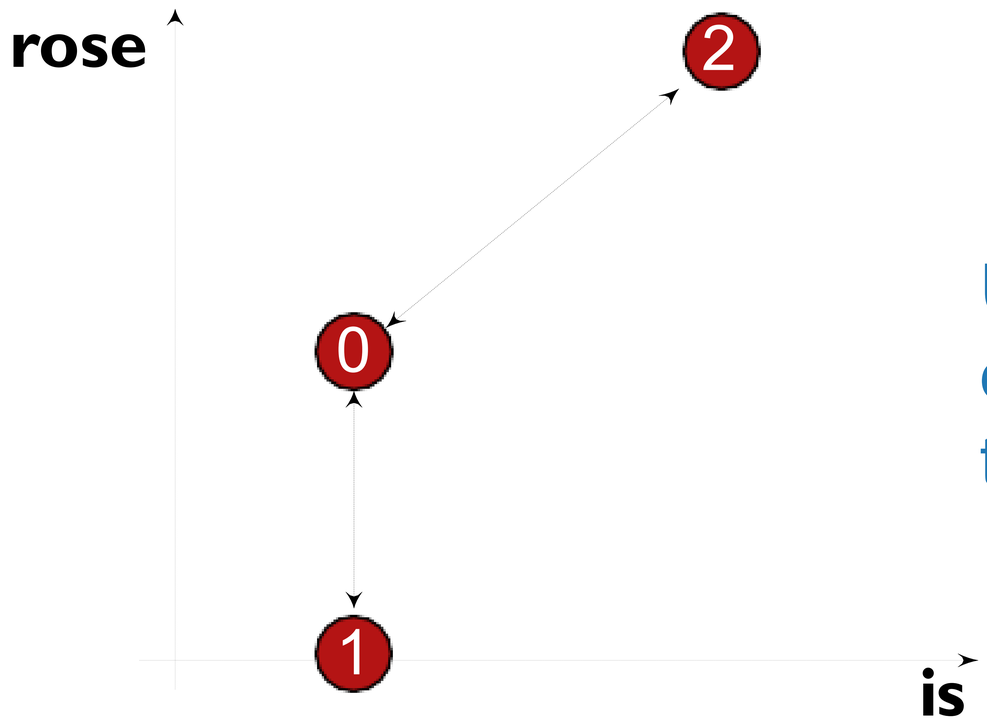
Textual data

1. "A Rose Is Still a Rose"
2. "There is no there there."
3. "Rose is a rose is a rose is a rose."

⇒

	is	rose	...
0	1	2	...
1	1	0	...
2	3	4	...

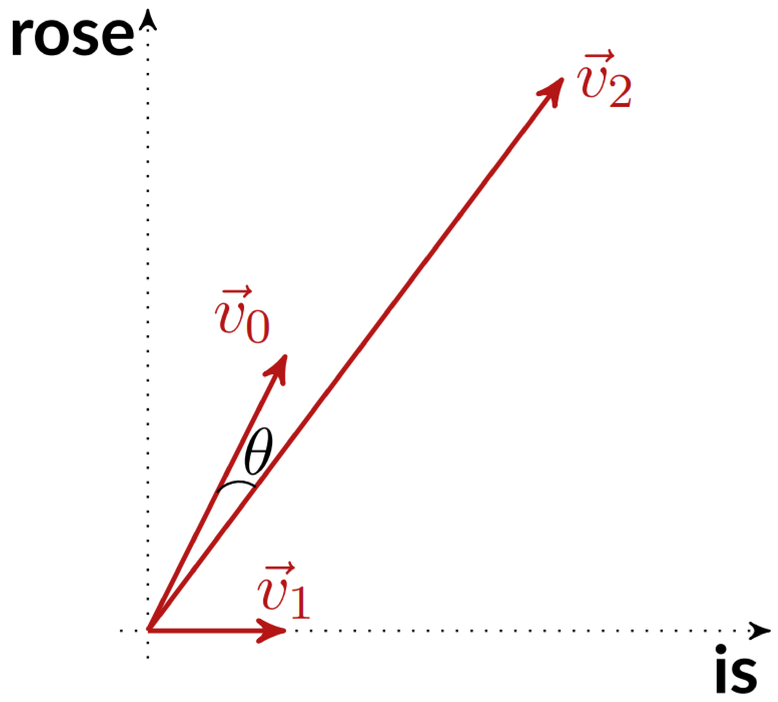
Which document is most similar to document 0?



Using Euclidean distance, document 1 appears closer than document 2!

Vector space model

In the **vector space model**, documents are represented *as vectors* instead of points.



The **length of a vector** is its distance from the origin $\vec{0}$:

$$||\vec{v}|| = \sqrt{\sum_{j=1}^{D} v_j^2}.$$

The distance between two vectors is the angle between them:

$$d(\vec{v}, \vec{w}) = 1 - \cos\theta = 1 - \frac{\text{sum of } v_j \cdot w_j}{||\vec{v}|| \cdot ||\vec{w}||}.$$

Using cosine distance, document 2 now appears closer!

Implementing a vector space model

```
corpus = ["a rose is still a rose", "there is no there there",
"rose is a rose is a rose is a rose"]
```

First, we use Pandas to get the term-frequency matrix.

```
import pandas as pd
from collections import Counter

tf = pd.DataFrame([pd.Series(Counter(doc.split())) for doc in corpus],
                  ).fillna(0)
```

tf

	a	rose	is	still	there	no
0	2.0	2.0	1.0	1.0	0.0	0.0
1	0.0	0.0	1.0	0.0	3.0	1.0
2	3.0	4.0	3.0	0.0	0.0	0.0

Now we just have to implement the formula for cosine distance.

Implementing a vector space model

tf =

	a	rose	is	still	there	no
0	2.0	2.0	1.0	1.0	0.0	0.0
1	0.0	0.0	1.0	0.0	3.0	1.0
2	3.0	4.0	3.0	0.0	0.0	0.0

Now we just have to implement the formula for cosine distance.

$$d(\vec{v}, \vec{w}) = 1 - \frac{\text{sum of } v_j \cdot w_j}{||\vec{v}|| \cdot ||\vec{w}||}.$$

```
import numpy as np

def length(v):
  return np.sqrt((v ** 2).sum())

def cos_dist(v, w):
  return 1 - (v * w).sum() / (length(v) * length(w))

cos_dist(tf.loc[0], tf.loc[1]), cos_dist(tf.loc[0], tf.loc[2])
```

```
(0.9046537410754407, 0.07804555427071147)
```

Vector space model in scikit-learn

It's easier to do it in Scikit-Learn.

```
from sklearn.feature_extraction.text import CountVectorizer

vec = CountVectorizer(token_pattern=r"(?u)\b\w+\b")
vec.fit(corpus)
tf_mat = vec.transform(corpus)
tf_mat.todense()
```

```
matrix([[2, 1, 0, 2, 1, 0],
        [0, 1, 1, 0, 0, 3],
        [3, 3, 0, 4, 0, 0]])
```

```
from sklearn.metrics import pairwise_distances
pairwise_distances(tf_mat[0, :], tf_mat[1:, :], metric="cosine")
```

```
array([[0.90465374, 0.07804555]])
```

TF-IDF

So far, we've simply counted the **term frequency** tf(d, t): how many times each term t appears in each document d.

Problem: Common words like "is" or "the" tend to dominate because they have high counts.

We need to adjust for how common each word is:

1. Count the fraction of documents the term appears in:

$$\mathrm{df}(t, D) = \frac{\#\text{ documents containing term } t}{\#\text{ documents}} = \frac{|d \in D : t \in d|}{|D|}$$

2. Invert and take a log to obtain **inverse document frequency**:

$$\mathrm{idf}(t, D) = 1 + \log \frac{1}{\mathrm{df}(t, D)}.$$

3. Multiply tf by idf to get tf-idf:

$$\text{tf-idf}(d, t, D) = \mathrm{tf}(d, t) \cdot \mathrm{idf}(t, D).$$

Now we can use the **tf-idf matrix** just like we used the term-frequency matrix.

TF-IDF example

0. `"A Rose Is Still a Rose"`
1. `"There is no there there."`
2. `"Rose is a rose is a rose is a rose."`

$\Rightarrow$

	is	rose	…
0	1	2	…
1	1	0	…
2	3	4	…

Now let's calculate the TF-IDF matrix!

1. Calculate the document frequencies:

$$\mathrm{df}(\text{“is”}, D) = \frac{3}{3} = 1 \qquad \mathrm{df}(\text{“rose”}, D) = \frac{2}{3}$$

2. Calculate the inverse document frequencies:

$$\mathrm{idf}(\text{“is”}, D) = 1 + \log 1 = 1 \quad \mathrm{idf}(\text{“rose”}, D) = 1 + \log 1.5 \approx 1.176$$

3. Multiply tf by idf to get tf-idf:

	is	rose	…
0	1	2.81	…
1	1	0	…
2	3	5.62	…

TF-IDF in scikit-learn

```
from sklearn.feature_extraction.text import TfidfVectorizer

# The options ensure that the numbers match our example above.
vec = TfidfVectorizer(smooth_idf=False, norm=None)
vec.fit(corpus)
tfidf_mat = vec.transform(corpus)
tfidf_mat.todense()
```

```
matrix([[1.        , 0.        , 2.81093022, 2.09861229, 0.        ],
        [1.        , 2.09861229, 0.        , 0.        , 6.29583687],
        [3.        , 0.        , 5.62186043, 0.        , 0.        ]])
```

```
pairwise_distances(tfidf_mat[0, :], tfidf_mat[1:, :], metric="cosine")
```

```
array([[0.95915143, 0.19106774]])
```

Outline: Text Data Mining

1. Introduction to Text Mining
2. Vector Space Model
3. Text Classification
4. Probabilistic Topic Models

Example

Is this spam?

YOUU Have Been PAID 📦 Check Yourr Account-Now 📦 💲 $1000.00 💲 User ID#372-75784 Spam ×

CashAP 💲 uqaxeuhkmhorygq@extentor.help via pm.mtasv.net
to contigome

Sat, Mar 18, 11:08 PM

-THIS MESSAGE WAS SENT FROM A TRUSTED SENDER.

C0NGRATULATIONS ****@gmail.com !

A.balance..0F **$1000.00** Is AVAILABLE F0R..your ***CashApp*.Accountt**

Thiss.TRANSACTION.may.0nly.appearr. 0n.your.ACC0UNTT..afterr.
VALIDATE.your.Info.

PAY0UT:

03/2023

$1000.000

FUNDING.For: **** EMAIL: ****@gmail.com

Confirm Here

Balance Amount: $1000.00

Memo *PAY0UT* SIGNATURE ****

Example

What is the subject of this article?

MEDLINE Article

Syntactic frame and verb bias in aphasia: Plausibility judgments of undergoer-subject sentences

?

MeSH Subject Category Hierarchy

- Antogonists and Inhibitors
- Blood Supply
- Chemistry
- Drug Therapy
- Embryology
- Epidemiology
- …

Example

- Assigning subject categories, topics, or genres
- Spam detection
- Authorship identification
- Age/gender identification
- Language Identification
- Sentiment analysis
- …

Text classification: definition

Classification Task:

- *Input*:
  - a document d
  - a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$
- *Output*: a predicted class $c \in C$

Methods:

- Any kind of classifier
  - Naïve Bayes
  - Logistic regression, maxent
  - Support-vector machines
  - k-Nearest Neighbors
  - …

Chain rule & Bayes rule

Chain rule:

$$P(X,Y) = P(X|Y)P(Y) = P(Y|X)P(X)$$

- **A**: It is raining.
- **B**: You carry an umbrella.
- **P(A)**: Probability that it is raining = 0.3 (30%)
- **P(B|A)**: Probability that you carry an umbrella given that it is raining = 0.9 (90%)
- **P(B|A^c)**: Probability that you carry an umbrella given that it is not raining = 0.1 (10%)

P(B, A) $= P(B|A)P(A) = 0.9 \times 0.3$

$= P(A|B) \times P(B)$ $= 0.27$

Chain rule & Bayes rule

Chain rule:

$$P(X,Y) = P(X|Y)P(Y) = P(Y|X)P(X)$$

Bayes rule:

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)}$$

- $P(\text{Disease}) = 0.01$
- $P(\text{No Disease}) = 0.99$
- $P(\text{Positive}|\text{Disease}) = 0.9$
- $P(\text{Positive}|\text{No Disease}) = 0.05$

Find $P(\text{Disease}|\text{Positive})$:

Using Bayes' Rule:

$$P(\text{Disease}|\text{Positive}) = \frac{P(\text{Positive}|\text{Disease}) \times P(\text{Disease})}{P(\text{Positive})}$$

$$P(\text{Positive}) = P(\text{Positive}|\text{Disease}) \times P(\text{Disease}) + P(\text{Positive}|\text{No Disease}) \times P(\text{No Disease})$$

$$= (0.9 \times 0.01) + (0.05 \times 0.99) = 0.009 + 0.0495 = 0.0585$$

Apply Bayes' Rule:

$$P(\text{Disease}|\text{Positive}) = \frac{0.9 \times 0.01}{0.0585} = \frac{0.009}{0.0585} \approx 0.1538$$

Bayesian learning

$\mathcal{D}$ is the measured data.
Our goal is to estimate parameter θ.

- Use Bayes rule:

$$P(\theta \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid \theta) P(\theta)}{P(\mathcal{D})}$$

- Or equivalently:

$$P(\theta \mid \mathcal{D}) \propto P(\mathcal{D} \mid \theta) P(\theta)$$

posterior likelihood prior

Bayesian classification

- Let set of categories be $\{c_1, c_2, \ldots c_n\}$
- Let E be description of an instance.
- Determine category of E by determining for each c_i

$$P(c_i \mid E) = \frac{P(c_i)P(E \mid c_i)}{P(E)}$$

- P(E) can be ignored since is factor $\forall$ categories

$$P(c_i \mid E) \sim P(c_i)P(E \mid c_i)$$

Bayesian classification

$$P(c_i \mid E) \sim P(c_i) P(E \mid c_i)$$

- Need to know:
  - Priors: P(c_i)
  - Conditionals: P($E \mid c_i$)
- P(c_i) are easily estimated from data.
  - If n_i of the examples in D are in c_i, then P(c_i) = $n_i / |D|$
- Assume instance is a conjunction of binary features:

$$E = e_1 \wedge e_2 \wedge \cdots \wedge e_m$$

- Too many possible instances (exponential in m) to estimate all P($E \mid c_i$)

Problem!

Naïve Bayesian rule

- Problem: Too many possible instances
  (exponential in m)
  to estimate all $P(E \mid c_i)$

- If we assume features of an instance are independent given the category (c_i) (*conditionally independent*).

$$P(E \mid c_i) = P(e_1 \wedge e_2 \wedge \cdots \wedge e_m \mid c_i) = \prod_{j=1}^{m} P(e_j \mid c_i)$$

- Therefore, we then only need to know $P(e_j \mid c_i)$ for each feature and category.

Example

Document	Content	Classification
Doc1	Promotion Bumper Winner	Spam
Doc2	Deadline Meeting Promotion	Non-Spam
Doc3	Bumper Lottery Winner	Spam

C={Spam, non-Spam}

Spam={Promotion, Bumper, Winner, Lottery}

Non-Spam={Deadline, Meeting, Promotion}

$$\hat{P}(c_j) = \frac{doccount(C = c_j)}{N_{doc}}$$

$$\hat{P}(w_i \mid c_j) = \frac{count(w_i, c_j)}{\sum_{w \in V} count(w, c_j)}$$

New Document: Promotion Bumper Lottery

- P(Spam) = 2/3
- P(Non-Spam) = 1/3

Word	Probability (P(Word | Spam))
Promotion	1/6
Bumper	2/6
Winner	2/6
Lottery	1/6

Word	Probability (P(Word | Non-Spam))
Deadline	1/3
Meeting	1/3
Promotion	1/3
Bumper	0 (Not present in Non-Spam docs)

Calculation for Spam:

$$P(Spam|NewDocument) \propto P(Spam) \times P(Promotion|Spam) \times P(Bumper|Spam) \times P(Lottery|Spam) \propto \frac{2}{3} \times \frac{1}{6} \times \frac{2}{6} \times \frac{1}{6} = \frac{1}{54}$$

Calculation for Non-Spam:

$$P(Non-Spam|NewDocument) \propto P(Non-Spam) \times P(Promotion|Non-Spam) \times P(Bumper|Non-Spam) \times P(Lottery|Non-Spam) \propto \frac{1}{3} \times \frac{1}{3} \times 0 \times (\text{unknown probability for Lottery}) = 0$$

Since $P(Bumper|Non-Spam) = 0$, the probability $P(Non-Spam|NewDocument)$ will also be zero.

Example

Ex	Size	Color	Shape	Category
1	small	red	circle	positive
2	large	red	circle	positive
3	small	red	triangle	negitive
4	large	blue	circle	negitive

Test Instance X:
<medium, red, circle> -> ?

Probability	positive	negative
P(Y)	0.5	0.5
P(small | Y)	0.5	0.5
P(medium | Y)	0.0	0.0
P(large | Y)	0.5	0.5
P(red | Y)	1.0	0.5
P(blue | Y)	0.0	0.5
P(green | Y)	0.0	0.0
P(square | Y)	0.0	0.0
P(triangle | Y)	0.0	0.5
P(circle | Y)	1.0	0.5

P(positive | X) = 0.5 * 0.0 * 1.0 * 1.0 = 0

P(negative | X) = 0.5 * 0.0 * 0.5 * 0.5 = 0

Smoothing

- To account for estimation from small samples, probability estimates are adjusted or *smoothed*.
- Laplace smoothing:
  - using an *m*-estimate assumes that each feature is given a prior probability, *p*, that is assumed to have been previously observed in a "virtual" sample of size *m*.

$$P(A_i = a_{ij} \mid Y = y_k) = \frac{n_{ijk} + mp}{n_k + m}$$

  - For binary features, *p* is simply assumed to be 0.5.

Laplace smoothing example

- Assume training set contains 10 positive examples:
  - 4: small
  - 0: medium
  - 6: large
- Estimate parameters as follows (if m=1, p=1/3)
  - P(small | positive) = (4 + 1/3) / (10 + 1) = 0.394
  - P(medium | positive) = (0 + 1/3) / (10 + 1) = 0.03
  - P(large | positive) = (6 + 1/3) / (10 + 1) = 0.576
  - P(small $\vee$ medium $\vee$ large | positive) = 1.0

Outline: Text Data Mining

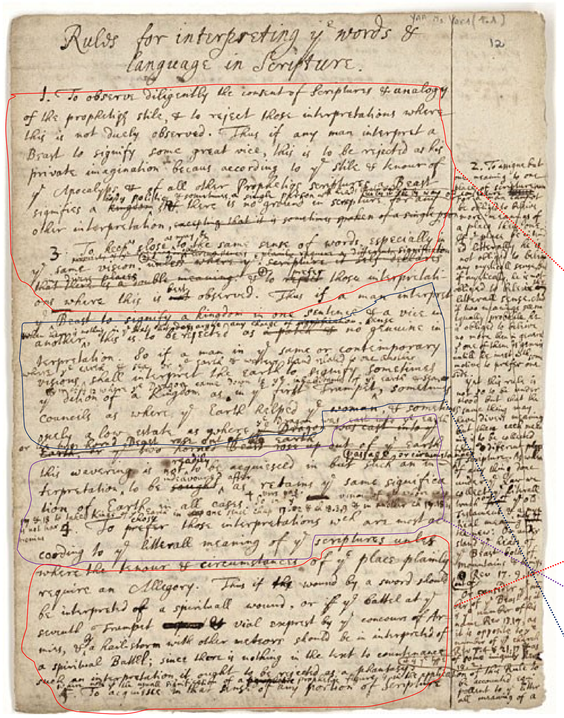
1. Introduction to Text Mining
2. Vector Space Model
3. Text Classification
4. Probabilistic Topic Models

Motivation

Suppose you're given a massive corpora and asked to carry out the following tasks

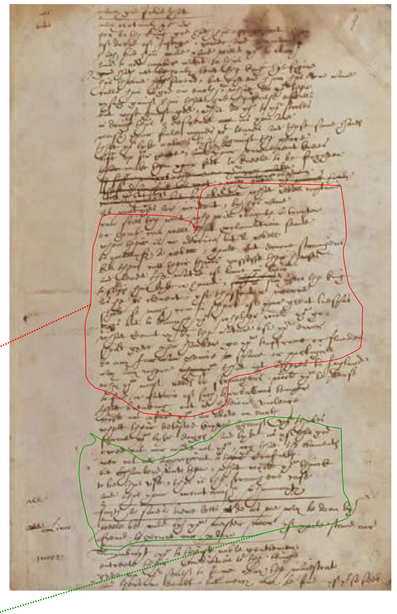
- **Organize** the documents into **thematic categories**
- **Describe** the evolution of those categories **over time**
- Enable a domain expert to **analyze and understand** the content
- Find **relationships** between the categories
- Understand how **authorship** influences the content

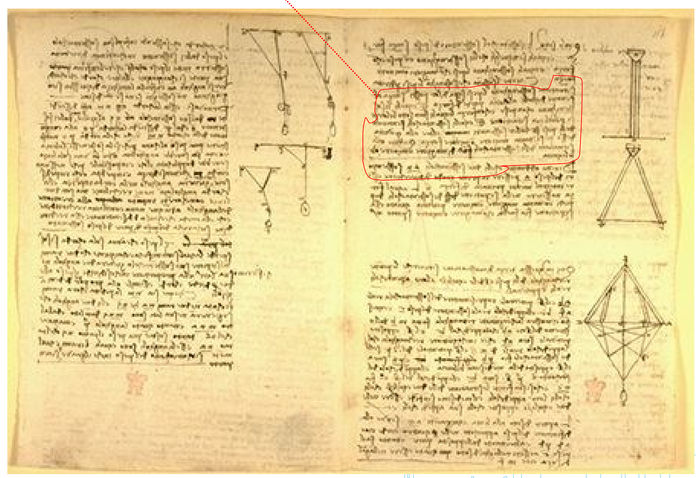
What is a “topic”



Representation: a probabilistic distribution over words.

retrieval	0.2
information	0.15
model	0.08
query	0.07
language	0.06
feedback	0.03
……	

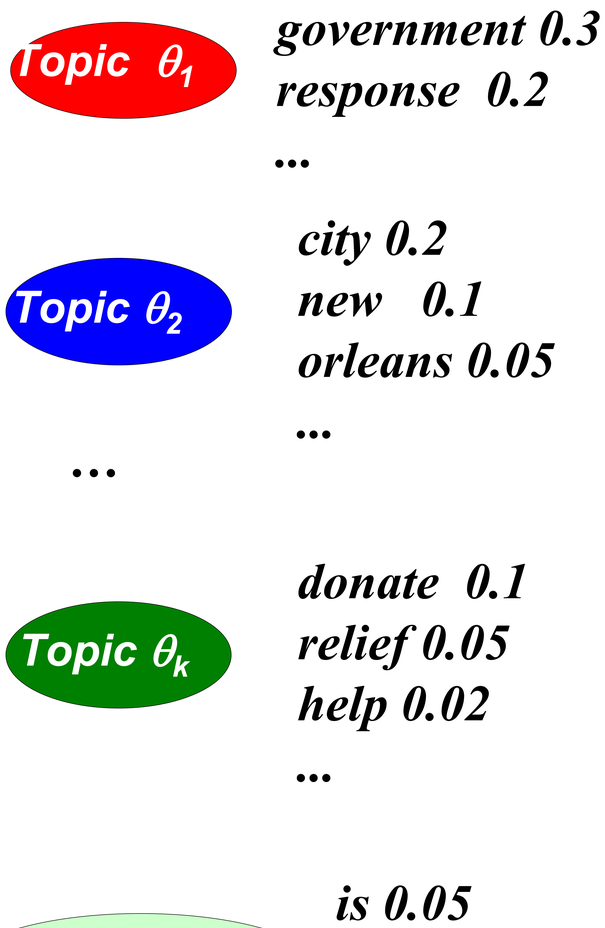




Topic: A broad concept/theme, semantically coherent, which is *hidden* in documents

e.g., politics; sports; technology; entertainment; education etc.

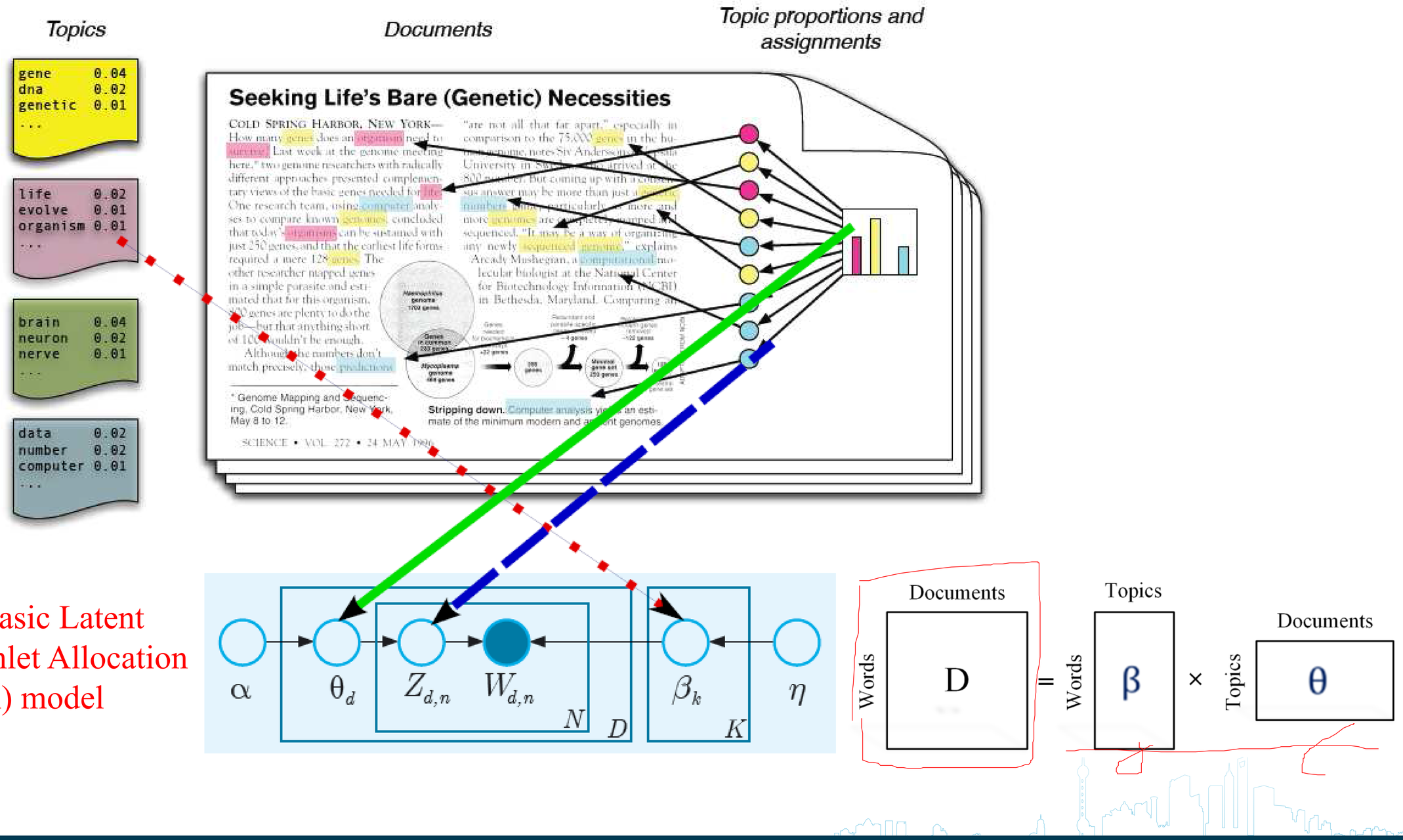
Document as a mixture of topics



[Criticism of government response to the hurricane primarily consisted of criticism of its response to the approach of the storm and its aftermath, specifically in the delayed response] to the *[flooding of New Orleans. … 80% of the 1.3 million residents of the greater New Orleans metropolitan area evacuated]* …*[Over seventy countries pledged monetary donations or other assistance]*. …

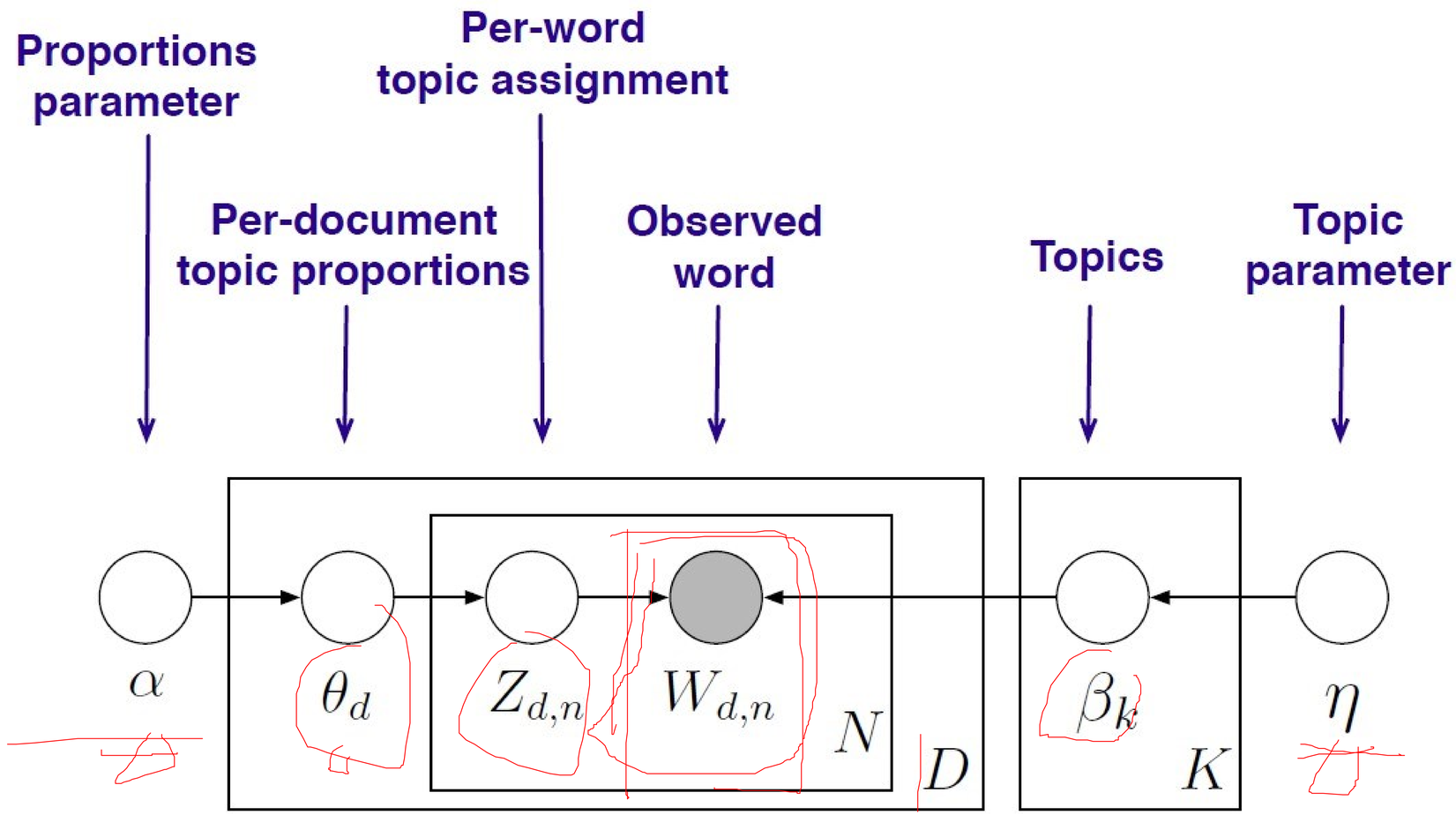
- How can we discover these topic-word distributions?
- Many applications would be enabled by discovering such topics
  - Summarize themes/aspects
  - Facilitate navigation/browsing
  - Retrieve documents
  - Segment documents
  - Many other text mining tasks

Latent Dirichlet Allocation (LDA)



The basic Latent Dirichlet Allocation (LDA) model

LDA



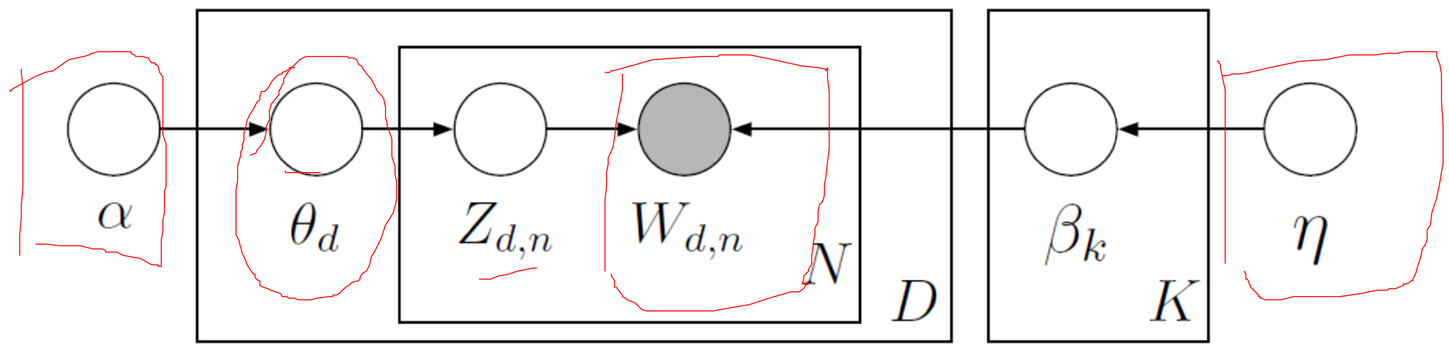
$\theta_d \sim Dirichlet\,(\alpha)$: **address topic distribution for unseen documents**

$\beta_k \sim Dirichlet\,(\boldsymbol{\eta})$: **smoothing over words**

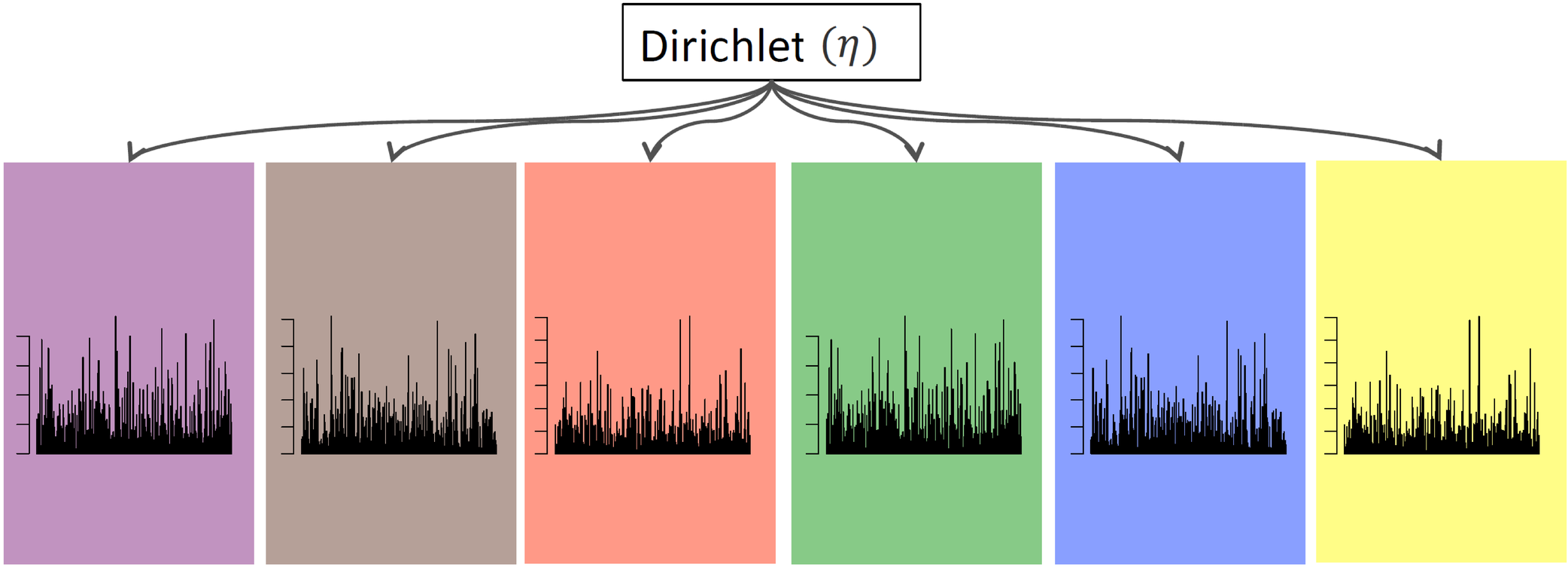
LDA

Generative Model for LDA

For each topic $k \in \{1, \ldots, K\}$:
  $\beta_k \sim \text{Dir}(\eta)$ [*draw distribution over words*]
For each document $d \in \{1, \ldots, D\}$
  $\boldsymbol{\theta}_d \sim \text{Dir}(\boldsymbol{\alpha})$ [*draw distribution over topics*]
  For each word $n \in \{1, \ldots, N_d\}$
    $z_{d,n} \sim \text{Mult}(1, \boldsymbol{\theta}_d)$ [*draw topic assignment*]
    $w_{d,n} \sim \theta_{z_{d,n}}$ [*draw word*]

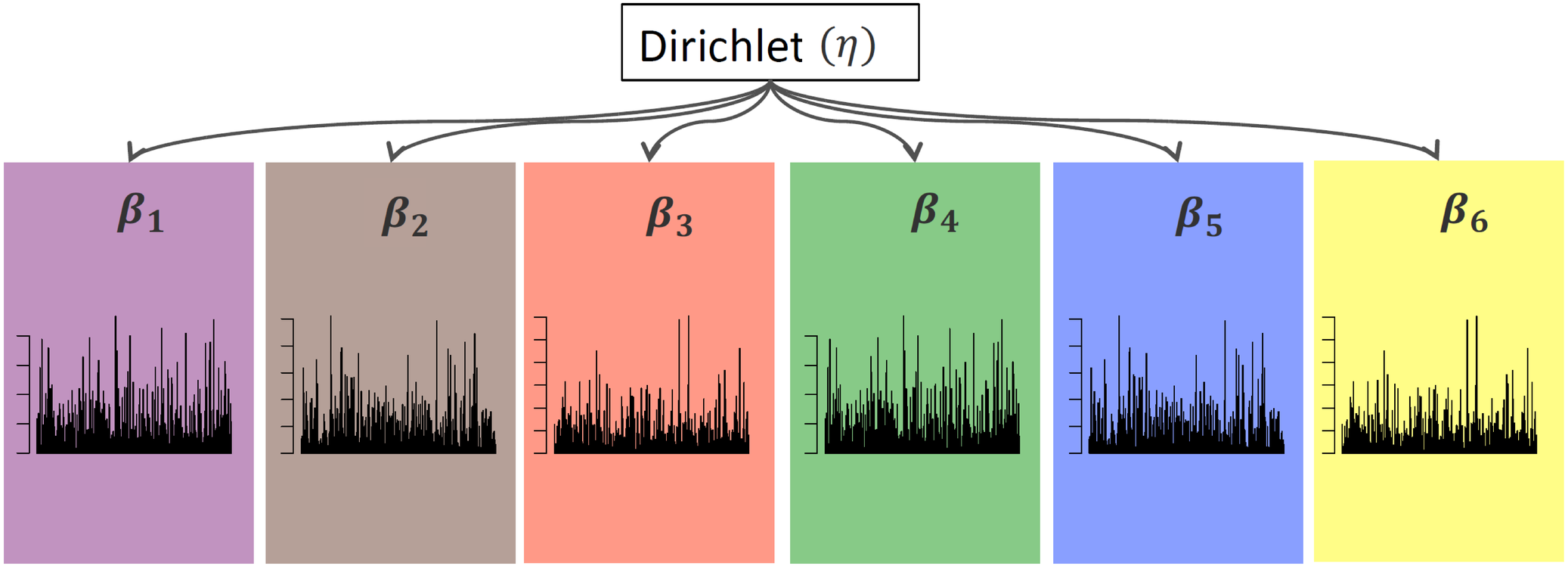


LDA for Topic Modeling



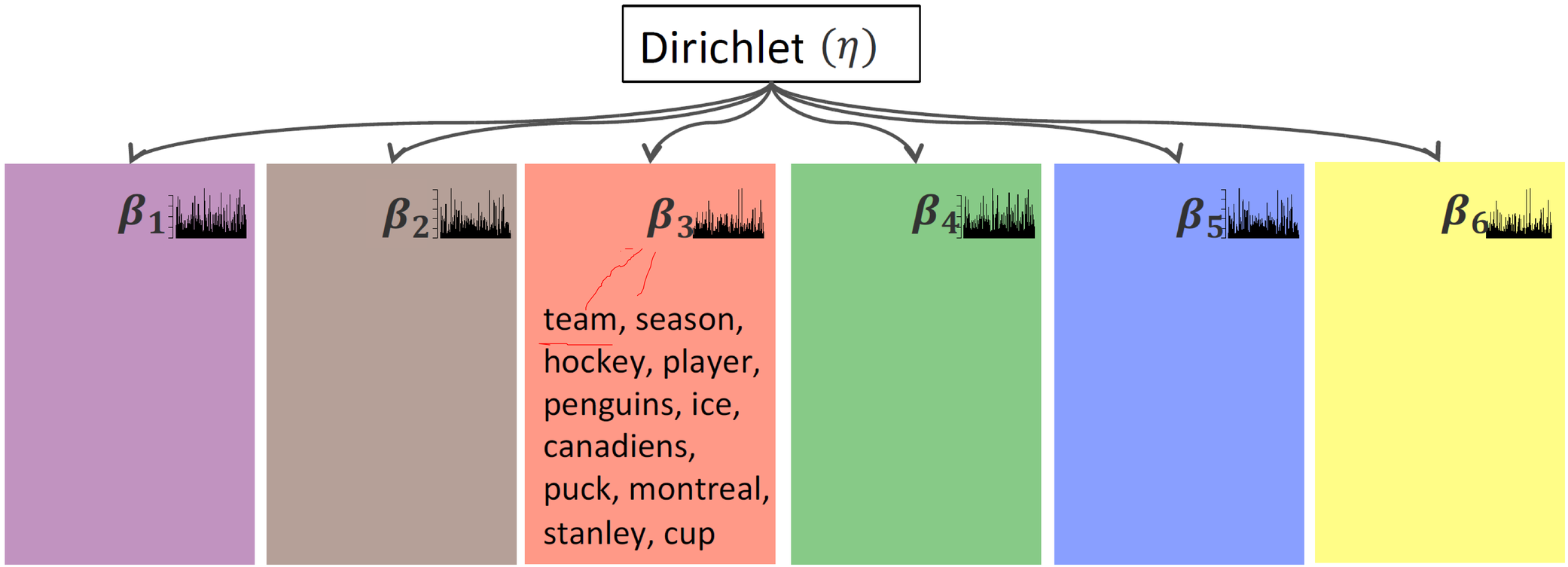
- The **generative story** begins with only a **Dirichlet prior** over the topics.
- Each **topic** is defined as a **Multinomial distribution** over the vocabulary, parameterized by $\boldsymbol{\beta_k}$

LDA for Topic Modeling



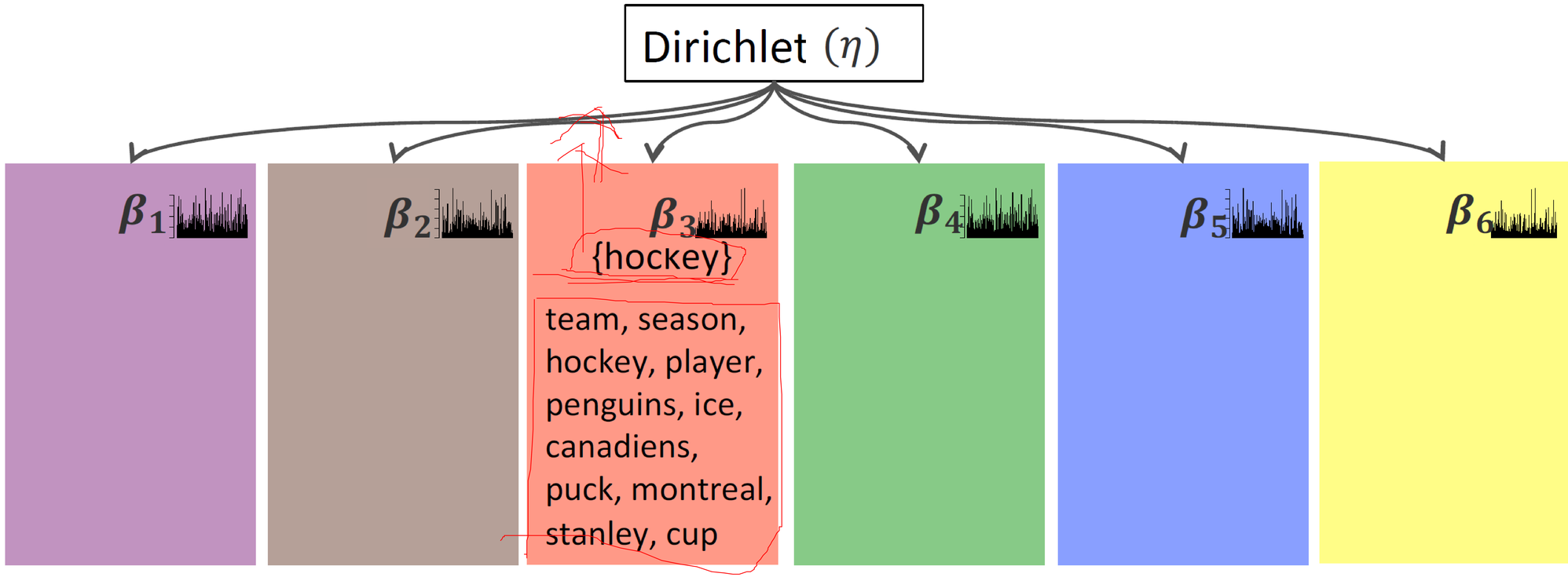
- The **generative story** begins with only a **Dirichlet prior** over the topics.
- Each **topic** is defined as a **Multinomial distribution** over the vocabulary, parameterized by β_k

LDA for Topic Modeling



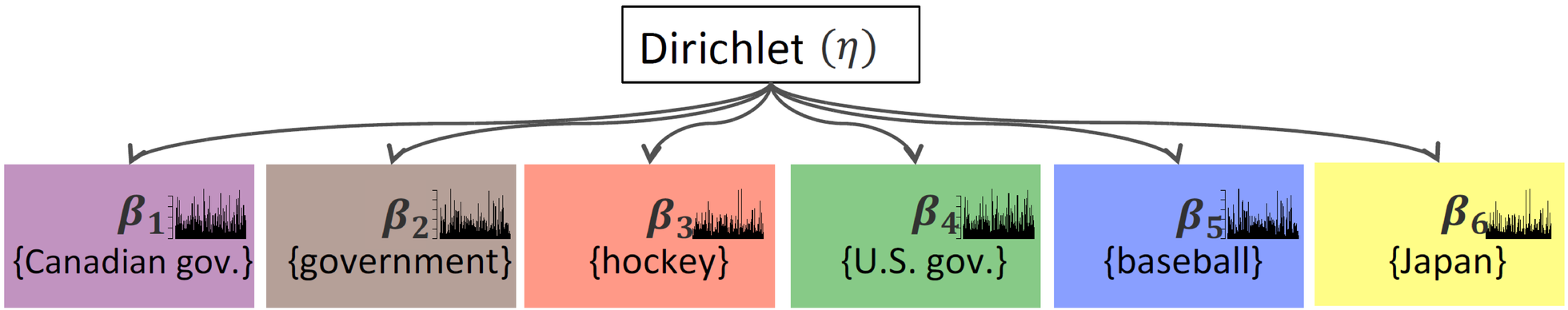
- A topic is visualized as its **high probability words**.

LDA for Topic Modeling



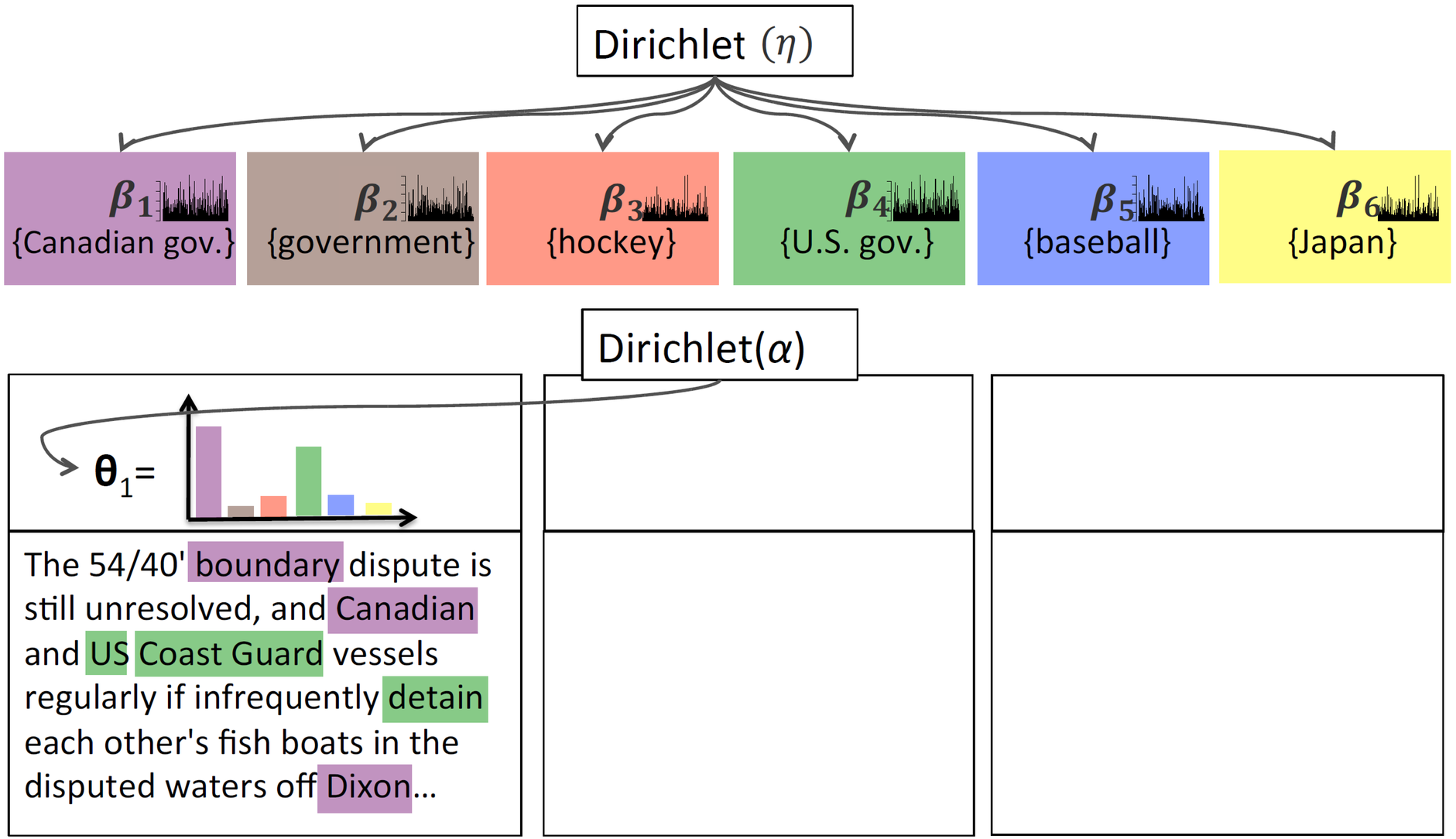
- A topic is visualized as its **high probability words**.
- A pedagogical **label** is used to identify the topic.

LDA for Topic Modeling

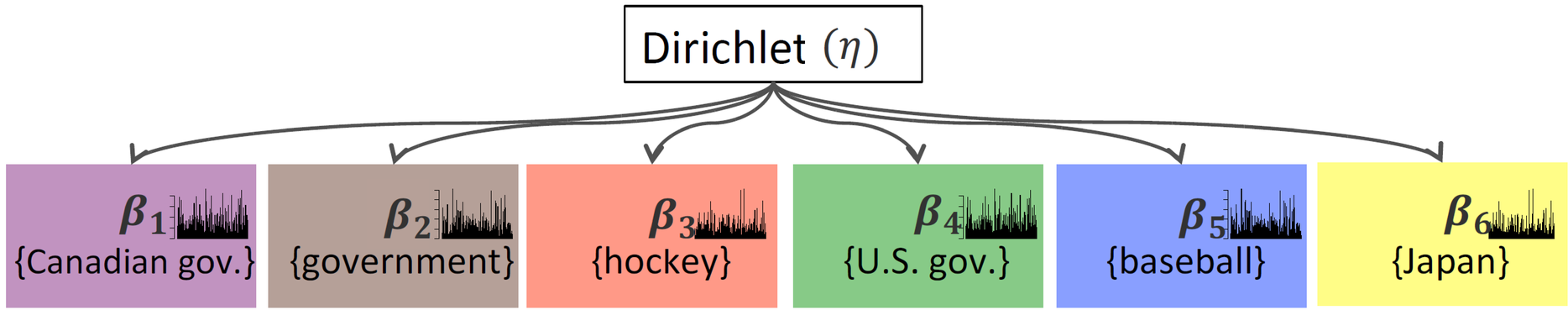


- A topic is visualized as its high probability words.
- A pedagogical **label** is used to identify the topic.

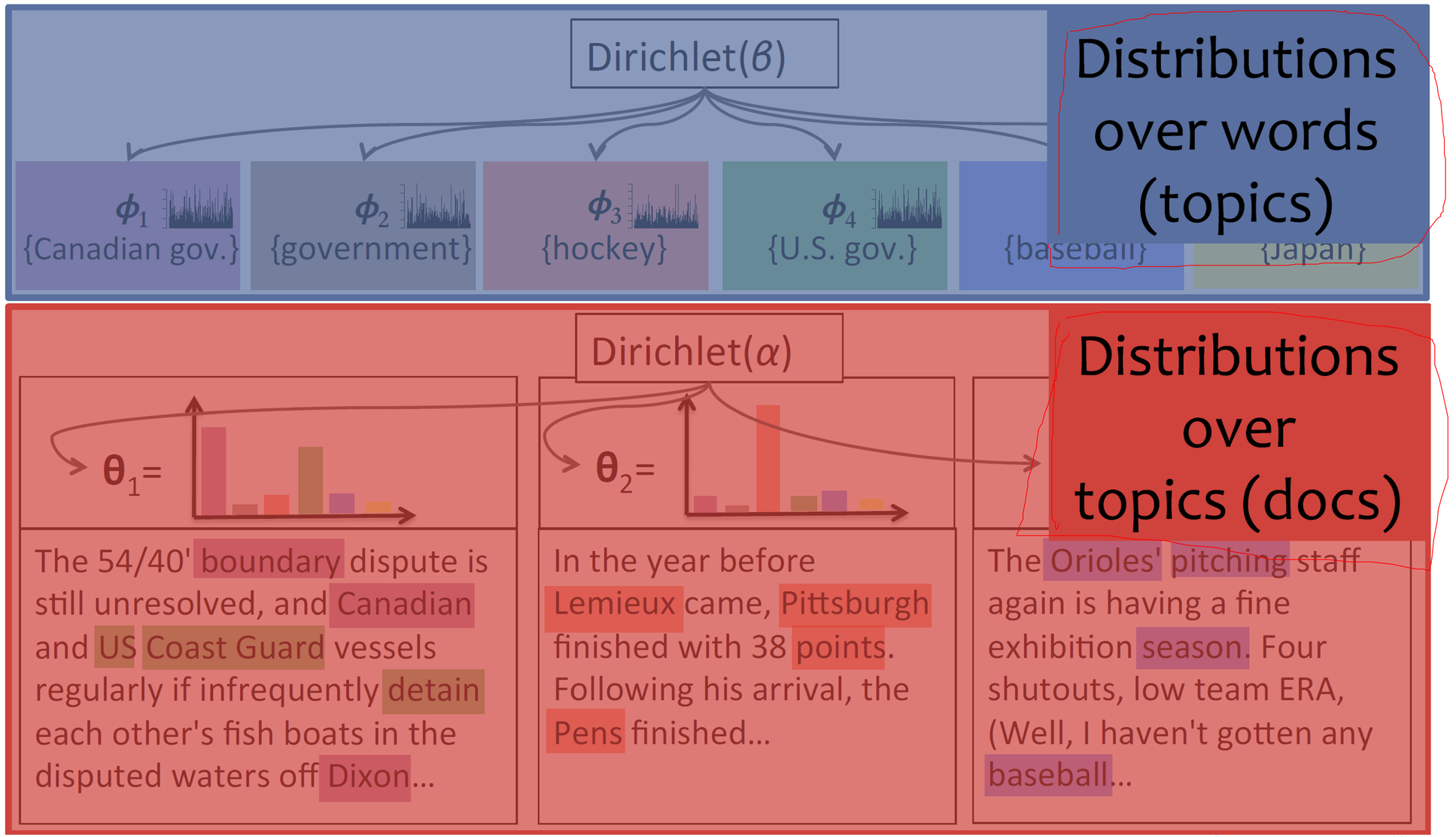
LDA for Topic Modeling



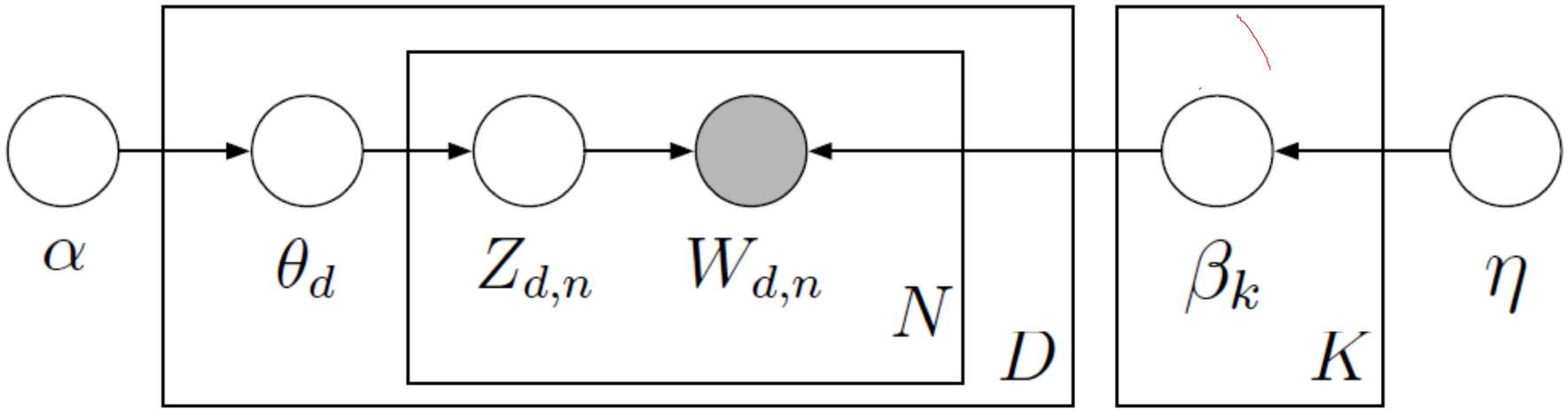
LDA for Topic Modeling



LDA for Topic Modeling



Joint distribution for LDA



- Joint distribution of latent variables and documents is:

$$p(\boldsymbol{\beta}_{1:K}, \boldsymbol{z}_{1:D}, \boldsymbol{\theta}_{1:D}, \boldsymbol{w}_{1:D} | \alpha, \eta) =$$

$$\prod_{i=1}^{K} p(\beta_i | \eta) \prod_{d=1}^{D} p(\theta_d | \alpha) \left(\prod_{n=1}^{N} p(z_{d,n} | \theta_d) p(w_{d,n} | \beta_{1:K}, z_{d,n}) \right)$$

LDA has been widely-used

[HTML] Discovering topics and trends in the field of Artificial Intelligence: Using **LDA** topic **modeling**

D Yu, B Xiang - Expert systems with applications, 2023 - Elsevier

… 2021, and applies the Latent Dirichlet allocation (**LDA**) **model** to extract the 40 topics from the abstracts… This study aggregates the results of the **LDA model** from the perspectives of year, …

☆ Save 99 Cite Cited by 50 Related articles All 2 versions

Technological topic analysis of standard-essential patents based on the improved Latent Dirichlet Allocation (**LDA**) **model**

C Tian, J Zhang, D Liu, Q Wang… - Technology Analysis & …, 2024 - Taylor & Francis

… of topics for the **LDA model** more accurately and thus improve the accuracy of technological topic analysis. Specifically, we apply the improved **LDA model** and visualisation to obtain …

☆ Save 99 Cite Cited by 11 Related articles Web of Science: 4

Topic Modelling and Opinion Analysis On Climate Change Twitter Data Using **LDA** And BERT **Model**.

SE Uthirapathy, D Sandanam - Procedia Computer Science, 2023 - Elsevier

… Using the well-known topic modelling technique **LDA**, this work investigates the various climate change topics discussed by the public via tweets. The BERT uncased **model** is then …

☆ Save 99 Cite Cited by 40 Related articles All 2 versions

Choosing the Number of Topics in **LDA Models**–A Monte Carlo Comparison of Selection Criteria

V Bystrov, V Naboka-Krell… - Journal of Machine …, 2024 - jmlr.org

… BIC that can be applied to singular statistical **models**. The comparison is based on Monte … for **LDA model** selection in applications are derived. Keywords: Topic **models**, text analysis, …

☆ Save 99 Cite Cited by 3 Related articles All 4 versions »

[HTML] An association-constrained **LDA model** for joint extraction of product aspects and opinions

C Wan, Y Peng, K Xiao, X Liu, T Jiang, D Liu - Information Sciences, 2020 - Elsevier

… Different from classic **LDA models** which are only based on a priori probability distribution, we develop an Association Constrained **LDA** (AC-**LDA**) **model**. Our **model** allows the …

☆ Save 99 Cite Cited by 41 Related articles All 2 versions Web of Science: 25

LDA-based topic **modeling** sentiment analysis using topic/document/sentence (TDS) **model**

A Farkhod, A Abdusalomov, F Makhmudov, YI Cho - Applied Sciences, 2021 - mdpi.com

… is based on joint sentiment topic (JST) and latent Dirichlet allocation (**LDA**) topic **modeling** … First, we applied the **LDA model** to discover topics from the reviews; then, the TDS **model** was …

☆ Save 99 Cite Cited by 54 Related articles All 7 versions Web of Science: 24 »

Applying **LDA** topic **modeling** in communication research: Toward a valid and reliable methodology

D Maier, A Waldherr, P Miltner… - Computational …, 2021 - taylorfrancis.com

… In summary, **LDA models** draw on an abstract hypothetical probabilistic process that implies different assumptions. It has proved to be a powerful approach to quickly identify major …

☆ Save 99 Cite Cited by 761 Related articles All 12 versions Web of Science: 393 »

Outline: Text Data Mining

1. Introduction to Text Mining
2. Vector Space Model
3. Text Classification
4. Probabilistic Topic Models

Email: min.shi@louisiana.edu