MACHINE LEARNING & PUBLIC POLICY

LECTURE 2: TEXT ANALYSIS

Professor Edward McFowland III

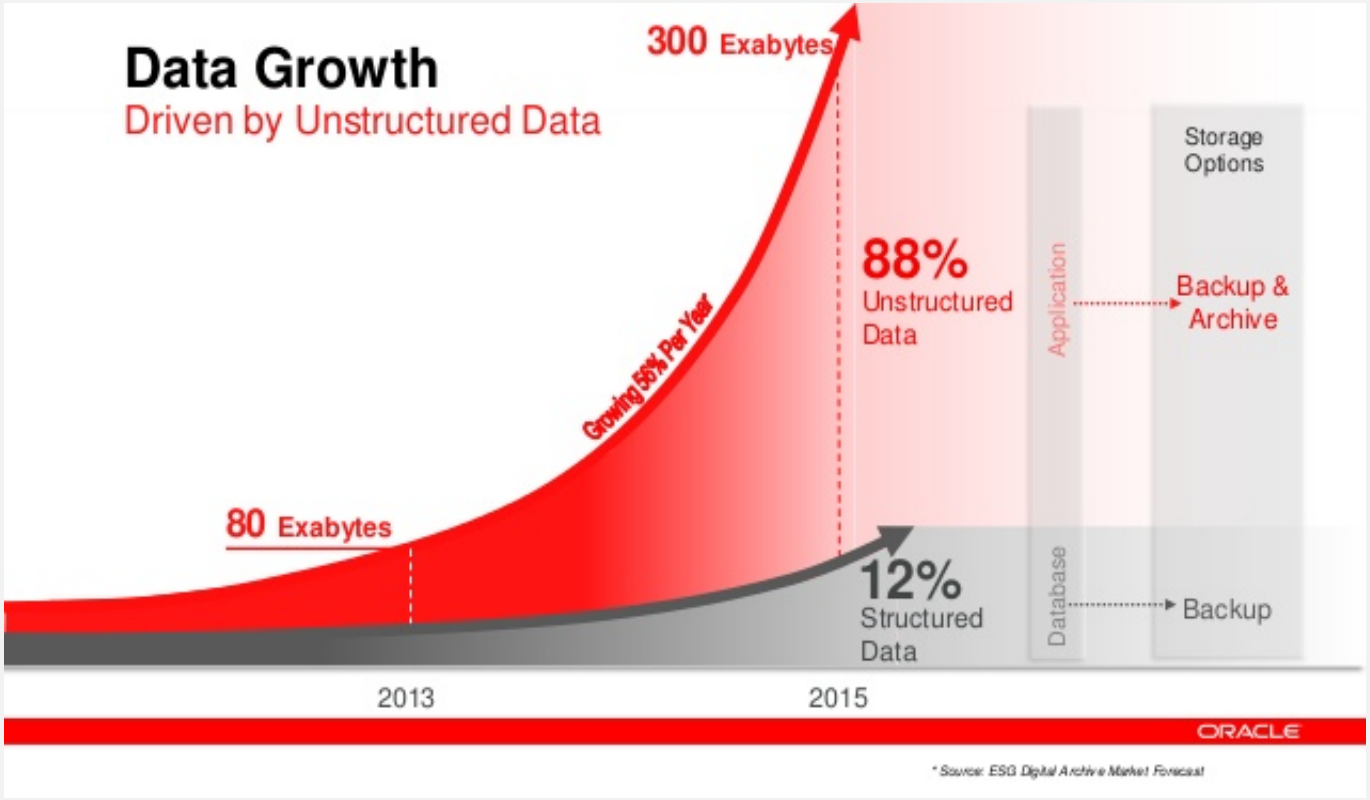
Information Systems and Decision Sciences
Carlson School of Managment

University of Minnesota

Heavily adapted from Panos Adamopoulos

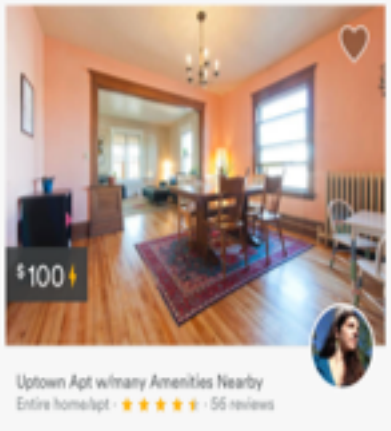
PREDICTION FOR POLICY

Observation 1: Sometimes correlation is valuable on its own


Data Growth
Driven by Unstructured Data
300 Exabytes
80 Exabytes
Growing 56% Per Year
88%
Unstructured
Data
12%
Structured
Data
Application
Database
Storage
Options
Backup &
Archive
Backup
2013
2015
ORACLE
* Source: ESG Digital Archive Market Forecast

User-Generated Content

Reviews Are a Key Element of e-Commerce

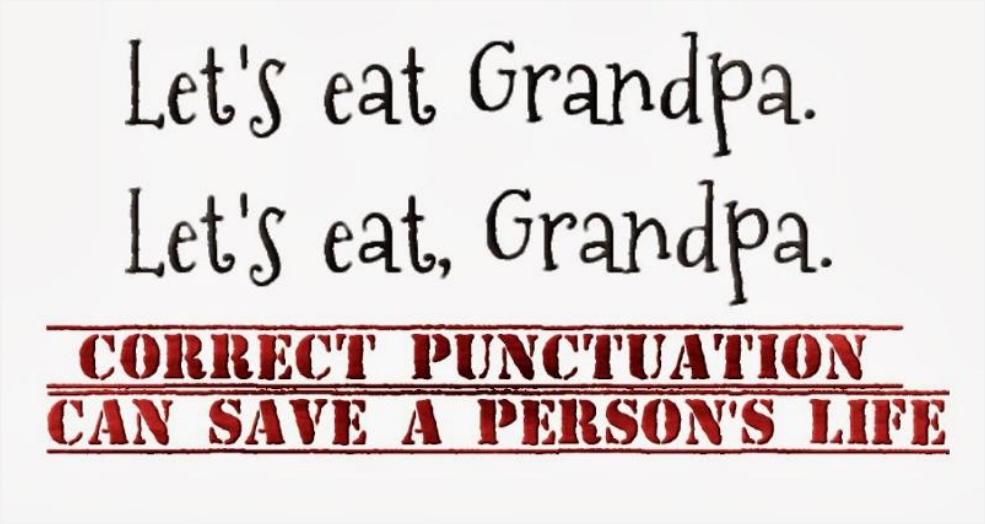


DEALING WITH TEXT

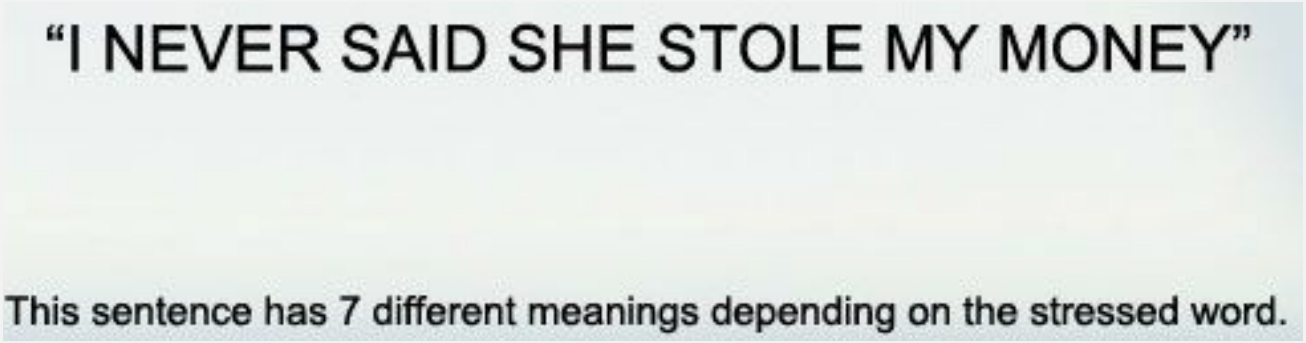
- Data are represented in ways natural to problems from which they were derived
- Vast amount of text.
- If we want to apply the many machine learning tools that we have at our disposal, we must
  - either engineer the data representation to match the tools (representation engineering), or
  - build new tools to match the data

WHY TEXT IS DIFFICULT?

WHY TEXT IS DIFFICULT?



WHY TEXT IS DIFFICULT?



WHY TEXT IS DIFFICULT?

- Word order, punctuation, and context matter
- Text is "unstructured"
  - Linguistic structure is intended for human communication and not computers
- Text can be dirty
  - People write ungrammatically, misspell words, abbreviate unpredictably, and punctuate randomly
  - synonyms, homograms, abbreviations, etc.

TEXT REPRESENTATION

- **Goal**: Take a set of documents--each of which is a relatively free-form sequence of words--and turn it into our familiar feature-vector form
- A collection of documents is called a *corpus*
- *A document is composed of individual tokens or terms*
- *Each document is one instance*
  - *but we don't know in advance what the features will be*

TEXT REPRESENTATION

Bag of Words (BOW)

"BAG OF WORDS"

- Treat every document as just a collection of individual words
  - Ignore grammar, word order, sentence structure, and (usually) punctuation
  - Treat every word in a document as a potentially important keyword of the document
- What will be a feature's value in a given document?
  - Each document is represented by a one (if the token is present in the document) or a zero (the token is not present in the document)
- Straightforward representation
- Inexpensive to generate
- Tends to work well for many tasks

PRE-PROCESSING OF TEXT

- The following steps should be performed:
- The case should be normalized
  - every term is in lowercase
- Words should be stemmed
  - suffixes are removed
  - E.g. noun plurals are transformed to singular forms
- Stop-words should be removed
  - A stop-word is a very common word in English (or whatever language is being parsed)
  - Typical words such as the words *the*, *and*, *of*, and *on* are removed

NORMALIZED TERM FREQUENCY

- Documents of various lengths
- Words of different frequencies
  - Words should not be *too common* or *too rare*
  - Both upper and lower limit on the number (or fraction) of documents in which a word may occur
  - Feature selection is often employed
- The raw term frequencies are normalized in some way,
  - such as by dividing each by the total number of words in the document
  - or the frequency of the specific term in the corpus

TF-IDF

$$\mathrm{TFIDF}(t, d) = \mathrm{TF}(t, d) \times \mathrm{IDF}(t)$$

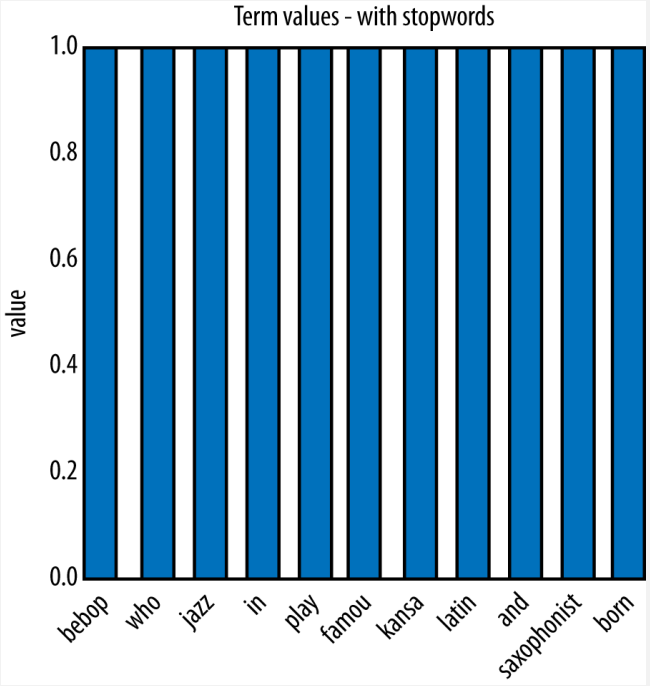
- Inverse Document Frequency (IDF) of a term

$$\mathrm{IDF}(t) = 1 + \log\left(\frac{\text{Total number of documents}}{\text{Number of documents containing } t}\right)$$

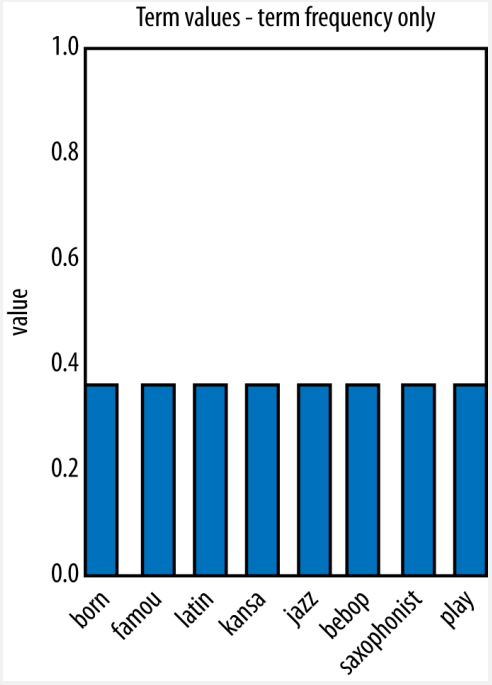
EXAMPLE: JAZZ MUSICIANS

- 16 prominent jazz musicians and excerpts of their biographies from Wikipedia
- Nearly 2,000 features after stemming and stop-word removal!
- Consider the sample phrase "Famous jazz saxophonist born in Kansas who played bebop and latin"

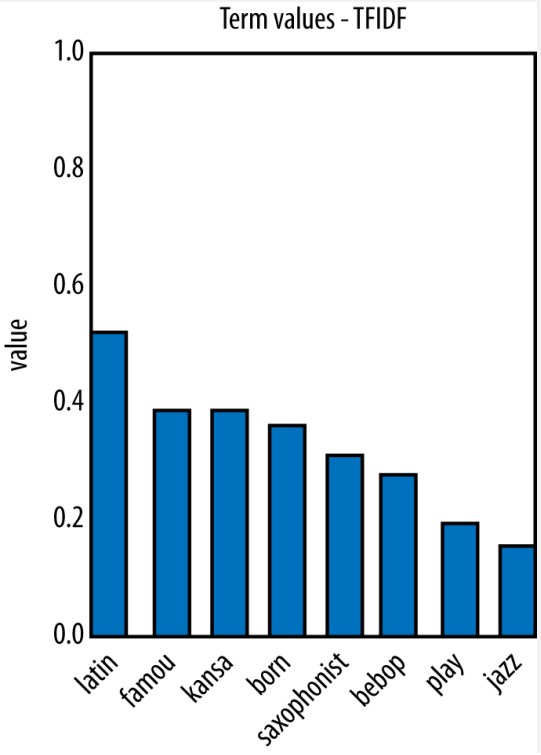
EXAMPLE: JAZZ MUSICIANS



EXAMPLE: JAZZ MUSICIANS



EXAMPLE: JAZZ MUSICIANS



EXAMPLE: JAZZ MUSICIANS

Musician	Similarity	Musician	Similarity
Charlie Parker	0.135	Count Basie	0.119
Dizzie Gillespie	0.086	John Coltrane	0.079
Art Tatum	0.050	Miles Davis	0.050
Clark Terry	0.047	Sun Ra	0.030
Dave Brubeck	0.027	Nina Simone	0.026
Thelonius Monk	0.025	Fats Waller	0.020
Charles Mingus	0.019	Duke Ellington	0.017
Benny Goodman	0.016	Louis Armstrong	0.012

EXAMPLE: JAZZ MUSICIANS

Charlie Parker

Parker at Three Deuces, New York in 1947

Background information	
Birth name	Charles Parker Jr.
Also known as	Bird, Yardbird
Born	August 29, 1920 Kansas City, Kansas, U.S.
Died	March 12, 1955 (aged 34) Manhattan, New York, U.S.
Genres	Jazz · bebop
Occupation(s)	Musician · composer
Instruments	Alto and tenor saxophone
Years active	1937–55
Labels	Savoy · Dial · Verve · Mercury · UK: Esquire · Vogue · EMI Columbia
Associated acts	Dizzy Gillespie · Max Roach · Miles Davis
Website	charliebirdparker.com

A MAJOR LIMITATION OF BOW

```
from scipy.spatial.distance import cosine

d1 = "Obama speaks to the media in Illinois"
d2 = "The President addresses the press in Chicago"

vect = CountVectorizer(stop_words="english").fit([d1, d2])
print("Features:",  ", ".join(vect.get_feature_names()))

v_1, v_2 = vect.transform([d1, d2])
v_1 = v_1.toarray().ravel()
v_2 = v_2.toarray().ravel()
print(v_1, v_2)
print("cosine(doc_1, doc_2) = {:.2f}".format(cosine(v_1, v_2)))
```

```
('Features:', u'addresses, chicago, illinois, media, obama, president, press, speaks')
(array([0, 0, 1, 1, 1, 0, 0, 1], dtype=int64), array([1, 1, 0, 0, 0, 1, 1, 0], dtype=int64))
cosine(doc_1, doc_2) = 1.00
```

TEXT REPRESENTATION

Beyond "Bag of Words"

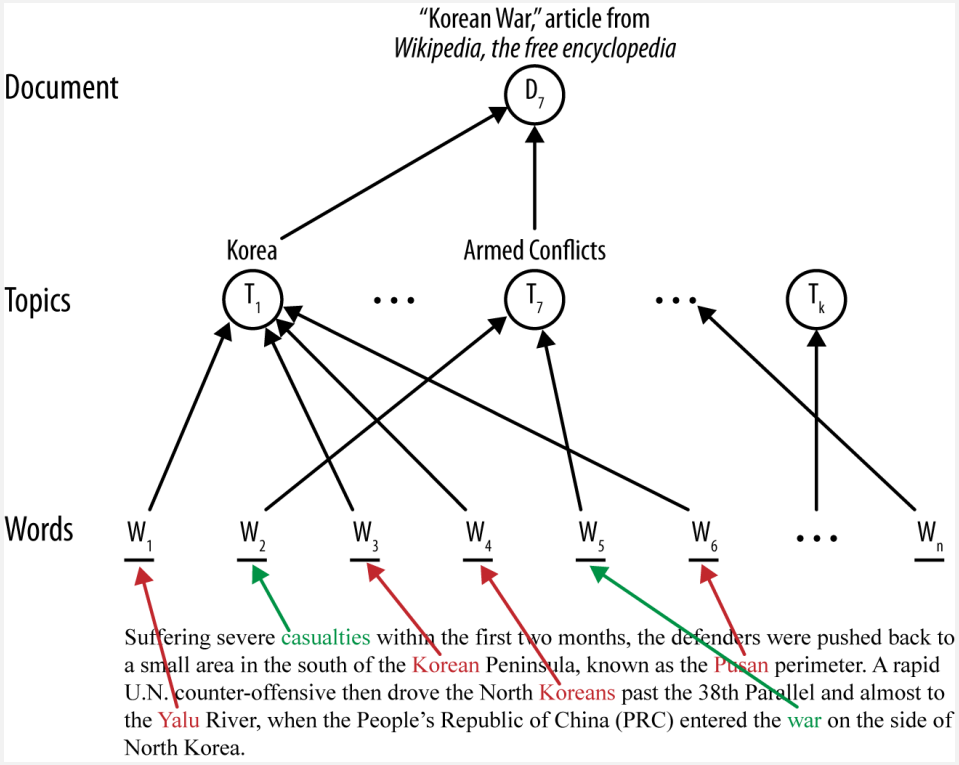
BEYOND "BAG OF WORDS"

- N-gram Sequences
- Named Entity Extraction
- Topic Models

N-GRAM SEQUENCES

- In some cases, **word order is important** and you want to preserve some information about it in the representation
- A next step up in complexity is to include sequences of adjacent words as terms
- Adjacent pairs are commonly called bi-grams
- Example: "The quick brown fox jumps"
  - it would be transformed into {quick, brown, fox, jumps, quick_brown, brown_fox, fox_jumps}
- N-grams greatly increase the size of the feature set

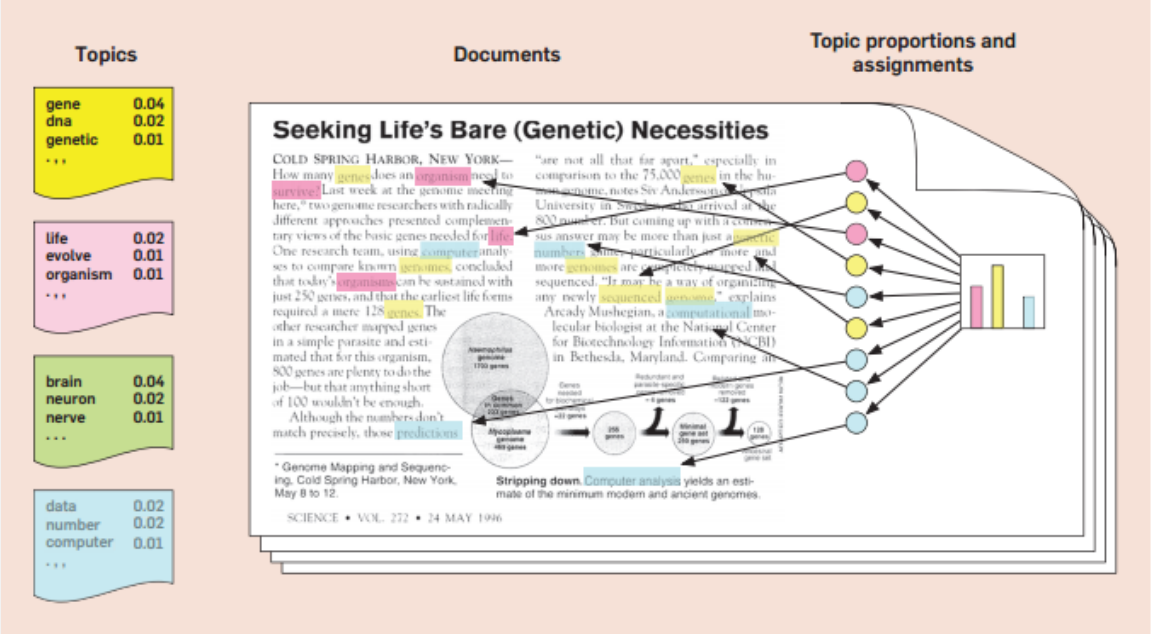
TOPIC MODELS



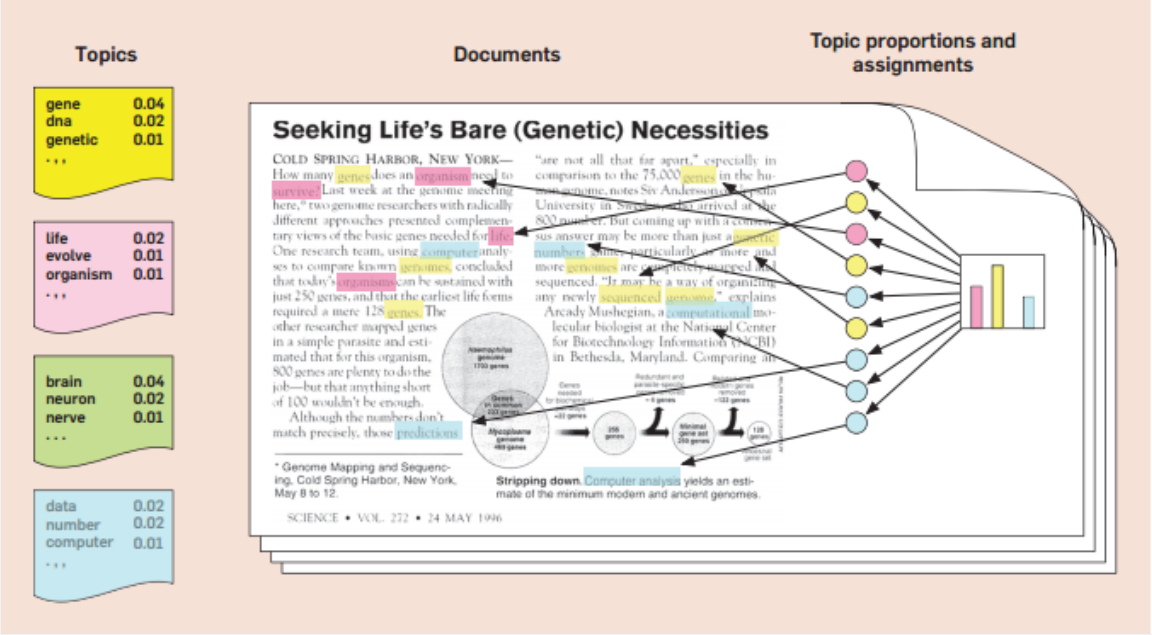
TOPIC MODELS

- Given size of online repositories → manual organization is not practical
- Our goal: To discover the hidden thematic structure with *hierarchical probabilistic models* called topic models
- Traditional applications: browsing, searching, content-based recommender systems
- Recent research:

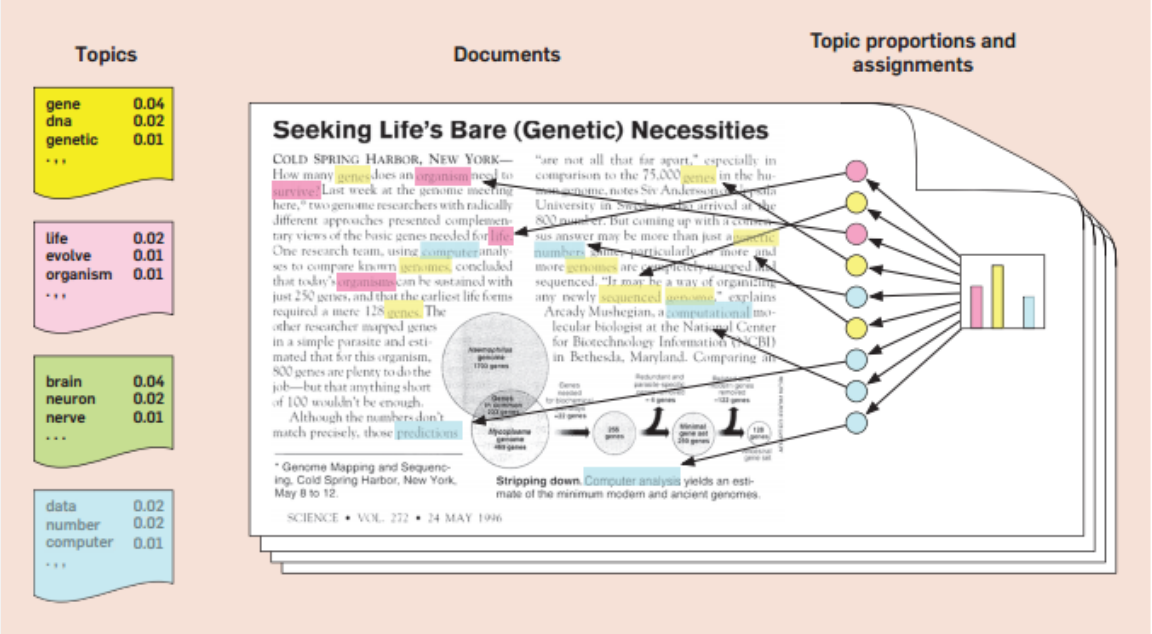
LATENT DIRICHLET ALLOCATION



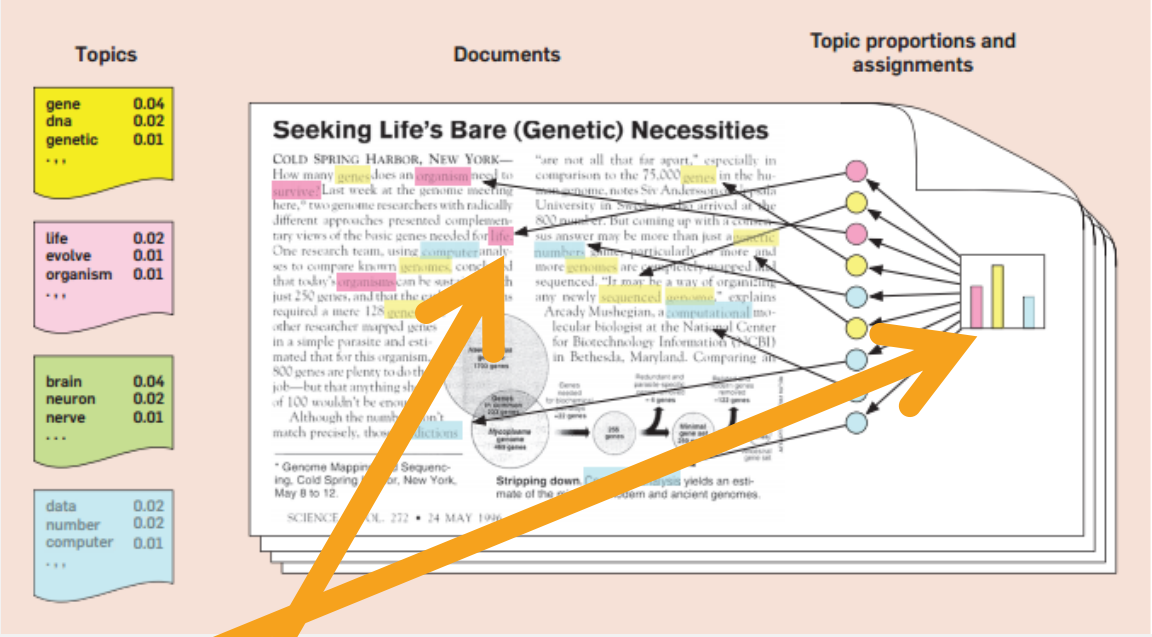
LATENT DIRICHLET ALLOCATION



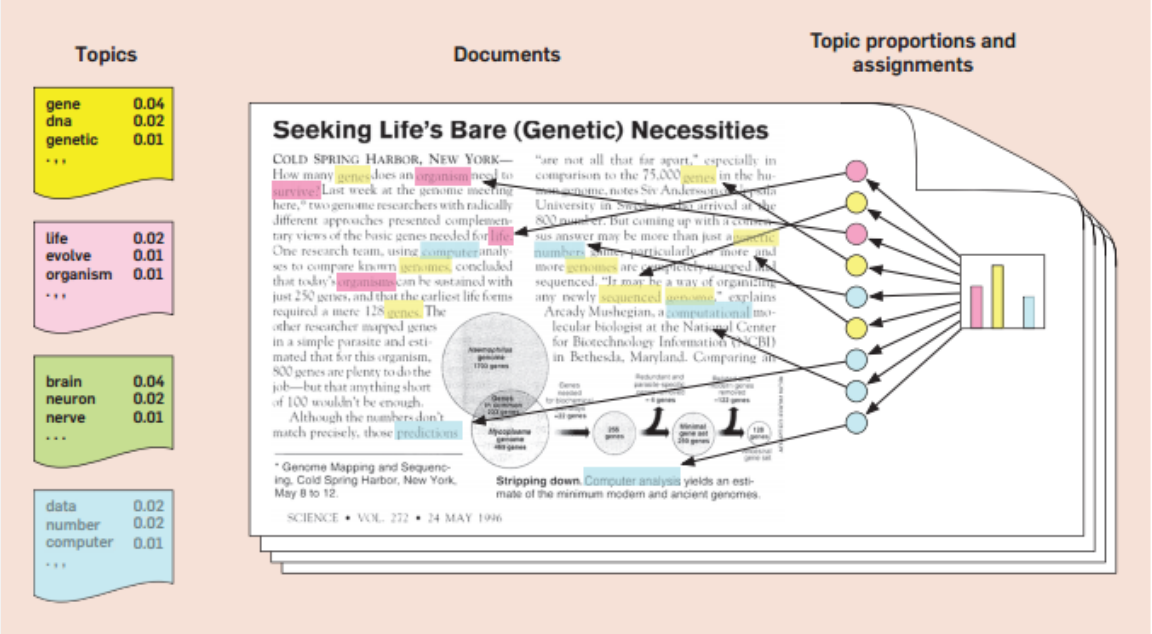
Imagine a generative probabilistic process for this document



- Imagine a generative probabilistic process for this document
- Lets say we have 100 distributions of words, ie 100 topics
  - Some have words about genetics with high probability
  - Some have words about computation with high probability
- Each document is a random mixture of these 100 topics
- Each word is a draw from one of these topics

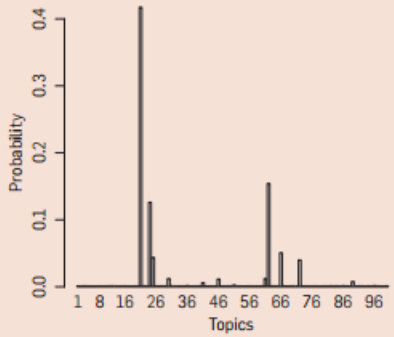


1. Choose a distribution over the distribution of words (aka topic)
2. For each word in the collection choose a topic and draw a word from the distribution
   - E.g. word 'organism' came from the pink topic
3. Repeat this process for every document



- In reality, we only observe the documents
- We need to discover the underlying topic structure
  - What are the topics?
  - How are the documents divided according to these topics? – topic proportion
  - How likely is a given word associated with a given topic? – topic (to word) assignment

100 TOPIC MODEL TO 17,000 *SCIENCE* ARTICLES

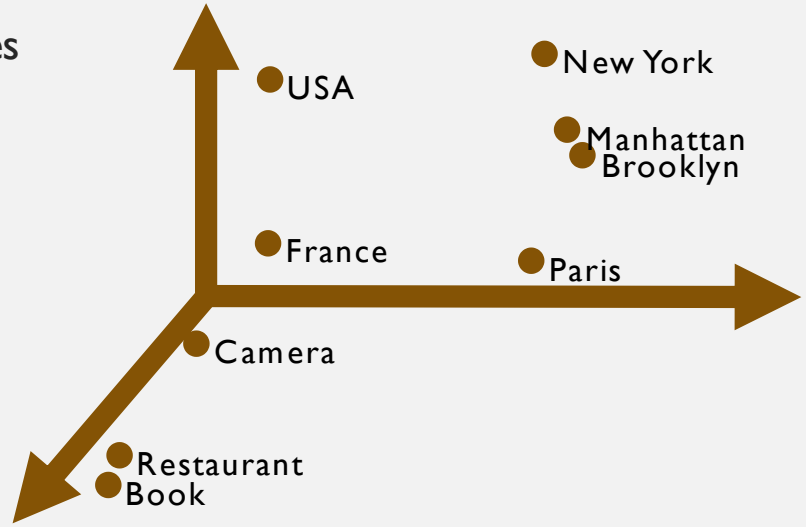


"Genetics"	"Evolution"	"Disease"	"Computers"
human	evolution	disease	computer
genome	evolutionary	host	models
dna	species	bacteria	information
genetic	organisms	diseases	data
genes	life	resistance	computers
sequence	origin	bacterial	system
gene	biology	new	network
molecular	groups	strains	systems
sequencing	phylogenetic	control	model
map	living	infectious	parallel
information	diversity	malaria	methods
genetics	group	parasite	networks
mapping	new	parasites	software
project	two	united	new
sequences	common	tuberculosis	simulations

- Left – Topic proportions for a given article
- Right – Top 15 most frequent words from the most frequent topics

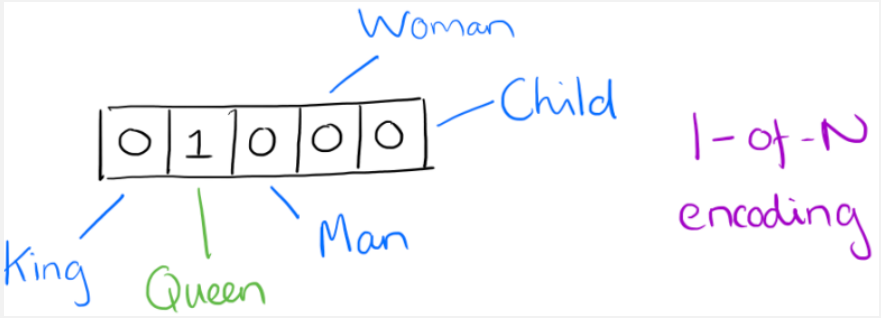
EMBEDDED (VECTOR) REPRESENTATION

- Words as dense numeric vectors
- Semantics of words (vs. bag-of-words)
  - distances between words and phrases



REPRESENTING THE MEANING OF WORDS

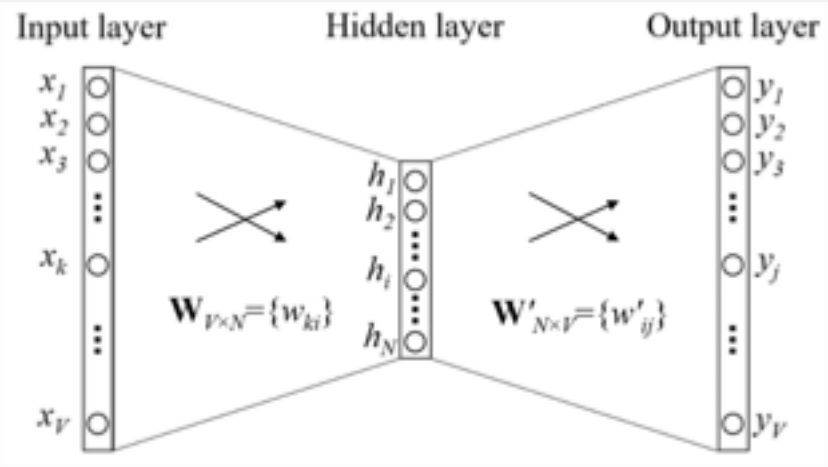
ONE-HOT



DISTRIBUTED REPRESENTATION

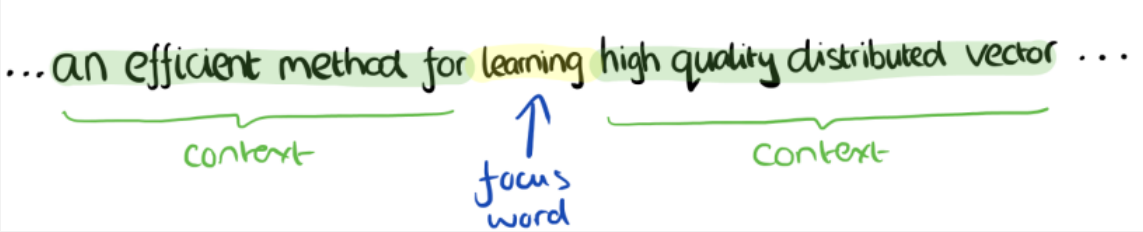
WORD2VEC

- Use Neural Network to embed words into smaller dimension (N = 100-700)



word2vec model architecture

CONTEXTUAL REPRESENTATION



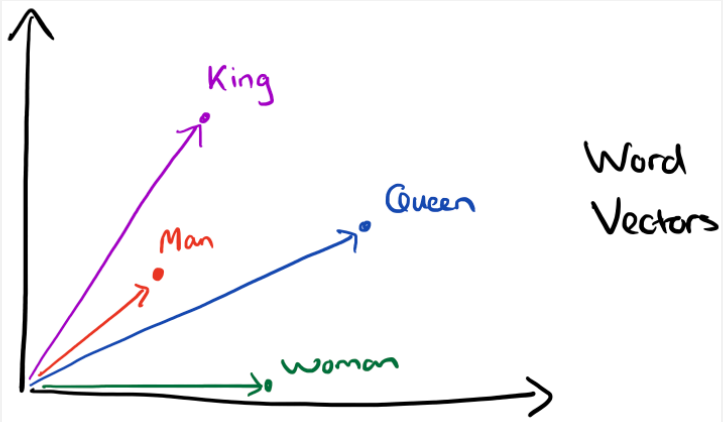
Word is represented by context in use:

I eat an **apple** every day.

I eat an **orange** every day.

I like driving my **car** to work.

WORD2VEC


King
Queen
Man
Woman
Word
Vectors


King
-Man
Queen
+Woman
Vector
Composition

WORD2VEC

Table 8: *Examples of the word pair relationships, using the best word vectors from Table 4 (Skip-gram model trained on 783M words with 300 dimensionality).*

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

WORD2VEC

Newspapers			
New York	New York Times	Baltimore	Baltimore Sun
San Jose	San Jose Mercury News	Cincinnati	Cincinnati Enquirer
NHL Teams			
Boston	Boston Bruins	Montreal	Montreal Canadiens
Phoenix	Phoenix Coyotes	Nashville	Nashville Predators
NBA Teams			
Detroit	Detroit Pistons	Toronto	Toronto Raptors
Oakland	Golden State Warriors	Memphis	Memphis Grizzlies
Airlines			
Austria	Austrian Airlines	Spain	Spainair
Belgium	Brussels Airlines	Greece	Aegean Airlines
Company executives			
Steve Ballmer	Microsoft	Larry Page	Google
Samuel J. Palmisano	IBM	Werner Vogels	Amazon

Table 2: Examples of the analogical reasoning task for phrases (the full test set has 3218 examples). The goal is to compute the fourth phrase using the first three. Our best model achieved an accuracy of 72% on this dataset.

WORD2VEC

Czech + currency	Vietnam + capital	German + airlines	Russian + river	French + actress
koruna	Hanoi	airline Lufthansa	Moscow	Juliette Binoche
Check crown	Ho Chi Minh City	carrier Lufthansa	Volga River	Vanessa Paradis
Polish zolty	Viet Nam	flag carrier Lufthansa	upriver	Charlotte Gainsbourg
CTK	Vietnamese	Lufthansa	Russia	Cecile De

Table 5: Vector compositionality using element-wise addition. Four closest tokens to the sum of two vectors are shown, using the best Skip-gram model.

WORD2VEC

- Unsupervised ML:
- *Berlin* is to *Germany* as *Paris* is to …?
- *king – man + woman* = ?
- "dinner cereal breakfast lunch", which word doesn't fit?

PREDICTION FOR POLICY

Aiding in Public Health

AIDING IN PUBLIC HEALTH

PREDICTING RESTAURANT SANITATION

- Outbreaks of illness commonly come from food prepared by restaurants
  - Estimated 75% of the outbreaks in US came from food prepared by caterers, delis, and restaurants.
- Health inspectors are a constrained resource but inspections are random
- Yelp collects rich data about customers' experiences at restaurants
  - Predominately through ratings and free text reviews
- Can Yelp's data make the process of sending Health Inspectors to restaurants more efficient?
  - Kang et al. found using Yelp's reviews data and past health inspection records can predict future inspection scores for restaurants 82% of the time.
  - Meja et al. shows that Yelp's reviews data can identify cases of hygiene violations in restaurants, even after the restaurant has been inspected and certified.

PREDICTING RESTAURANT SANITATION

- **Target variable**: *binary*
  - whether the restaurant will have a (class minor, major, or severe) violation on a given inspection date
- **Prediction**: *class probability estimation*
  - The probability that a restaurant will have a given violation on the inspection date
- **Targeting**: target restaurants that are more likely to have (more extreme) violations
- **Features**: review, reviewer, and restaurant characteristics.

COMBINING EVIDENCE PROBABILISTICALLY

- The unstructured nature of text makes things difficult
- It is all but certain we will not have the exact same review repeated
- We will consider the different pieces of evidence separately, and then combine the evidence
  - "Evidence" here represents the marginal correlation of the use of a word, in a review, with the probability of a violation

BAYES' RULE

- Joint probability using conditional probability:

$$p(A \cap B) = p(A) \times p(B|A) = p(B) \times p(A|B)$$

[Similarly, $p(A \cap B|C) = p(A|C) \times p(B|A \cap C)$]

- This means:

$$p(B|A) = \frac{p(A|B) \times p(B)}{p(A)}$$

BAYES RULE FOR CLASSIFICATION

$$p(C = c|\boldsymbol{E}) = \frac{p(\boldsymbol{E}|C = c) \times p(C = c)}{p(\boldsymbol{E})}$$

E: Full Text of Review

C: Health Violation

- $p(C = c|\boldsymbol{E})$ is the *posterior* probability
  - the probability that the target variable C takes on the class of interest c after taking the evidence E
- $p(C = c)$ is the prior probability of the class
  - the probability we would assign to the class before seeing any evidence
- $p(\boldsymbol{E}|C = c)$ is the likelihood of seeing the evidence $\boldsymbol{E}$ when the class $C = c$
- $p(\boldsymbol{E})$ is the likelihood of the evidence

BAYES RULE FOR CLASSIFICATION

$$p(\boldsymbol{E}|c) = p(e_1 \wedge e_2 \wedge \ldots \wedge e_k|c)$$

e_k : k^{th} word in Review

- **Bayesian methods** deal with this issue by making assumptions of probabilistic independence

CONDITIONAL INDEPENDENCE AND NAÏVE BAYES

- Mathematically, assuming conditional independence:

$$p(\boldsymbol{E}|c) = p(e_1 \wedge e_2 \wedge \ldots \wedge e_k|c)$$
$$= p(e_1|c) \times p(e_2|c) \times \cdots \times p(e_k|c)$$

- Hence, the posterior probability is given by:

$$p(c_0|\boldsymbol{E}) = \frac{p(e_1|c_0) \times p(e_2|c_0) \times \cdots \times p(e_k|c_0) \times p(c_0)}{p(e_1|c_0) \times \cdots \times p(e_k|c_0) + p(e_1|c_1) \times \cdots \times p(e_k|c_1)}$$

ADVANTAGES AND DISADVANTAGES OF NAÏVE BAYES

- Very simple classifier
- Efficient in terms of both storage space and computation time
- Performs well in many real-world applications
- Non-accurate class probability estimation
- Incremental learnerss