Session 9

Text Mining in Marketing

Alain Lemaire

Text Mining, Automated text analysis is a tool for discovery and measurement in textual data of prevalent attitudes, concepts, or events.

O’connor, Bamman & Smith 2011

Text Mining, Automated text analysis is a **tool** for **discovery and measurement** in textual data of prevalent **attitudes**, concepts, or events.

O'connor, Bamman & Smith 2011

Steps


Acquiring Text


Representing Text


Analyzing Text


Validating Results


Managing Data

Steps

Acquiring Text

Validating Results

Representing Text

Managing Data

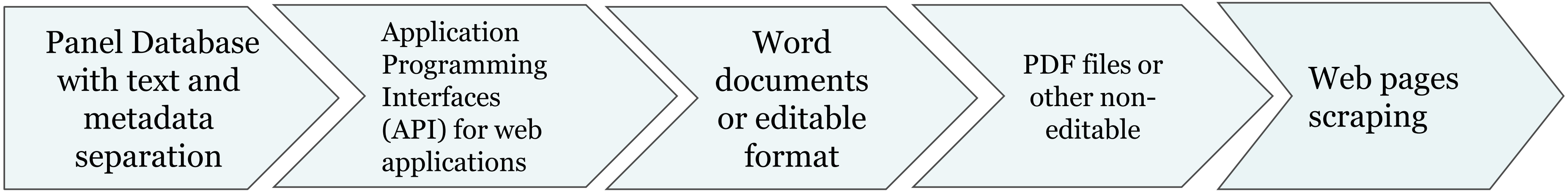
Analyzing Text


KAREN ROBARDS
PHILIP ROTH

Acquiring Text

Data collection……………..

Sources


Panel Database with text and metadata separation
Application Programming Interfaces (API) for web applications
Word documents or editable format
PDF files or other non-editable
Web pages scraping

Sources

- Panel Database with text and metadata separation
- Application Programming Interfaces (API) for web applications
- Word documents or editable format
- PDF files or other non-editable
- Web pages scraping

Increasing degree of difficulty....

Panel Database with text and metadata separation → Application Programming Interfaces (API) for web applications → Word documents or editable format → PDF files or other non-editable → Web pages scraping

Pros

Well organized and ready for analysis

Easily accessible

No Barrier to entry

Cons

Hard to obtain

Panel Database with text and metadata separation → **Application Programming Interfaces (API) for web applications** → Word documents or editable format → PDF files or other non-editable → Web pages scraping

Pros

Well organized and ready for analysis

Easily accessible ;but …

Cons

Beginner’s programing level experience

Rate limit requirement

- Panel Database with text and metadata separation
- Application Programming Interfaces (API) for web applications
- Word documents or editable format
- PDF files or other non-editable
- Web pages scraping

Pros

Easy to view, manage, and format

Cons

Data cleaning required(time intensive)

Intermediate level Programming experience

Panel Database with text and metadata separation → Application Programming Interfaces (API) for web applications → Word documents or editable format → **PDF files or other non-editable** → Web pages scraping

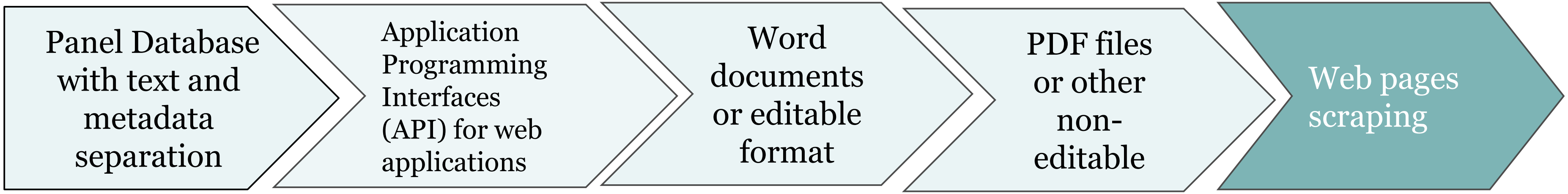
Pros

Easy to view

Cons

Data cleaning required

Intermediate level programming experience



Pros

Lots of interesting data (most interesting data) is on the web.

Cons

Require some HTML knowledge.

Lots of patience to write program.

Intermediate level programming experience.

Luck...

My Takeaway...

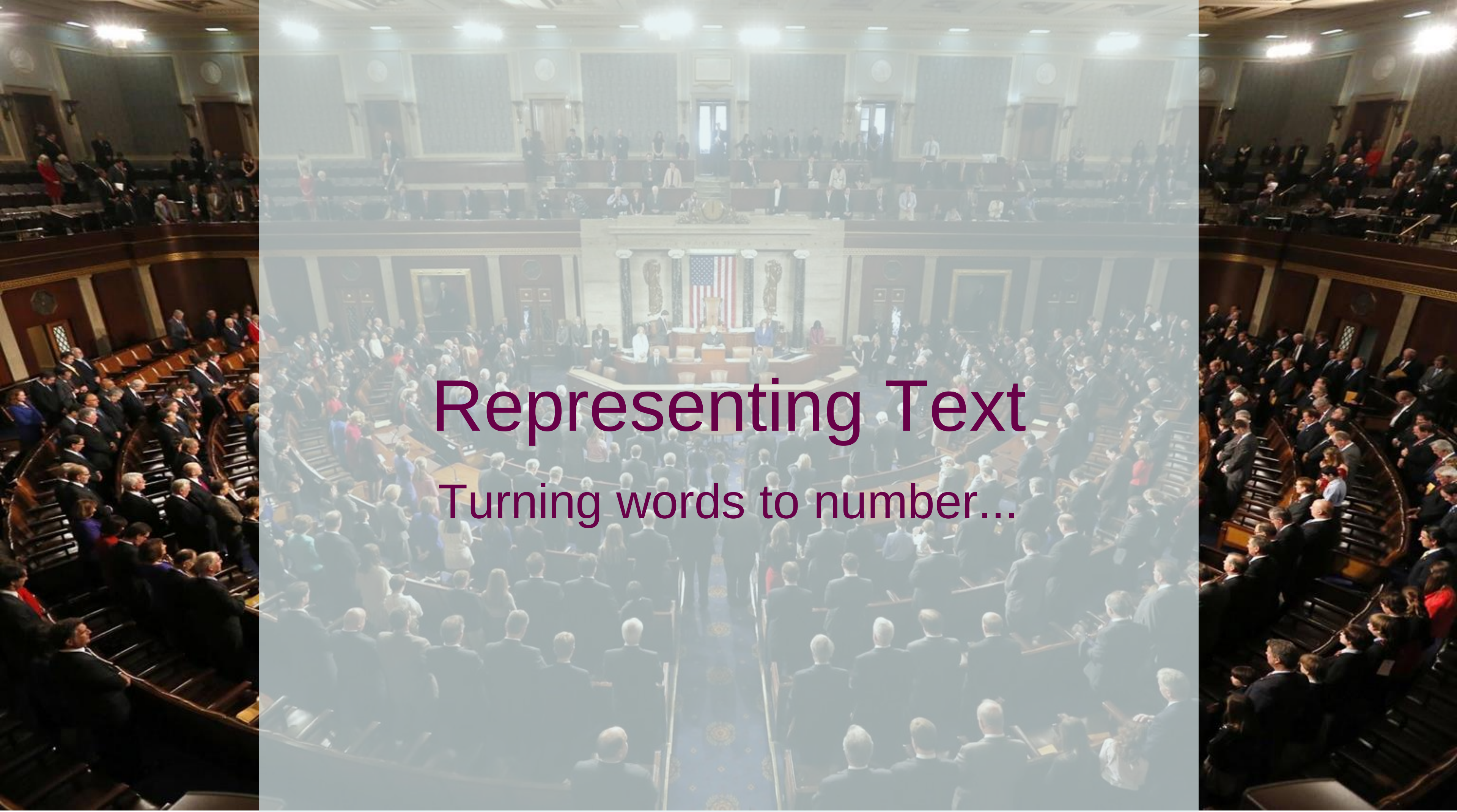
Better to be in left side of line, but more likely going to end up in the right.

Supplement I

http://www.columbia.edu/~apl2122/Phd_Scraping.html

Representing Text

Turning words to number...



Text Processing Steps

1 Tokenizing

2 Stopwords Removal

3 Lemmatization, Stemming

4 Convert data to numbers

Text Processing Steps

1. Break statements in tokens
2. Decide which tokens you want to keep
3. Decide whether and how you want to transform tokens
4. Convert data to numbers

Tokenization

Tokenizing a string is breaking it up into its linguistic elements (tokens): words, number, punctuation.

Stopword Removal

Stopwords are common words(the, a) that we remove from text as part of pre-processing. These words can be found online and are modifiable.

Linguistic Roots

Stemming :is the act of reducing a word (tokens) to its roots

Ex: Worked, working,works => work

Conversion

Word Counts

Term Frequency

$$\text{TF-IDF} = \text{TF} * \log\left(\frac{N}{n_t}\right)$$

Illustration

Examples...

5/10/2015

Eduardo A.
PACOIMA, CA
12 friends
29 reviews

Share review
Compliment
Send message
Follow Eduardo A.

I'm visiting this state with part of my family, and I wanted to give a try to this place. Attention is super great, the environment make us feel very comfortable, the service received was exceptional. Food is a mix of central american, dominican and the taste of New York of course.

Tokenization

tokens tokens tokens tokens tokens tokens tokens tokens

I'm | visiting | this | state | with | part | of | my | family, | | and | I
| wanted | to | give | a | try | to | this | place. | Attention | is |
super | great, | the | environment | make | us | feel | very |
comfortable, | the | service | received | was | exceptional. |
Food | is | a | mix | of | central | american, | dominican | and |
the | taste | of | New | York | of | course.

Stopwords Removal

stopwords stopwords stopwords

~~I'm~~ | visiting | ~~this~~ | state | ~~with~~ | part | ~~of~~ | ~~my~~ | family, | | ~~and~~ | ~~I~~ | wanted | ~~to~~ | give | ~~a~~ | try | ~~to~~ | ~~this~~ | place. | Attention | ~~is~~ | super | great, | ~~the~~ | environment | make | us | feel | ~~very~~ | comfortable, | ~~the~~ | service | received | ~~was~~ | exceptional. | Food | ~~is~~ | ~~a~~ | mix | ~~of~~ | central | american, | dominican | ~~and~~ | ~~the~~ | taste | ~~of~~ | New | York | ~~of~~ | course.

visiting | state | part |family, | wanted | give | try |place. |
Attention | super | great, | environment | make | us | feel |
comfortable, | service | received | exceptional. | Food | mix |
central | american, | dominican | taste | New | York | course.

Stemming

visit~~ing~~ | state | part |famili, | want~~ed~~ | give | tri |place. |
Attent~~ion~~ | super | great, | environ~~ment~~ | make | us | feel |
comfort~~able,~~ | servic~~e~~ | receiv~~ed~~ | except~~ional.~~ | Food | mix |
central | american, | dominican | tast~~e~~ | New | York | cours~~e~~.

Conversion

Turning words to number...

visit | state | part |famili, | want | give | tri |place. | Attent | super | great, | environ | make | us | feel | comfort | servic | receiv | except | Food | mix | central | american, | dominican | tast | New | York | cours.

visit	:	1
state	:	1
part	:	1
famili	:	1
want	:	1
give	:	1
tri	:	1
place	:	1
attent	:	1
super	:	1
great	:	1
environ	:	1
make	:	1
us	:	1

feel	:	1
comfort	:	1
servic	:	1
receiv	:	1
except	:	1
food	:	1
mix	:	1
central	:	1
american	:	1
dominican	:	1
tast	:	1
new	:	1
york	:	1
cours	:	1

visit :	1/28=.0357
state :	0.0357
part :	0.0357
famili :	0.0357
want :	0.0357
give :	0.0357
tri :	0.0357
place :	0.0357
attent :	0.0357
super :	0.0357
great :	0.0357
environ :	0.0357
make :	0.0357
us :	0.0357
feel :	0.0357
comfort:	0.0357
servic :	0.0357
receiv :	0.0357
except :	0.0357
food :	0.0357
mix :	0.0357
central :	0.0357
american:	0.0357
dominican:	0.0357
tast :	0.0357
new :	0.0357
york :	0.0357
cours :	0.0357

Term-Document-Matrix

	tast	new	york	cours	mix	food	receiv	famili	great	super
Eduardo	1	1	1	1	1	1	1	1	1	1
Doc 2										
Doc 3										

	tast	new	york	cours	mix	food	receiv	famili	great	super
Doc 1	1	1	1	1	1	1	1	1	1	1
Doc 2	2	0	0	1	5	6	1	0	0	0
Doc 3	0	0	1	2	0	1	0	3	1	0

	tast	new	york	cours	mix	food	receiv	famili	great	super
Doc 1	1	1	1	1	1	1	1	1	1	1
Doc 2	2	0	0	1	5	6	1	0	0	0
Doc 3	0	0	1	2	0	1	0	3	1	0

**Assume Doc2 has 30 words , doc 3 has 15 words

	tast	new	york	cours	mix	food	receiv	famili	great	super
Doc 1	.0357	.0357	.0357	.0357	.0357	.0357	.0357	.0357	.0357	.0357
Doc 2	.0667	0	0	.0333	.1666	.2	.0333	0	0	0
Doc 3	0	0	.0667	.1333	0	.1333	0	.2	.0667	0

**Assume Doc 2 has 30 words , doc 3 has 15 words

	tast	new	york	cours	mix	food	receiv	famili	great	super
Doc 1	.0357*log(3/2)	.0392	.01447	0	.01447	0	.01447	.01447	.01447	.0392
Doc 2	.00270	0	0	0	.06759	0	.01350	0	0	0
Doc 3	0	0	.0270	0	0	0	0	.08109	.02704	0

**Assume Doc 2 has 30 words , doc 3 has 15 words

My Takeaway...

1)Tokenizing is the only required steps, all the other steps are optional and dependent on your research question.
2) there are packages, that will performed those steps for you.
3) These steps creates large and sparse Term Document Matrices

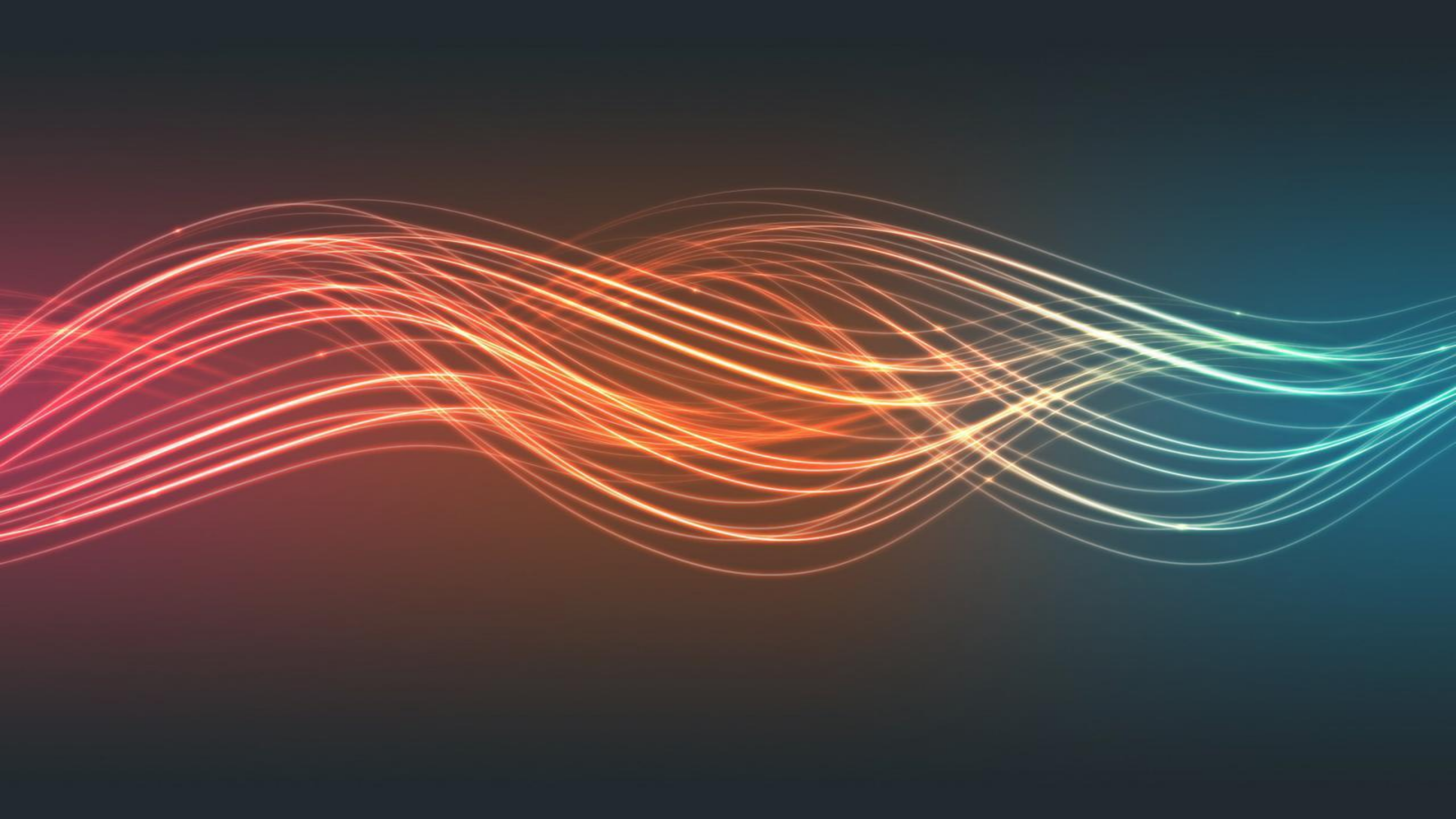
Data science is 80% cleaning, 20% analysis.

unknown

Data science is 75% cleaning, 10% analysis, 15% presentation

Supplement II

http://www.columbia.edu/~apl2122/text_analysis.html



Processing Text

Analysis...

Tools

- Marketing Research:
  - PCA
  - Cluster Analysis
  - Linear Regression
  - Logistic Regression
  - Discriminant Analysis

Tools

- Machine Learning:
  - Latent Dirichlet Allocation
  - Cluster Analysis
  - Penalized Linear Regression
  - Penalized Logistic Regression or Support Vector Machine
  - Multinomial Naive Bayes

Tools

- Machine Learning:
    - **Latent Dirichlet Allocation**
    - Cluster Analysis
    - Penalized Linear Regression
    - Penalized Logistic Regression or Support Vector Machine
    - Multinomial Naive Bayes

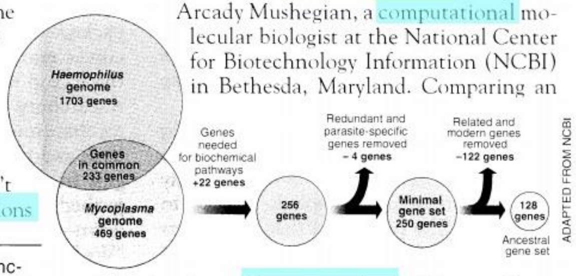
Latent Dirichlet Allocation

Slides taken From David Blei

Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK— How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

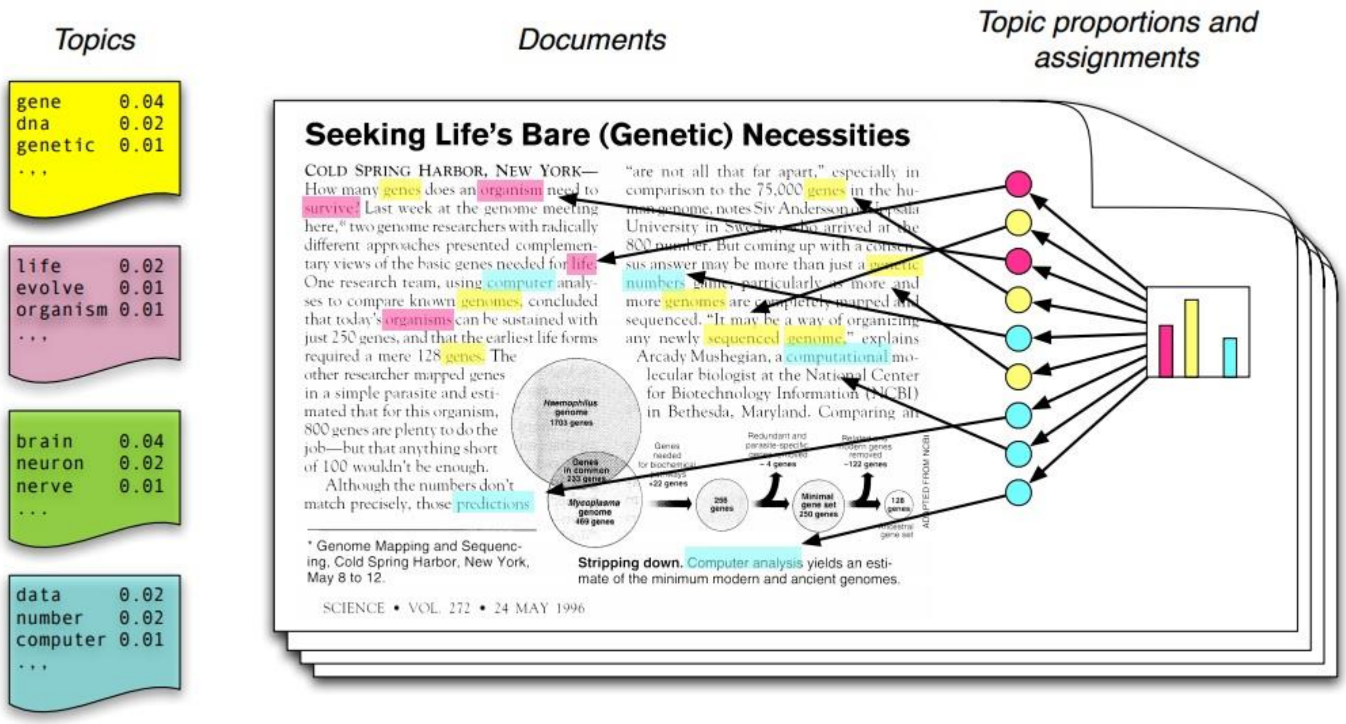
Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an



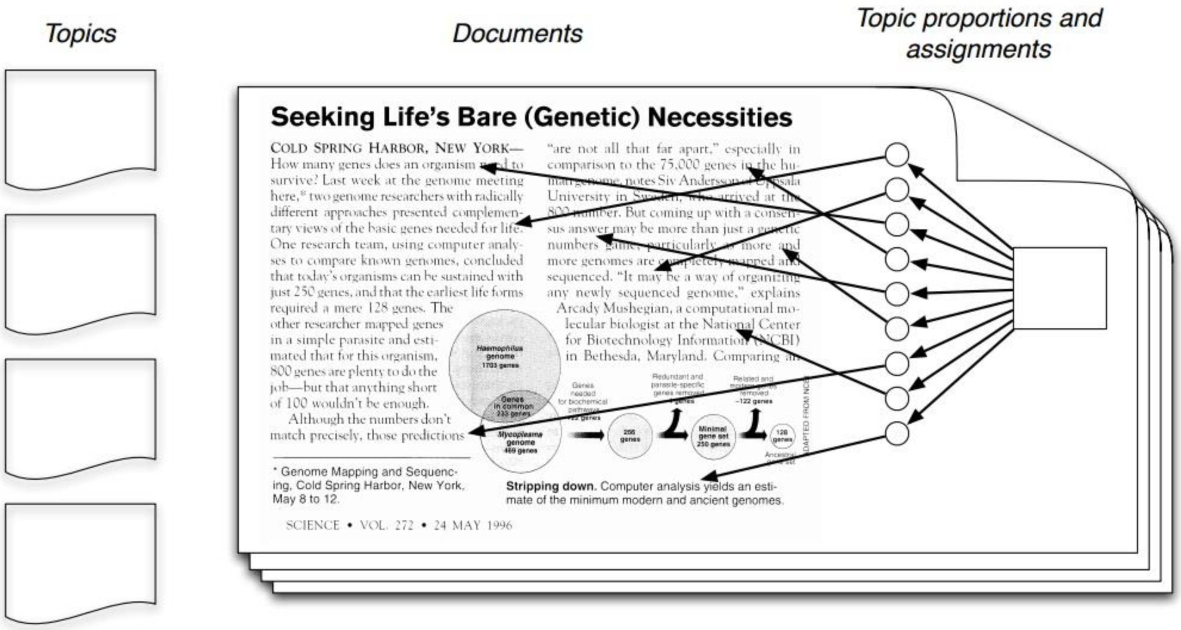
Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

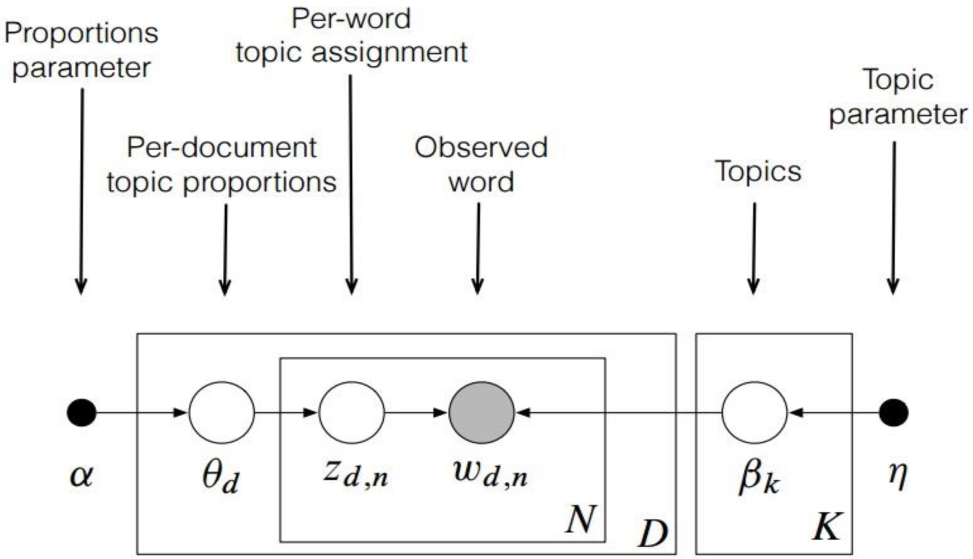
Documents exhibit multiple topics.



Latent Dirichlet Allocation

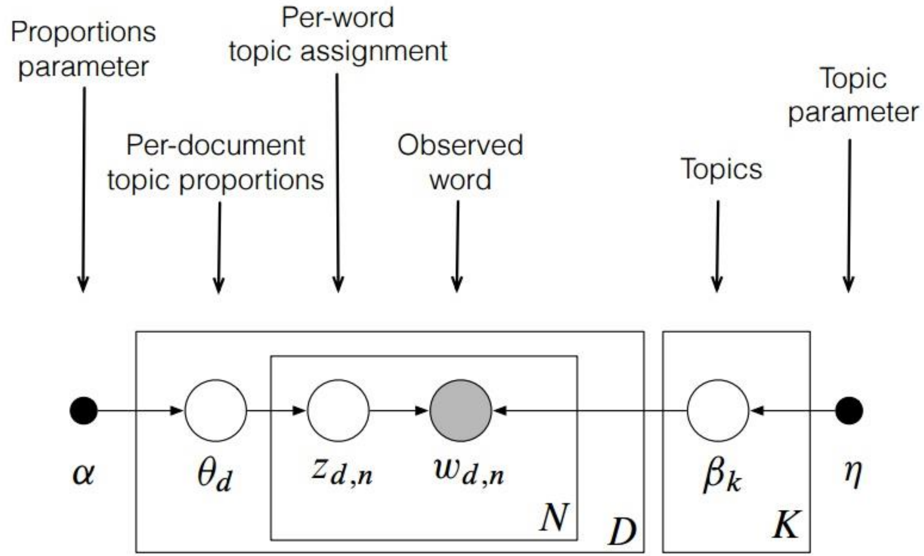


Latent Dirichlet Allocation



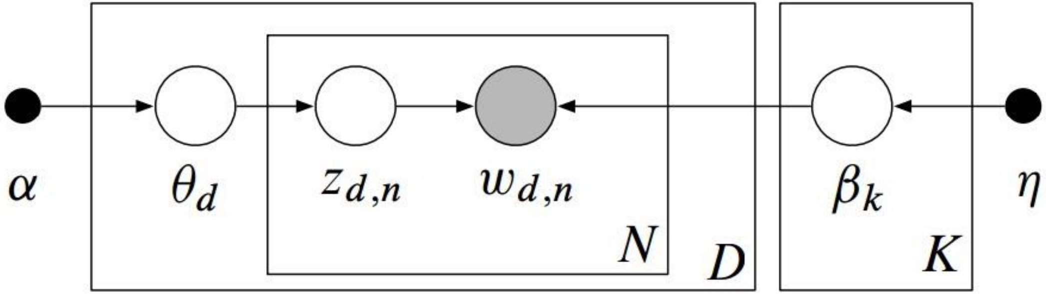
LDA as a graphical model

- Nodes are random variables; edges indicate dependence.
- Shaded nodes are observed; unshaded nodes are hidden.
- Plates indicate replicated variables.



LDA as a graphical model

- Encodes independence assumptions about the variables
- Defines a factorization of the joint probability distribution
- Connects to algorithms for computing with data



- The joint defines a posterior, $p(\theta, z, \beta \mid w)$.
- From a collection of documents, infer
  - Per-word topic assignment $z_{d,n}$
  - Per-document topic proportions θ_d
  - Per-corpus topic distributions β_k
- Then use posterior expectations to perform the task at hand: information retrieval, document similarity, exploration, and others.

How does LDA "work"?

- LDA trades off two goals.
  1. In each **document**, allocate its words to **few topics**.
  2. In each **topic**, assign high probability to **few terms**.
- These goals are at odds.
  - Putting a document in a single topic makes #2 hard:
    All of its words must have probability under that topic.
  - Putting very few words in each topic makes #1 hard:
    To cover a document's words, it must assign many topics to it.
- Trading off these goals finds groups of tightly co-occurring words.

Collapsed Gibbs Sampling

Input: words $\mathbf{w} \in$ documents $\mathbf{d}$
Output: topic assignments $\mathbf{z}$ and counts $n_{d,k}, n_{k,w}$, and n_k
begin
    randomly initialize $\mathbf{z}$ and increment counters
    foreach *iteration* **do**
        for $i = 0 \to N - 1$ **do**
            $word \leftarrow w[i]$
            $topic \leftarrow z[i]$
            $n_{d,topic}$-=1; $n_{word,topic}$-=1; n_{topic}-=1
            for $k = 0 \to K - 1$ **do**
                $p(z = k|\cdot) = (n_{d,k} + \alpha_k)\frac{n_{k,w}+\beta_w}{n_k+\beta\times W}$
            end
            $topic \leftarrow$ sample from $p(z|\cdot)$
            $z[i] \leftarrow topic$
            $n_{d,topic}$+=1; $n_{word,topic}$+=1; n_{topic}+=1
        end
    end
    return $\mathbf{z}$, $n_{d,k}, n_{k,w}, n_k$
end

Algorithm 1: LDA Gibbs Sampling

That’s all folks!
Thank You!

Collapsed Gibbs Sampling

$w_{d,n}$	Word n in document d
$z_{d,n}$	Topic allocation of word n in document d
$v_{d,n}$	Token index of word n in document d
Dirichlet Distributions	
β_k	Term distribution for each topic k
η	Dirichlet hyperparameter associated with term distributions
θ_d	Topic distribution for each document d
α	Dirichlet hyperparameter associated with topic distributions
Counts	
m_k^d	Count of words in document d allocated to topic k
m_v^k	Count of times token v is allocated to topic k
$m_{k,-n}^d$	Excluding token n, count of words in document d allocated to topic k
$m_{v,-(d,n)}^k$	Excluding token n in document d, count of times unique token v is allocated to topic k

- Compute the counts above from this naïve topic model, then for each word document pair,
  - Delete the word-doc pair from the counts, and then reallocate it into a topic randomly by the distribution

$$\Pr\left[\, z_{d,n} = k \mid z_{-(d,n)}, \mathbf{w} \,\right] \propto \frac{m_{v_{d,n},-(d,n)}^k + \eta}{\sum_{v=1}^V (m_{v,-(d,n)}^k + \eta)} \left(m_{k,-n}^d + \alpha\right)$$

  - Recount and move to the next word document pair
- Words will tend to be reassigned to topics in which they have a high relative frequency across documents