Topic Identification, Topic Modeling

IV161 Natural Language Processing in Practice

Zuzana Nevěřilová, Aleš Horák

NLP Centre, FI MU, Brno

October 20, 2025

I've seen this problem several times. It was always the result of the little rollers inside the mouse becomming dirty- they are good at collecting grime. the solution is simple: remove the ball to reveal the two rollers. Carefully clean them and the ball.

Windows was working just great. I had a Bus mouse and mother board problem. DELL replaced the mouse, gave me a newer mouse driver for windows and replaced the motherboard.

Does anyone know how to configure a DOS app in Progman so that only one instance of it can be running at a time?

I think you would need a DOS macro program. Superkey (by Borland?) comes to mind. I don't think Windows is capable of sending keystrokes to a DOS window.

Using a Windows 3.1 printer driver, I would like to "print to a file", with output as a Postscript file. Later, I would like to take this Postscript file to a machine with an attached Hewlett Packard 4M laser printer,

However, a printer definitely worth looking at is the new inkjet from Epson. This printer is faster, cheaper, and capable of producing laser-like quality on normal copier paper. Can't remember the model #, LX - something I think?

. . . 18,000 more files . . .

I've seen this problem several times. It was always the result of the little rollers inside the mouse becomming dirty- they are good at collecting grime. the solution is simple: remove the ball to reveal the two rollers. Carefully clean them and the ball.

Windows was working just great. I had a Bus mouse and mother board problem. DELL replaced the mouse, gave me a newer mouse driver for windows and replaced the motherboard.

Does anyone know how to configure a DOS app in Progman so that only one instance of it can be running at a time?

I think you would need a DOS macro program. Superkey (by Borland?) comes to mind. I don't think Windows is capable of sending keystrokes to a DOS window.

Using a Windows 3.1 printer driver, I would like to "print to a file", with output as a Postscript file. Later, I would like to take this Postscript file to a machine with an attached Hewlett Packard 4M laser printer,

However, a printer definitely worth looking at is the new inkjet from Epson. This printer is faster, cheaper, and capable of producing laser-like quality on normal copier paper. Can't remember the model #, LX - something I think?

. . . 18,000 more files . . .


wrong
kiss
attentive
wait
night
visit
town
two
high
seat
sit
nice
busy
minute
leave
day
start
drive
staff
year
friendly
friend
live
hand
long
fire
last
first
strip
people
spot
cut
local
joint
waitress
fast
meal
fun
bar
big
full
week
worth
top
new
enjoy
drink
fry
dinner
flavor
table
dish
henderson
area
japanese
portion
super
ambiance
edamame
lunch
hot
rice
serve
another
huge
large
offer
quick
tasty
tempura
salad
fantastic
couple
item
yummy
salmon
specialty
better
review
yellowtail
sashimi
shrimp
happy
nigiri
kimono
pack
excellent
bad
awesome

1. Topic Modeling

2. Topic Modeling Approaches
   - Latent Semantic Analysis – LSA
   - Latent Dirichlet Allocation – LDA
   - Topic Modeling with Word Embeddings

3. Topic Evaluation

4. Topic Labeling

Topic modeling

- **organize, summarize, and understand** large collections of documents
  with no a priori knowledge
- discover unknown topical patterns in collection of documents
- dimensionality reduction – instead of taking into account every word in the document, take into account only words representing the document topics
- **topic** – group of related words representing concepts ($\rightarrow$ document tagging)
- statistical, unsupervised modeling

Topic Modeling and Topic Classification

topic modeling – find document representation by discovering topics present in the document + how much they are present (e.g. 10% horror, 70% fun, 25% Australia, 30% nature)

topic classification – categorize documents into a set of (predefined) topics

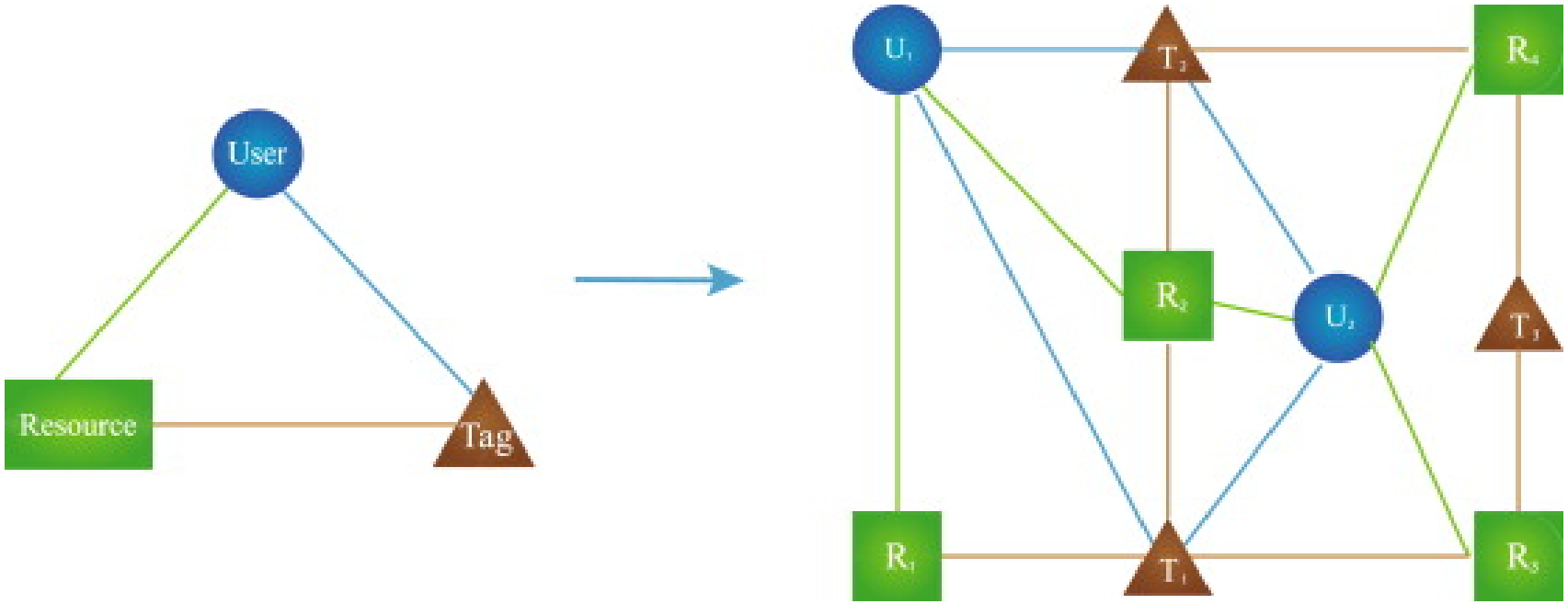
- supervised method
- best approach is to train for a specific set of documents, e.g.,
  - cluster company documents into invoices, contracts, purchase orders, delivery notes, other
  - cluster customer emails into customer complaints, request for contract end, relocation notice, other

Topic Modeling – Applications

- recommender systems
- document classification (one or more categories a document fits into)
- bio-informatics (interpret biological data)
- chatbots, topic tracking in dialogues
- document summarization (via topic names, a document is seen as a collection of topics, each with a weight)

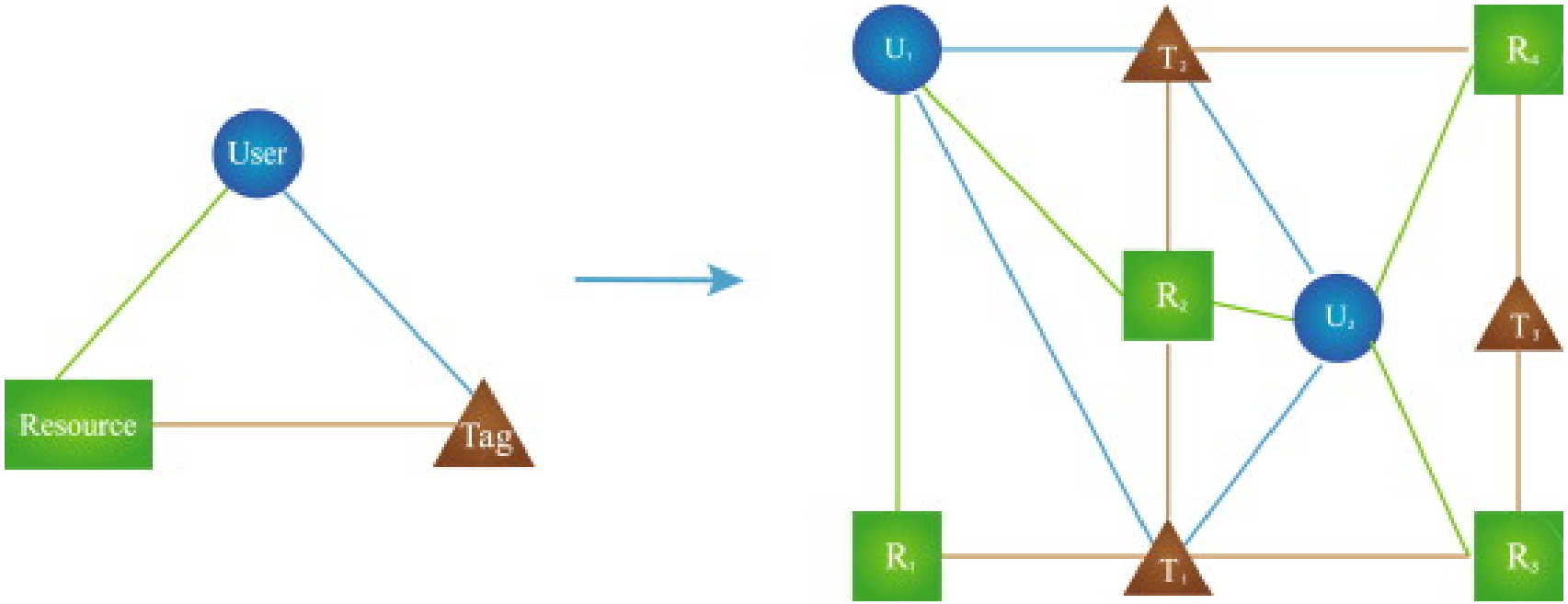
Recommender Systems

- recommend the best product for the user
- clusters of users, based on preference
- clusters of products



Recommender Systems

- recommend the best product for the user
- clusters of users, based on preference
- clusters of products

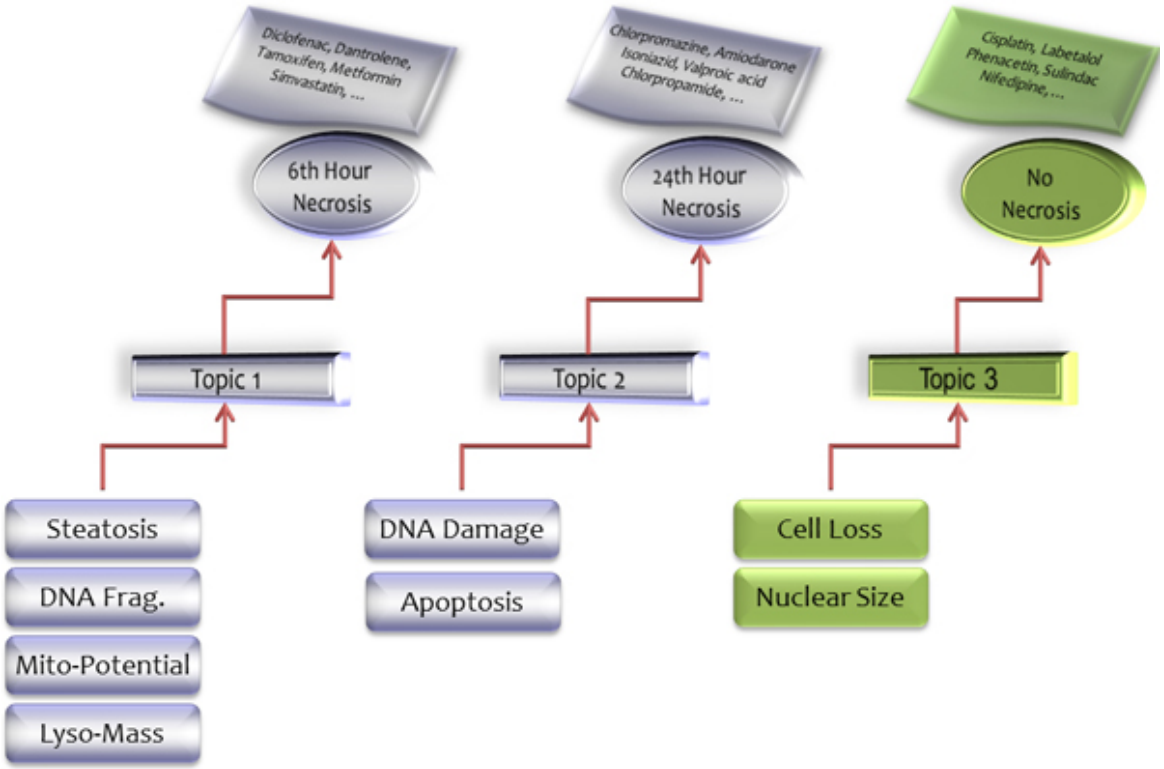


e.g. Netflix prize – \$1 million prize for improving Netflix recommender by 10%
training data set of 100,480,507 ratings that 480,189 users gave to 17,770 movies

Bio-informatics

applications of topic classification:

- categorize patients into risk groups based on text protocols
- detect common genomic features based on gene sequence data
- group drugs by diagnosis



Topic Modeling Approaches

- Latent Semantic Analysis, Latent Semantic Indexing (LSA/LSI) – matrix factorization
- Probabilistic Latent Semantic Analysis (pLSA) – probabilistic decomposition
- Latent Dirichlet Allocation (LDA) – iterative probabilistic method
- other decomposition techniques (e.g., Non-negative Matrix Factorization, NMF)
- other clustering techniques (e.g., K-means clustering of word vectors)

Latent Semantic Analysis

Works because the **distributional hypothesis** works.

> *. . . words that occur in the same contexts tend to have similar meanings*
>
> *(Harris, 1954)*[1]

LSA takes input of how frequently words occur in:

- documents
- the whole corpus

. . . and assumes that similar documents have similar **distribution of word frequencies**

(syntax + semantics are ignored)

[1] `https://aclweb.org/aclwiki/Distributional_Hypothesis`

LSA – step 1

- create **document-term matrix** (word frequencies in documents)
- rows = terms (words or multi-word expressions), columns = documents
- sparse matrix

*term**document*	D1	D2	D3	D4	D5	D6	D7	D8
abnormality	0	0	0	1	0	1	1	0
blood	0	1	1	2	1	0	1	1
culture	3	0	0	0	0	0	0	0
disease	0	2	3	0	1	1	0	0
rate	0	3	7	0	0	3	1	0

LSA – step 2

- **weighting** matrix elements
- most popular TF-IDF
  Term Frequency × Inverse Document Frequency

$$TF - IDF(t, d) = TF(t, d) \times \log(\frac{N}{n_t})$$

- term occurring in many documents is not interesting for analysis

$term \backslash document$	D1	D2	D3	D4	D5	D6	D7	D8
abnormality	0	0	0	1	0	1	1	0
blood	0	1	1	2	1	0	1	1
culture	3	0	0	0	0	0	0	0
disease	0	2	3	0	1	1	0	0
rate	0	3	7	0	0	3	1	0

LSA – step 2

- **weighting** matrix elements
- most popular TF-IDF
  Term Frequency × Inverse Document Frequency

$$TF - IDF(t, d) = TF(t, d) \times \log(\frac{N}{n_t})$$

- term occurring in many documents is not interesting for analysis

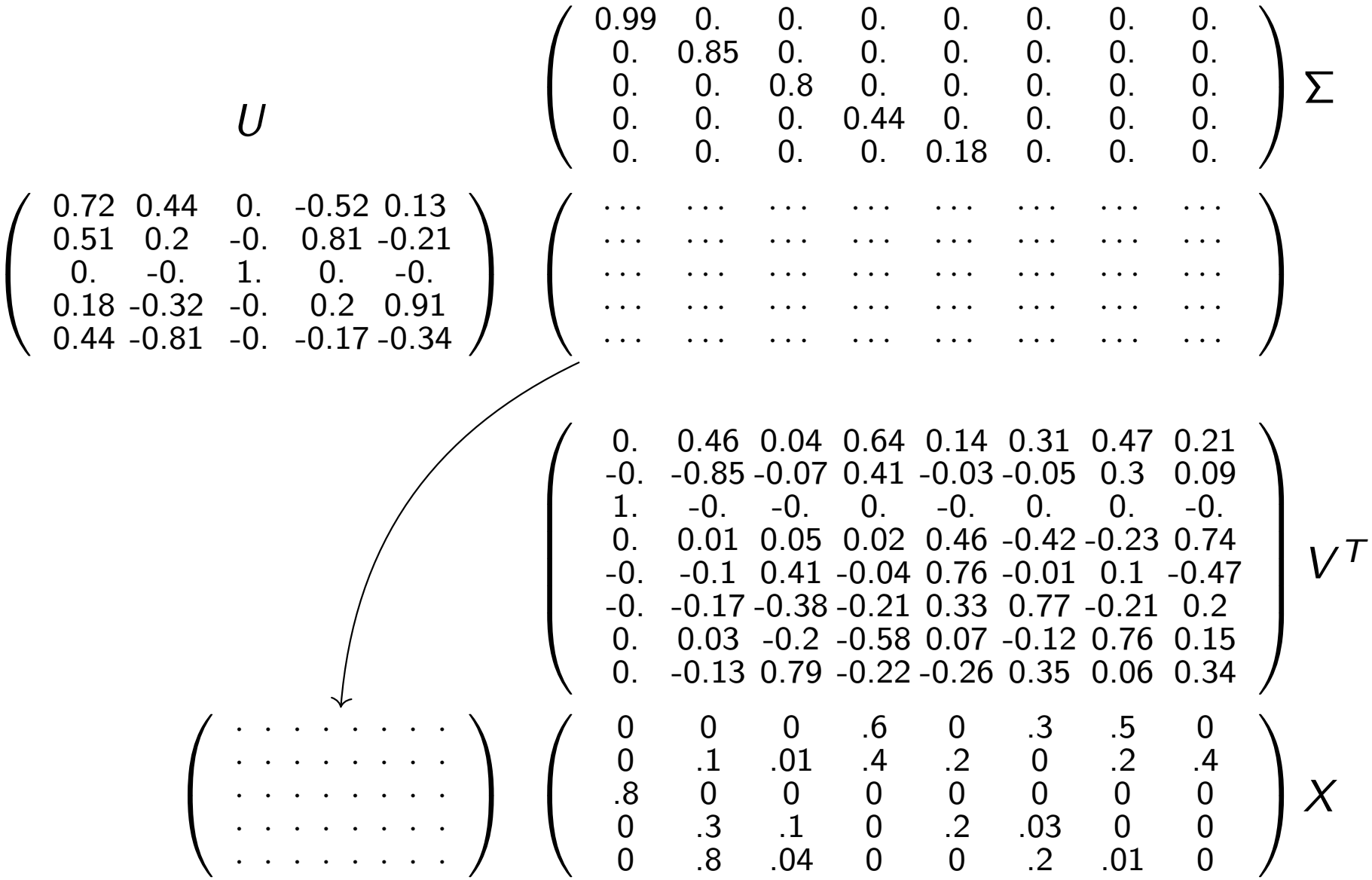
$term \backslash document$	D1	D2	D3	D4	D5	D6	D7	D8
abnormality	0	0	0	.6	0	.3	.5	0
blood	0	.1	.01	.4	.2	0	.2	.4
culture	.8	0	0	0	0	0	0	0
disease	0	.3	.1	0	.2	.03	0	0
rate	0	.8	.04	0	0	.2	.01	0

LSA – step 3

- Singular Value Decomposition (SVD), suitable decomposition for sparse data
  document-term matrix $(m \times n)$ is decomposed into the product of 3 matrices $X = U \cdot \Sigma \cdot V$, where
  - U – term-topic matrix $m \times m$
  - V – document-topic matrix $n \times n$
  - Σ – diagonal matrix

  U, V are unitary matrices ($U \cdot U^T = \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & 1 & \vdots \\ 0 & \cdots & 1 \end{pmatrix}$, identity matrix)

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



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

$$\Sigma = \begin{pmatrix} \mathbf{0.99} & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & \mathbf{0.85} & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & \mathbf{0.8} & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & \mathbf{0.44} & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & \mathbf{0.18} & 0. & 0. & 0. \end{pmatrix}$$

numbers on the diagonal – **singular values** of X
they express the strength of each (latent) topic.

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

term-topic matrix

$$U = \begin{matrix} \textbf{abnormality} \\ \textbf{blood} \\ \textbf{culture} \\ \textbf{disease} \\ \textbf{rate} \end{matrix} \begin{pmatrix} 0.72 & 0.44 & 0. & -0.52 & 0.13 \\ 0.51 & 0.2 & -0. & 0.81 & -0.21 \\ 0. & -0. & 1. & 0. & -0. \\ 0.18 & -0.32 & -0. & 0.2 & 0.91 \\ 0.44 & -0.81 & -0. & -0.17 & -0.34 \end{pmatrix}$$

each column represents a latent topic

each row – how the term relates to each topic

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

document-topic matrix

$$V^T = \begin{array}{c} \begin{array}{cccccccc} \textbf{D1} & \textbf{D2} & \textbf{D3} & \textbf{D4} & \textbf{D5} & \textbf{D6} & \textbf{D7} & \textbf{D8} \end{array} \\ \left(\begin{array}{cccccccc} 0. & 0.46 & 0.04 & 0.64 & 0.14 & 0.31 & 0.47 & 0.21 \\ -0. & -0.85 & -0.07 & 0.41 & -0.03 & -0.05 & 0.3 & 0.09 \\ 1. & -0. & -0. & 0. & -0. & 0. & 0. & -0. \\ 0. & 0.01 & 0.05 & 0.02 & 0.46 & -0.42 & -0.23 & 0.74 \\ -0. & -0.1 & 0.41 & -0.04 & 0.76 & -0.01 & 0.1 & -0.47 \\ -0. & -0.17 & -0.38 & -0.21 & 0.33 & 0.77 & -0.21 & 0.2 \\ 0. & 0.03 & -0.2 & -0.58 & 0.07 & -0.12 & 0.76 & 0.15 \\ 0. & -0.13 & 0.79 & -0.22 & -0.26 & 0.35 & 0.06 & 0.34 \end{array} \right) \end{array}$$

each column – how a document relates to the latent topics
absolute value gives the involvement, the sign indicates opposites within the topic

LSA – step 4

dimensionality reduction – throw away the least important topics
Keep first t **topics** – singular values (and therefore first t columns from U and first t rows from V)
we get a rank t approximation to X

$t = 3$ *topics*

$\sigma = (0.99, 0.85, 0.8)$, ~~*0.44*~~, ~~*0.18*~~

$$U = \begin{array}{l} \textit{abnormality} \\ \textit{blood} \\ \textit{culture} \\ \textit{disease} \\ \textit{rate} \end{array} \begin{pmatrix} 0.72 & 0.44 & 0. \\ 0.51 & 0.2 & -0. \\ 0. & -0. & 1. \\ 0.18 & -0.32 & -0. \\ 0.44 & -0.81 & -0. \end{pmatrix}$$

$$V = \begin{array}{c} \begin{array}{cccccccc} D1 & D2 & D3 & D4 & D5 & D6 & D7 & D8 \end{array} \\ \begin{pmatrix} 0. & 0.46 & 0.04 & 0.64 & 0.14 & 0.31 & 0.47 & 0.21 \\ -0. & -0.85 & -0.07 & 0.41 & -0.03 & -0.05 & 0.3 & 0.09 \\ 1. & -0. & -0. & 0. & -0. & 0. & 0. & -0. \end{pmatrix} \end{array}$$

LSA – step 4

dimensionality reduction – throw away the least important topics
Keep first t **topics** – singular values (and therefore first t columns from U and first t rows from V)
we get a rank t approximation to X

$t = 3$ *topics*

$\sigma = (0.99, 0.85, 0.8)$, ~~*0.44*~~, ~~*0.18*~~

$$U = \begin{array}{l} \textbf{\textit{abnormality}} \\ \textbf{\textit{blood}} \\ \textit{culture} \\ \textit{disease} \\ \textit{rate} \end{array} \begin{pmatrix} \textbf{0.72} & 0.44 & 0. \\ \textbf{0.51} & 0.2 & -0. \\ \textbf{0.} & -0. & 1. \\ \textbf{0.18} & -0.32 & -0. \\ \textbf{0.44} & -0.81 & -0. \end{pmatrix}$$

$$V = \begin{array}{c} D1 \quad D2 \quad D3 \quad D4 \quad D5 \quad D6 \quad D7 \quad D8 \end{array} \begin{pmatrix} \textbf{0.} & \textbf{0.46} & \textbf{0.04} & \textbf{0.64} & \textbf{0.14} & \textbf{0.31} & \textbf{0.47} & \textbf{0.21} \\ -0. & -0.85 & -0.07 & 0.41 & -0.03 & -0.05 & 0.3 & 0.09 \\ 1. & -0. & -0. & 0. & -0. & 0. & 0. & -0. \end{pmatrix}$$

LSA – step 4

dimensionality reduction – throw away the least important topics
Keep first t **topics** – singular values (and therefore first t columns from U and first t rows from V)
we get a rank t approximation to X

$t = 3$ *topics*

$\sigma = (0.99, 0.85, 0.8)$, ~~*0.44*~~, ~~*0.18*~~

$$U = \begin{array}{l} \textbf{\textit{abnormality}} \\ \textit{blood} \\ \textit{culture} \\ \textit{disease} \\ \textbf{\textit{rate}} \end{array} \begin{pmatrix} 0.72 & \textbf{\textit{0.44}} & 0. \\ 0.51 & \textbf{\textit{0.2}} & -0. \\ 0. & \textbf{\textit{-0.}} & 1. \\ 0.18 & \textbf{\textit{-0.32}} & -0. \\ 0.44 & \textbf{\textit{-0.81}} & -0. \end{pmatrix}$$

$$V = \begin{array}{c} D1 \quad \textbf{\textit{D2}} \quad D3 \quad \textbf{\textit{D4}} \quad D5 \quad D6 \quad D7 \quad D8 \end{array} \begin{pmatrix} 0. & 0.46 & 0.04 & 0.64 & 0.14 & 0.31 & 0.47 & 0.21 \\ \textbf{\textit{-0.}} & \textbf{\textit{-0.85}} & \textbf{\textit{-0.07}} & \textbf{\textit{0.41}} & \textbf{\textit{-0.03}} & \textbf{\textit{-0.05}} & \textbf{\textit{0.3}} & \textbf{\textit{0.09}} \\ 1. & -0. & -0. & 0. & -0. & 0. & 0. & -0. \end{pmatrix}$$

LSA – step 4

dimensionality reduction – throw away the least important topics
Keep first t **topics** – singular values (and therefore first t columns from U and first t rows from V)
we get a rank t approximation to X

$t = 3$ *topics*

$\sigma = (0.99, 0.85, 0.8)$, ~~*0.44*~~, ~~*0.18*~~

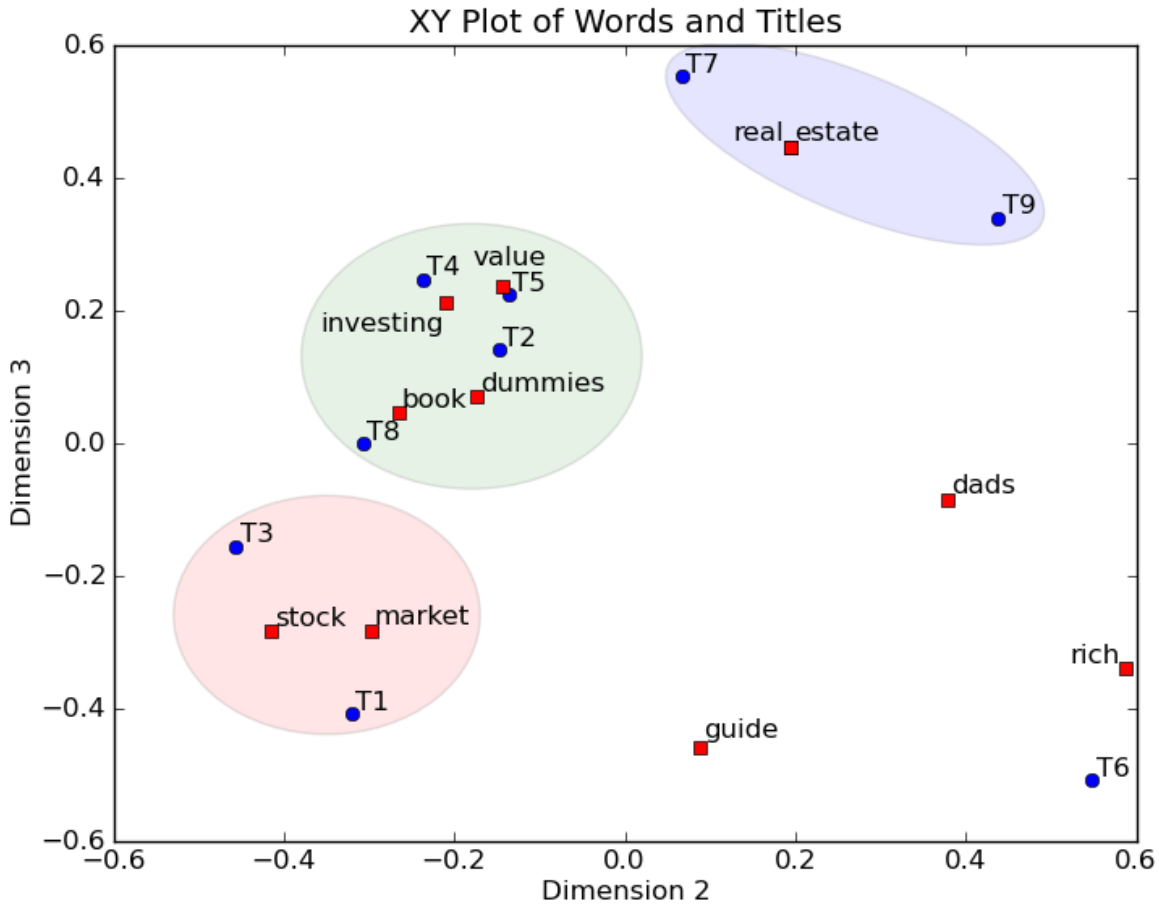
$$U = \begin{array}{l} \textit{abnormality} \\ \textit{blood} \\ \textbf{\textit{culture}} \\ \textit{disease} \\ \textit{rate} \end{array} \begin{pmatrix} 0.72 & 0.44 & \textbf{\textit{0.}} \\ 0.51 & 0.2 & \textbf{\textit{-0.}} \\ 0. & -0. & \textbf{\textit{1.}} \\ 0.18 & -0.32 & \textbf{\textit{-0.}} \\ 0.44 & -0.81 & \textbf{\textit{-0.}} \end{pmatrix}$$

$$V = \begin{array}{c} \textbf{\textit{D1}} \\ \\ \\ \\ \\ \\ \\ \end{array} \begin{array}{c} D2 \\ \\ \\ \\ \\ \\ \\ \end{array} \begin{array}{c} D3 \\ \\ \\ \\ \\ \\ \\ \end{array} \begin{array}{c} D4 \\ \\ \\ \\ \\ \\ \\ \end{array} \begin{array}{c} D5 \\ \\ \\ \\ \\ \\ \\ \end{array} \begin{array}{c} D6 \\ \\ \\ \\ \\ \\ \\ \end{array} \begin{array}{c} D7 \\ \\ \\ \\ \\ \\ \\ \end{array} \begin{array}{c} D8 \\ \\ \\ \\ \\ \\ \\ \end{array}$$

$$\begin{pmatrix} 0. & 0.46 & 0.04 & 0.64 & 0.14 & 0.31 & 0.47 & 0.21 \\ -0. & -0.85 & -0.07 & 0.41 & -0.03 & -0.05 & 0.3 & 0.09 \\ \textbf{\textit{1.}} & \textbf{\textit{-0.}} & \textbf{\textit{-0.}} & \textbf{\textit{0.}} & \textbf{\textit{-0.}} & \textbf{\textit{0.}} & \textbf{\textit{0.}} & \textbf{\textit{-0.}} \end{pmatrix}$$

LSA – step 5

cluster close vectors (documents and terms)



Latent Semantic Analysis

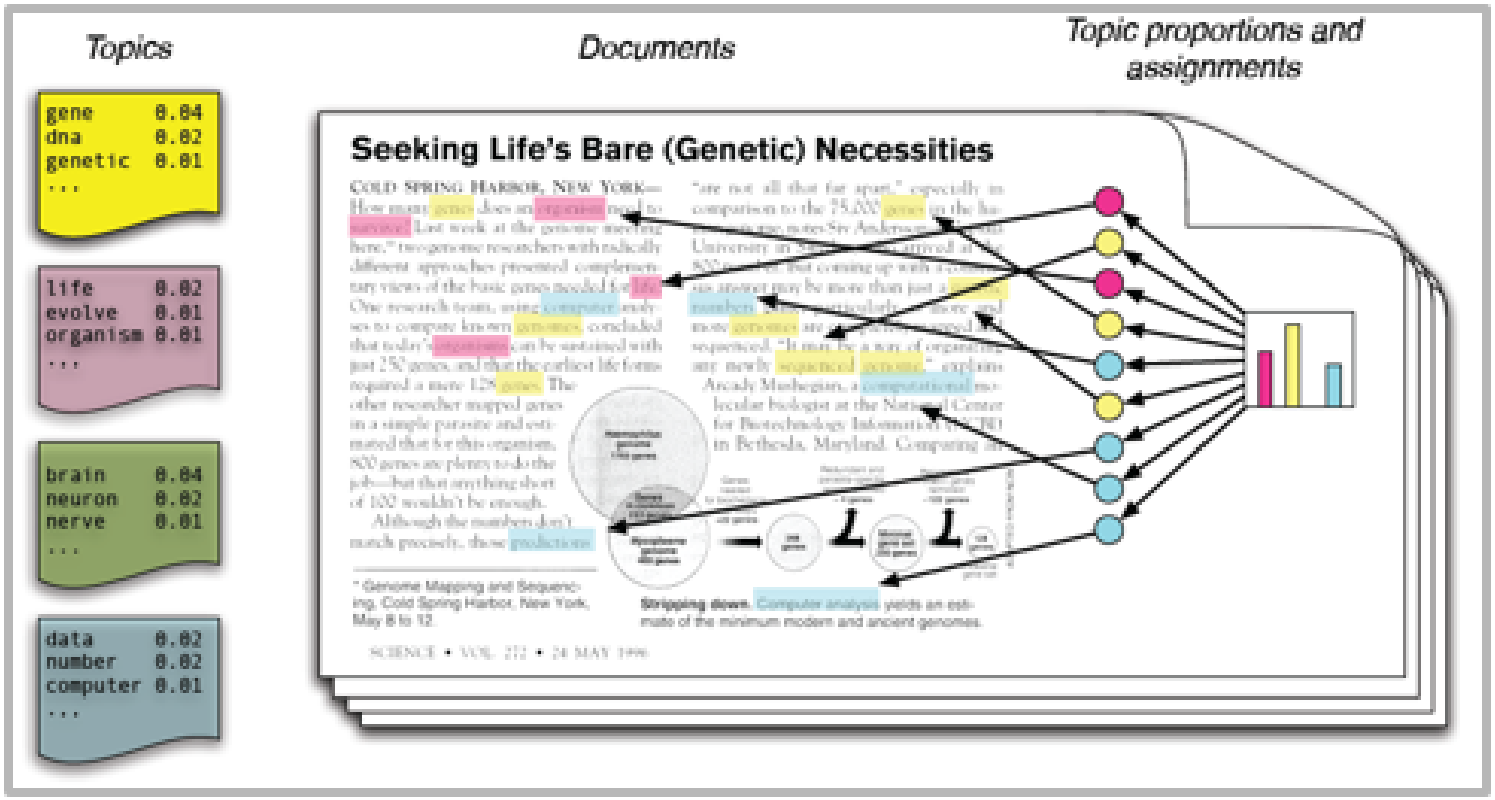
summary

- document = bag of words
- computes t latent topics
- the topics are expresses as **vector representations** of words and/or documents
- allows to cluster documents with similar topic

see (Ioana, 2020) for detailed explanation.

Latent Dirichlet Allocation

- same assumptions as in LSA (distributional hypothesis + mixture of topics in one document) but using **generation**
- each document is a **mixture of topics** = distributions over words
- **LDA** discovers topics and their ratio by modeling how documents could have been produced
- each word in document was generated by one of the topics



Example

Document 1: I like to eat **broccoli** and **bananas**.
Document 2: I ate a **banana** and spinach smoothie for **breakfast**.
Document 3: Chinchillas and kittens are cute.
Document 4: My sister adopted a kitten yesterday.
Document 5: Look at this cute hamster **munching** on a piece of **broccoli**.

Example

Topic A: 30% broccoli, 15% bananas, 10% breakfast, 10% munching
Topic B: 20% chinchillas, 20% kittens, 20% cute, 15% hamster

Example

Document 1 and 2: 100% **Topic A**
Document 3 and 4: 100% Topic B
Document 5: 53% **Topic A**, 47% Topic B

LDA outputs compared to LSA

with both approaches we obtain term-topic relations

LSA: term–topic vectors

- Singular Value Decomposition of the term–document matrix
- a **topic** = dimension (principal component) in a latent semantic space
- term-topic weights are not probabilities
- interpreting a topic = looking at the terms with highest absolute weights

LDA: topic–word distributions

- obtained by fitting a probabilistic generative model to documents
- each **topic** is an explicit probability distribution over the vocabulary
- term weights = probabilities (non-negative and sum to 1 for each topic)
- topics are interpretable clusters of semantically related words:

> If you were talking about this topic, these are the probabilities with which you would use each word.

LDA process

- pick **fixed number of topics** K
- for each document $d \in D$, randomly assign topic to each word
- improve, for each document d, word w and topic t:
  - assume all topic assignments are correct, except for current word
  - calculate $p(\textit{topic } t | \textit{document } d)$ – how many words in document have topic t?
  - calculate $p(\textit{word } w | \textit{topic } t)$ – how many times word w appears in topic t?
  - new topics = $p(\textit{topic } t | \textit{document } d; \alpha) \times p(\textit{word } w | \textit{topic } t; \beta)$
- repeat and reach almost steady state

LDA parameters

parameters K, α, and β
parametrized vector computations of topics and documents
(α and β are concentration parameters)

low $\alpha \rightarrow$ fewer topics are assigned to a document

low $\beta \rightarrow$ fewer words model a topic

LDA outputs

topical characteristic of the document colection D:

- a distribution of words for each topic $t \in K$
  $t_1 = [0.1, 0.0, 0.4, 0.1, \ldots]$
  $t_2 = [0.0, 0.2, 0.2, 0.0, \ldots]$
  $\ldots$
- a distribution of topics for each document $d \in D$
  vector containing coverage of every topic for the document
  $d_1 = [0.3, 0.4, 0.1, \ldots]$
  $d_2 = [0.2, 0.3, 0.3, \ldots]$
  $\ldots$

LSA and LDA – practical assumptions

- preprocessing: lowercase, punctuation removal, stopword removal, (stemming or lemmatization)
- both LDA and LSA ignore the syntactic structure
- the number of topics K is the input parameter
- LDA assumes probabilistic distributions of words in topics and of topics in documents $\rightarrow$ topics are more interpretable
- output: wordcloud
- human readable topic labels are difficult (and not part of LSA/LDA)

Weaknesses of LSA and LDA

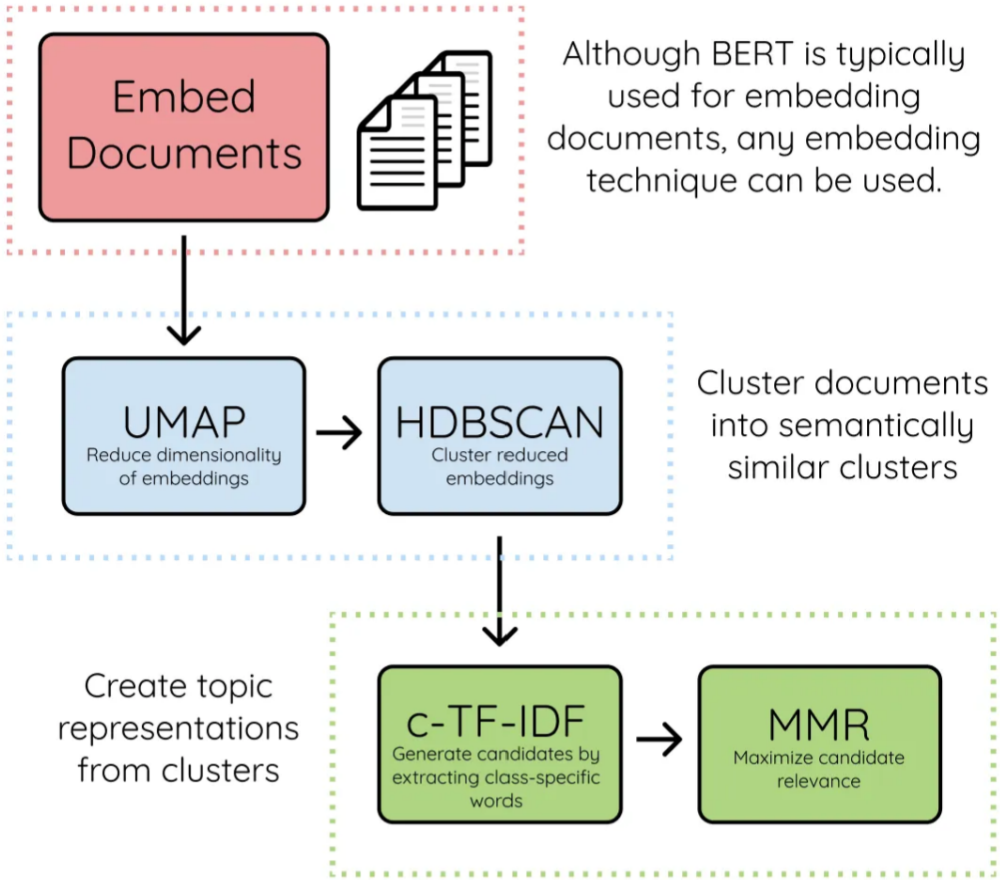
- how to choose number of topics K
- how to select vocabulary – stopwords, stemming/lemmatization
- how to interpret the topics
- ignoring text structure

some solutions:

- Hierarchical Dirichlet Process (HDP) – number of topics learnt from data
- top2vec: word + document embeddings (Angelov, 2020) – captures the document semantics using word embeddings
- BERTopic – c-TF-IDF (class-based TF-IDF) + embeddings + document structure – clustering and topic names

BERTopic

- input: documents + embeddings, output: clusters/topics + cluster keywords
- adjustable – reduce number of topics, topic hierarchy, predefined topic names



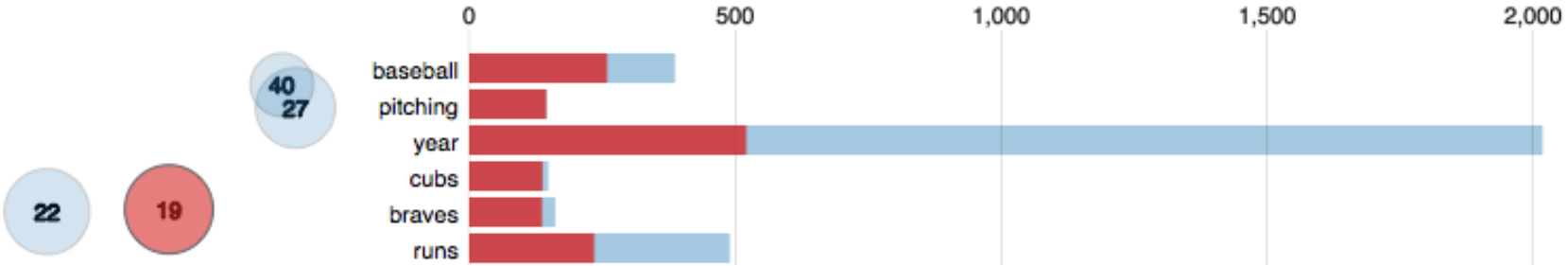
see `https://medium.com/data-reply-it-datatech/bertopic-topic-modeling-as-you-have-never-seen-it-before-abb48bbab2b2`

Topic Evaluation Methods

Good topics = **interpretable** topics

Evaluation methods:

- human judgement – top terms, visual inspection (`pyLDAVis`)
- intrinsic methods – perplexity (on *test*), coherence measures
- extrinsic methods – how does the resulting model influence subsequent task



Topic Coherence

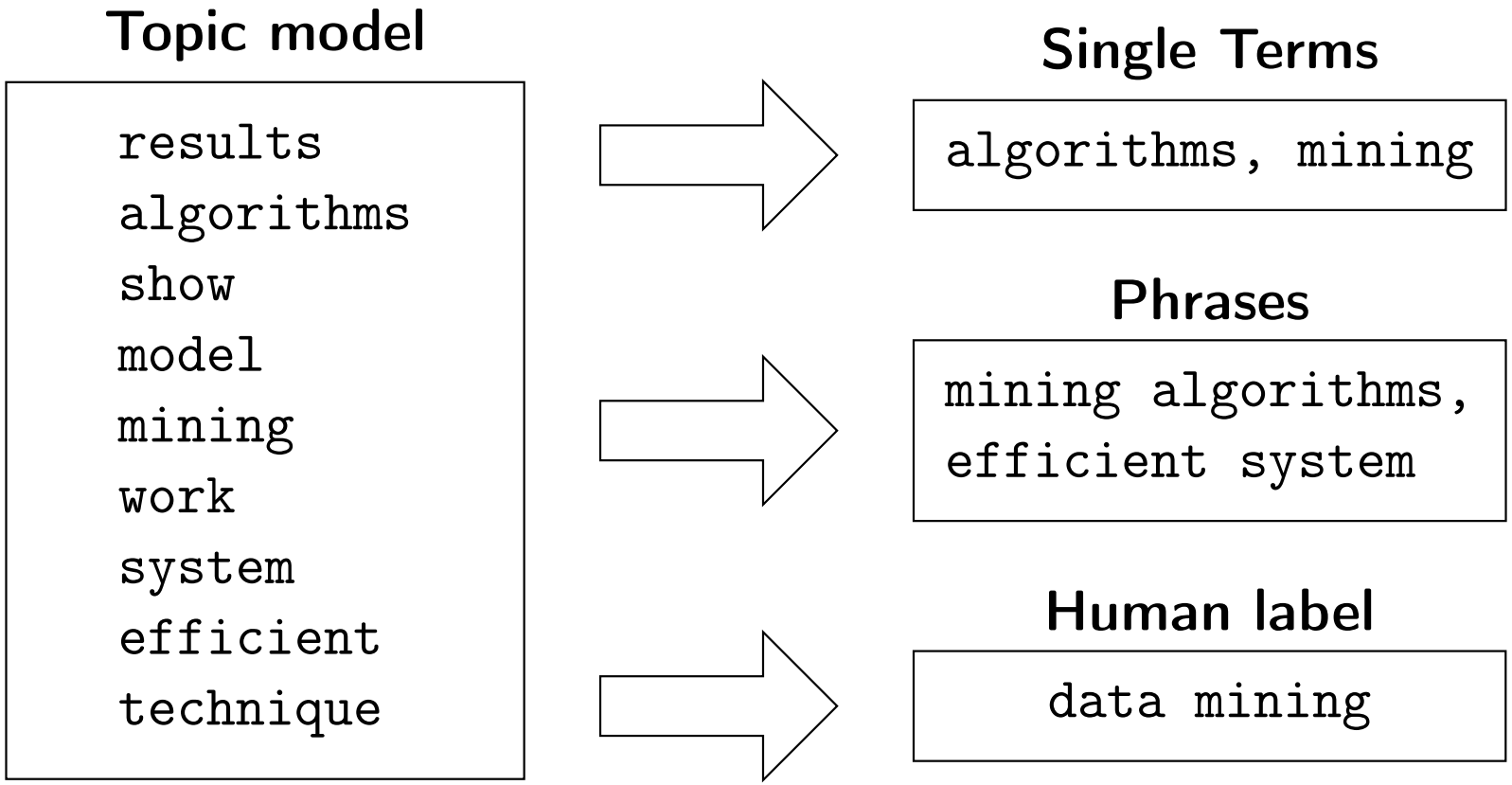
Algorithms and interpretation:

- **UMass**:
  - co-occurrence counts and conditional probability
  - values between -14 and 14
  - the greater number, the better.
  - easier to implement than C_V
- **C_V**:
  - co-occurrence counts in a sliding window and normalized point-wise mutual information (NPMI)
  - values between 0 and 1
  - the greater number, the better
  - higher correlation to human judgment

Topic Labeling

represent topic with **human-friendly label**

- straightforward – choose the most salient words
- better interpretation – find more general terms
  - find Wikipedia articles based on word list
  - find centers of word semantic clusters
  - document summarization from topic documents

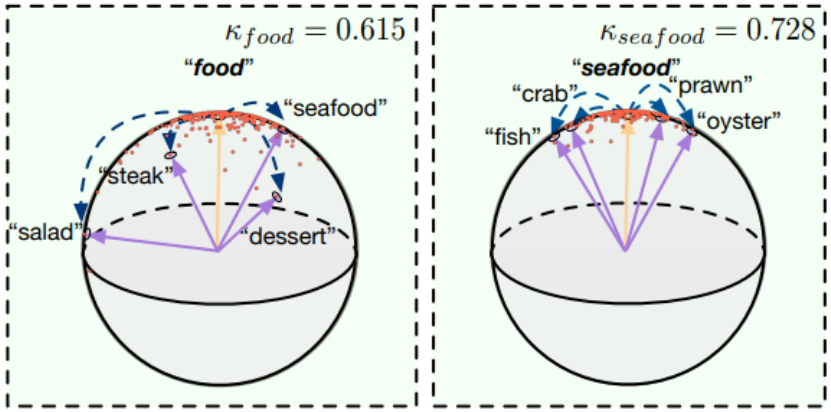


Topic labels from word embeddings

Word Distributional Specificity

for each word w learn a distributional specificity score $\kappa_w \geq 0$ – how specific the word meaning is

the bigger κ_w, the more specific meaning word w has



food = less specific, $\kappa_w = 0.615$

seafood = more specific, $\kappa_w = 0.728$

selecting Category Representative Words:

- high meaning vector similarity
- low distributional specificity

see (Meng et al., 2020)

References I

Angelov, D. (2020). Top2vec: Distributed representations of topics.

Blair, S. J., Bi, Y., and Mulvenna, M. D. (2020). Aggregated topic models for increasing social media topic coherence. *Applied Intelligence*, 50(1):138–156.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993 – 1022.

Castellanos, A., Cigarrn, J., and Garca-Serrano, A. (2017). Formal concept analysis for topic detection. *Inf. Syst.*, 66(C):24–42.

Curiskis, S. A., Drake, B., Osborn, T. R., and Kennedy, P. J. (2020). An evaluation of document clustering and topic modelling in two online social networks: Twitter and reddit. *Information Processing & Management*, 57(2):102034.

Grootendorst, M. (2022). Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

References II

Ioana (2020). Latent semantic analysis: intuition, math, implementation. Available at `https://towardsdatascience.com/latent-semantic-analysis-intuition-math-implementation-a194aff870f8`.

Lim, K. H., Karunasekera, S., and Harwood, A. (2017). Clustop: A clustering-based topic modelling algorithm for twitter using word networks. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 2009–2018. IEEE.

Lü, L., Medo, M., Yeung, C. H., Zhang, Y.-C., Zhang, Z.-K., and Zhou, T. (2012). Recommender systems. *Physics Reports*, 519(1):1 – 49. Recommender Systems.

Meng, Y., Huang, J., Wang, G., Wang, Z., Zhang, C., Zhang, Y., and Han, J. (2020). Discriminative topic mining via category-name guided text embedding. In *Proceedings of The Web Conference 2020*, WWW '20, page 2121–2132, New York, NY, USA. Association for Computing Machinery.

References III

Röder, M., Both, A., and Hinneburg, A. (2015). Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, pages 399–408.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2006). Hierarchical Dirichlet processes . *Journal of the American Statistical Association*, 101:1566 – 1581.

Wan, X. and Wang, T. (2016). Automatic labeling of topic models using text summaries. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2297–2305.

Xie, P. and Xing, E. P. (2013). Integrating document clustering and topic modeling. *CoRR*, abs/1309.6874.