- Jurafsky, D. and Martin, J. H. (2018) Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Third Edition, Draft: Chapter 6
- D. Blei. Introduction to probabilistic topic models. Communications of the ACM, 2012
- Mark Steyvers; Tom Griffiths (2007) "Probabilistic Topic Models" In: T. Landauer, D McNamara, S. Dennis, and W. Kintsch (eds), Handbook of Latent Semantic Analysis, Psychology Press.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. ICLR Workshop, 2013.
- IR Book Chapter 18: https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf

LSA, Generative Topic Models (LDA), Neural Word Embeddings (word2vec)

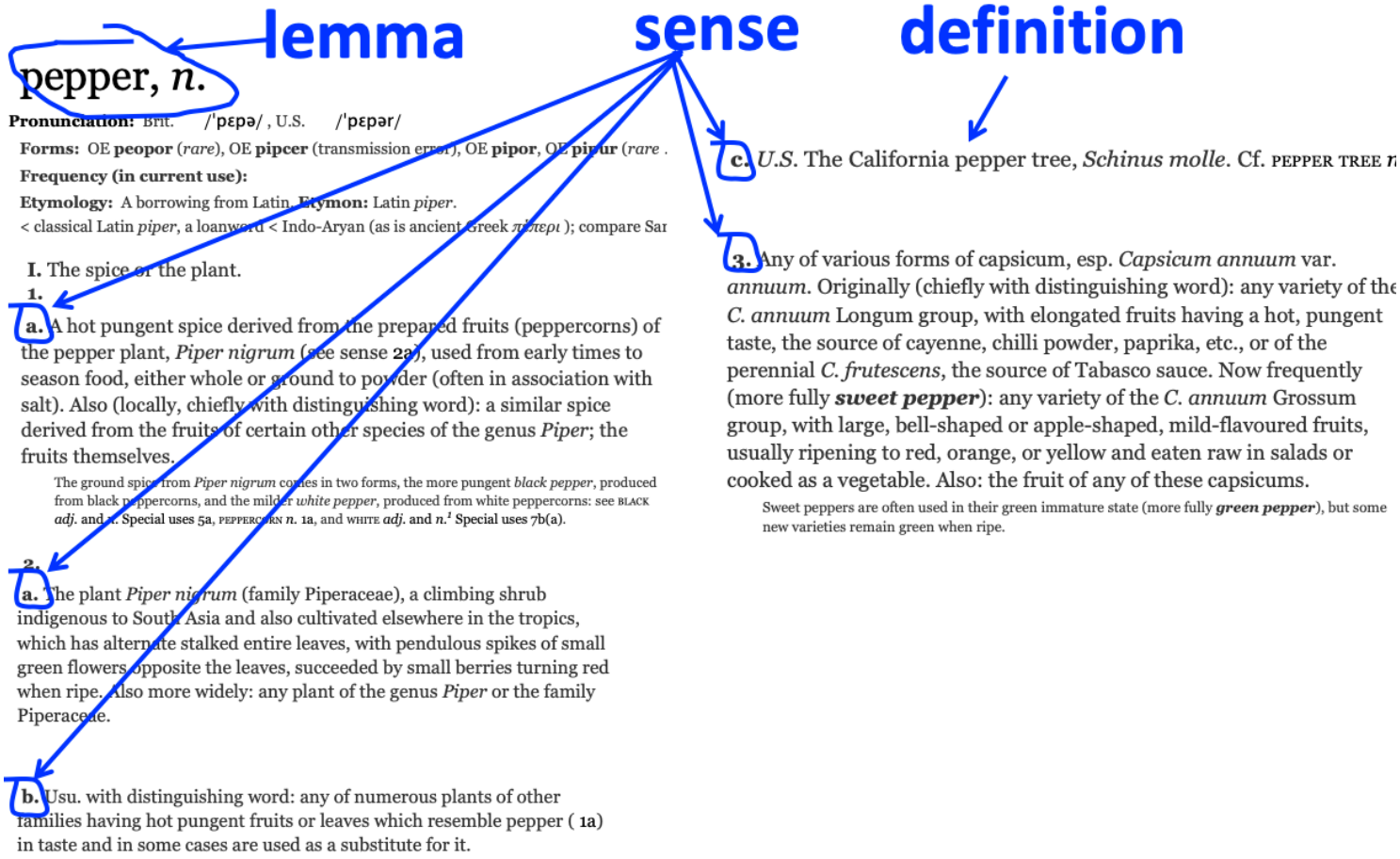
DENSE VECTOR REPRESENTATIONS

Slides credit: Martin Riedl, Alexander Panchenko

WHAT DO WORDS MEAN?

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

- First thought: look in a dictionary, e.g.: http://www.oed.com



ASK HUMANS HOW SIMILAR TWO WORDS ARE

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

word1	word2	similarity
vanish	disappear	9.8
behave	obey	7.3
belief	impression	5.95
muscle	bone	3.65
modest	flexible	0.98
hole	agreement	0.3

SimLex-999 dataset (Hill et al., 2015)

Felix Hill, Roi Reichart and Anna Korhonen (2015): SimLex-999: Evaluating Semantic Models with (Genuine) Similarity Estimation. Computational Linguistics 41(4): 665-695

CLASSICAL ("ARISTOTELIAN") THEORY OF CONCEPTS

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

The meaning of a word: a concept defined by **necessary** and **sufficient** conditions

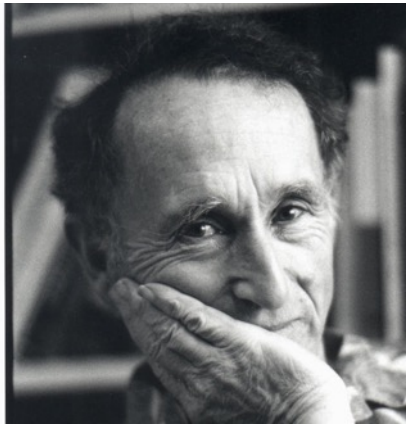
- A **necessary** condition for being an X is a condition C that X must satisfy in order for it to be an X.
  - If not C, then not X
  - "Having four sides" is necessary to be a square.

- A **sufficient** condition for being an X is condition such that if something satisfies condition C, then it must be an X.
  - If and only if C, then X
  - The following necessary conditions, jointly, are sufficient to be a square
    - x has (exactly) four sides
    - each of x's sides is straight
    - x is a closed figure
    - x lies in a plane
    - each of x's sides is equal in length to each of the others
    - each of x's interior angles is equal to the others (right angles)
    - the sides of x are joined at their ends

WILLIAM LABOV'S (1975) DEFINITION OF CUP

The term *cup* is used to denote round containers with a ratio of depth to width of $1 \pm r$ where $r \leq r_b$, and $r_b = \alpha_1 + \alpha_2 + \ldots \alpha_\upsilon$ and α_1 is a positive quality when the feature i is present and 0 otherwise.

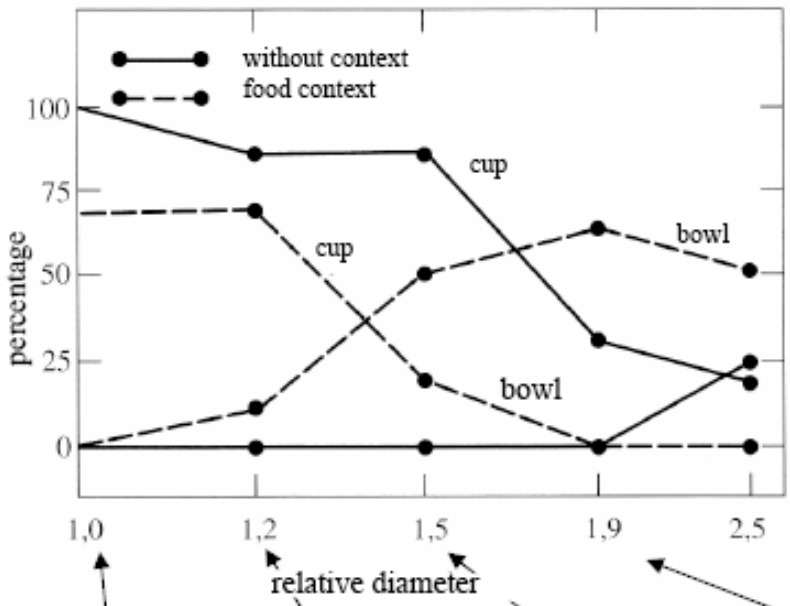
feature
1 = with one handle
2 = made of opaque vitreous material
3 = used for consumption of food
4 = used for the consumption of liquid food
5 = used for consumption of hot liquid food
6 = with a saucer
7 = tapering
8 = circular in cross-section



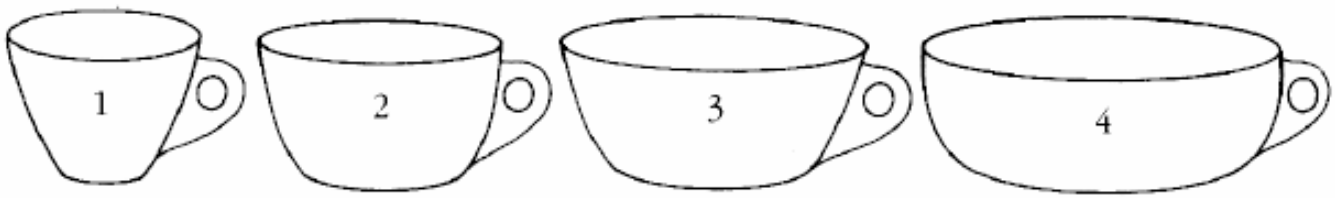
Cup is used variably to denote such containers with ratios width to depth $1 \pm r$ where $r_b \leq r \leq r_1$ with a probability of $r_1 - r/r_t - r_b$. The quantity $1 \pm r_b$ expresses the distance from the modal value of width to height.

THE CATEGORY DEPENDS ON THE CONTEXT! IF THERE IS FOOD IN IT, IT'S A BOWL

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG



Boundaries between cups and bowls are context sensitive

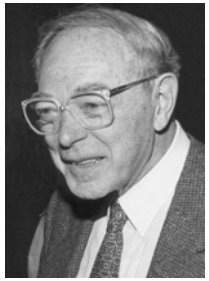


"DATA-DRIVEN APPROACH" TO DERIVATION OF WORD MEANING

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

- Ludwig Wittgenstein (1945): "**The meaning of a word is its use in the language**"

- Zellig Harris (1954): "**If A and B have almost identical environments we say that they are synonyms**"



- John Firth (1957): "**You shall know the word by the company it keeps.**"

WHAT DOES “ONG CHOI” MEAN?

WHAT DOES "ONG CHOI" MEAN?

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Suppose you see these sentences:

- Ong choi is delicious **sautéed with garlic**.
- Ong choi is superb **over rice**
- Ong choi **leaves** with salty sauces ..

WHAT DOES "ONG CHOI" MEAN?

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Suppose you see these sentences:

- Ong choi is delicious **sautéed with garlic**.
- Ong choi is superb **over rice**
- Ong choi **leaves** with salty sauces

- And you've also seen these:
  - ...spinach **sautéed with garlic over rice**
  - Chard stems and **leaves** are **delicious**
  - Collard greens and other **salty** leafy greens

WHAT DOES "ONG CHOI" MEAN?

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Suppose you see these sentences:

- Ong choi is delicious **sautéed with garlic**.
- Ong choi is superb **over rice**
- Ong choi **leaves** with salty sauces

- And you've also seen these:
  - ...spinach **sautéed with garlic over rice**
  - Chard stems and **leaves** are **delicious**
  - Collard greens and other **salty** leafy greens

- Conclusion:
  - Ong choi is a leafy green like spinach, chard, or collard greens

ONG CHOI: IPOMOEA AQUATICA "WATER SPINACH"

Yamaguchi, Wikimedia Commons, public domain

REPRESENTATION OF DOCUMENTS: THE VECTOR SPACE MODEL (VSM)

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

- (a.k.a. term-document matrix in Information Retrieval)
- word vectors: characterizing word with the documents they occur in
- document vectors: characterizing documents with their words

Documents

Words	d1	d2	…	di	…	dn
w1						
w2						
…						
wj				n(di,wj)		
…						
wm						

n(di, wj):= (number of words wj in document di) * term weighting

MORE COMMON: WORD-WORD (TERM-CONTEXT) MATRIX

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

- Two words are similar in meaning if their context vectors are similar

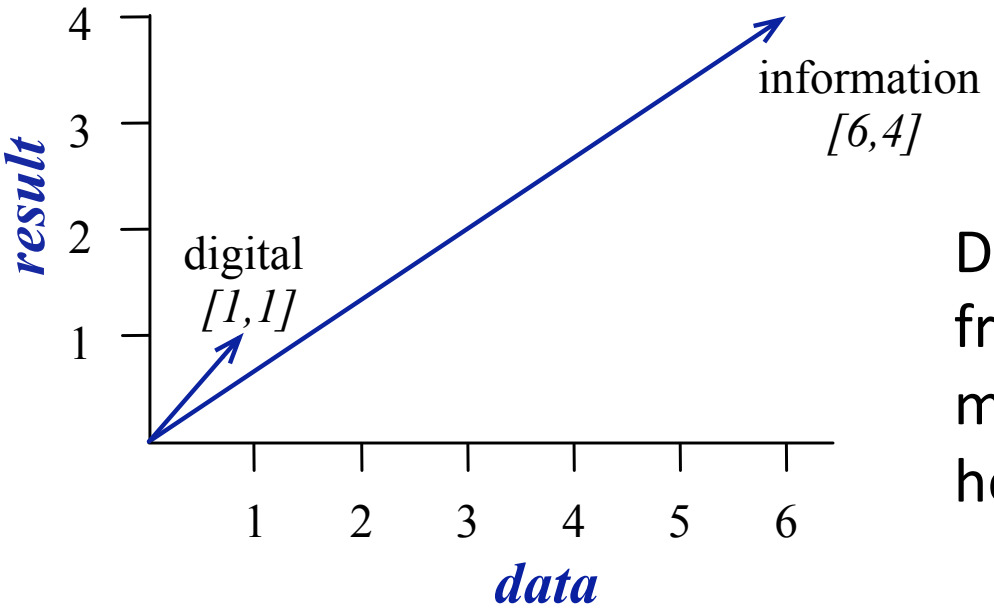
sugar, a sliced lemon, a tablespoonful of **apricot** jam, a pinch each of,
their enjoyment. Cautiously she sampled her first **pineapple** and another fruit whose taste she likened
well suited to programming on the digital **computer**. In finding the optimal R-stage policy from
for the purpose of gathering data and **information** necessary for the study authorized in the

	aardvark	computer	data	pinch	result	sugar	...
apricot	0	0	0	1	0	1	
pineapple	0	0	0	1	0	1	
digital	0	2	1	0	1	0	
information	0	1	6	0	4	0	

MORE COMMON: WORD-WORD (TERM-CONTEXT) MATRIX

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

	aardvark	computer	data	pinch	result	sugar	…
apricot	0	0	0	1	0	1	
pineapple	0	0	0	1	0	1	
digital	0	2	1	0	1	0	
information	0	1	6	0	4	0	



Do raw frequencies make sense here?

REMINDERS FROM LINEAR ALGEBRA

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

$$\text{dot-product}(\vec{v},\vec{w}) = \vec{v}\cdot\vec{w} = \sum_{i=1}^{N} v_i w_i = v_1 w_1 + v_2 w_2 + \ldots + v_N w_N$$

$$\text{vector length} \quad |\vec{v}| = \sqrt{\sum_{i=1}^{N} v_i^2}$$

$$\text{cosine}(\vec{v},\vec{w}) = \frac{\vec{v}\cdot\vec{w}}{|\vec{v}||\vec{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$$

POINTWISE MUTUAL INFORMATION

- Ask whether a context word is **particularly informative** about the target word.

Pointwise mutual information:

Do events x and y co-occur more than if they were independent?

$$\mathrm{PMI}(X,Y) = \log_2 \frac{P(x,y)}{P(x)P(y)}$$

PMI between two words:

Do words x and y co-occur more than if they were independent?

$$\mathrm{PMI}(word_1,word_2) = \log_2 \frac{P(word_1,word_2)}{P(word_1)P(word_2)}$$

POSITIVE POINTWISE MUTUAL INFORMATION

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

- PMI ranges from -∞ to +∞
- But the negative values are problematic
  - Things are co-occurring **less than** we expect by chance
  - Unreliable without enormous corpora
    - Imagine w1 and w2 whose probability is each 10^{-6}
    - Hard to be sure p(w1,w2) is significantly different than 10^{-12}
  - "unrelatedness" is a strange notion, given the size of the vocabulary; how does tiger-window vs. wall-tree compare?
- Positive PMI (PPMI) between word1 and word2:

$$\text{PPMI}(word_1, word_2) = \max\left(\log_2 \frac{P(word_1, word_2)}{P(word_1)P(word_2)}, 0\right)$$

WEIGHTING PMI: GIVING RARE WORDS HIGHER PROBABILITY

- Raise the context probabilities to e.g. α=0.75:

$$\text{PPMI}_\alpha(w,c) = \max(\log_2 \frac{P(w,c)}{P(w)P_\alpha(c)}, 0)$$

$$P_\alpha(c) = \frac{count(c)^\alpha}{\sum_c count(c)^\alpha}$$

- This helps because $P_\alpha(c) > P(c)$ for rare c

- Consider two events, P(a) = .99 and P(b)=.01

$$P_\alpha(a) = \frac{.99^{.75}}{.99^{.75}+.01^{.75}} = .97 \qquad P_\alpha(b) = \frac{.01^{.75}}{.99^{.75}+.01^{.75}} = .03$$

CHALLENGES IN THE PLAIN VSM

PPMI and other plain vectors are long (length |V|= 20,000 to 50,000) and sparse (most elements are zero)

- High dimensionality
  - large storage needed
  - slow processing
- Sparse Matrix
  - Power Law Distribution of frequencies: many words occur only once
  - many zeros
- Lack of Generalization
  - different words are distinct symbols, no matter how similar in meaning
  - e.g. synonyms are represented as different words, but should they be not mapped onto the same concept?

LSA – LATENT SEMANTIC ANALYSIS

WHAT IF THE MATRIX WAS SMALLER?

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

- Map Vector Space Model to a fixed lower number of dimensions
- Mapping reflects semantic association
  - words with similar context should have a similar profile in the reduced dimension space

➔Map terms with similar meaning (in best case synonyms) to similar locations in a low dimensional space

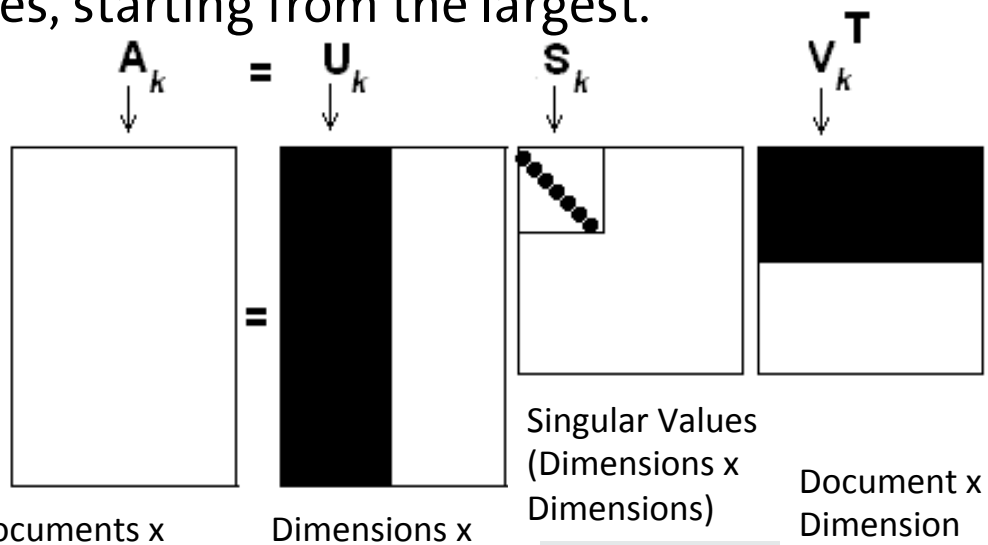
➔Extract main dimensions of meaning and remove noisy dimensions

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Richard Harshman (1990): Indexing by latent semantic analysis. Journal of the Association for Information Science and Technology 41(6):391-407

SINGULAR VALUE DECOMPOSITION (SVD)

THE METHOD USED BY LSA

- term-document matrix **A** can be decomposed by Singular Value Decomposition (SVD) into:
  $\mathbf{A} = \mathbf{U} \bullet \mathbf{S} \bullet \mathbf{V}^T$ where
  - **U**: columns are the Eigenvectors of $\mathbf{A} \bullet \mathbf{A}^T$ ("left" EVs)
  - **S**: diagonal matrix of *singular values*: square roots of Eigenvalues
  - **V**: columns are the "right" Eigenvectors of $\mathbf{A}^T \bullet \mathbf{A}$ ("right" EVs)
- can order the singular values in decreasing order. There are methods to iteratively compute singular values, starting from the largest.
- approximating the original matrix by the k most important dimensions (topics) in the SVD representation:
  $\mathbf{A}_k = \mathbf{U}_k \bullet \mathbf{S}_k \bullet \mathbf{V}_k^T$
  effectively, set all other values of **s** to 0.

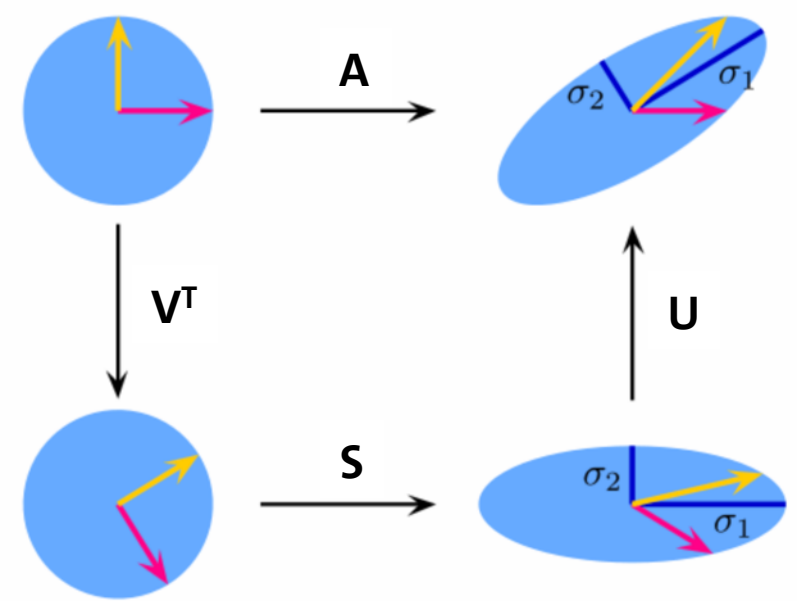


SVD EXPLAINED

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

$A = U \cdot S \cdot V^T$

- U: rotate the axes into the direction of maximal variance
- S: rescale the axes to achieve equal variance
- V : rotate the input vectors according to the new axes



https://technologiesrunning.blogspot.de/2016/12/10-algoritmos-de-aprendizaje-automatico.html

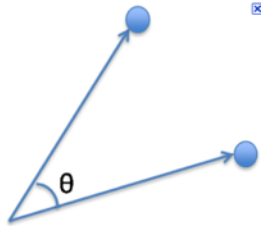
SVD is an orthogonal transform that de-correlates the variables.

Approximation to k dimensions keeps the ones with the largest variance.

Retain only U for word representations

LSA IN PRACTICE

$$sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$



- Compare words, compare documents
  - VSM: e.g. compute similarity between document vectors or word vectors using the cosine similarity
  - In LSA: compare reduced word vectors; project document vector into reduced document vector, then use cosine similarity
- Building LSA matrices is typically done using a large background corpus with k=100~1000 dims (subject to hyperparameter tuning)

Why this works:

- LSA can generalize over similar words
- LSA serves as a noise reduction technique
- LSA can fold in knowledge about lexical similarities beyond what is learnable from smaller labelled/annotated training data

EXAMPLE CORPUS

words

documents

b	b	b	b	m	m	m	m	m	m	l	l	l	l	l	l
b	b	b	b	b	m	m	m	m	m	m	m	l	l	l	l
b	b	b	b	b	b	b	m	m	m	m	m	l	l	l	l
b	b	b	b	b	b	b	m	m	m	m	m	m	l	l	l
b	b	b	b	b	b	b	m	m	l	l	l	l	l	l	l
b	b	b	b	b	b	b	b	b	m	m	m	l	l	l	l
r	b	b	b	b	m	m	m	m	m	m	m	l	l	l	l
r	s	s	b	b	b	b	b	b	m	m	m	m	l	l	l
r	s	s	s	b	b	b	b	b	b	m	m	m	m	l	l
r	r	s	s	s	b	b	b	b	b	b	m	l	l	l	l
r	r	s	s	s	b	b	b	b	b	b	b	m	m	m	l
r	r	r	s	s	s	s	s	s	b	b	b	b	b	b	m
r	r	r	r	r	r	s	s	s	b	b	b	b	b	b	l
r	r	s	s	s	s	s	s	s	s	b	b	b	b	b	b
r	r	r	r	s	s	s	s	s	s	s	b	b	b	b	b
r	r	r	r	r	s	s	s	s	s	s	s	b	b	b	b

With:

b: bank
m: money
l: loan
r: river
s: stream

LSA EXAMPLE

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

- Perceive Corpus as VSM

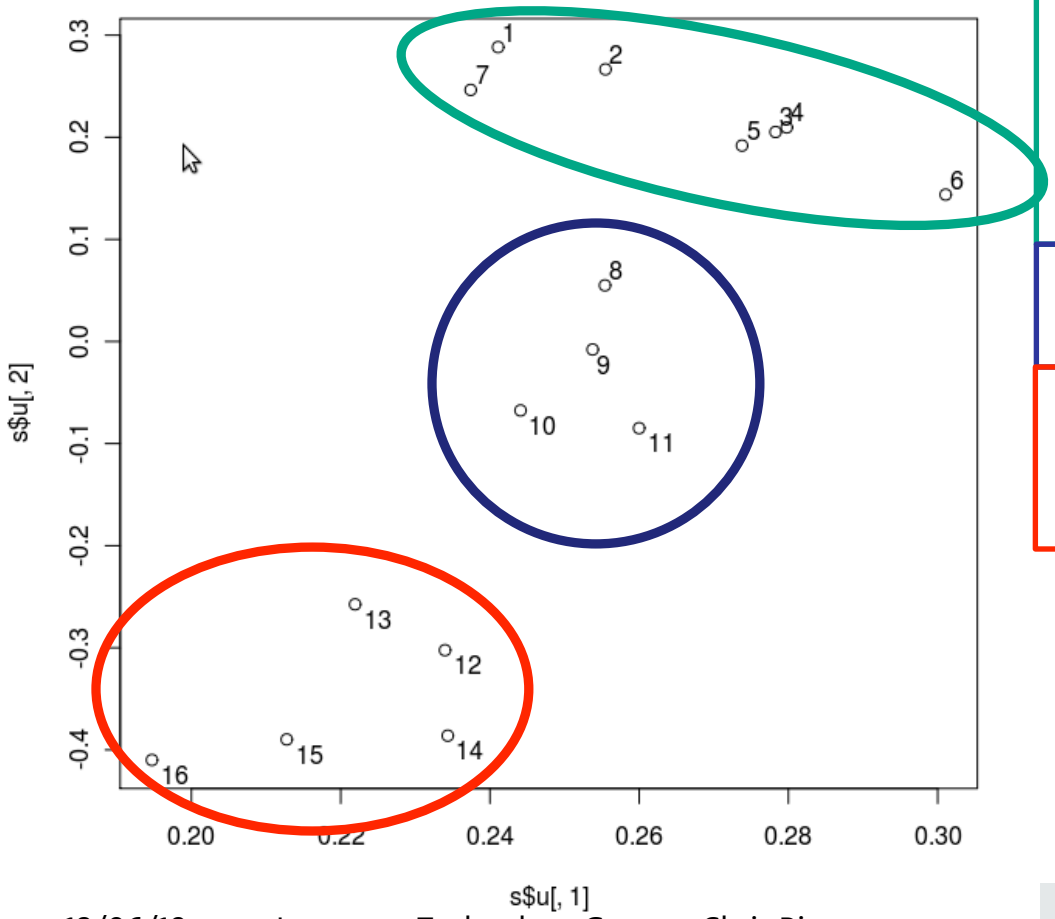
ID	b	r	s	m	l
1	4	0	0	6	6
2	5	0	0	7	4
3	7	0	0	5	4
4	7	0	0	6	3
5	7	0	0	2	7
6	9	0	0	3	4
7	4	1	0	7	4
8	6	1	2	4	3
9	6	1	3	4	2
10	6	2	3	1	4
11	7	2	3	3	1
12	6	3	6	1	0
13	6	6	3	0	1
14	6	2	8	0	0
15	5	4	7	0	0
16	4	5	7	0	0

With:
b: bank
m: money
l: loan
r: river
s: stream

LSA EXAMPLE

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Plot of the first two dimensions of V



1	b	b	b	b	m	m	m	m	m	m	l	l	l	l	l	l
2	b	b	b	b	b	m	m	m	m	m	m	m	l	l	l	l
3	b	b	b	b	b	b	b	m	m	m	m	m	l	l	l	l
4	b	b	b	b	b	b	b	m	m	m	m	m	m	l	l	l
5	b	b	b	b	b	b	b	m	m	l	l	l	l	l	l	l
6	b	b	b	b	b	b	b	b	b	m	m	m	l	l	l	l
7	r	b	b	b	b	m	m	m	m	m	m	m	l	l	l	l
8	r	s	s	b	b	b	b	b	b	m	m	m	m	l	l	l
9	r	s	s	s	b	b	b	b	b	b	m	m	m	m	l	l
10	r	r	s	s	s	b	b	b	b	b	b	m	l	l	l	l
11	r	r	s	s	s	b	b	b	b	b	b	b	m	m	m	l
12	r	r	r	s	s	s	s	s	s	b	b	b	b	b	b	m
13	r	r	r	r	r	r	s	s	s	b	b	b	b	b	b	l
14	r	r	s	s	s	s	s	s	s	s	b	b	b	b	b	b
15	r	r	r	r	s	s	s	s	s	s	s	b	b	b	b	b
16	r	r	r	r	r	s	s	s	s	s	s	s	b	b	b	b

With:

b: bank
m: money
l: loan
r: river
s: stream

SUMMARY ON LSA

- Advantages:
  - Exact mathematical computation – however, expensive for large collections
  - Comparison between documents/words is possible
  - Reduction to n dimensions („semantic categories“)
- Disadvantages:
  - There is no statistical theory why this works
  - Unclear how many dimensions are good a priori
  - Each word can have only one representation: polysemy not modeled
  - „semantic categories“ is an overstatement

DISTRIBUTIONAL HYPOTHESIS

HARRIS, 1954

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

The **Distributional Hypothesis** in linguistics is the theory that words that occur in similar contexts tend to have similar meanings.

The Distributional Hypothesis is the basis for **Statistical Semantics**. It states that the meaning of a word can be defined in terms of its context.

Distributional Hypothesis:

- words are characterized by their (typical) contexts: meaning of a word can be defined in terms of its context
- words are more similar, the more contexts they share

Any process that builds a structure on sentences can be used as a source for contexts

GENERATIVE MODELS

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

- …are models for generating observable data randomly, mainly using some hidden parameters
- **Example:** Throw a fair die and observe the output
  - Output:
    - 2 6 2 3 4 4 3 3 1 6
    - 3 4 6 3 2 1 3 2 5 2
    - 3 3 1 5 5 3 5 3 2 4
- Other generative models:
  - Hidden Markov Chain (HMM)
  - LDA (Latent Dirichlet Allocation)

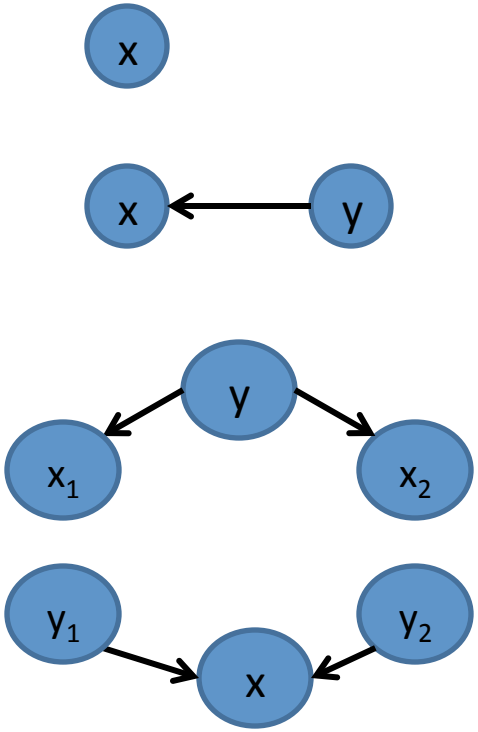
GRAPHICAL NOTATION FOR RANDOM VARIABLES

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Algebraic

- $P(x)$
- $P(x|y)$
- $P(x_1,x_2|y)$
- $P(x|y_1,y_2)$

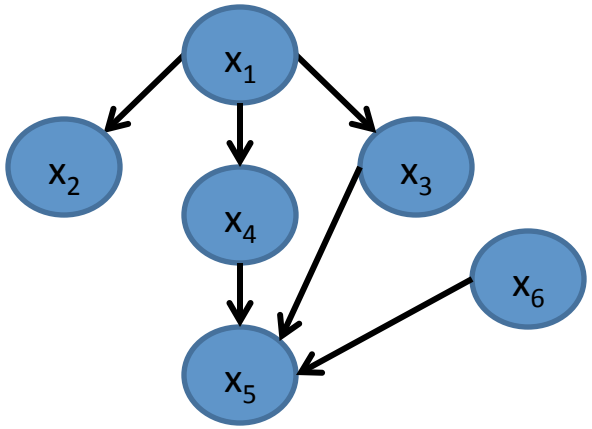
Graphical



Examples from “Graphical Representation – Graphical Models - Gibbs Sampling” by Han Xiao, Ping Luo

JOINT DISTRIBUTION

$$p(x_1, x_2, \ldots, x_D) = \prod_{i=1}^{D} p(x_i \mid \text{parents of } x_i)$$



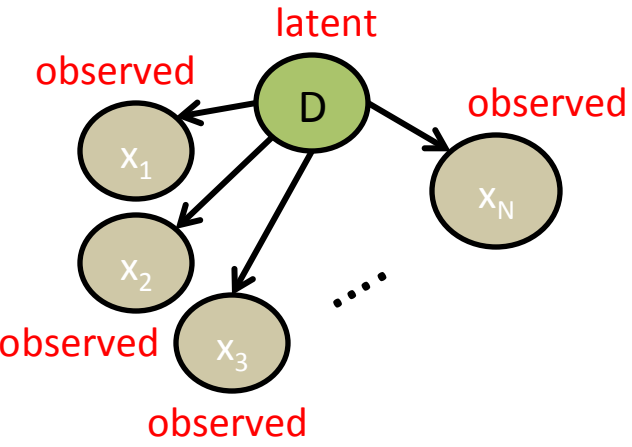
EXAMPLE: DICE MODEL

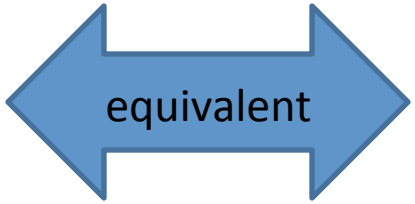
- generation process:
  - Sample a number by throwing the die
- a multinomial distribution D

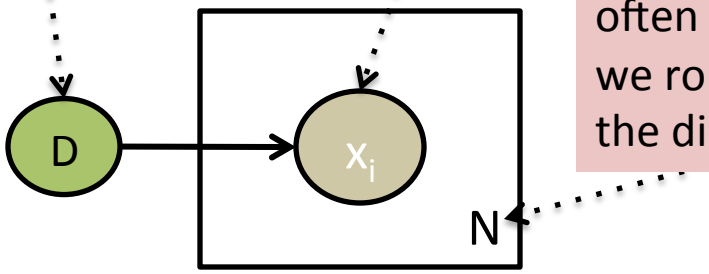
D defines the probability for rolling 1,2,3,4,5,6 and is a hidden (latent) distribution

x_i is the number we rolled

N defines how often we roll the die







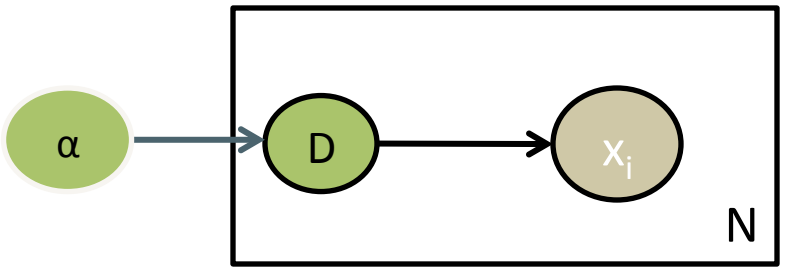
$$P(X \mid D) = \prod_i P(x_i \mid D)$$

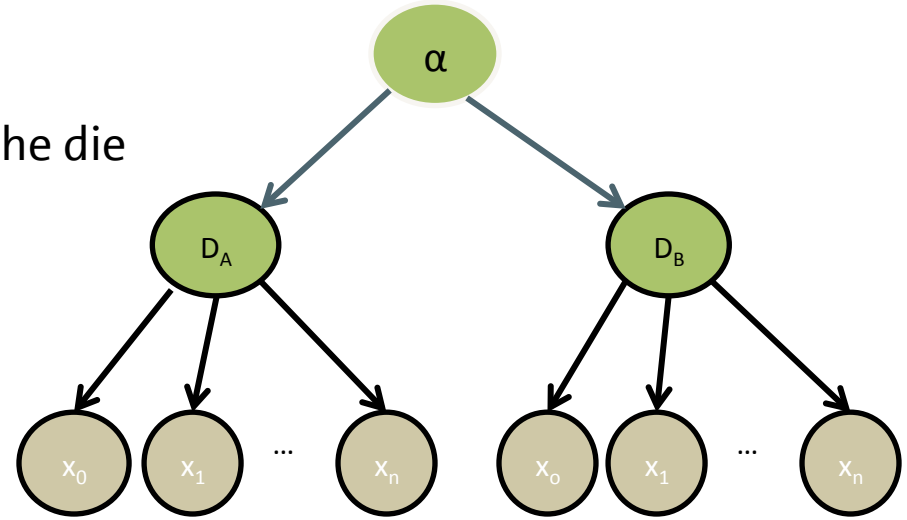
$$P(X, D) = \prod_i P(x_i \mid D) P(D)$$

Examples adapted from „Generative Probabilistic Models“ from Thijs Westerveld, Arjen de Vries and Franciska de Jong

DICE MODEL WITH TWO DICE

- Next assume:
  We have two dice: Die A is a standard die faced with A=(1,2,3,4,5,6) and die B has faces B=(1,1,1,1,1,1). We want to sample with repeated choice of the dice
- Generation Story:
  - Pick a dice randomly
    - Sample a number by throwing the die



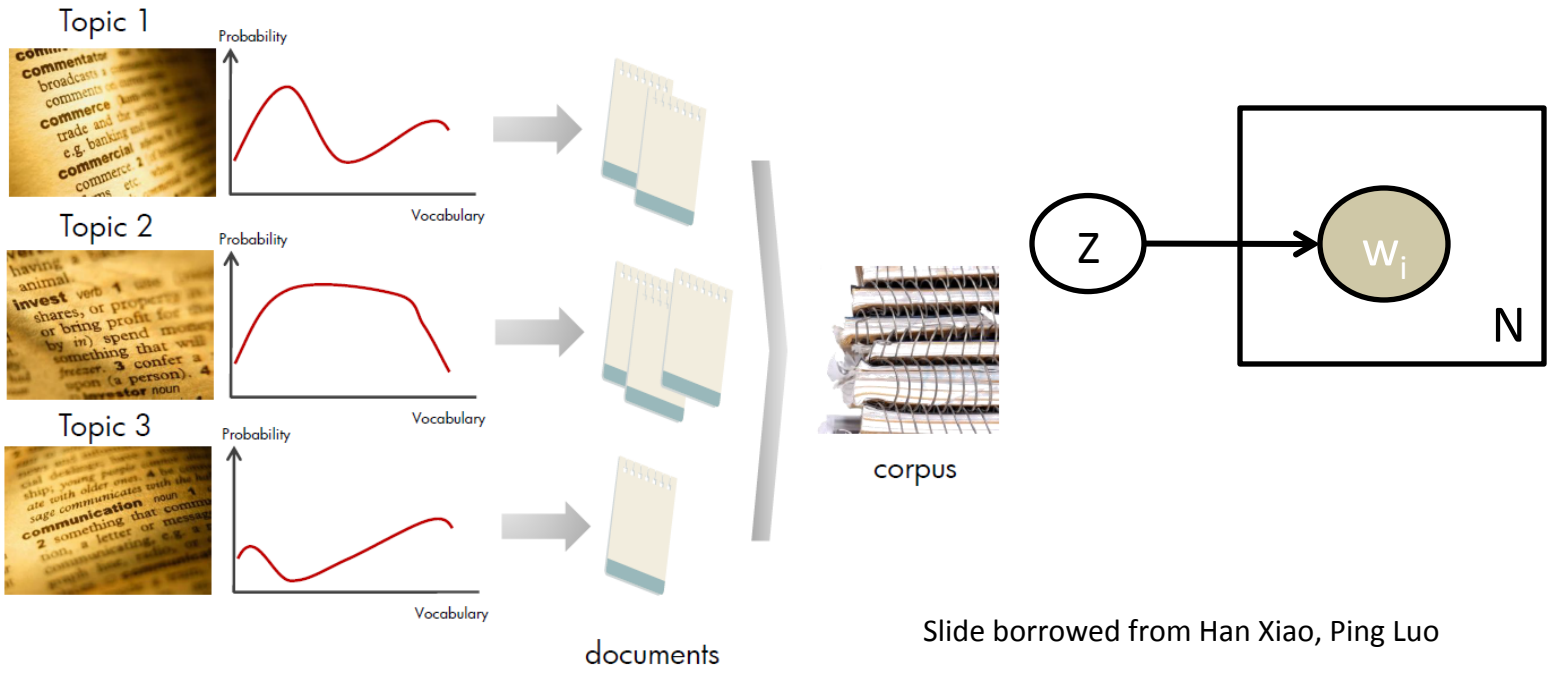


For completeness we also need a prior probability to choose a die

UNIGRAM MODEL

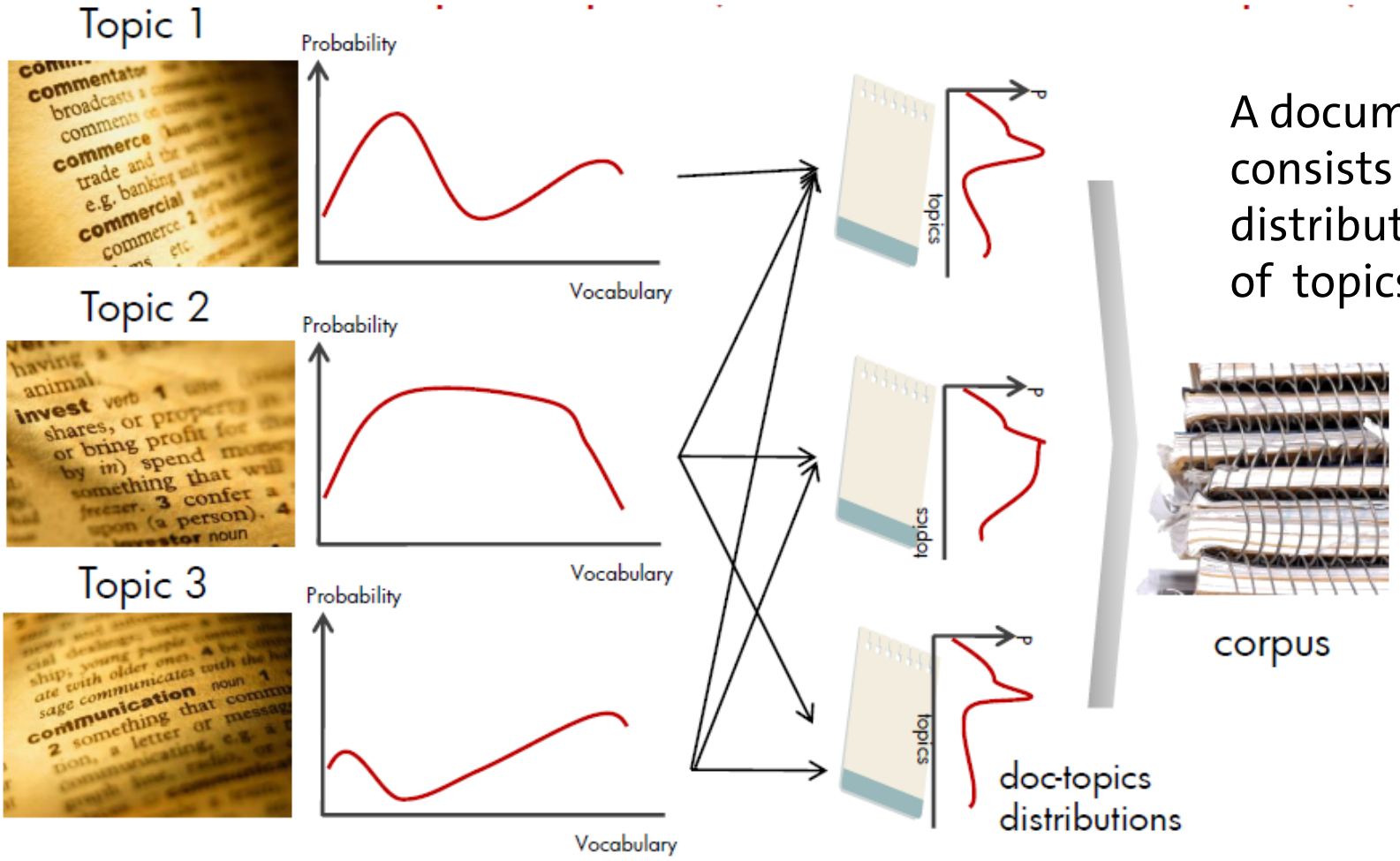
Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

- Generative Story
  - Pick a language model Z for each document
    - Draw N $\mathbf{w_i}$ from multinomial Z



Slide borrowed from Han Xiao, Ping Luo

LATENT DIRICHLET ALLOCATION (BLEI ET AL. 2003)



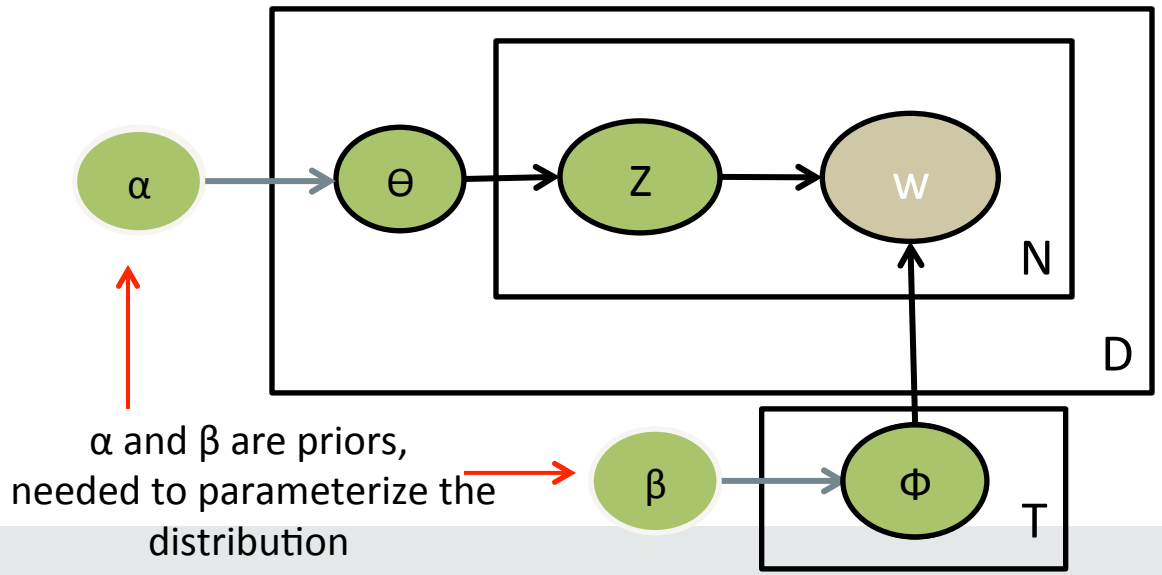
Slide borrowed from Han Xiao, Ping Luo

LATENT DIRICHLET ALLOCATION (BLEI ET AL. 2003)

- In LDA, the topic mixture proportions for each document are drawn from some distribution
  - ➔So we need a multinomial distribution (k-tuples of non-negative numbers that sum up to one)
- Geometrically, the space of these multinomials can be interpreted as a (k-1) simplex, which is a generalization of a triangle to (k-1) dimensions
- For the multinomial, a Dirichlet distribution is used as prior (which makes this approach computable and acts like a smoothing parameter)

LATENT DIRICHLET ALLOCATION (BLEI ET AL. 2003)

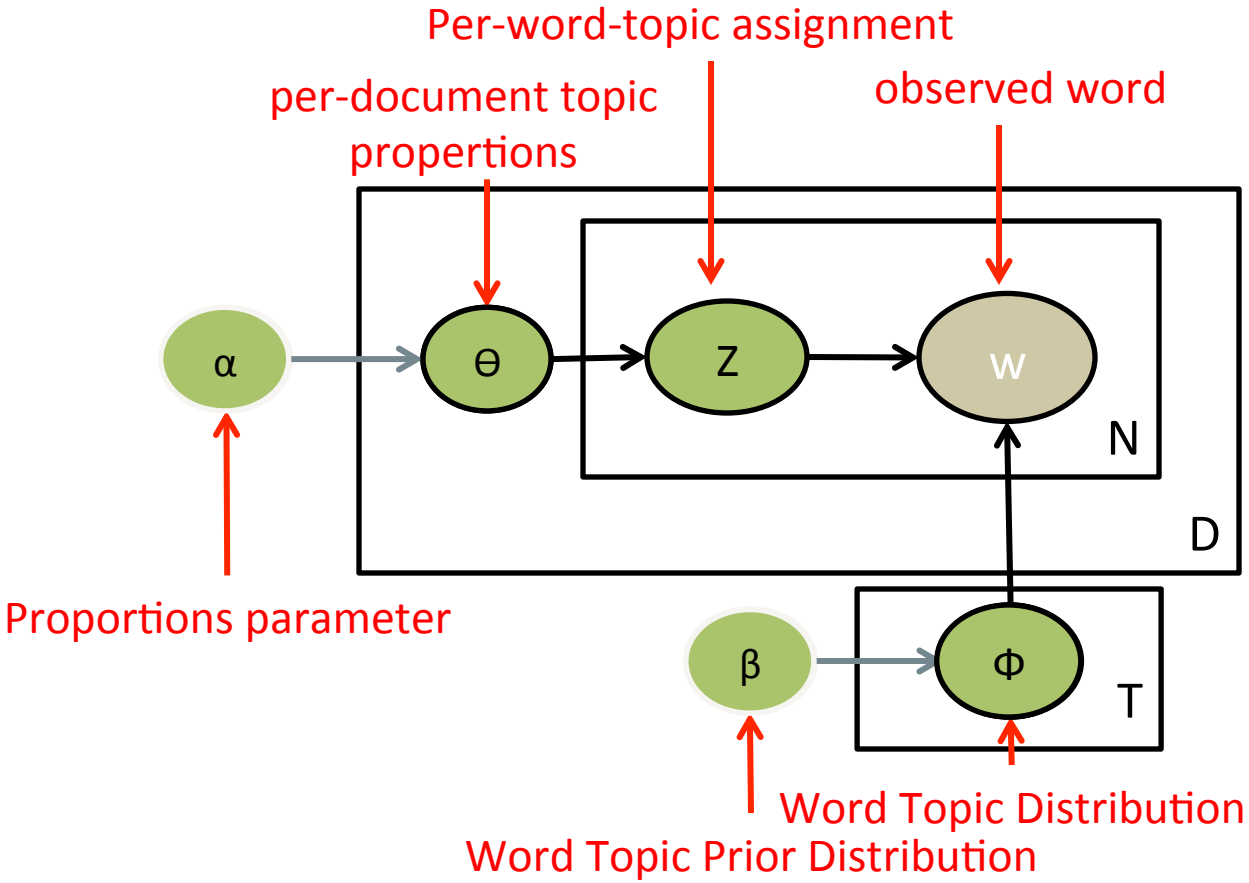
- Generative Story
  - For each document **d** in the corpus D
    - Draw a multinomial distribution **Θ** (topic distribution used for the document)
      - Draw each word in document d
        - » Draw a topic **Z** from multinomial **Θ**
          - Draw **w** from the multinomial **Φ**



Each document is generated based on a topic distribution

α and β are priors, needed to parameterize the distribution

LATENT DIRICHLET ALLOCATION (BLEI ET AL. 2003)



$$\prod_{t=1}^{T} p(\phi \mid \beta) \prod_{d=1}^{D} p(\theta \mid \alpha) \prod_{n=1}^{N} p(z \mid \theta)\, p(w \mid z, \phi)$$

LATENT DIRICHLET ALLOCATION (LDA)

- Example topics generated with LDA using a Wikipedia corpus

Topic 7th:
- Corp. 0.02833210
- technology 0.02470010
- computer 0.02451910
- market 0.01888910
- computers 0.01798110
- Computer 0.01489410
- Inc. 0.01434910
- products 0.01380410
- Compaq 0.01162510
- chips 0.01126110
- company 0.01053510
- machines 0.01035310
- Technology 0.00962710
- Sun 0.00944510
- sales 0.00908210
- personal 0.00908210
- next 0.00853710
- electronic 0.00835610
- Intel 0.00835610

Topic 11th:
- dollar 0.05227010
- U.S. 0.03612410
- currency 0.02312510
- was 0.02203110
- yen 0.02148310
- marks 0.01806310
- rates 0.01765210
- late 0.01628410
- trade 0.01587310
- Lawson 0.01382110
- Thatcher 0.01231610
- interest 0.01204210
- exchange 0.01190510
- rate 0.01149510
- economic 0.01122110
- pound 0.01081110
- Bank 0.00985310
- Britain 0.00957910
- British 0.00916910

Topic 92th:
- years 0.02375010
- may 0.02233410
- cost 0.02154810
- provide 0.02139110
- service 0.02139110
- result 0.01808810
- services 0.01635810
- additional 0.01588610
- based 0.01384110
- period 0.01305510
- number 0.01305510
- most 0.01195410
- terms 0.01164010
- information 0.01116810
- time 0.01069610
- provided 0.01053910
- future 0.01038210
- figure 0.01038210
- financial 0.01022410

LEARNING THE LDA MODEL

- The exact calculation is intractable

- Approximation algorithms:
  - Mean field variational inference
  - Expectation propagation
  - Collapsed Gibbs Sampling
  - Collapsed Variational Inference

COLLAPSED GIBBS SAMPLING

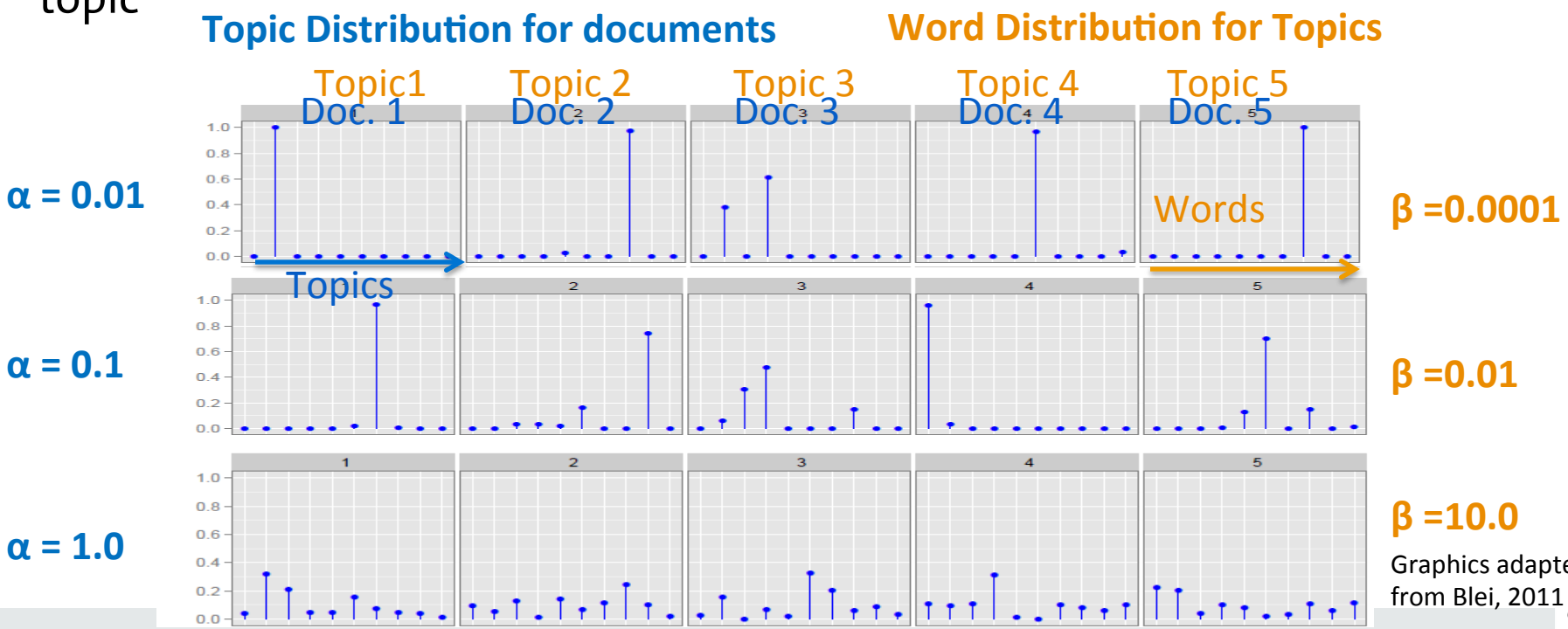
Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

- Gibbs Sampling can be used to maximize $P(x) = P(x_1, ..., x_D)$
- P(x) does not need to be known, but it must be possible to draw samples from the full conditional probability function $P(x_d|x_{-d})$ for all $1 \leq d \leq D$
- Algorithm:
  - Randomly initialize $X^{(0)} = \{x_1^{(0)}, ..., x_D^{(0)}\}$
  - For t=1...T *(T is the number of iterations)*
    - For d=1...D *(e.g. all words of a document)*
      - Sample $X \sim P(x_d|x_{1,..,d-1}^{(t)}, .., x_{d+1,..,D}^{(t-1)})$
        (Assign X a value according to the probability of all other assigned values)
  - *Use n last assignments to calculate the probabilities*

PRIORS DISTRIBUTION PARAMETERS α AND β

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

- **α:** regulates the sparseness of the topic-per-document distribution. Lower values result in documents being represented by fewer topics
- **β:** regulates the sparsity of topics, by assigning fewer terms to each topic, which is correlated to how related words need to be for being assigned to a topic



Graphics adapted from Blei, 2011

EXAMPLE OF TRAINING A LDA MODEL

- Example: We want to train 2 topics with $\alpha=1$ and $\beta=0.1$

Doc 1	a	a	b	b	a	a
Doc 2	a	b	b	c	c	a
Doc 3	d	d	d	c	c	
Doc 4	d	c				

EXAMPLE OF TRAINING A LDA MODEL

- Example: We want to train 2 topics with $\alpha=1$ and $\beta=0.1$

Initialize topics randomly

Doc 1	a	a	b	b	a	a
	2	2	1	1	2	1
Doc 2	a	b	b	c	c	a
	2	1	1	1	2	2
Doc 3	d	d	a	c	c	
	1	2	1	2	1	
Doc 4	d	c				
	1	1				

EXAMPLE OF TRAINING A LDA MODEL

- Example: We want to train 2 topics with α=1 and β=0.1

Sample new topic for first word in first document
→Delete topic of actual word

Doc 1	a	a	b	b	a	a
	???	2	1	1	2	1
Doc 2	a	b	b	c	c	a
	2	1	1	1	2	2
Doc 3	d	d	a	c	c	
	1	2	1	2	1	
Doc 4	d	c				
	1	1				

topic 1 (z1) with "a" divided by #topic 1

topic 1 (z1) in doc 1 divided by topics in Doc 1

$$P(z=1) \propto \frac{n_{z1}^{("a")} + \beta}{\sum_{t=1}^{V=4}\left(n_{z1}^{(t)} + \beta\right)} \bullet \frac{n_{z1}^{(d1)} + \alpha}{[\sum_{z=1}^{2}(n_{z}^{(d1)} + \alpha\ )]-1}$$

$$P(z=2) \propto \frac{n_{z2}^{("a")} + \beta}{\sum_{t=1}^{V=4}(n_{z2}^{(t)} + \beta)} \bullet \frac{n_{z2}^{(d1)} + \alpha}{[\sum_{z=1}^{2}(n_{z}^{(d1)} + \alpha\ )]-1}$$

EXAMPLE OF TRAINING A LDA MODEL

- Example: We want to train 2 topics with α=1 and β=0.1

Sample new topic for first word in first document

Doc 1	a	a	b	b	a	a
	???	2	1	1	2	1
Doc 2	a	b	b	c	c	a
	2	1	1	1	2	2
Doc 3	d	d	a	c	c	
	1	2	1	2	1	
Doc 4	d	c				
	1	1				

topic 1 (z1) with "a" divided by #topic 1

topic 1 (z1) in doc 1 divided by topics in Doc 1

$$P(z=1) \propto \frac{2+0.1}{11+4*0.1} \bullet \frac{3+1}{5+2*1}$$

$$P(z=2) \propto \frac{4+0.1}{7+4*0.1} \bullet \frac{2+1}{5+2*1}$$

EXAMPLE OF TRAINING A LDA MODEL

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

- Example: We want to train 2 topics with α=1 and β=0.1

Sample new topic for first word in first document

Doc 1	a	a	b	b	a	a
	???	2	1	1	2	1
Doc 2	a	b	b	c	c	a
	2	1	1	1	2	2
Doc 3	d	d	a	c	c	
	1	2	1	2	1	
Doc 4	d	c				
	1	1				

$P(z = 1) \propto 0.11$
$P(z = 2) \propto 0.23$

Randomly Sample a number [0;0.34]:

0-0.11: Topic 1
Else: Topic 2

0.3 → Topic 2

EXAMPLE OF TRAINING A LDA MODEL

- Example: We want to train 2 topics with α=1 and β=0.1

Sample next topic for each word in each document and repeat the process N times (until no substantial changes)

Doc 1	a	a	b	b	a	a
	2	???	1	1	2	1
Doc 2	a	b	b	c	c	a
	2	1	1	1	2	2
Doc 3	d	d	a	c	c	
	1	2	1	2	1	
Doc 4	d	c				
	1	1				

EXAMPLE OF TRAINING A LDA MODEL

- Example: We want to train 2 topics with α=1 and β=0.1

  After 1000 iterations

Doc 1	a	a	b	b	a	a
	1	1	1	1	1	1
Doc 2	a	b	b	c	c	a
	1	1	1	2	2	1
Doc 3	d	d	a	c	c	
	2	2	2	2	2	
Doc 4	d	c				
	2	2				

EXAMPLE OF TRAINING A LDA MODEL

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

- Example: We want to train 2 topics with α=1 and β=0.1

Now we can calculate the word topic distribution

Doc 1	a	a	b	b	a	a
	1	1	1	1	1	1
Doc 2	a	b	b	c	c	a
	1	1	1	2	2	1
Doc 3	d	d	a	c	c	
	2	2	2	2	2	
Doc 4	d	c				
	2	2				

1) Count #topic/term

2) Use Smoothing factor β for zero counts

	Topic 1	Topic 2
a	0.59	0.11
b	0.39	0.01
c	0.01	0.55
d	0.01	0.33

3) Normalize per topic

EXAMPLE OF TRAINING A LDA MODEL

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

- Example: We want to train 2 topics with α=1 and β=0.1

We can also calculate the document topic distribution

Doc 1	a	a	b	b	a	a
	1	1	1	1	1	1
Doc 2	a	b	b	c	c	a
	1	1	1	2	2	1
Doc 3	d	d	a	c	c	
	2	2	2	2	2	
Doc 4	d	c				
	2	2				

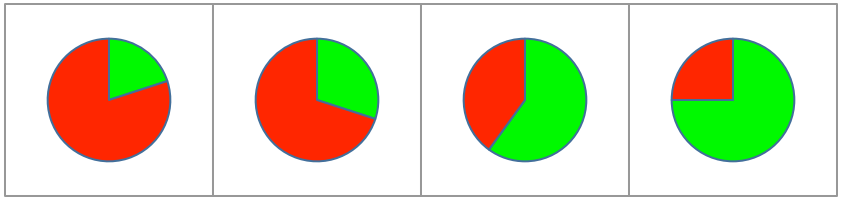
1) Count #topic/document
2) Add smoothing for zero values (α)
3) Normalize per document

	Topic 1	Topic 2
Doc 1	0.85	0.15
Doc 2	0.67	0.33
Doc 3	0.17	0.83
Doc 4	0.33	0.76

EXAMPLE OF TRAINING A LDA MODEL

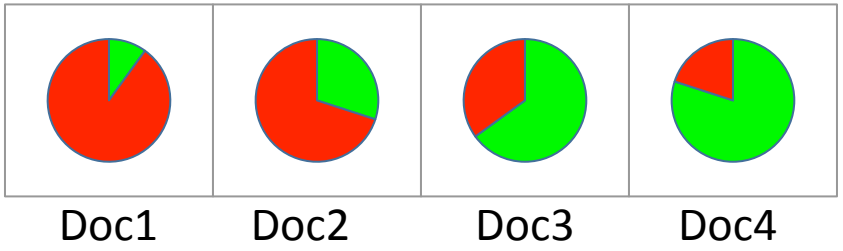
Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

- Output of LDA
  - A word topic distribution



Topic1	Topic2
a:80%	d:75%
b:65%	c:60%
c:40%	b:35%
d:25%	a:20%

  - A topic word distribution
  - A document topic distribution



  - A document where each word is assigned to a topic (see previous slide)

APPLICATIONS USING LDA

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

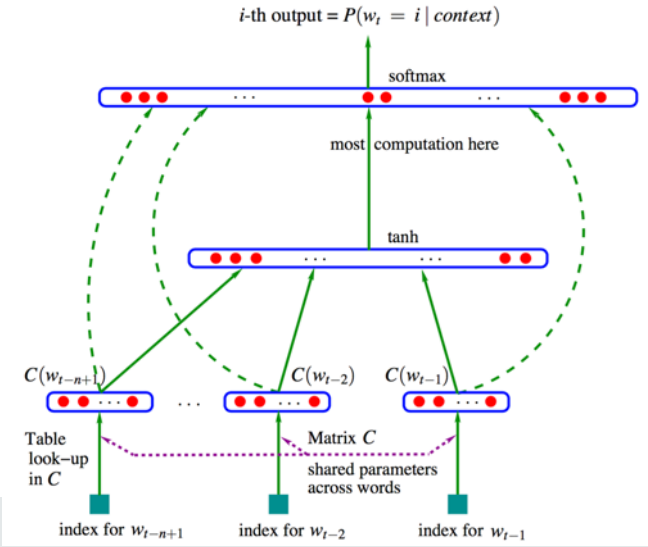
- Text clustering
- Characterize content of a collection, popular in social sciences and the Digital Humanities
- Find similar texts a user is interested in, e.g. news recommendation system
- Illustrate changing topic distribution over time
- Text Segmentation in topical segments

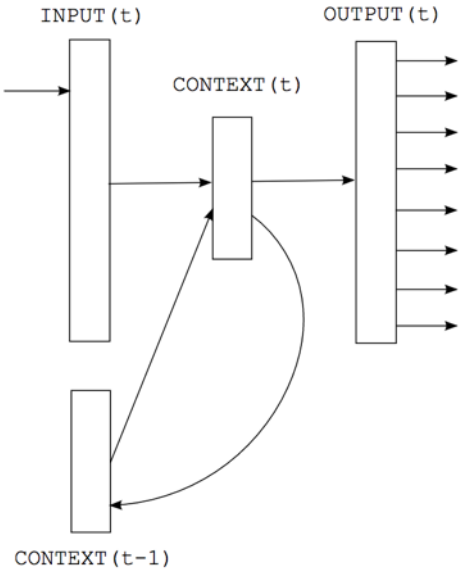
NEURAL WORD EMBEDDINGS

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Recap: Neural Language Models

- Need continuous representations of words
- These are trained in order to predict the next word, given previous words





NEURAL VERSION OF DISTRIBUTIONAL HYPOTHESIS

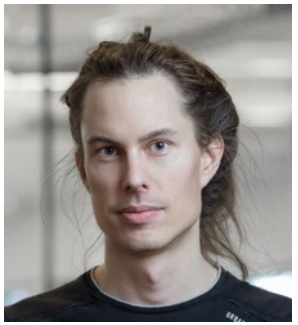
- A word can be **predicted** by its context
- OR: its context can be **predicted** by a word
- words and contexts are represented **by dense vectors**
- that are trained as to **maximize the probability** of a corpus

$$\arg\max_{\theta} \prod_{w \in Text} \left[\prod_{c \in C(w)} p(c|w;\theta) \right]$$

- assuming that this happens when **similar words get similar dense vector embeddings**

SOME POPULAR PRE-TRAINED DENSE WORD EMBEDDINGS

- **word2vec** (Mikolov et al., 2013)
  - https://code.google.com/archive/p/word2vec/

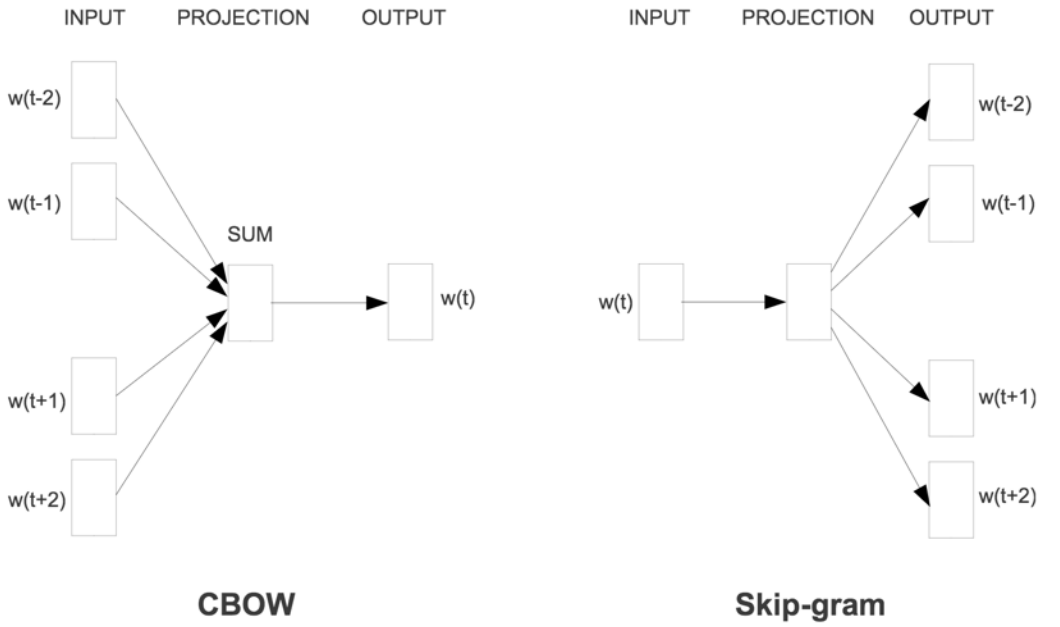


- **fastText** (Bojanowski et. al., 2017)
  - http://www.fasttext.cc

- **GloVe** (Pennington et al., 2014)
  - http://nlp.stanford.edu/projects/glove

WORD2VEC

https://code.google.com/archive/p/word2vec/



Two architectures; both work:

- CBOW: predict word, given its close context. Bag-of-words within context
- SKIP-gram: predict context, given a word. Takes order into account.

Mikolov, T., Chen, K., Conrado, G., Dean, J. (2013) Efficient Estimation of Word Representations in Vector Space. Proceedings of the Workshop at ICLR, Scottsdale, pp. 1-12.

WORD2VEC: SKIP-GRAM TASK

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

- word2vec provides a **variety of options**. Here in more detail: "skip-gram with negative sampling" (SGNS)

Skip-gram algorithm:

1. Treat the target word and a neighboring context word as positive examples.
2. Randomly sample other words in the lexicon to get negative samples
3. Use logistic regression to train a classifier to distinguish those two cases
4. Use the weights as the embeddings

SKIP-GRAM TRAINING DATA

Training sentence: Assume context words are those in +/- 2 word window.

... lemon, a tablespoon of **apricot** jam a pinch ...
c1 c2 target c3 c4

Given a tuple (t,c) = target, context

- (apricot , jam)
- (apricot, aadvark)

Return probability that c is a real context word:

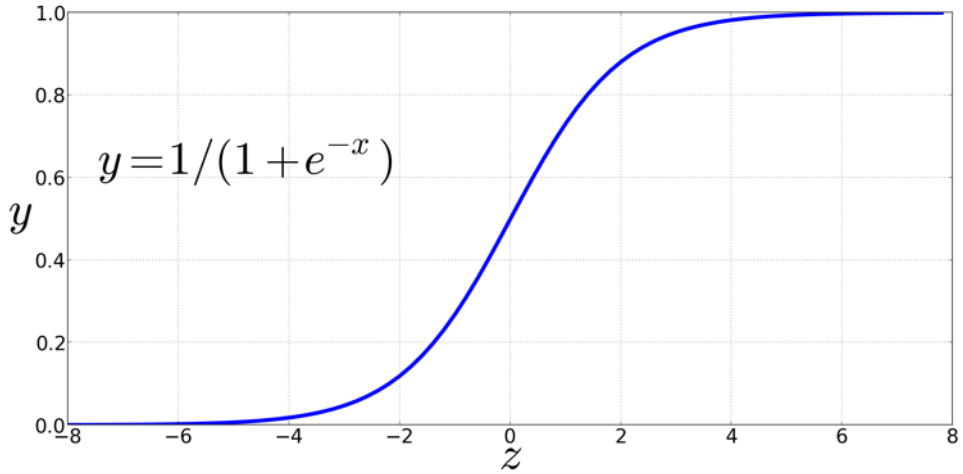
$P(+|t,c)$

$P(-|t,c) = 1-P(+|t,c)$

How to compute P(+|t,c)?

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

- Intuition:
  - Words are likely to appear near similar words
  - Model similarity with dot-product
  - Similarity(t,c) $\propto t \cdot c$
  - Turning dot product into a probability



$$\sigma(x) = \frac{1}{1+e^{-x}}$$

COMPUTING PROBABILITIES

Turning dot product into a probability:

$$P(+|t,c) = \frac{1}{1+e^{-t\cdot c}}$$

$$\begin{aligned} P(-|t,c) &= 1 - P(+|t,c) \\ &= \frac{e^{-t\cdot c}}{1+e^{-t\cdot c}} \end{aligned}$$

Assume all context words are independent:

$$P(+|t,c_{1:k}) = \prod_{i=1}^{\kappa} \frac{1}{1+e^{-t\cdot c_i}}$$

$$\log P(+|t,c_{1:k}) = \sum_{i=1}^{k} \log \frac{1}{1+e^{-t\cdot c_i}}$$

SKIP-GRAM TRAINING

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Training sentence: Assume context words are those in +/- 2 word window.

... lemon, a tablespoon of **apricot** jam a pinch ...
c1 c2 target c3 c4

positive examples +

t	c
apricot	tablespoon
apricot	of
apricot	preserves
apricot	or

negative examples -

t	c	t	c
apricot	aardvark	apricot	twelve
apricot	puddle	apricot	hello
apricot	where	apricot	dear
apricot	coaxial	apricot	forever

NEGATIVE SAMPLING: CHOOSING NOISE WORDS

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Need to prevent that all words get the same vector.

- For each observed word-context pair, also 'observe' random contexts
- change objective function to assign a low probability to random contexts and high probability to real observations

- Could pick w according to their unigram frequency P(w)
- More common to chosen then according to $p_\alpha(w)$

$$P_\alpha(w) = \frac{count(w)^\alpha}{\sum_w count(w)^\alpha}$$

- α= 0.75 works well because it gives rare noise words slightly higher probability

OBJECTIVE FUNCTION

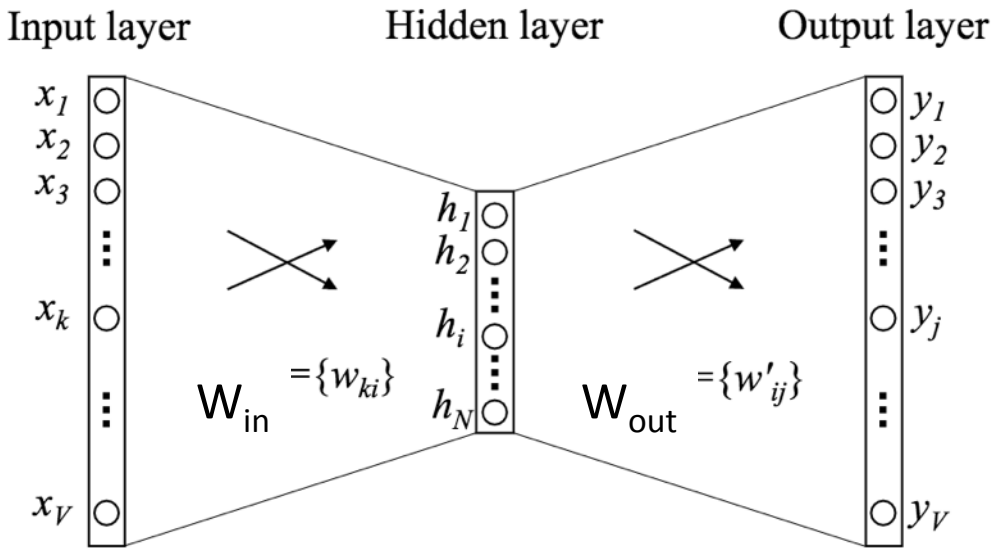
- We want to maximize...

$$\sum_{(t,c)\in+} logP(+|t,c) + \sum_{(t,c)\in-} logP(-|t,c)$$

- Maximize the + label for the pairs from the positive training data, and the – label for the negative samples.

$$\begin{aligned} L(\theta) &= \log P(+|t,c) + \sum_{i=1} \log P(-|t,n_i) \\ &= \log \sigma(c \cdot t) + \sum_{i=1}^{k} \log \sigma(-n_i \cdot t) \\ &= \log \frac{1}{1+e^{-c \cdot t}} + \sum_{i=1}^{k} \log \frac{1}{1+e^{n_i \cdot t}} \end{aligned}$$

EMBEDDINGS:
WEIGHTS TO/FROM PROJECTION LAYER

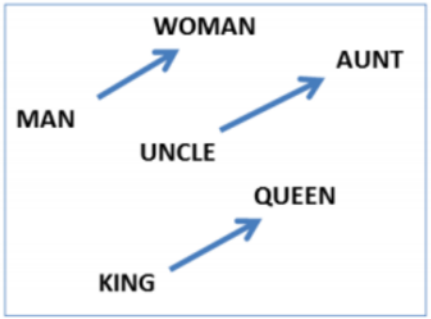


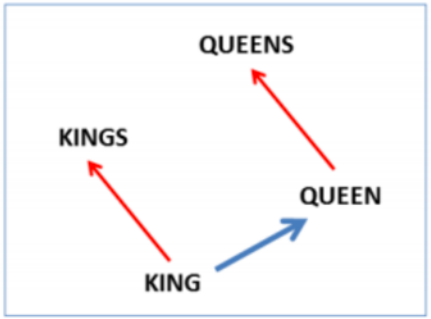
- Example: simple continuous BOW bigram model
- W_{in} and W_{out}^T: V x N matrices
- every word is embedded in N dimensions, which is the size of the hidden layer
- Note: embeddings for words and contexts differ

VARIANTS OF WORD2VEC

- subsampling of frequent words (not informative contexts)
- rare-word pruning
- dynamic window size
- hierarchical softmax (speedup)

VECTOR ALGEBRA FOR ANALOGY QUESTIONS





- Observation: words in the same relation have similar vector differences
- Syntactic analogy questions: "a is to b as c is to ..." (rough is to rougher as tough is to ...)
- This is a wildly overestimated feature: the same semantic relation has DIFFERENT directions in subspaces

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

Method	Adjectives	Nouns	Verbs	All
LSA-80	9.2	11.1	17.4	12.8
LSA-320	11.3	18.1	20.7	16.5
LSA-640	9.6	10.1	13.8	11.3
RNN-80	9.3	5.2	30.4	16.2
RNN-320	18.2	19.0	45.0	28.5
RNN-640	21.0	25.2	54.8	34.7
RNN-1600	**23.9**	**29.2**	**62.2**	**39.6**

Table 2: Results for identifying syntactic regularities for different word representations. Percent correct.

Mikolov, T., Yih, W., Zweig, G. (2013): Linguistic Regularities in Continuous Space Word Representations. Proc. HLT-NAACL '13, pp. 746-751

COUNT OR PREDICT?

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

- Much previous debate why prediction-based models seem to perform better than count-based models
- In reality, they are the same, as shown by Levy & Goldberg (2014):
  - Skip-gram negative sampling performs implicit matrix factorization with PMI weighting
  - This is equivalent to an approximation of SVD, shifted by a global constant
- Advances on tasks largely attributed to more parameter tuning

"If you want to get good results, you should tune your hyperparameters. And if you want to make good science, don't forget to tune your baselines' hyperparameters too!"
- Omer Levy, pers. communication

Levy, O. and Goldberg, Y. (2014): Neural word embedding as implicit matrix factorization. Advances in neural information processing systems, pp. 2177-2185

CURRENT DIRECTIONS

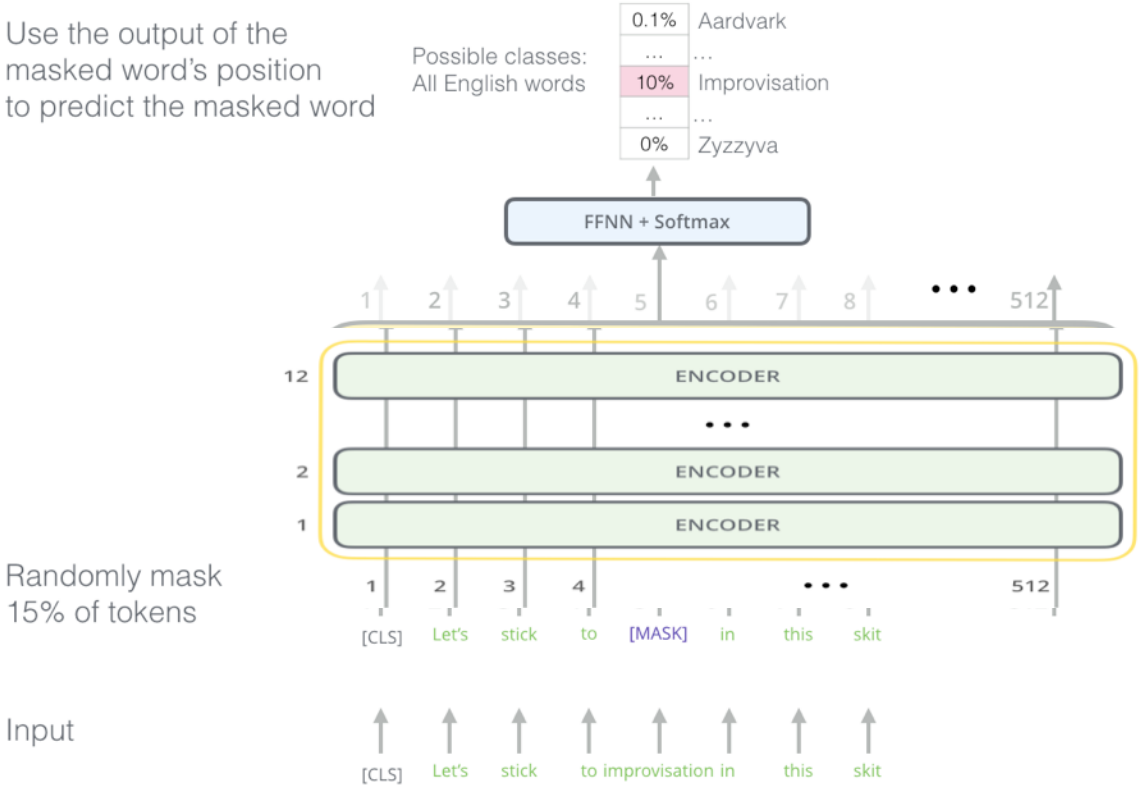
Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

- Better address OOV problems: use character-n-gram-based models, e.g. fasttext https://github.com/facebookresearch/fastText
- Use grammatical context, e.g. word2vecf https://bitbucket.org/yoavgo/word2vecf
- Train your specialized embeddings vs. learn a projection on 'default' embeddings, e.g. http://ahogrammer.com/2017/01/20/the-list-of-pretrained-word-embeddings/
- From word embeddings to sense embeddings
- Contextualized embeddings, e.g. BERT, GTP, ELMo, e.g. https://github.com/google-research/bert

There is a large variety of models that learn embeddings.

ILLUSTRATING CONTEXTUALIZED BERT EMBEDDINGS

- Training: Masked word prediction (a form of language modelling)
- Use: all or just some of the 12/24 transformer layers per token



http://jalammar.github.io/illustrated-bert/

USING CLMS IN TASKS

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

1 - Semi-supervised training on large amounts of text (books, wikipedia..etc).

The model is trained on a certain task that enables it to grasp patterns in language. By the end of the training process, BERT has language-processing abilities capable of empowering many models we later need to build and train in a supervised way.

Semi-supervised Learning Step

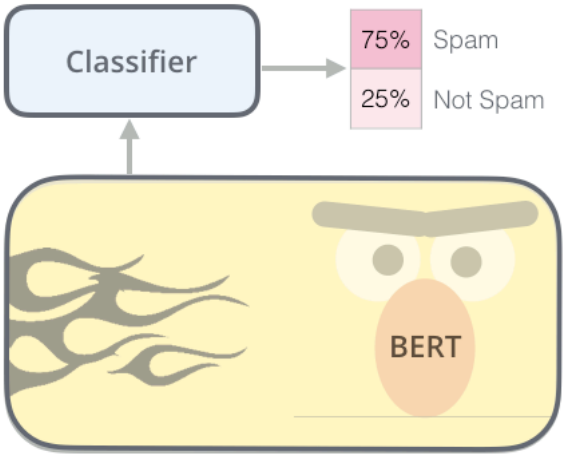
Model: 

Dataset: 

Objective: Predict the masked word (langauge modeling)

2 - Supervised training on a specific task with a labeled dataset.

Supervised Learning Step

Model: (pre-trained in step #1) 

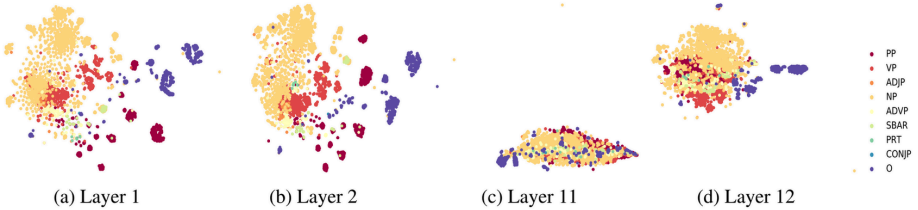
Dataset:

Email message	Class
Buy these pills	Spam
Win cash prizes	Spam
Dear Mr. Atreides, please find attached…	Not Spam

http://jalammar.github.io/illustrated-bert/

SOME BERT INSIGHTS

- different layers seem to model different linguistic structures. here: grammar:



(a) Layer 1 (b) Layer 2 (c) Layer 11 (d) Layer 12

Figure 1: 2*D* t-SNE plot of span embeddings computed from first and last two layers of BERT.

layer	1	2	3	4	5	6	7	8	9	10	11	12
NMI	**0.38**	0.37	0.35	0.3	0.24	0.2	0.19	0.16	0.17	0.18	0.16	0.19

Table 1: Clustering performance of span representations obtained from different layers of BERT.

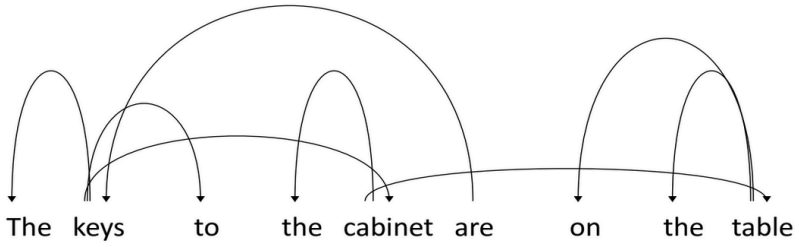


Figure 2: Dependency parse tree induced from attention head #11 in layer #2 using gold root ('are') as starting node for maximum spanning tree algorithm.

Ganesh Jawahar, Benoît Sagot, Djamé Seddah. What does BERT learn about the structure of language?. ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Jul 2019, Florence, Italy

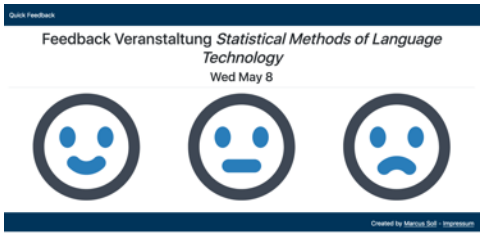
SUMMARY ON DENSE VECTOR REPRESENTATIONS

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

- Core idea: make documents or words comparable by real-valued representations
- Similar items should receive similar representations, which serves as a layer of abstraction
- Representations can be mathematically computed (LSA), sampled (LDA) or learned (word2vec)
- Dense vectors required as input layer for neural methods, but also suitable as features in any ML-based NLP architecture
- Principles have been known in the NLP community for over 25 years
- neural embeddings gained a lot of attention is due to more efficient approximate implementations
- recent advances on contextualized embeddings make a huge difference

IMMEDIATE FEEDBACK

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG



- Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems, 30(1-7), 107–117.
- Erkan, G. and Radev, D. R. (2004). Lexrank: Graph-based lexical centrality a salience in text summarization. Journal of Artificial In ligence Research IR), 22, 45 .
- Mihalcea, R. and Tarau, P. (2004). Textra ging ord tex cedi the conference on Empirical Methods in Natural L s 404–411, Barcelona, Spain.
- Mihalcea, R. and Radev, D. (2011). Graph mation Retrieval. Cambridge University Press

coming up next

graphs, networks, PageRank, graph cluste

GRAPH-BASED METHODS FOR NLP