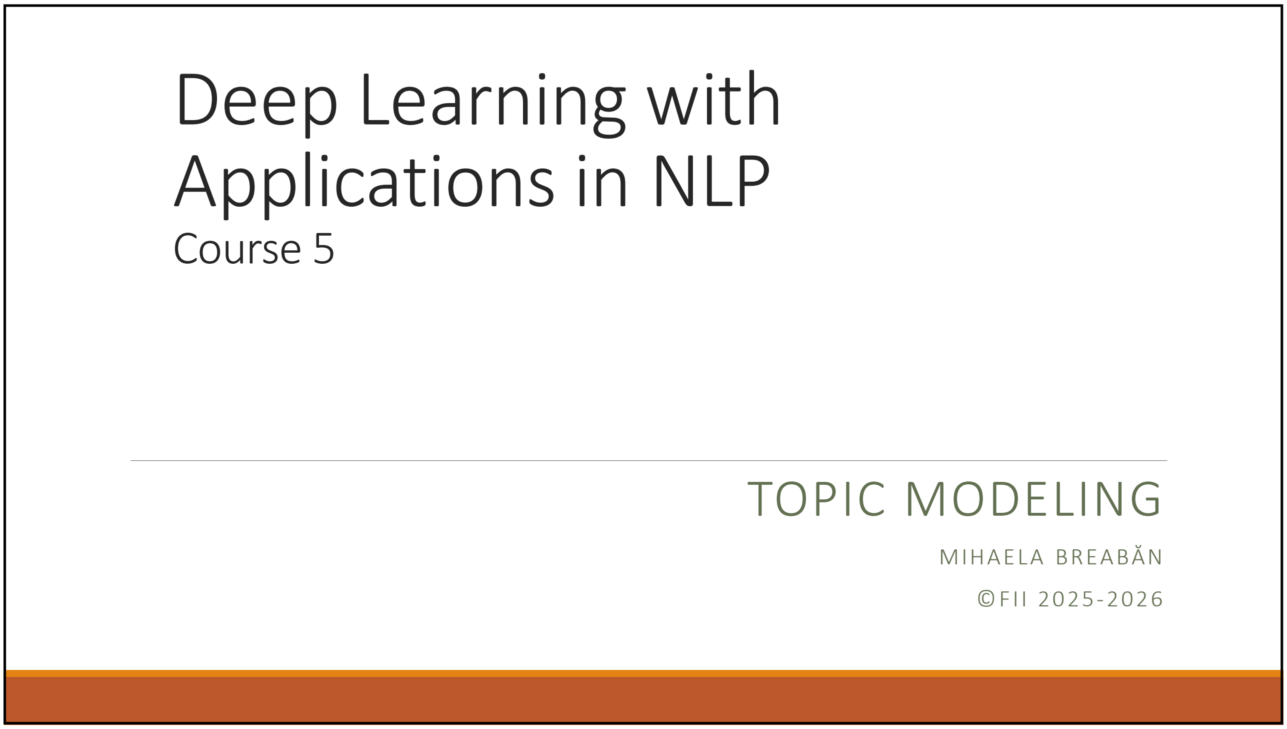


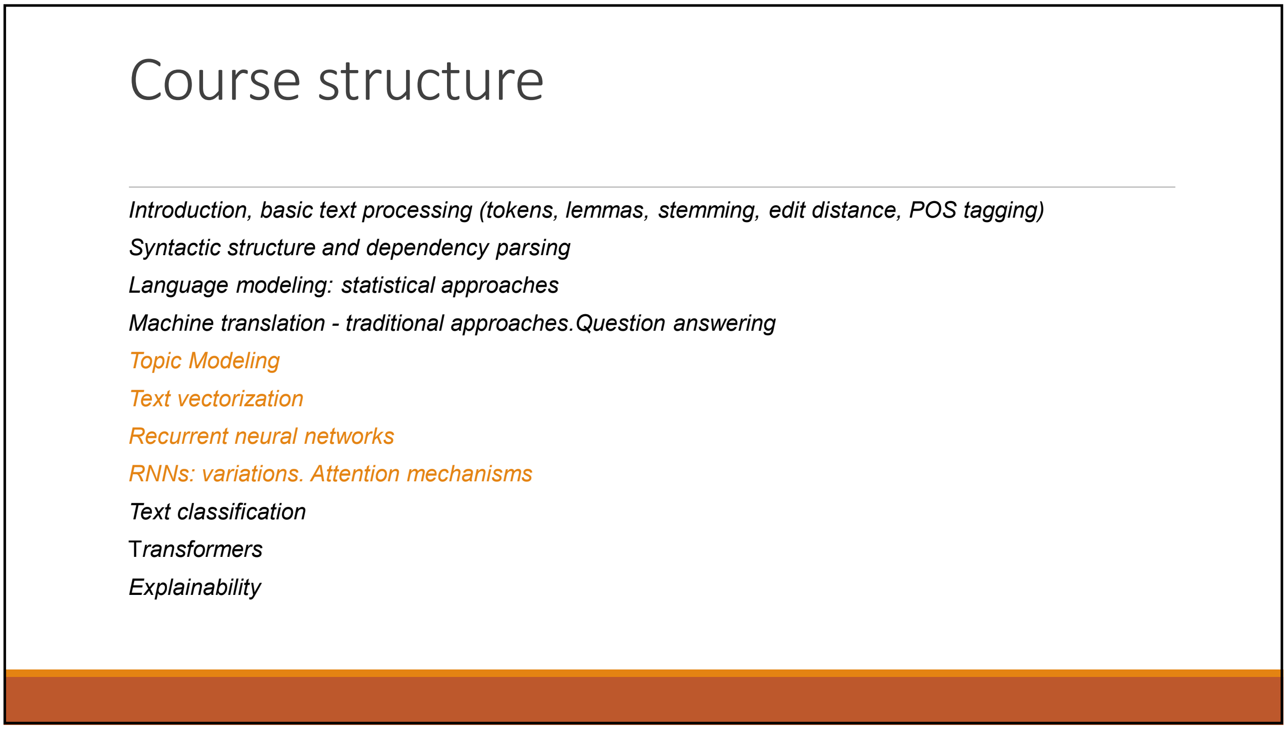
Deep Learning with Applications in NLP

Course 5

TOPIC MODELING

MIHAELA BREABĂN

1



Course structure

Introduction, basic text processing (tokens, lemmas, stemming, edit distance, POS tagging)

Syntactic structure and dependency parsing

Language modeling: statistical approaches

Machine translation - traditional approaches.Question answering

Topic Modeling

Text vectorization

Recurrent neural networks

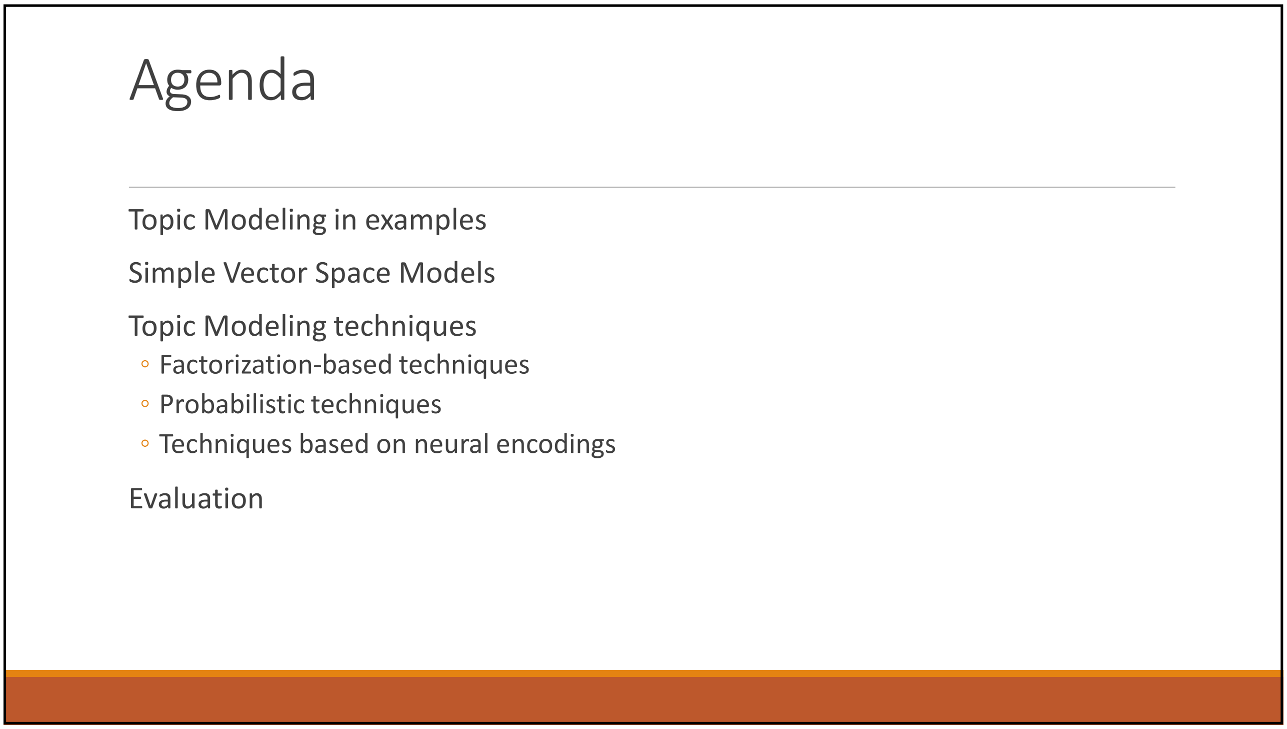
RNNs: variations. Attention mechanisms

Text classification

Transformers

Explainability

2



Agenda

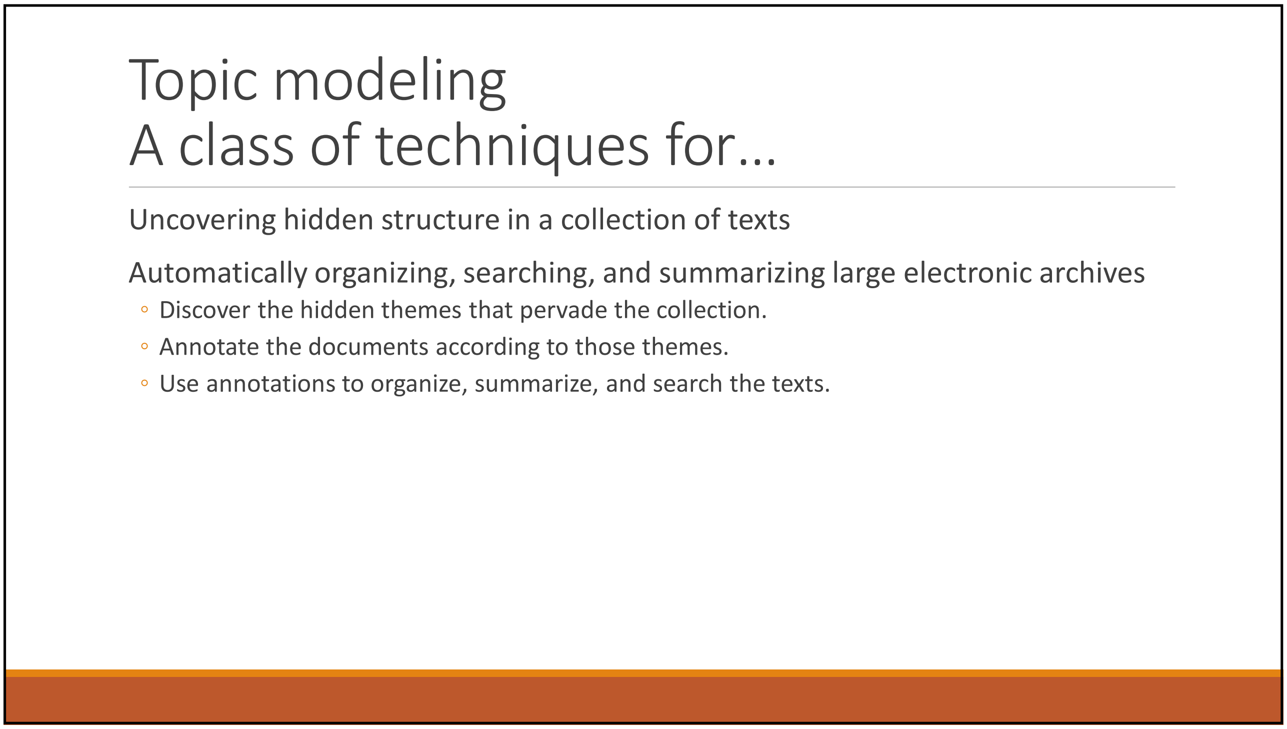
Topic Modeling in examples

Simple Vector Space Models

Topic Modeling techniques
- Factorization-based techniques
- Probabilistic techniques
- Techniques based on neural encodings

Evaluation

3



Topic modeling
A class of techniques for…

Uncovering hidden structure in a collection of texts

Automatically organizing, searching, and summarizing large electronic archives
- Discover the hidden themes that pervade the collection.
- Annotate the documents according to those themes.
- Use annotations to organize, summarize, and search the texts.

4

Topic modeling
is basically exploratory

- the topics are automatically detected (no prior assumptions on the set of topics)
- a topic is described by (or discovered as) a collection of semantically related words

Example: **Topic Modeling Martha Ballard's Diary –** courtesy to Cameron Blevins
https://www.cameronblevins.org/posts/topic-modeling-martha-ballards-diary/
The challenge: daily entries over the course of 27 years (1785-1812)

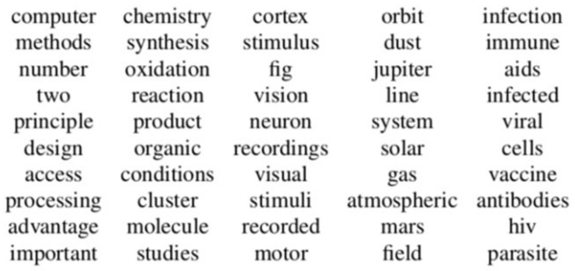
DEATH: day yesterday informd morn years death ye hear expired expird weak dead las past heard days drowned departed
GARDENING: gardin sett worked clear beens corn warm planted matters cucumbers gatherd potatoes plants ou sowd door squash wed seeds
SHOPPING**:** lb made brot bot tea butter sugar carried oz chees pork candles wheat store pr beef spirit churnd flower
ILLNESS: unwell mr sick gave dr rainy easier care head neighbor feet relief made throat poorly takeing medisin ts stomach

5

Topic modeling
is basically exploratory

- the topics are automatically detected (no prior assumptions on the set of topics)
- a topic is basically described by (or discovered as) a collection of semantically related words

Example: **Five topics from a 50-topic LDA model fit to *Science* articles from 1980-2002**
–David M. Blei, John D. Lafferty. *Topic Models. 2011*



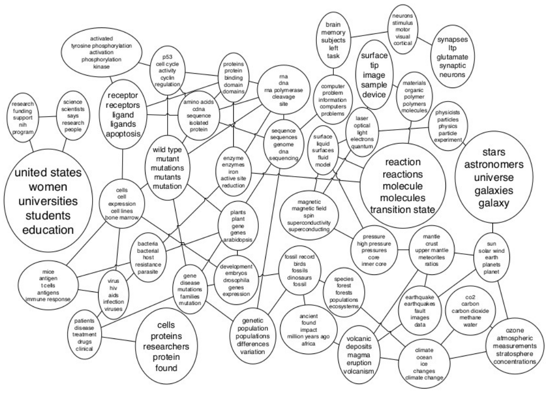
computer	chemistry	cortex	orbit	infection
methods	synthesis	stimulus	dust	immune
number	oxidation	fig	jupiter	aids
two	reaction	vision	line	infected
principle	product	neuron	system	viral
design	organic	recordings	solar	cells
access	conditions	visual	gas	vaccine
processing	cluster	stimuli	atmospheric	antibodies
advantage	molecule	recorded	mars	hiv
important	studies	motor	field	parasite

6

Topic modeling is basically exploratory

- the topics are automatically detected (no prior assumptions on the set of topics)
- a topic is basically described by (or discovered as) a collection of semantically related words

Example: A portion of the topic graph learned from the 16,351 OCR articles from Science (1990-1999). Each topic node is labeled with its most probable phrases and has font proportional to its popularity in the corpus.
–Blei, David M., and John D. Lafferty. "A correlated topic model of science." *The annals of applied statistics* 1.1 (2007)

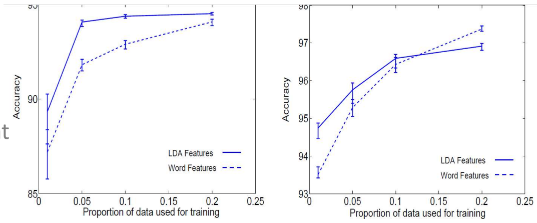

activated tyrosine phosphorylation activation phosphorylation kinase
p53 cell cycle activity cyclin regulation
proteins protein binding domain domains
rna dna rna polymerase cleavage site
brain memory subjects left task
neurons stimulus motor visual cortical
synapses ltp glutamate synaptic neurons
research funding support nih program
science scientists says research people
receptor receptors ligand ligands apoptosis
amino acids cdna sequence isolated protein
computer problem information computers problems
surface tip image sample device
materials organic polymer polymers molecules
physicists particles physics particle experiment
sequence sequences genome dna sequencing
surface liquid surfaces fluid model
laser optical light electrons quantum
united states women universities students education
wild type mutant mutations mutants mutation
enzyme enzymes iron active site reduction
reaction reactions molecule molecules transition state
stars astronomers universe galaxies galaxy
cells cell expression cell lines bone marrow
plants plant gene genes arabidopsis
magnetic magnetic field spin superconductivity superconducting
bacteria bacterial host resistance parasite
pressure high pressure pressures core inner core
mantle crust upper mantle meteorites ratios
sun solar wind earth planets planet
mice antigen t cells antigens immune response
virus hiv aids infection viruses
gene disease mutations families mutation
development embryos drosophila genes expression
fossil record birds fossils dinosaurs fossil
species forest forests populations ecosystems
earthquake earthquakes fault images data
co2 carbon carbon dioxide methane water
patients disease treatment drugs clinical
cells proteins researchers protein found
genetic population populations differences variation
ancient found impact million years ago africa
volcanic deposits magma eruption volcanism
climate ocean ice changes climate change
ozone atmospheric measurements stratosphere concentrations

Topic modeling can assist predictive/inference tasks

- As input for (supervised) classification tasks

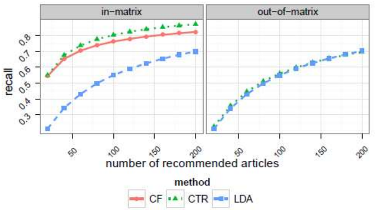
Example: Classification results on two binary classification problems from the Reuters-21578

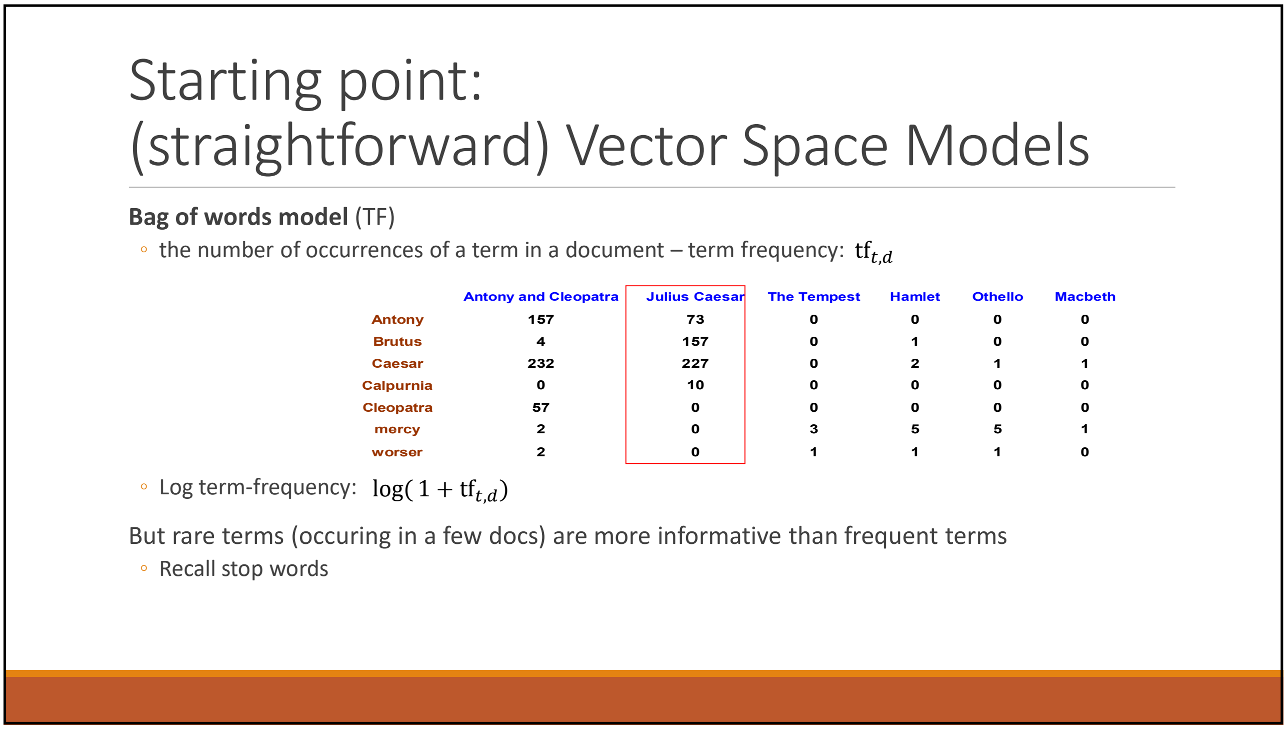
- Blei David, Andrew Y. Ng, Michael I. Jordan. "Latent Dirichlet Allocation." *The Journal of machine Learning research* 3 (2003)



- In recommender systems

- Wang, Chong, and David M. Blei. "Collaborative topic modeling for recommending scientific articles." *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2011.





Starting point: (straightforward) Vector Space Models

Bag of words model (TF)

- the number of occurrences of a term in a document – term frequency: $\mathrm{tf}_{t,d}$

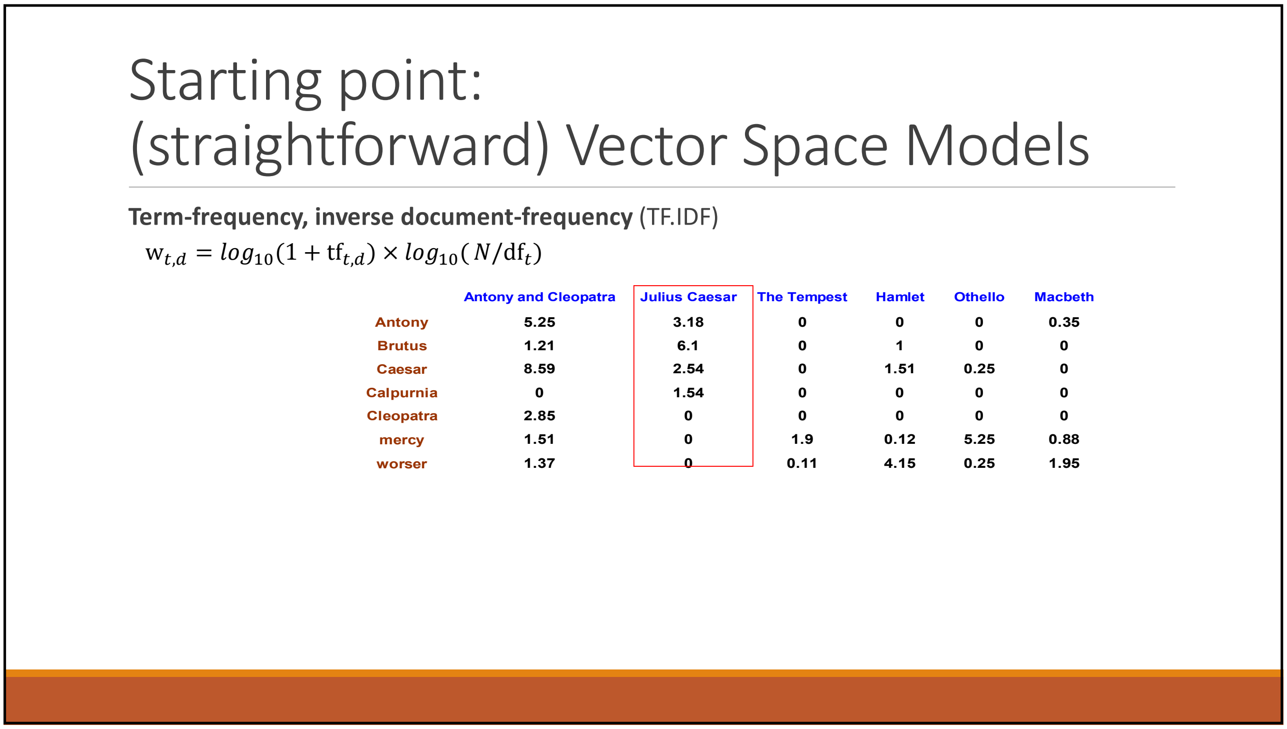
	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	157	73	0	0	0	0
Brutus	4	157	0	1	0	0
Caesar	232	227	0	2	1	1
Calpurnia	0	10	0	0	0	0
Cleopatra	57	0	0	0	0	0
mercy	2	0	3	5	5	1
worser	2	0	1	1	1	0

- Log term-frequency: $\log(1 + \mathrm{tf}_{t,d})$

But rare terms (occuring in a few docs) are more informative than frequent terms

- Recall stop words

9



Starting point: (straightforward) Vector Space Models

Term-frequency, inverse document-frequency (TF.IDF)

$$\mathrm{w}_{t,d} = log_{10}(1 + \mathrm{tf}_{t,d}) \times log_{10}(N/\mathrm{df}_t)$$

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	5.25	3.18	0	0	0	0.35
Brutus	1.21	6.1	0	1	0	0
Caesar	8.59	2.54	0	1.51	0.25	0
Calpurnia	0	1.54	0	0	0	0
Cleopatra	2.85	0	0	0	0	0
mercy	1.51	0	1.9	0.12	5.25	0.88
worser	1.37	0	0.11	4.15	0.25	1.95

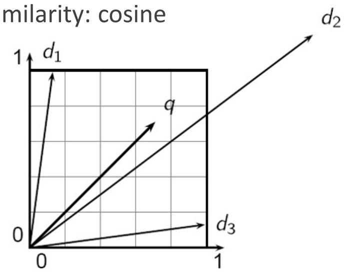
10

Search in such a Vector Space

Query q – encoded in the same VS as document d

Distance: the angle between the vectors; corresponding similarity: cosine

$$\cos(\vec{q},\vec{d}) = \frac{\vec{q}\bullet\vec{d}}{|\vec{q}||\vec{d}|} = \frac{\vec{q}}{|\vec{q}|}\bullet\frac{\vec{d}}{|\vec{d}|} = \frac{\sum_{i=1}^{|V|} q_i d_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2}\sqrt{\sum_{i=1}^{|V|} d_i^2}}$$



Why not Euclidean distance?

It is large even though the distribution of terms in the query and the distribution of terms in the document are very similar.

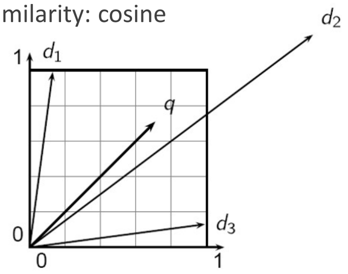
Experiment: take a document *d* and append it to itself. Call this document *d*′. “Semantically” d and d′ have the same content. The Euclidean distance can be large while the angle is 0

Search in such a Vector Space

Query q – encoded in the same VS as document d

Distance: the angle between the vectors; corresponding similarity: cosine

$$\cos(\vec{q},\vec{d}) = \frac{\vec{q}\bullet\vec{d}}{|\vec{q}||\vec{d}|} = \frac{\vec{q}}{|\vec{q}|}\bullet\frac{\vec{d}}{|\vec{d}|} = \frac{\sum_{i=1}^{|V|} q_i d_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2}\sqrt{\sum_{i=1}^{|V|} d_i^2}}$$



What about synonyms? Query has different terms compared to document, even though semantically are the same.

What about polysemes? Query has the same terms compared to document, but they are semantically different.

Topic Modeling
Objectives/Uses

Semantic text vectorization

- Extract and represent the contextual-usage meaning of words in a numerical “semantic space”
- Determine the similarity of meaning of words/passages/documents

Dimensionality reduction for text representation

- From words (vocabulary size) to topics (#domains/subjects)

Increase performance in search

- exploits the meaning of words by removing “linguistic noise” that is present due to the variability in term choice
  - Synonymy (teacher/instructor/educator) -> increase recall
  - Polysemy (sound) -> increase precision

Perform fuzzy (bi-)clustering

- Cluster both words and documents

13

Topic Modeling
Formal definition

Input:

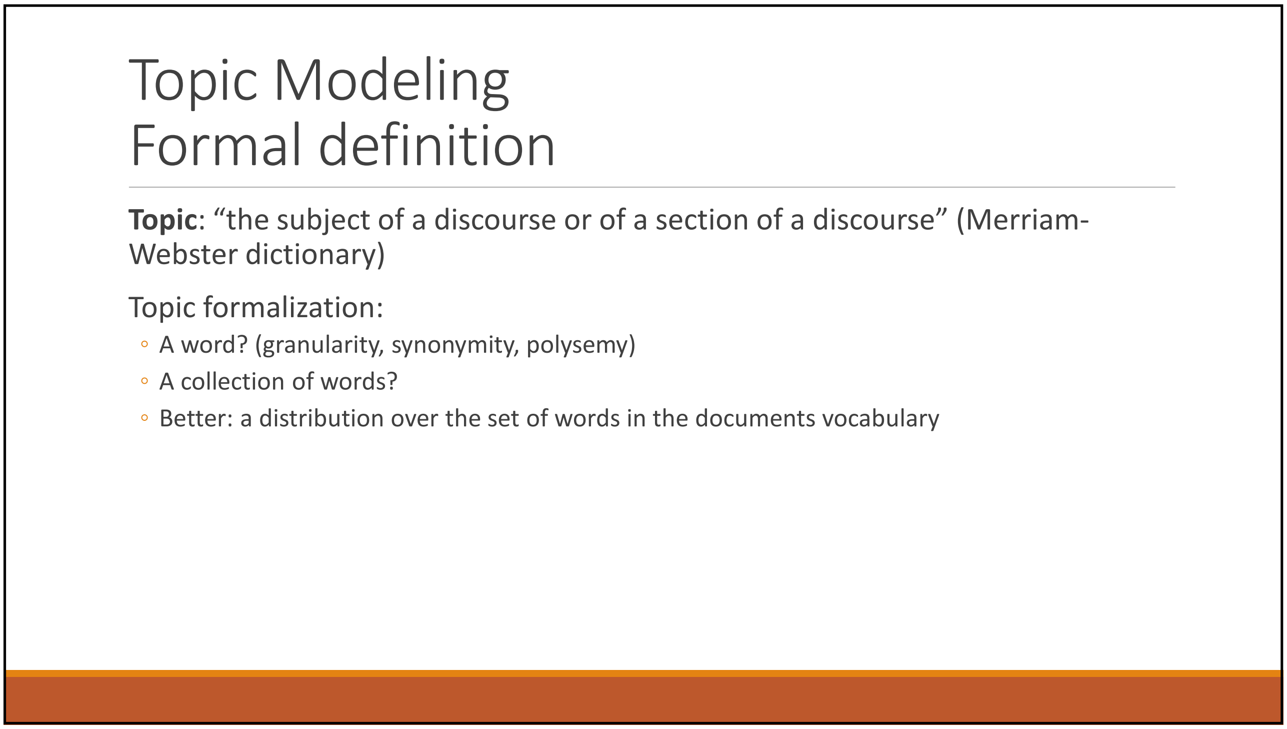
- A collection of N documents
- Number of topics k

Output:

- k topics
- Occurrence of the k topics in each of the N documents
  - Desirable as a distribution for each document

But how model a topic?

14



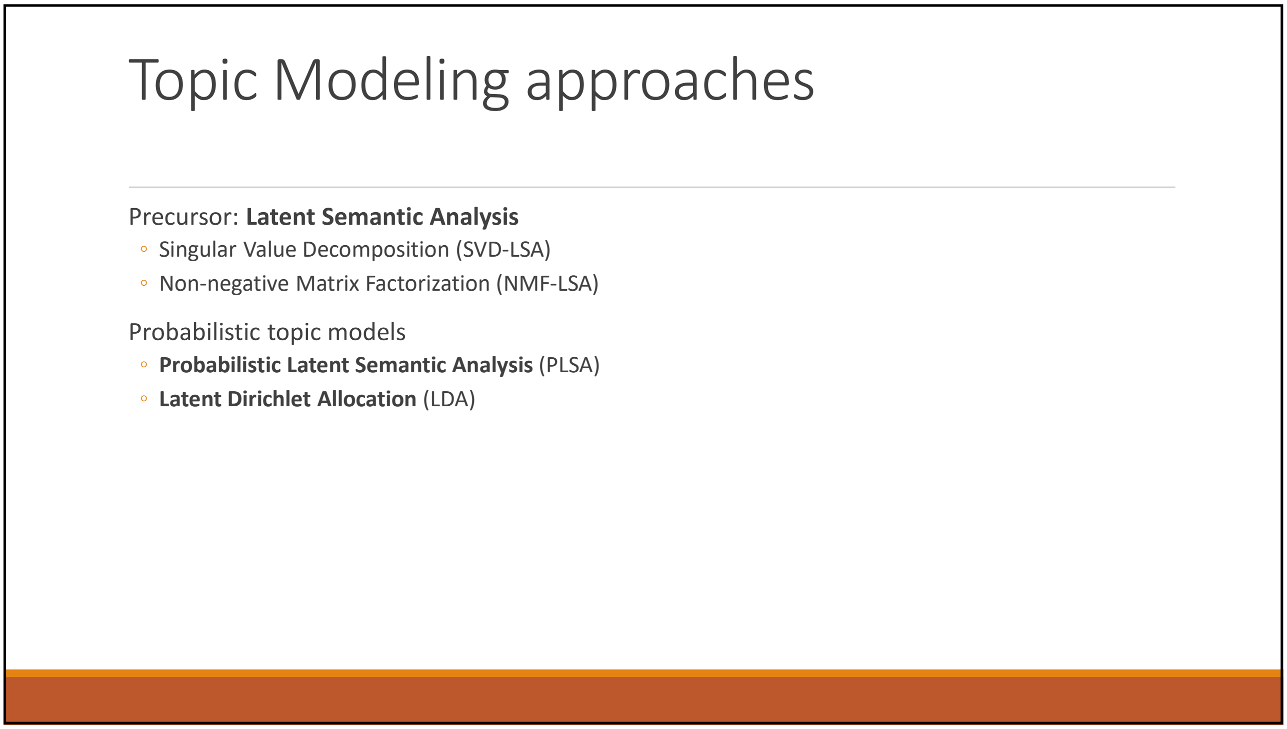
Topic Modeling
Formal definition

Topic: “the subject of a discourse or of a section of a discourse” (Merriam-Webster dictionary)

Topic formalization:

- A word? (granularity, synonymity, polysemy)
- A collection of words?
- Better: a distribution over the set of words in the documents vocabulary

15



Topic Modeling approaches

Precursor: **Latent Semantic Analysis**

- Singular Value Decomposition (SVD-LSA)
- Non-negative Matrix Factorization (NMF-LSA)

Probabilistic topic models

- **Probabilistic Latent Semantic Analysis** (PLSA)
- **Latent Dirichlet Allocation** (LDA)

16

Latent Semantic Analysis (LSA) [Landauer & Dumais 1997]

A form of factor analysis – behind the observable data there are some latent factors that generate/influence/explain what we see

- Data: matrix
- Procedure: matrix decomposition/factorization

Observable data in LSA: the document collection, in the form of a numerical matrix

- Bag of words: tf
- Weighted bag of words: tf.idf

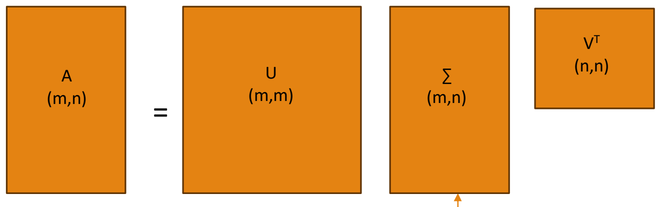
Originates in Landauer and Dumais work (1997), based on SVD

17

Singular value decomposition (SVD)

Generalizes the eigendecomposition of a square normal matrix with an orthonormal eigenbasis to any m × n matrix

$$\boxed{\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T}$$



Orthogonal matrixes
- The columns of **U** are orthonormal eigenvectors of $\mathbf{A}\mathbf{A}^T$
- The columns of the orthogonal matrix **V** are eigenvectors of$\mathbf{A}^T\mathbf{A}$.

Diagonal matrix
(singular values – squared root of eigenvalues of $\mathbf{A}\mathbf{A}^T$)

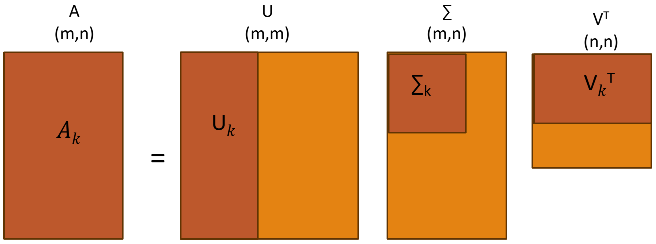
* The SVD is not unique. We choose the one where the singular values are in descending order!

18

SVD
+ truncation

Use a rank-k approximation $A_k = \sum_{i=1}^{k} u_i \sigma_i v_i^T$

This is the best or closest (distance is minimized) rank *k* approximation to the original matrix **A** (minimizes the Frobenius norm of the difference between A and A_k for a fixed k)



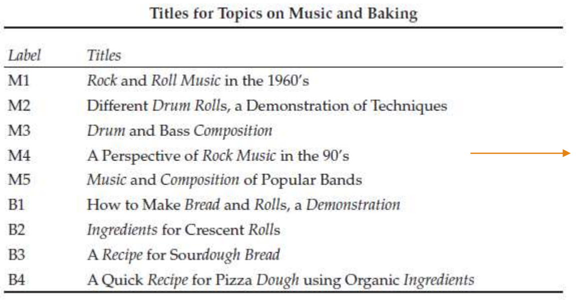
This *k*-dimensional vector space is the foundation for the semantic structures LSA exploits

SVD-LSA

Truncating the SVD and creating $\mathbf{A}_k$ is what captures the important underlying semantic structure of words and documents.

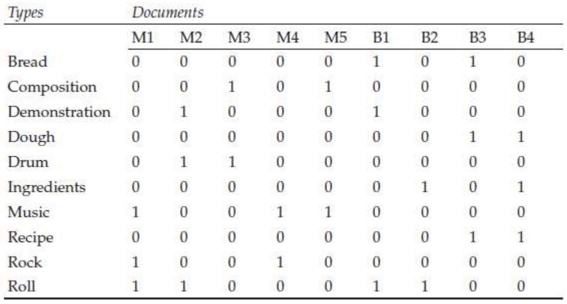
- Terms similar in meaning are "near" each other in *k*-dimensional vector space even if they never co-occur in a document
- Documents similar in conceptual meaning are near each other even if they share no words in common (Berry et al., 1995).

SVD-LSA
Example – the data



Titles for Topics on Music and Baking

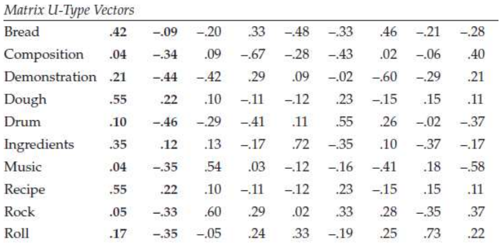
Label	*Titles*
M1	*Rock* and *Roll Music* in the 1960's
M2	Different *Drum Rolls*, a Demonstration of Techniques
M3	*Drum* and Bass *Composition*
M4	A Perspective of *Rock Music* in the 90's
M5	*Music* and *Composition* of Popular Bands
B1	How to Make *Bread* and *Rolls*, a *Demonstration*
B2	*Ingredients* for Crescent *Rolls*
B3	A *Recipe* for Sourdough *Bread*
B4	A Quick *Recipe* for Pizza *Dough* using Organic *Ingredients*



Types	*Documents*								
	M1	M2	M3	M4	M5	B1	B2	B3	B4
Bread	0	0	0	0	0	1	0	1	0
Composition	0	0	1	0	1	0	0	0	0
Demonstration	0	1	0	0	0	1	0	0	0
Dough	0	0	0	0	0	0	0	1	1
Drum	0	1	1	0	0	0	0	0	0
Ingredients	0	0	0	0	0	0	1	0	1
Music	1	0	0	1	1	0	0	0	0
Recipe	0	0	0	0	0	0	0	1	1
Rock	1	0	0	1	0	0	0	0	0
Roll	1	1	0	0	0	1	1	0	0

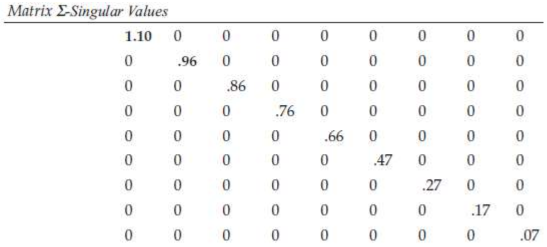
Landauer, T. K., McNamara, D. S., Dennis, S., & Kintsch, W. (Eds.). (2013). Handbook of latent semantic analysis. Psychology Press.

SVD-LSA
Example – the SVD



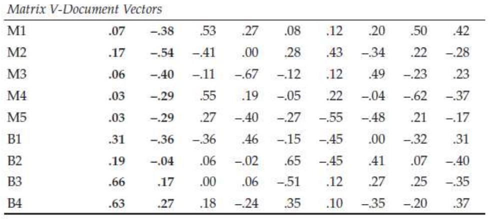
Matrix U-Type Vectors

Bread	.42	–.09	–.20	.33	–.48	–.33	.46	–.21	–.28
Composition	.04	–.34	.09	–.67	–.28	–.43	.02	–.06	.40
Demonstration	.21	–.44	–.42	.29	.09	–.02	–.60	–.29	.21
Dough	.55	.22	.10	–.11	–.12	.23	–.15	.15	.11
Drum	.10	–.46	–.29	–.41	.11	.55	.26	–.02	–.37
Ingredients	.35	.12	.13	–.17	.72	–.35	.10	–.37	–.17
Music	.04	–.35	.54	.03	–.12	–.16	–.41	.18	–.58
Recipe	.55	.22	.10	–.11	–.12	.23	–.15	.15	.11
Rock	.05	–.33	.60	.29	.02	.33	.28	–.35	.37
Roll	.17	–.35	–.05	.24	.33	–.19	.25	.73	.22



Matrix Σ-Singular Values

1.10	0	0	0	0	0	0	0	0
0	.96	0	0	0	0	0	0	0
0	0	.86	0	0	0	0	0	0
0	0	0	.76	0	0	0	0	0
0	0	0	0	.66	0	0	0	0
0	0	0	0	0	.47	0	0	0
0	0	0	0	0	0	.27	0	0
0	0	0	0	0	0	0	.17	0
0	0	0	0	0	0	0	0	.07



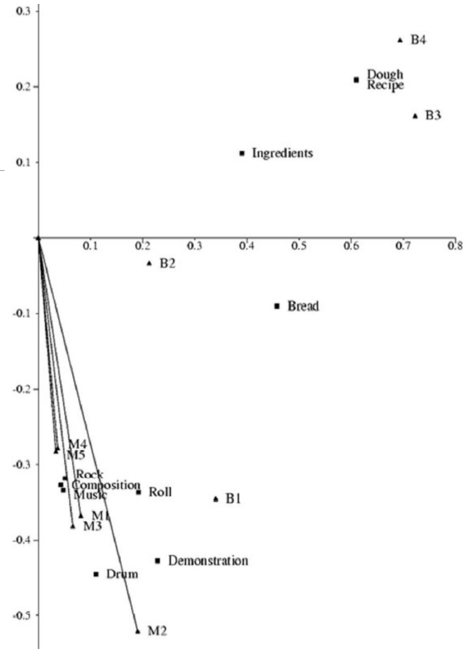
Matrix V-Document Vectors

M1	.07	–.38	.53	.27	.08	.12	.20	.50	.42
M2	.17	–.54	–.41	.00	.28	.43	–.34	.22	–.28
M3	.06	–.40	–.11	–.67	–.12	.12	.49	–.23	.23
M4	.03	–.29	.55	.19	–.05	.22	–.04	–.62	–.37
M5	.03	–.29	.27	–.40	–.27	–.55	–.48	.21	–.17
B1	.31	–.36	–.36	.46	–.15	–.45	.00	–.32	.31
B2	.19	–.04	.06	–.02	.65	–.45	.41	.07	–.40
B3	.66	.17	.00	.06	–.51	.12	.27	.25	–.35
B4	.63	.27	.18	–.24	.35	.10	–.35	–.20	.37

Landauer, T. K., McNamara, D. S., Dennis, S., & Kintsch, W. (Eds.). (2013). Handbook of latent semantic analysis. Psychology Press.

SVD-LSA Example

The semantic space



Landauer, T. K., McNamara, D. S., Dennis, S., & Kintsch, W. (Eds.). (2013). Handbook of latent semantic analysis. Psychology Press.

Latent Semantic space Updates

Add new term

- Initial representation: vector t containing zero and nonzero elements where the nonzero elements are (weighted) elements corresponding to the documents that contain the term.
- obtain t_{new} in the semantic space by computing

$$\acute{t}_{new} = tV_k \Sigma_k^{-1}$$

Add new document

- Initial representation: vector d containing zero and nonzero elements where the nonzero correspond to the (weighted) terms frequency contained in the document
- obtain d_{new} in the semantic space by computing

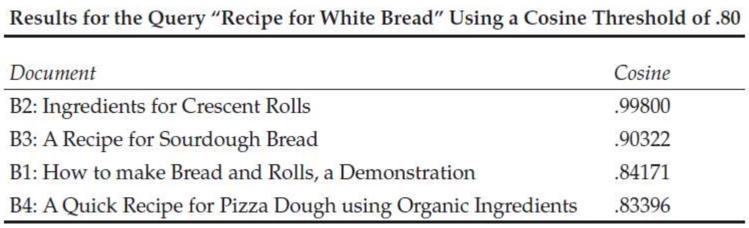
$$\bar{d}_{new} = d^T U_k \Sigma_k^{-1}$$

SVD-LSA
Use in search

- Given a query q (pseudo-documents), embed it in the semantic space by computing

$$query = q^T U_k \Sigma_k^{-1}$$

- Use the cosine similarity to rank documents (or terms)



Results for the Query "Recipe for White Bread" Using a Cosine Threshold of .80

Document	*Cosine*
B2: Ingredients for Crescent Rolls	.99800
B3: A Recipe for Sourdough Bread	.90322
B1: How to make Bread and Rolls, a Demonstration	.84171
B4: A Quick Recipe for Pizza Dough using Organic Ingredients	.83396

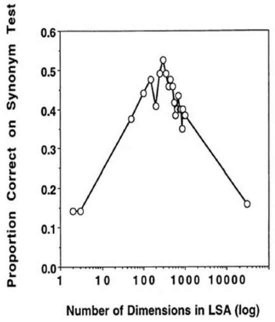
Landauer, T. K., McNamara, D. S., Dennis, S., & Kintsch, W. (Eds.). (2013). Handbook of latent semantic analysis. Psychology Press.

25

SVD-LSA
What about k?

Depends on the task and on the data

Example-Experimental task: choose from four alternatives the one that is closest in meaning to a tarhet word. With LSA, the cosine between the target word and each of the alternatives is calculated and the alternative with the highest cosine is chosen.*



Performance (corrected for guessing) on 80 retired items from the synonym component of the Test of English as a Foreign Language (TOEFL; Landauer & Dumais, 1997)

26

SVD-LSA
Pros and Cons

Strengths

- Filter out noise(synonyms, polysemes)
- Dimension reduction - considers only essential components of term-document matrix
- Reduces storage

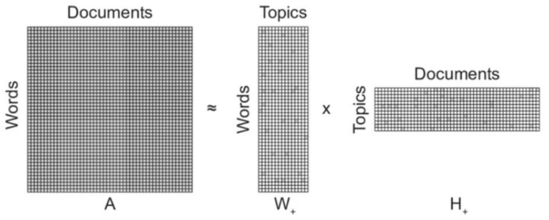
Weaknesses

- Interpretation impossible: mixed signs
- Good truncation point k is hard to determine.

27

Non-negative Matrix Factorization
NMF-LSA

$$\mathbf{A}_k = \mathbf{W}_k \mathbf{H}_k, \quad \mathbf{W}_k, \mathbf{H}_\mathrm{k} \geq \mathbf{0}$$



Defines a *topic space of dimensionality k*

W=basis vectors (the topics)

Each column of A a_i=W h_i

No orthogonal restriction on basis vector

Easy interpretation

- Elements of W and H are all non-negative.
- W_{ij} reflects how much basis vector w_j is related to topic t_i
- H_{ij} reflects how much document d_j points to the direction of basis vector w_i.

28

NMF-LSA
Algorithms

Minimize the reconstruction error (no analytical solution)

$$[\mathbf{W},\mathbf{H}] = \min \left\| \mathbf{A} - \mathbf{WH} \right\|_F^2, \text{s.t. } \mathbf{W},\mathbf{H} \geq \mathbf{0}$$

- **multiplicative update rule (**Lee and Seung, 1999**)**
  - initialize **W** and **H** non negative
  - update the values in **W** and **H** by computing, iteratively, the following

$$\mathbf{H}^{n+1}_{[i,j]} \leftarrow \mathbf{H}^{n}_{[i,j]} \frac{((\mathbf{W}^n)^T \boldsymbol{A})_{[i,j]}}{((\mathbf{W}^n)^T \mathbf{W}^n \mathbf{H}^n)_{[i,j]}}$$

and

$$\mathbf{W}^{n+1}_{[i,j]} \leftarrow \mathbf{W}^{n}_{[i,j]} \frac{(\boldsymbol{A}(\mathbf{H}^{n+1})^T)_{[i,j]}}{(\mathbf{W}^n \mathbf{H}^{n+1} (\mathbf{H}^{n+1})^T)_{[i,j]}}$$

- **non-negative least squares**:
  - H is fixed and W found by a non-negative least squares solver,
  - W is fixed and H is found analogously
  - repeat

29

NMF-LSA
Pros and Cons

Strengths

- Good interpretability
- Improved performance for document clustering comparing to LSA

Weaknesses

- Factorization is not unique
- Local minimum problem

30

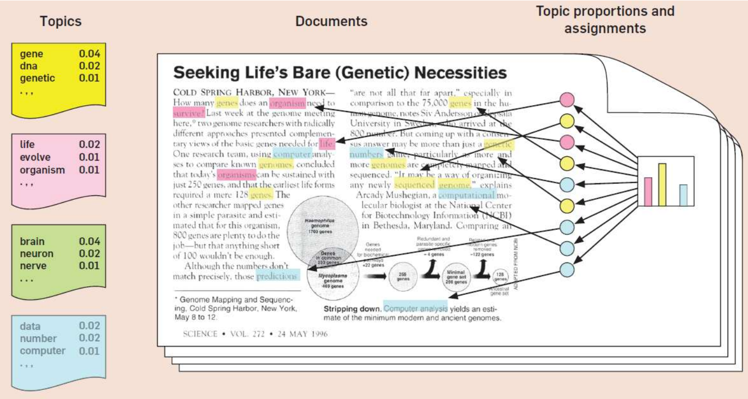
Generative approach for Topic Modeling

Assumptions:

- Documents are mixtures of topics
- A topic is a probability distribution over words

To generate a document

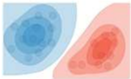
1. Choose a distribution over topics
2. Choose a topic according to the distribution and draw a word from that topic



Blei, David M. "Probabilistic topic models." *Communications of the ACM* 55.4 (2012): 77-84.

Generative vs. discriminative models in Machine Learning

Generative models:



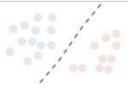
- model the process that generates the observable data (X)

- work by estimating P(X) or P(X/Y)
  - Maximum Likelihood Estimation (EM)
  - Bayes formula for computing P(Y/X)

Examples: Naïve Bayes, Gaussian mixture models, mostly unsupervised methods

They can generate data

Discriminative models:



- do not try to understand the data generation process

- generally work by identifying rules that discriminate between classes/clusters or directly estimate P(Y/X)

Most ML algorithms are discriminative

- Identify class boundaries: decision trees, kNN SVMs, logistic regression, NNs, …
- Identify cluster representatives: k-Means,…

They can not generate data

Probabilistic LSA [Hoffman, '99]

A document is generated from a mix of K multinomial distributions

- A topic is seen as a multinomial distribution over the vocabulary
  - A generalization of the binomial distribution: n independent trials, each trial resulting in the success of exactly one category from |V| categories, each category having a given fixed success probability

Each document has its specific weight vector for mixing/covering the k topics

Besides the k topics we may add also a "background" distribution to "attract" background words

33

Probabilistic LSA Notation

k – number of topics

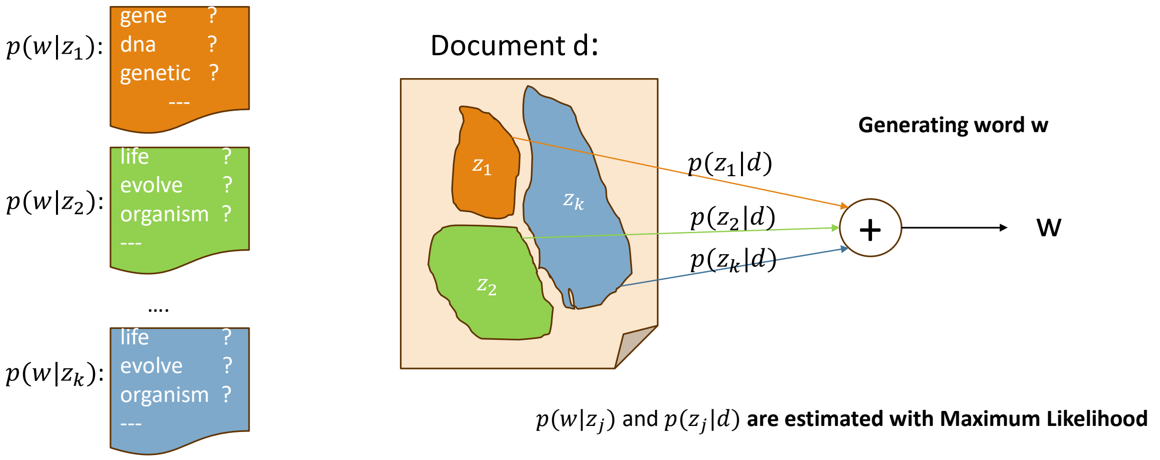
d – document

w – word

$z_1, z_2, \ldots, z_k$ - topics

34

PLSA as a mixture model



$p(w|z_1)$: gene ? dna ? genetic ? ---

$p(w|z_2)$: life ? evolve ? organism ? ---

....

$p(w|z_k)$: life ? evolve ? organism ? ---

Document d:

z_1 z_k z_2

Generating word w

$p(z_1|d)$ $p(z_2|d)$ $p(z_k|d)$ + W

$p(w|z_j)$ and $p(z_j|d)$ **are estimated with Maximum Likelihood**

PLSA as a mixture model

Probability to generate the word w in a document d generated from k topics:

$$p(w|d) = \sum_{j=1}^{k} \mathrm{p}(\mathrm{w}|z_j)p(z_j|d)$$

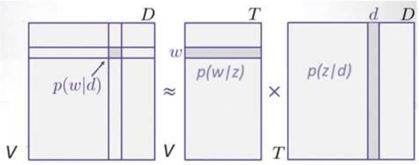
Log-probability to generate document d:

$$\log p(d) = \sum_{w\in V} n(w,d) \log \sum_{j=1}^{k} \mathrm{p}(\mathrm{w}|z_j)p(z_j|d)$$

where n(w,d) is the number of occurrences of w in d

Log-probability to generate the entire collection of documents C:

$$\log p(C) = \sum_{d\in C} \sum_{w\in V} n(w,d) \log \sum_{j=1}^{k} \mathrm{p}(\mathrm{w}|z_j)p(z_j|d)$$



Constrained optimization:

$$\theta^*, \phi^* = argmax_{\theta,\phi} \log p(C|\theta,\phi)$$

- $\forall j \in \{1..k\}, \sum_{j=1}^{k} \mathrm{p}(\mathrm{w}|\phi_j) = 1$
- $\forall d \in C, \sum_{j=1}^{k} \theta_{d,j} = 1$

Parameter estimation by Maximum Likelihood Estimation

E-step:

$$p(z_i|w,d) = \frac{p(w|z_i)p(d|z_i)p(z_i)}{\sum_{j=1}^{k} \mathrm{p}(w|z_j)p(d|z_j)p(z_j)}$$

M-step:

$$p(w|z_i) = \frac{\sum_{d\in D} n(d,w)p(z_i|w,d)}{\sum_{w\in V}\sum_{d\in D} n(d,w)p(z_i|w,d)}$$ - sum over each document where w occurs

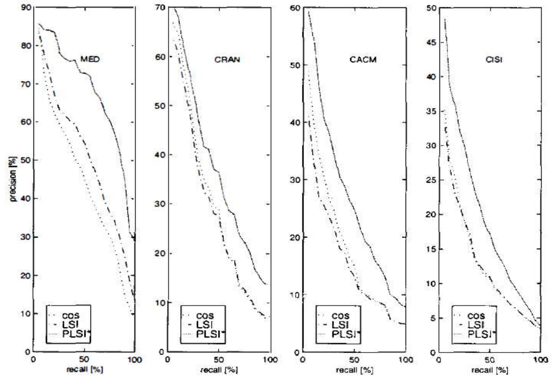
$$p(d|z_i) = \frac{\sum_{w\in V} n(d,w)p(z_i|w,d)}{\sum_{d\in D}\sum_{w\in V} n(d,w)p(z_i|w,d)}$$ - sum over all words in d

$$p(z_i) = \frac{\sum_{d\in D}\sum_{w\in V} n(d,w)p(z_i|d,w)}{\sum_{d\in D}\sum_{w\in V}\sum_{z_j} n(d,w)p(z_j|d,w)}$$ - sum over all words in all documents

where n(w,d) is the number of occurrences of w in d

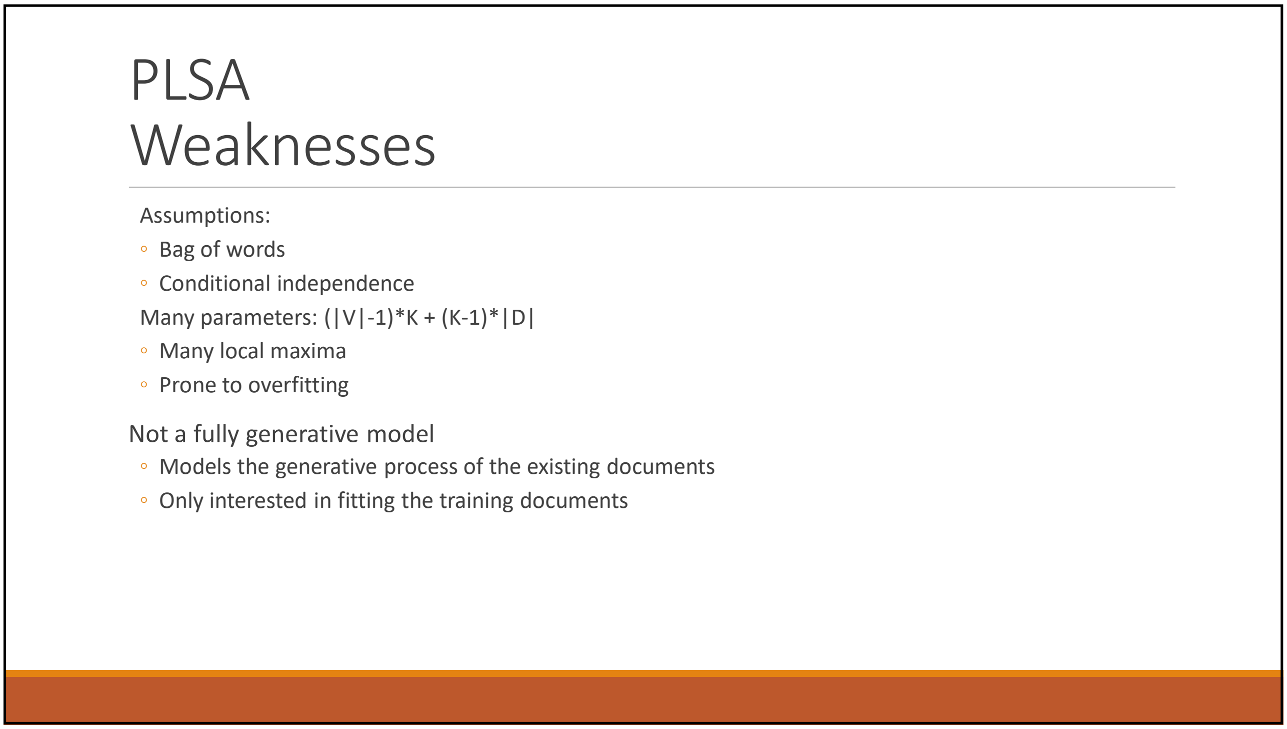
37

Comparative evaluation



(Hoffman, 1999)

38



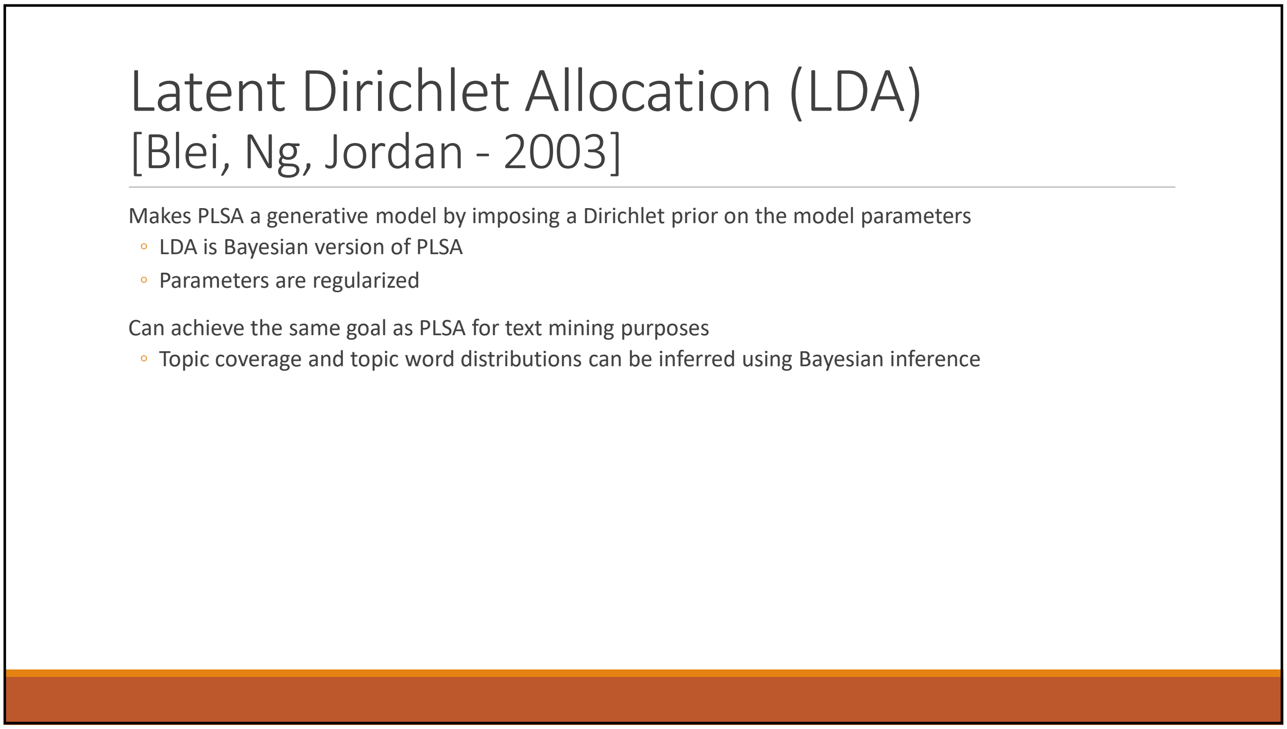
PLSA Weaknesses

Assumptions:
- Bag of words
- Conditional independence

Many parameters: (|V|-1)*K + (K-1)*|D|
- Many local maxima
- Prone to overfitting

Not a fully generative model
- Models the generative process of the existing documents
- Only interested in fitting the training documents

39



Latent Dirichlet Allocation (LDA) [Blei, Ng, Jordan - 2003]

Makes PLSA a generative model by imposing a Dirichlet prior on the model parameters
- LDA is Bayesian version of PLSA
- Parameters are regularized

Can achieve the same goal as PLSA for text mining purposes
- Topic coverage and topic word distributions can be inferred using Bayesian inference

40

LDA Notation

k – number of topics

d – document

w – word

$z_1, z_2, \dots, z_k$ - topics

$\theta_d = \{\theta_{d,1}, \theta_{d,2}, \dots, \theta_{d,k}\} = \{p(z_1|d), p(z_2|d), \dots, p(z_k|d)\}$ - distribution of topics in document d

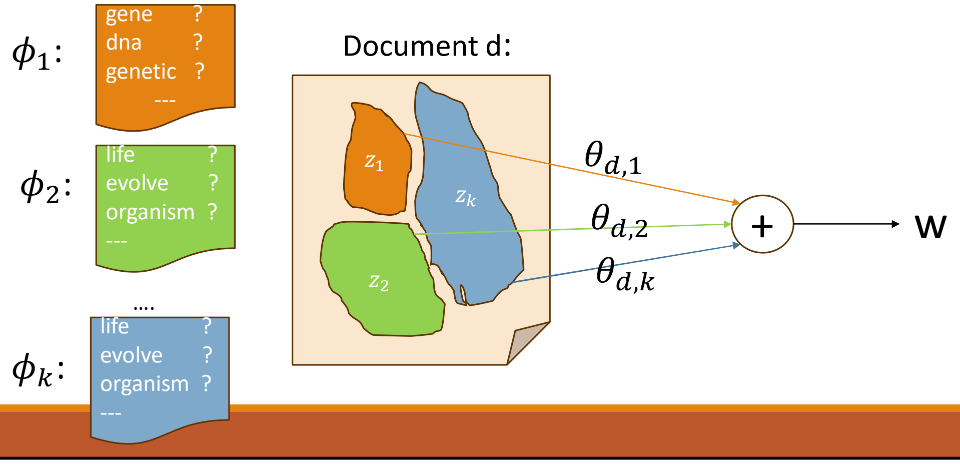
$\phi_1, \phi_2, \dots, \phi_k$– distribution of words in each topic

$$\phi_j = \{p(w|z_j)\ for\ every\ w \in V\}$$

41

LDA as a generative model

Sample ϕs from a Dirichlet distribution of parameter $\vec{\beta} = (\beta_1, \beta_2, \dots, \beta_{|V|})$
Sample θ from a Dirichlet distribution of parameter $\vec{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_k)$

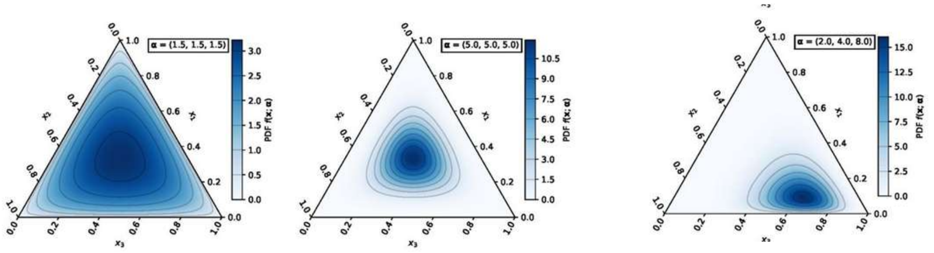


42

Dirichlet priors

The Dirichlet distribution determines the mixture proportions of the topics in the documents and the words in each topic

$\alpha_i > 1, i = 1..3$



https://en.wikipedia.org/wiki/Dirichlet_distribution

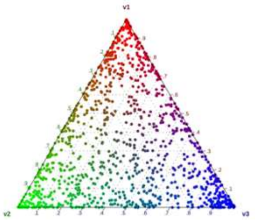
43

Dirichlet priors

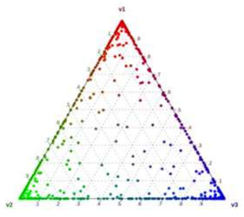
The Dirichlet distribution determines the mixture proportions of the topics in the documents and the words in each topic

```
install.packages('DirichletReg')
library(DirichletReg)
```

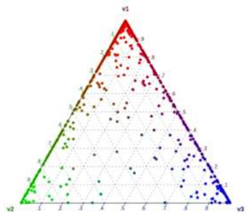
$\alpha_i < 1, i = 1..3$



```
diri = rdirichlet(1000,
         alpha = c(0.5,0.5,0.5))
dr = DR_data(diri)
plot(dr)
```



```
diri = rdirichlet(1000,
         alpha = c(0.1,0.1,0.1))
dr = DR_data(diri)
plot(dr)
```



```
diri = rdirichlet(1000,
         alpha = c(0.5,0.1,0.1))
dr = DR_data(diri)
plot(dr)
```

44

LDA in formulas

Probability of a document:

$$p(\text{'d}\mid\alpha,\beta)=\int p(\theta\mid\alpha)\left(\prod_{n=1}^{N}\sum_{z_n}p(z_n\mid\theta)p(w_n\mid z_n,\beta)\right)d\theta$$

Probability of the corpus:

$$p(\text{ c}\mid\alpha,\beta)=\prod_{d=1}^{M}\int p(\theta_d\mid\alpha)\left(\prod_{n=1}^{N_d}\sum_{z_{dn}}p(z_{dn}\mid\theta_d)p(w_{dn}\mid z_{dn},\beta)\right)d\theta_d$$

45

Parameter estimation in LDA

Use ML estimator

- Less parameters compared to PLSA

$$(\hat{\vec{\alpha}},\hat{\vec{\beta}})=\arg\max_{\vec{\alpha},\vec{\beta}}\log p(C\mid\vec{\alpha},\vec{\beta})$$

Methods:

- Gibbs sampling
- Variational inference

46

PLSA vs. LDA

LDA adds a Dirichlet distribution on top of PLSA to regularize the model

- Estimation of LDA is more complicated than PLSA

LDA is a generative model, while PLSA isn't

PLSA is more likely to over-fit the data than LDA

Which one to use?

- If you need generalization capacity, LDA
- If you want to mine topics from a collection, PLSA may be better (we want overfitting!)

47

Deep learning methods BERTopic

Combines transformer embeddings with clustering and class-based TF-IDF to generate interpretable topics.

Pipeline overview:

1. Generate document embeddings → *Sentence-BERT*
2. Reduce dimensionality → *UMAP*
3. Cluster semantically similar documents → *HDBSCAN*
4. Extract representative words → *c-TF-IDF (class-based TF-IDF)*

48

Evaluation

Topic modeling is **unsupervised**, so there are no “ground truth” labels.
Evaluation focuses on how **coherent, distinct, and interpretable** the discovered topics are.

Main goals:

- Are the topics semantically coherent?
- Are they distinct from each other?
- Can humans interpret and label the topics easily?
- Do documents fit well into their assigned topics?

Quantitative Evaluation Metrics

Topic Coherence

- Evaluates how semantically related the top words in a topic are.
- Reflects **human judgment of topic quality**.

C_v topic coherence metric (Röder et al. 2015) –

- most effective measures for correlating with human judgment
- measures how semantically similar a topic's top words are to each other

Topic diversity

$$\text{Diversity} = \frac{|V_{\text{unique}}|}{|V|}$$

-the total number of **unique** words across the top-N words of all topics

-the total number of words across the top-N words of all topics

Quantitative Evaluation Metrics
C_V computation

- Use a fixed window size (e.g., 110 words). Take N top-words in the topic and make pairs. A word pair (wi,wj) is said to co-occur if they appear within the same sliding window.

$$P(w_i, w_j) = \frac{count(w_i, w_j, \text{sliding window})}{count(\text{sliding window})}$$

$$\text{NPMI}(w_i, w_j) = \frac{\text{PMI}(w_i, w_j)}{-\log_2(P(w_i, w_j))} = \frac{\log_2\left(\frac{P(w_i,w_j)}{P(w_i)P(w_j)}\right)}{-\log_2(P(w_i, w_j))}$$

$$\vec{v}_{w_i} = (\text{NPMI}(w_i, w_1), \text{NPMI}(w_i, w_2), \ldots, \text{NPMI}(w_i, w_N))$$

$$\vec{v}_W = \sum_{i=1}^{N} \vec{v}_{w_i}$$

$$C_V = \frac{1}{N} \sum_{i=1}^{N} \text{CosineSim}(\vec{v}_{w_i}, \vec{v}_W)$$

References

LSA

- Landauer, Thomas K., and Susan T. Dumais. "A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge." *Psychological review* 104.2 (1997). http://cetus.stat.cmu.edu/~cshalizi/350/2008/readings/Landauer-Dumais.pdf

NMF

- Lee, Daniel D., and H. Sebastian Seung. "Learning the parts of objects by non-negative matrix factorization." *Nature* 401.6755 (1999): 788-791. http://lsa.colorado.edu/LexicalSemantics/seung-nonneg-matrix.pdf

Probabilistic LSA

- Hofmann, Thomas. "Unsupervised learning by probabilistic latent semantic analysis." *Machine learning* 42.1 (2001): 177-196. https://link.springer.com/content/pdf/10.1023/A:1007617005950.pdf

LDA

- Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." *the Journal of machine Learning research* 3 (2003): 993-1022. https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf
- https://miningthedetails.com/LDA_Inference_Book/

Evaluation

- Röder, M., Both, A., & Hinneburg, A. (2015). *Exploring* the *space of topic coherence measures*. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining* (pp. 399–408). ACM