Dokument als Vector

$$LDA(\underline{t}) := P(d|\underline{t})$$

Beispiel:

P(topic1|diese_Vorlesung) = 0.001
P(topic2|diese_Vorlesung) = 0.0
P(topic3|diese_Vorlesung) = 0.4
P(topic4|diese_Vorlesung) = 0.4
P(topic5|diese_Vorlesung) = 0.1
…

P(topic1|Abenteuer_Roman) = 0.2
P(topic2|Abenteuer_Roman) = 0.2
P(topic3|Abenteuer_Roman) = 0.0
P(topic4|Abenteuer_Roman) = 0.3
P(topic5|Abenteuer_Roman) = 0.2
…

Dokument als Vector

$$\mathrm{LDA}(\underline{t}) := P(d|\underline{t})$$

Was „topic1“ tatsächlich ist,
weis man nicht.
Das könnte alles sein!

Beispiel:

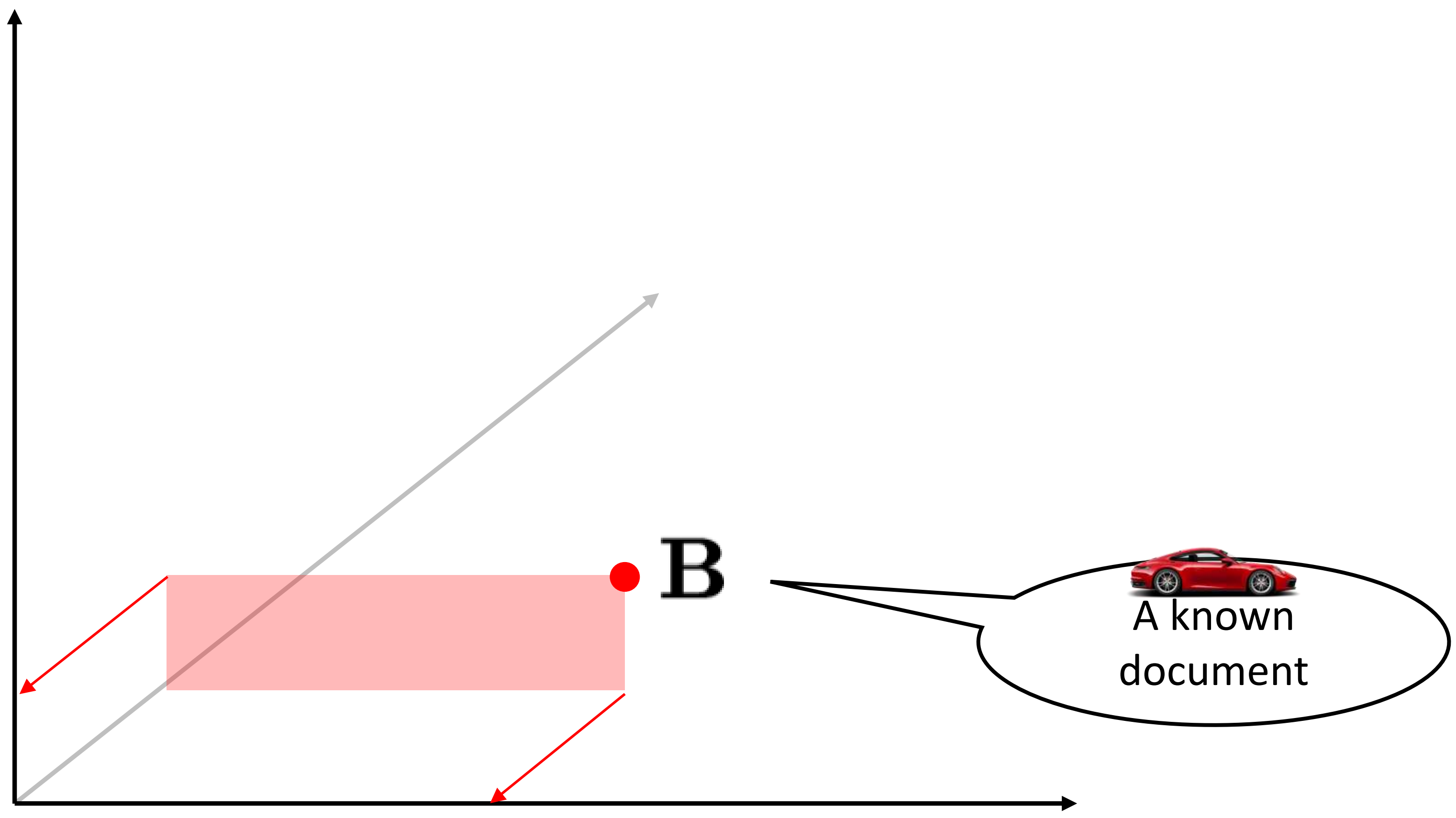
P(topic1|diese_Vorlesung) = 0.001
P(topic2|diese_Vorlesung) = 0.0
P(topic3|diese_Vorlesung) = 0.4
P(topic4|diese_Vorlesung) = 0.4
P(topic5|diese_Vorlesung) = 0.1
…

$$\mathbf{A} = \begin{pmatrix} 0.001 \\ 0.0 \\ 0.4 \\ 0.4 \\ 0.1 \\ \dots \end{pmatrix}$$

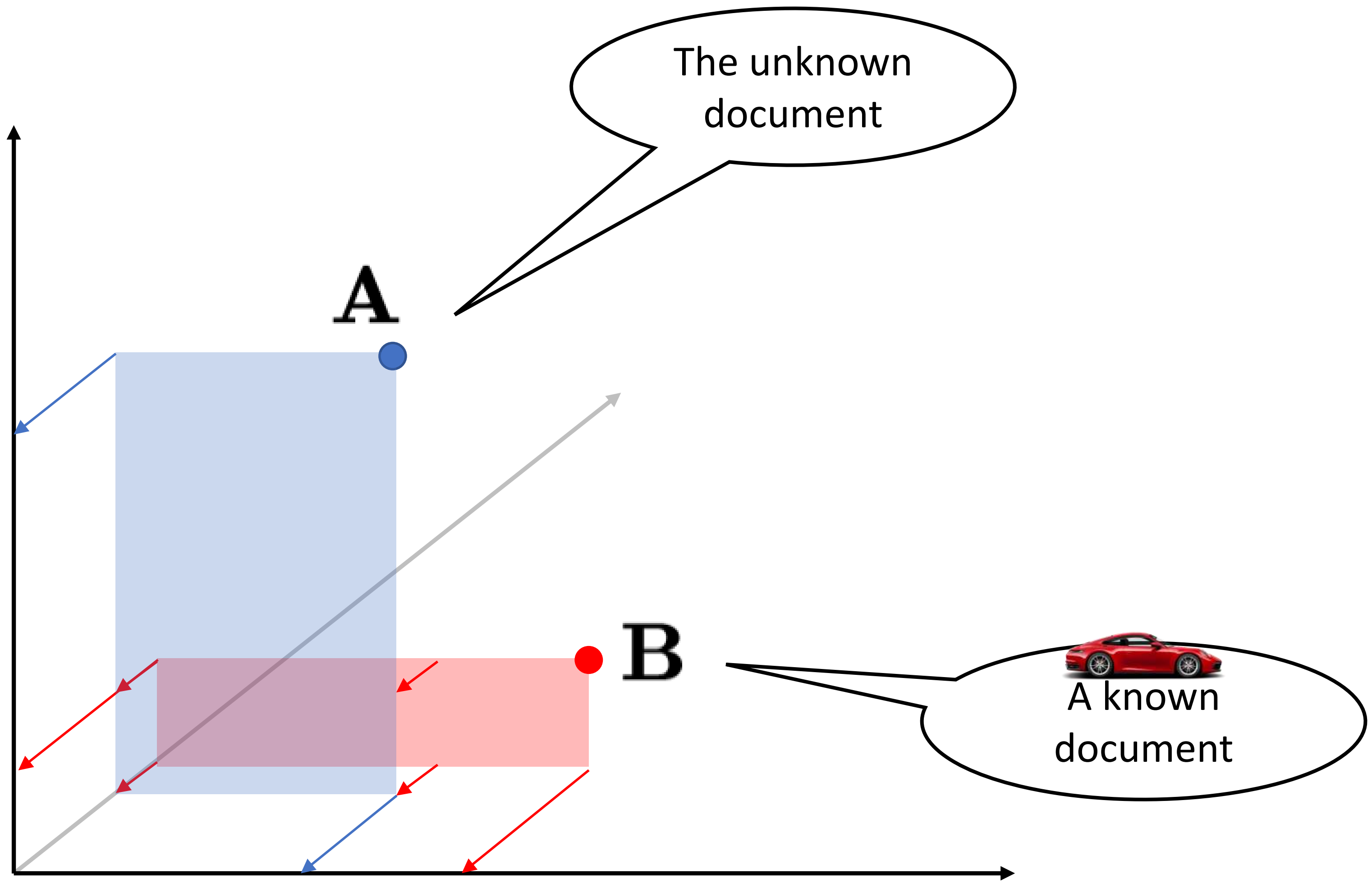
P(topic1|Abenteuer_Roman) = 0.2
P(topic2|Abenteuer_Roman) = 0.2
P(topic3|Abenteuer_Roman) = 0.0
P(topic4|Abenteuer_Roman) = 0.3
P(topic5|Abenteuer_Roman) = 0.2
…

$$\mathbf{B} = \begin{pmatrix} 0.2 \\ 0.2 \\ 0.0 \\ 0.3 \\ 0.2 \\ \dots \end{pmatrix}$$

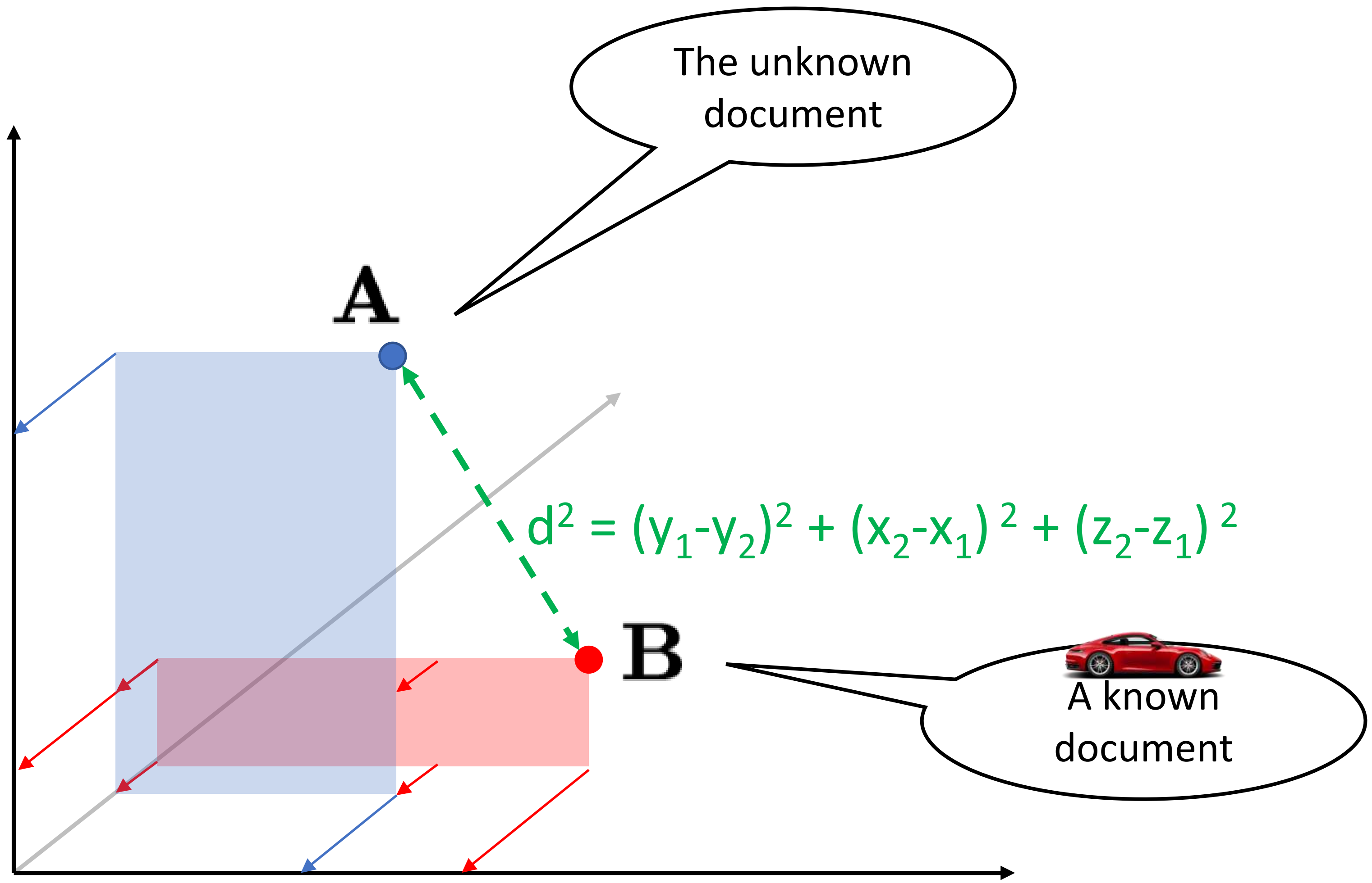
Dokument als Vector



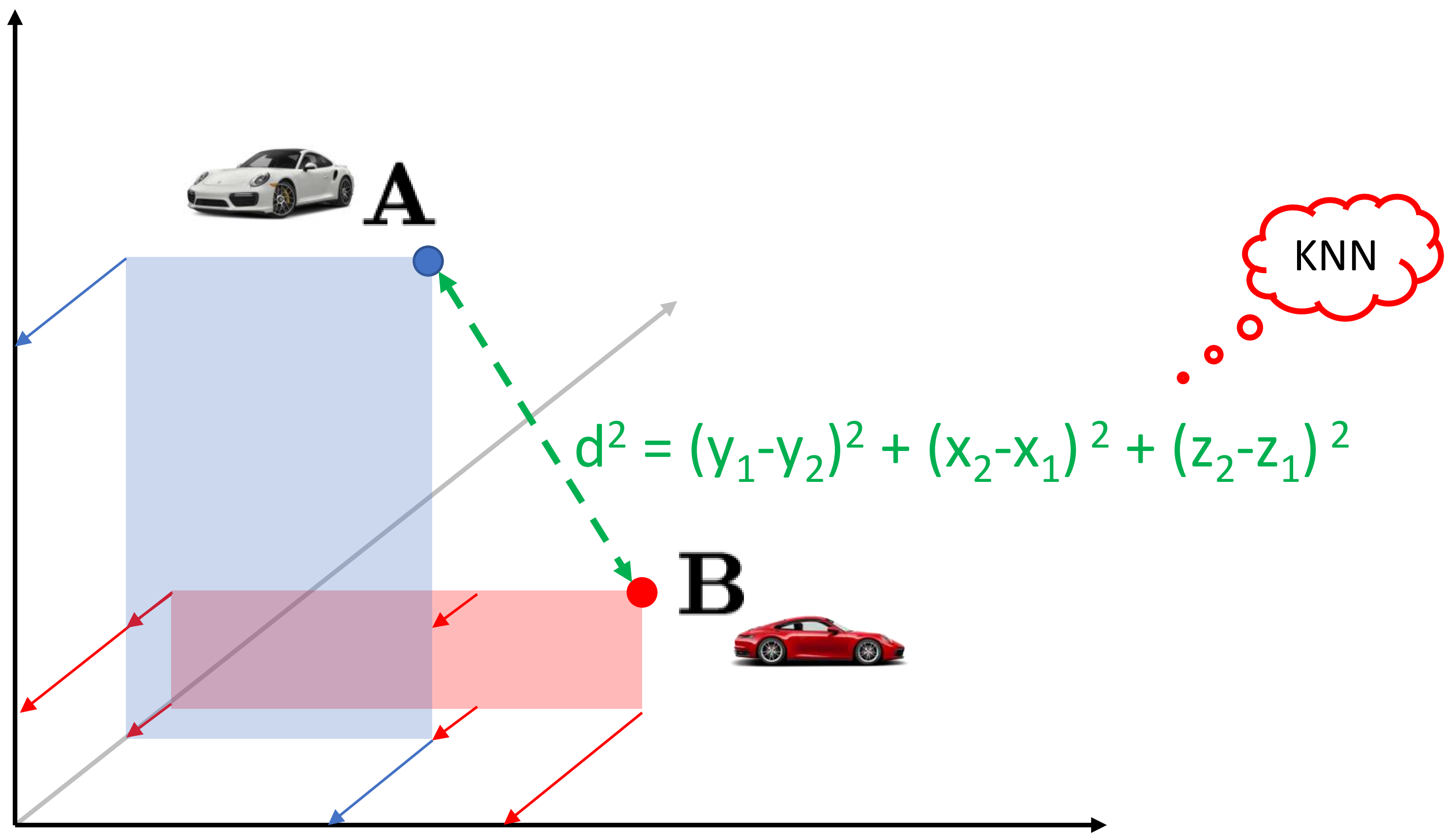
Dokument als Vector



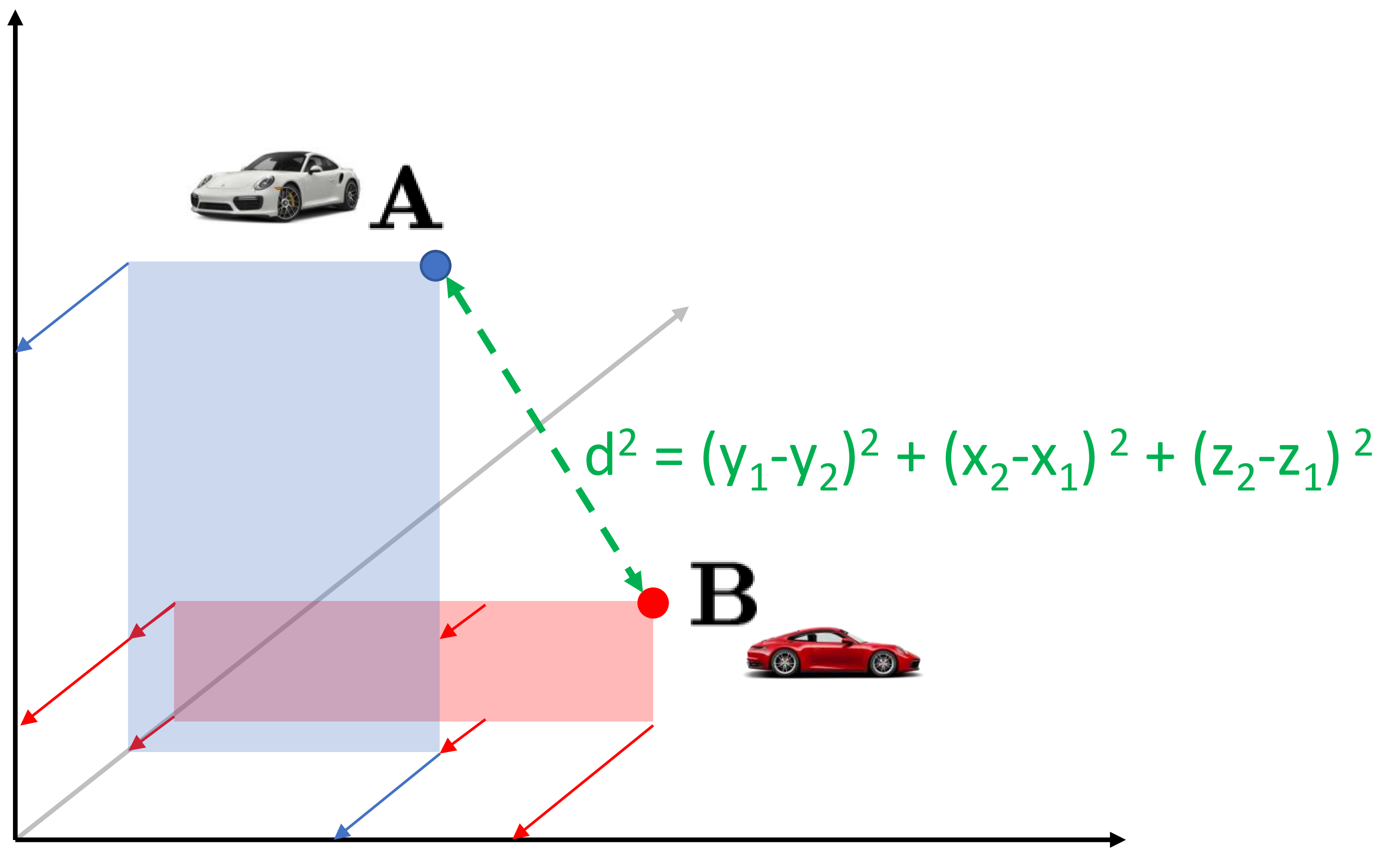
Dokument als Vector



Dokument als Vector

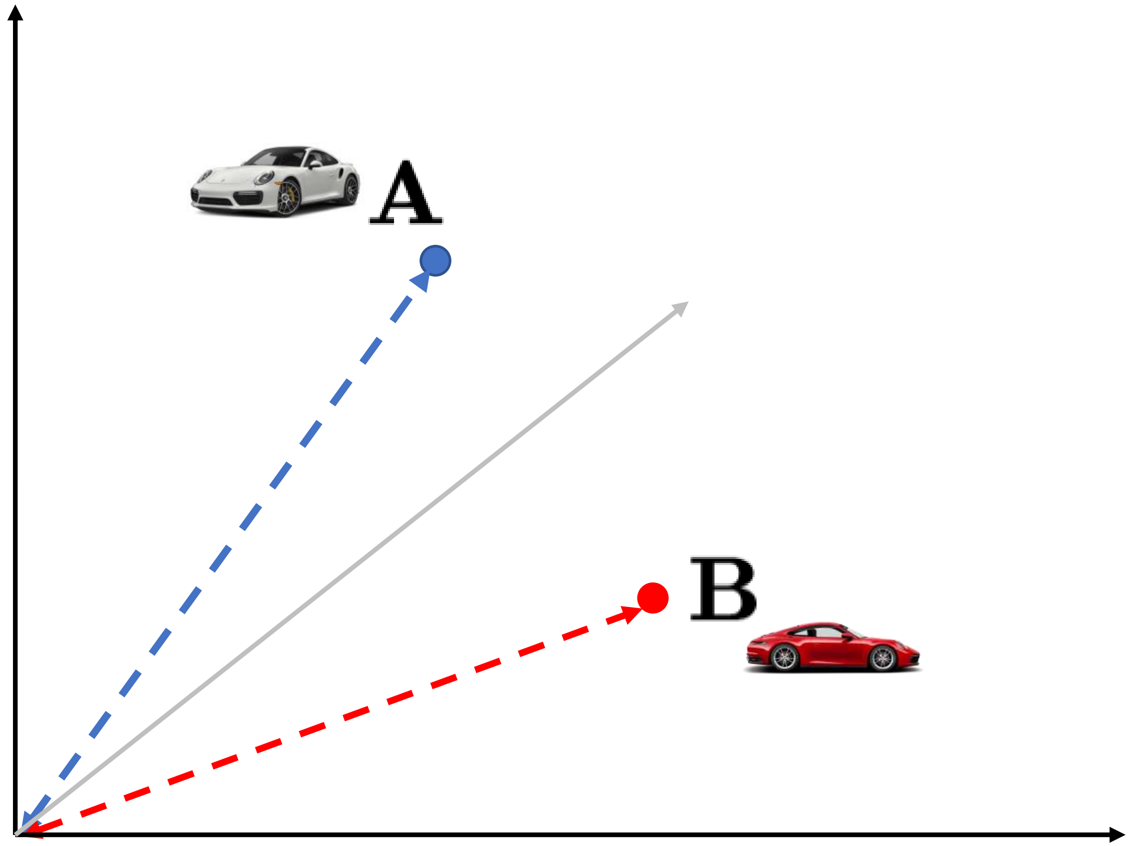


Dokument als Vector

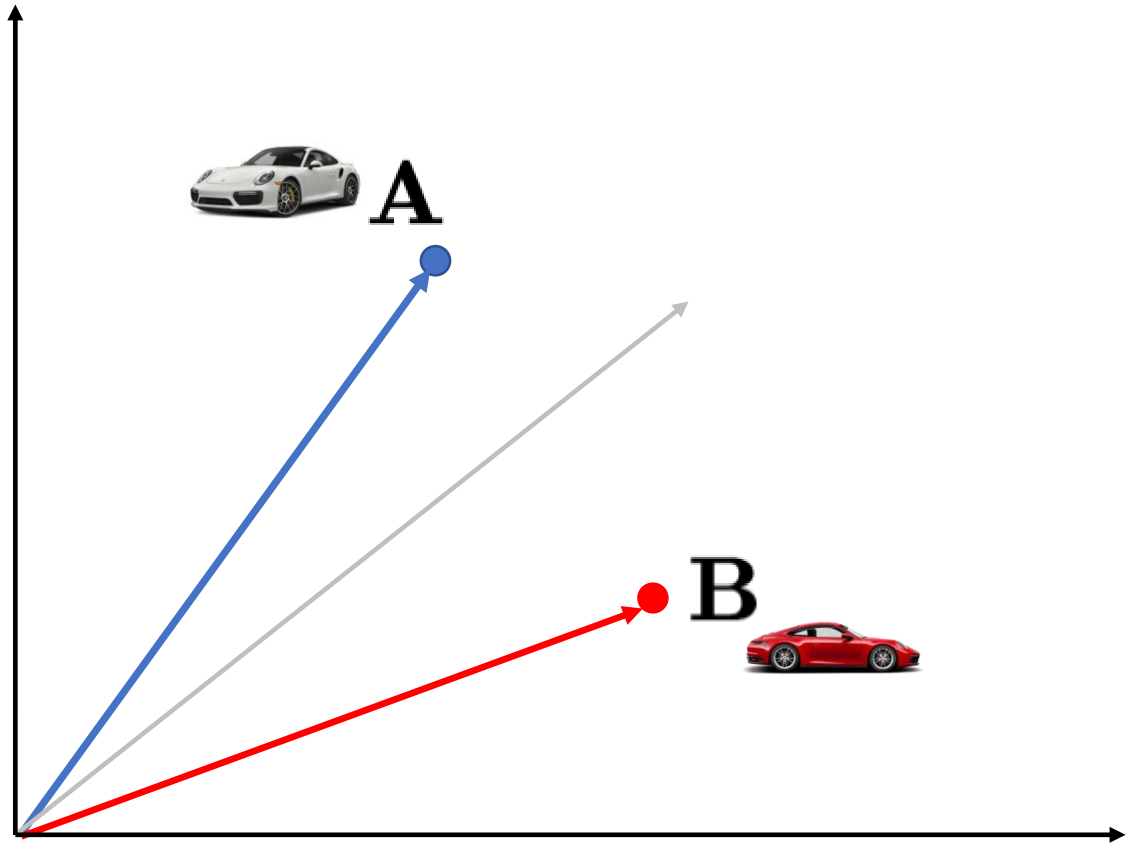


Technische Hochschule Ingolstadt

Dokument als Vector

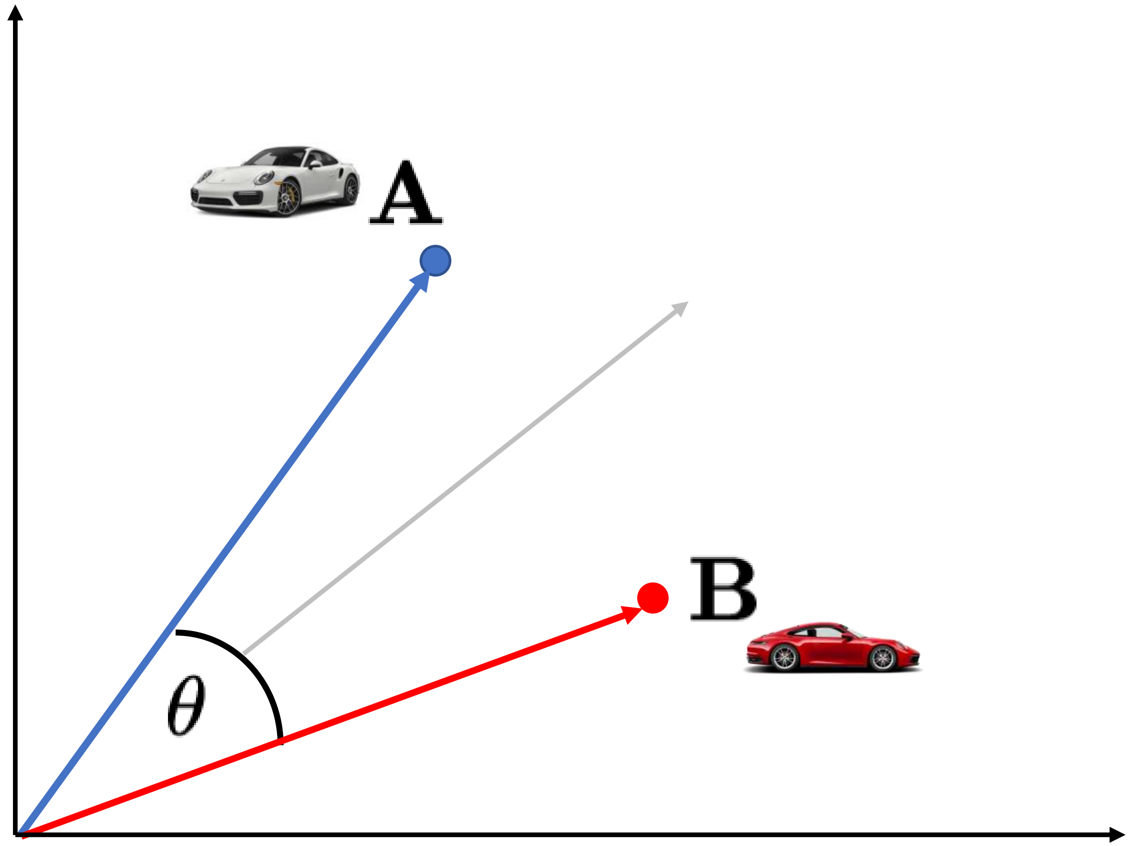


Dokument als Vector



Document as Vector

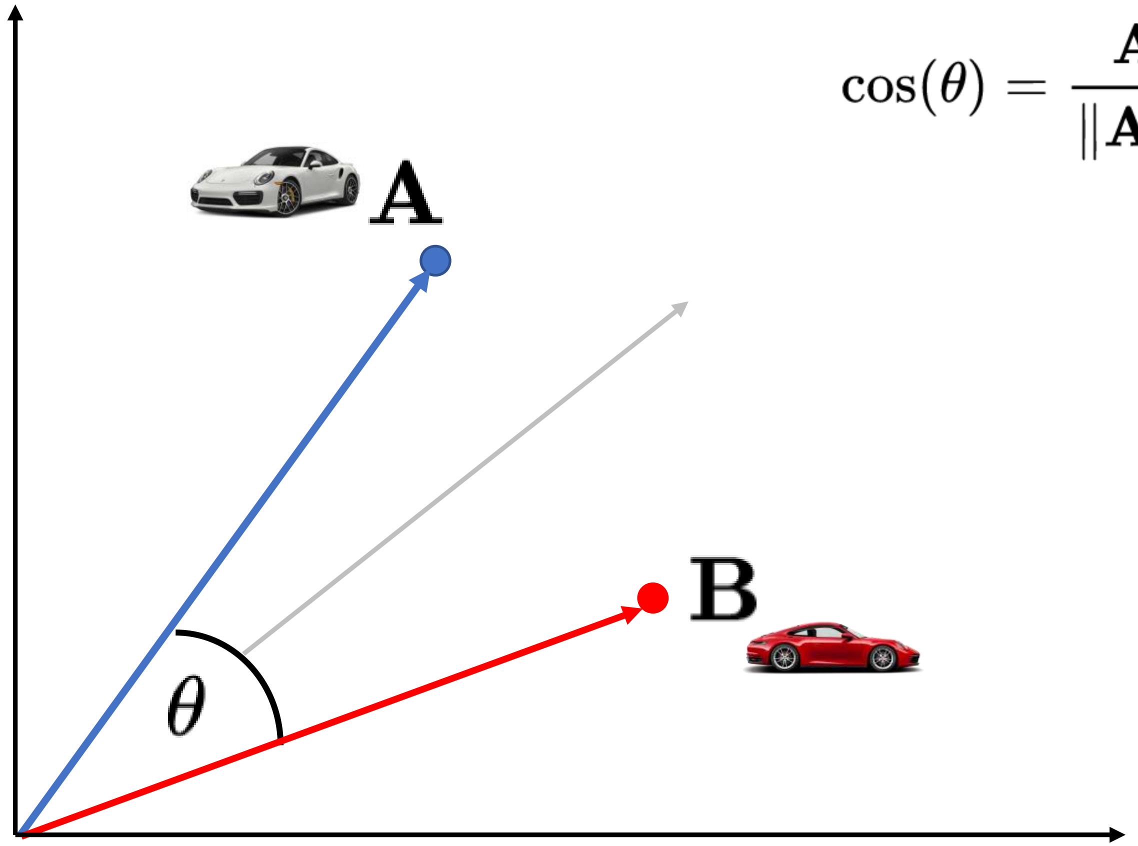
$$\mathbf{A} \cdot \mathbf{B} = \|\mathbf{A}\| \, \|\mathbf{B}\| \cos\theta$$



Document as Vector

$$\mathbf{A} \cdot \mathbf{B} = \|\mathbf{A}\| \, \|\mathbf{B}\| \cos\theta$$

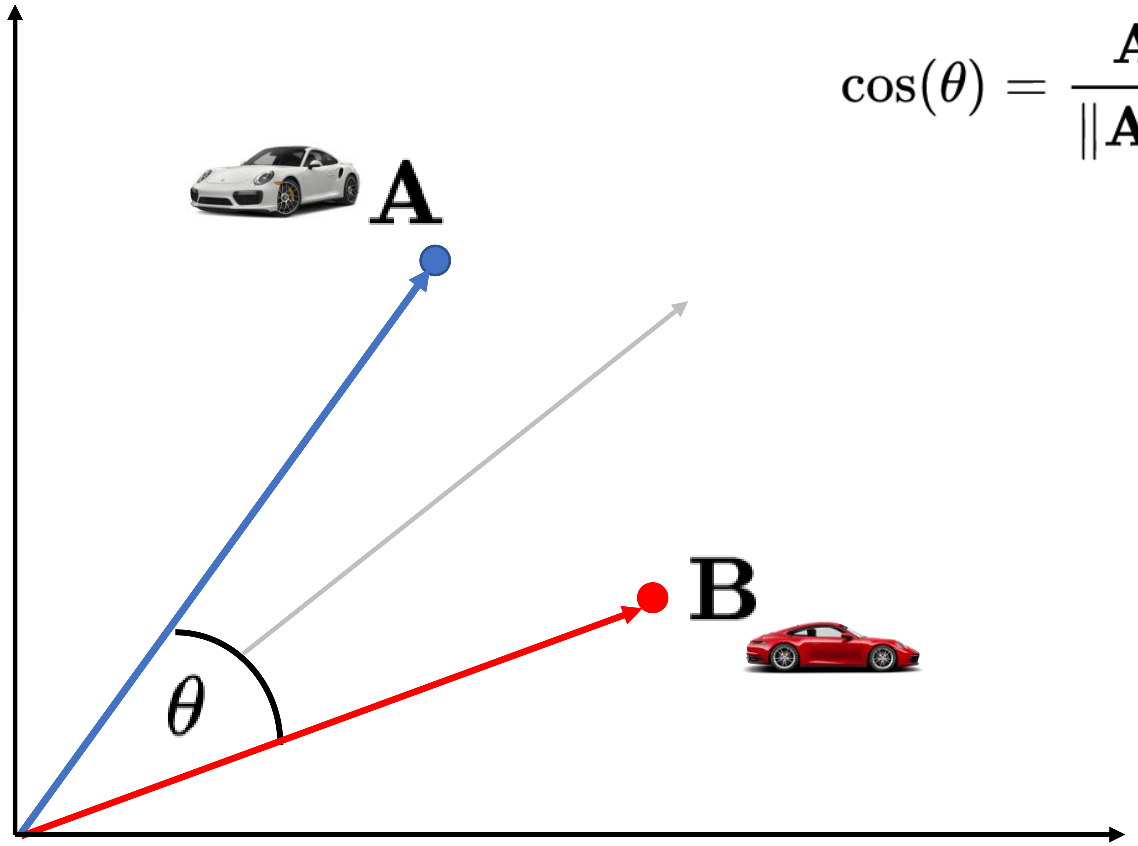
$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|}$$



Document as Vector

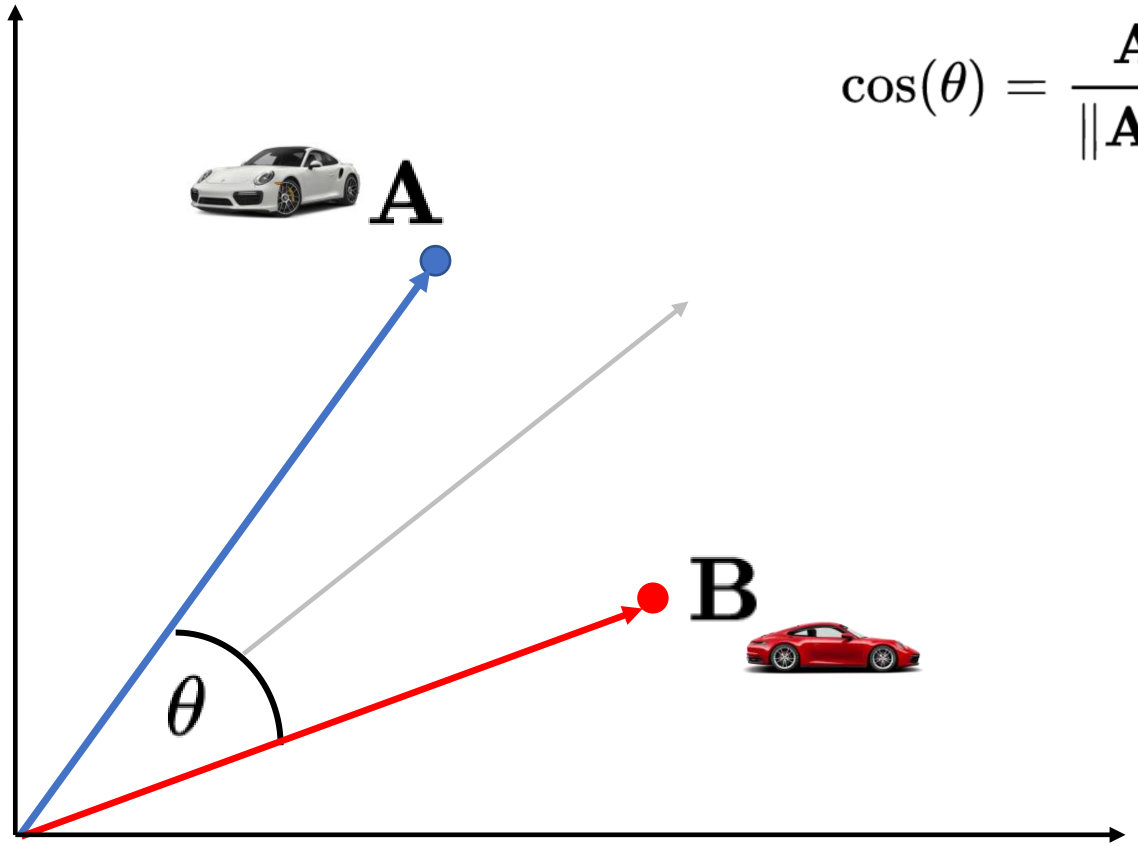
$$\mathbf{A} \cdot \mathbf{B} = \|\mathbf{A}\| \, \|\mathbf{B}\| \cos\theta$$

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\quad}$$



Technische Hochschule Ingolstadt

Document as Vector



$$\mathbf{A} \cdot \mathbf{B} = \|\mathbf{A}\| \, \|\mathbf{B}\| \cos\theta$$

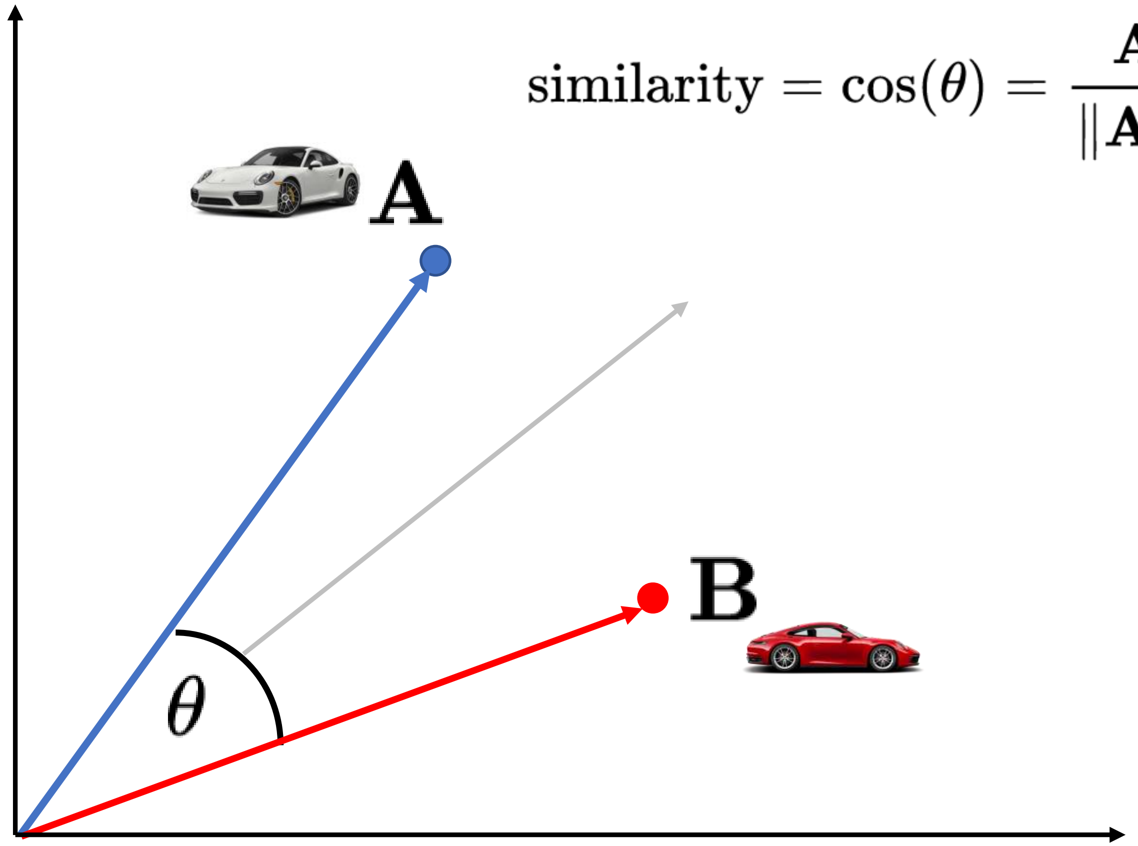
$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2} \sqrt{\sum\limits_{i=1}^{n} B_i^2}}$$

Technische Hochschule Ingolstadt

Cosine Similarity

$$\mathbf{A} \cdot \mathbf{B} = \|\mathbf{A}\| \, \|\mathbf{B}\| \cos\theta$$

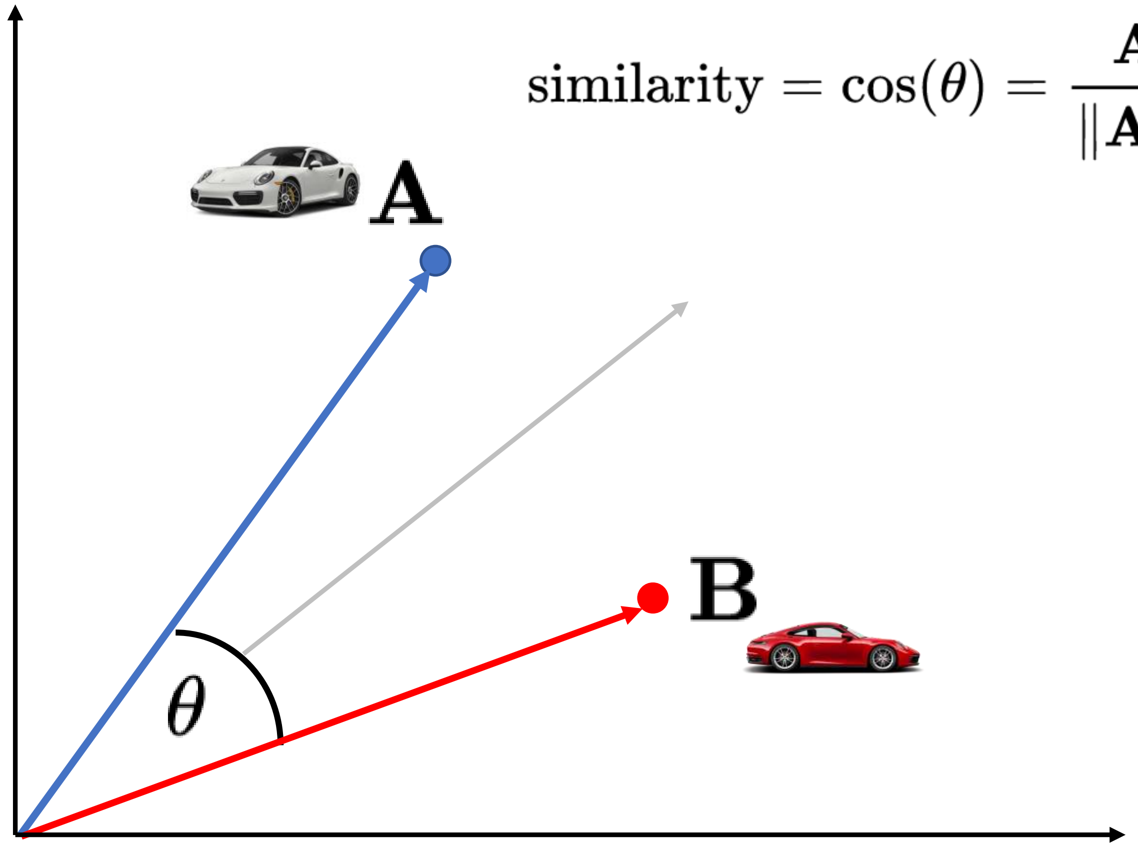
$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2} \sqrt{\sum\limits_{i=1}^{n} B_i^2}}$$



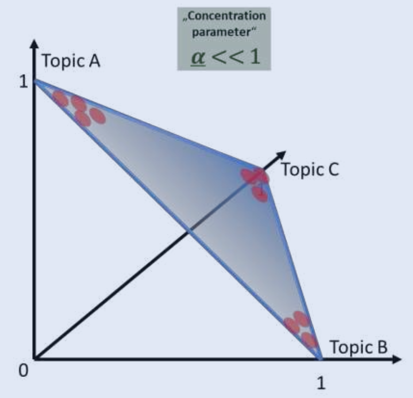
Cosine Similarity

$$\mathbf{A} \cdot \mathbf{B} = \|\mathbf{A}\| \, \|\mathbf{B}\| \cos\theta$$

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$



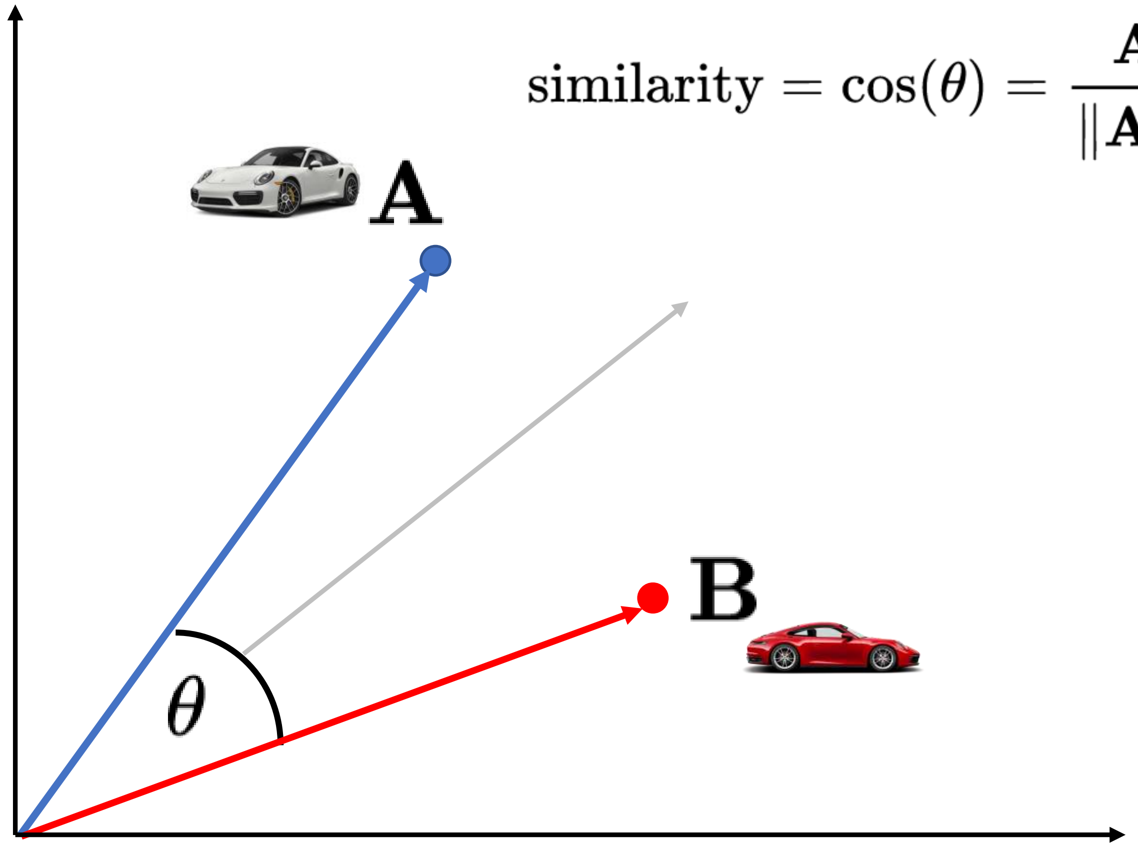
Remember:



Cosine Similarity

$$\mathbf{A} \cdot \mathbf{B} = \|\mathbf{A}\| \, \|\mathbf{B}\| \cos\theta$$

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2} \sqrt{\sum\limits_{i=1}^{n} B_i^2}}$$



The length of the Vector doesn't matter!

Latent Semantic Analysis

“Analyzing relationships between a set of documents and the words they contain by producing a set of concepts related to the documents and terms”

US Patent 4,839,853 issued 1988

Repetition

	Dokument 1	Dokument 2	Dokument 3
Auto	1	0	0
Boot	1	0	0
Kuh	0	1	1
Esel	0	1	0
Tomate	0	0	1
Zwiebel	0	0	1
Traktor	1	0	1
Giraffe	0	1	0
Schwein	0	1	1
Motor	1	0	0

=

?

https://en.wikipedia.org/wiki/Latent_semantic_analysis#Term-document_matrix

Repetition

Topics

A

	Dokument 1	Dokument 2	Dokument 3
Auto	1	0	0
Boot	1	0	0
Kuh	0	1	1
Esel	0	1	0
Tomate	0	0	1
Zwiebel	0	0	1
Traktor	1	0	1
Giraffe	0	1	0
Schwein	0	1	1
Motor	1	0	0

=

φ

	Topic 1	Topic 2
Auto	1	0
Boot	1	0
Kuh	0	1
Esel	0	1
Tomate	0	0
Zwiebel	0	0
Traktor	1	0
Giraffe	0	1
Schwein	0	1
Motor	1	0

ϑ

	Dokument 1	Dokument 2	Dokument 3
Topic 1	1	0	0
Topic 2	1	0	0

ϑ („**documents** vs. **topics**“)

φ („**words/n-grams** vs. **topics**“)

Latent Dirichlet Allocation

φ := phi
ϑ := theta

Technische Hochschule Ingolstadt

Repetition

Topics

A

	Dokument 1	Dokument 2	Dokument 3
Auto	1	0	0
Boot	1	0	0
Kuh	0	1	1
Esel	0	1	0
Tomate	0	0	1
Zwiebel	0	0	1
Traktor	1	0	1
Giraffe	0	1	0
Schwein	0	1	1
Motor	1	0	0

=

φ

	Topic 1	Topic 2
Auto	1	0
Boot	1	0
Kuh	0	1
Esel	0	1
Tomate	0	0
Zwiebel	0	0
Traktor	1	0
Giraffe	0	1
Schwein	0	1
Motor	1	0

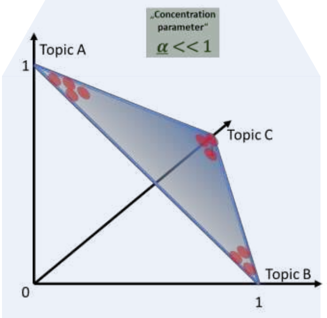
ϑ

	Dokument 1	Dokument 2	Dokument 3
Topic 1	1	0	0
Topic 2	1	0	0

ϑ („**documents** vs. **topics**“)

φ („**words/n-grams** vs. **topics**“)

Latent Dirichlet Allocation



φ := phi
ϑ := theta

Be Creative!

	Dokument 1	Dokument 2	Dokument 3
Auto	1	0	0
Boot	1	0	0
Kuh	0	1	1
Esel	0	1	0
Tomate	0	0	1
Zwiebel	0	0	1
Traktor	1	0	1
Giraffe	0	1	0
Schwein	0	1	1
Motor	1	0	0

= ?

https://en.wikipedia.org/wiki/Latent_semantic_analysis#Term-document_matrix

Latent Semantic Analysis

A

	Dokument 1	Dokument 2	Dokument 3
Auto	1	0	0
Boot	1	0	0
Kuh	0	1	1
Esel	0	1	0
Tomate	0	0	1
Zwiebel	0	0	1
Traktor	1	0	1
Giraffe	0	1	0
Schwein	0	1	1
Motor	1	0	0

=

U

	Hidden 1	Hidden 2
Auto	1	0
Boot	1	0
Kuh	0	1
Esel	0	1
Tomate	0	0
Zwiebel	0	0
Traktor	1	0
Giraffe	0	1
Schwein	0	1
Motor	1	0

S

	Hidden 1	Hidden 2
Hidden 1	1	0
Hidden 2	0	1

V*

	Dokument 1	Dokument 2	Dokument 3
Hidden 1	1	0	0
Hidden 2	1	[illegible]	0

Latent Semantic Analysis

A

	Dokument 1	Dokument 2	Dokument 3
Auto	1	0	0
Boot	1	0	0
Kuh	0	1	1
Esel	0	1	0
Tomate	0	0	1
Zwiebel	0	0	1
Traktor	1	0	1
Giraffe	0	1	0
Schwein	0	1	1
Motor	1	0	0

=

U

	Hidden 1	Hidden 2
Auto	1	0
Boot	1	0
Kuh	0	1
Esel	0	1
Tomate	0	0
Zwiebel	0	0
Traktor	1	0
Giraffe	0	1
Schwein	0	1
Motor	1	0

S

	Hidden 1	Hidden 2
Hidden 1	1	0
Hidden 2	0	1

V*

	Dokument 1	Dokument 2	Dokument 3
Hidden 1	1	0	0
Hidden 2	1	[illegible]	0

$\underbrace{U \; S \; V^*}_{\text{Singular Value Decomposition}}$

Singular Value Decomposition

Technische Hochschule Ingolstadt

Singular Value Decomposition

A

	Dokument 1	Dokument 2	Dokument 3
Auto	1	0	0
Boot	1	0	0
Kuh	0	1	1
Esel	0	1	0
Tomate	0	0	1
Zwiebel	0	0	1
Traktor	1	0	1
Giraffe	0	1	0
Schwein	0	1	1
Motor	1	0	0

=

U

	Hidden 1	Hidden 2
Auto	1	0
Boot	1	0
Kuh	0	1
Esel	0	1
Tomate	0	0
Zwiebel	[illegible]	0
Traktor	1	0
Giraffe	0	1
Schwein	0	1
Motor	1	0

S

	Hidden 1	Hidden 2
Hidden 1	1	0
Hidden 2	0	1

V*

	Dokument 1	Dokument 2	Dokument 3
Hidden 1	1	0	0
Hidden 2	1	[illegible]	0

Dimension reduction via eigenvalues

http://lsa.colorado.edu/papers/JASIS.lsi.90.pdf

Singular Value Decomposition

A

	Dokument 1	Dokument 2	Dokument 3
Auto	1	0	0
Boot	1	0	0
Kuh	0	1	1
Esel	0	1	0
Tomate	0	0	1
Zwiebel	0	0	1
Traktor	1	0	1
Giraffe	0	1	0
Schwein	0	1	1
Motor	1	0	0

=

U

	Hidden 1	Hidden 2
Auto	1	0
Boot	1	0
Kuh	0	1
Esel	0	1
Tomate	0	0
Zwiebel	0	0
Traktor	1	0
Giraffe	0	1
Schwein	0	1
Motor	1	0

S

	Hidden 1	Hidden 2
Hidden 1	1	0
Hidden 2	0	1

V*

	Dokument 1	Dokument 2	Dokument 3
Hidden 1	1	0	0
Hidden 2	1	[illegible]	0

Dimension reduction via eigenvalues

„The dimensionally reduced space reflects the underlying structure of the documents, i.e. their semantics. “

aka all unnecessary information (noise or grammatical constructs) has been removed as they don't contribute to semantics!

http://lsa.colorado.edu/papers/JASIS.lsi.90.pdf

Technische Hochschule Ingolstadt

Singular Value Decomposition

A

	Dokument 1	Dokument 2	Dokument 3
Auto	1	0	0
Boot	1	0	0
Kuh	0	1	1
Esel	0	1	0
Tomate	0	0	1
Zwiebel	0	0	1
Traktor	1	0	1
Giraffe	0	1	0
Schwein	0	1	1
Motor	1	0	0

=

U

	Hidden 1	Hidden 2
Auto	1	0
Boot	1	0
Kuh	0	1
Esel	0	1
Tomate	0	0
Zwiebel	0	0
Traktor	1	0
Giraffe	0	1
Schwein	0	1
Motor	1	0

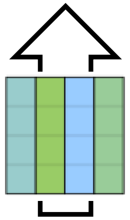
S

	Hidden 1	Hidden 2
Hidden 1	1	0
Hidden 2	0	1

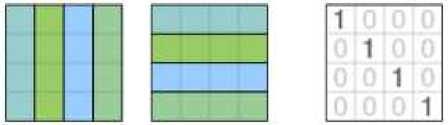
V*

	Dokument 1	Dokument 2	Dokument 3
Hidden 1	1	0	0
Hidden 2	1	[illegible]	0

$m \times m$



Unäre Matrix



$$U \; U^* = I_m$$

http://lsa.colorado.edu/papers/JASIS.lsi.90.pdf

Technische Hochschule Ingolstadt

Singular Value Decomposition

A

	Dokument 1	Dokument 2	Dokument 3
Auto	1	0	0
Boot	1	0	0
Kuh	0	1	1
Esel	0	1	0
Tomate	0	0	1
Zwiebel	0	0	1
Traktor	1	0	1
Giraffe	0	1	0
Schwein	0	1	1
Motor	1	0	0

=

U

	Hidden 1	Hidden 2
Auto	1	0
Boot	1	0
Kuh	0	1
Esel	0	1
Tomate	0	0
Zwiebel	0	0
Traktor	1	0
Giraffe	0	1
Schwein	0	1
Motor	1	0

$m \times m$

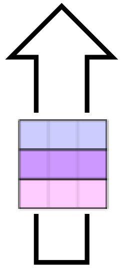
S

	Hidden 1	Hidden 2
Hidden 1	1	0
Hidden 2	0	1

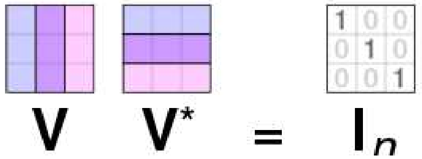
V*

	Dokument 1	Dokument 2	Dokument 3
Hidden 1	1	0	0
Hidden 2	1	[illegible]	0

$n \times n$



Adjungierte unitäre Matrix



http://lsa.colorado.edu/papers/JASIS.lsi.90.pdf

Singular Value Decomposition

A

	Dokument 1	Dokument 2	Dokument 3
Auto	1	0	0
Boot	1	0	0
Kuh	0	1	1
Esel	0	1	0
Tomate	0	0	1
Zwiebel	0	0	1
Traktor	1	0	1
Giraffe	0	1	0
Schwein	0	1	1
Motor	1	0	0

=

U

	Hidden 1	Hidden 2
Auto	1	0
Boot	1	0
Kuh	0	1
Esel	0	1
Tomate	0	0
Zwiebel	[illegible]	0
Traktor	1	0
Giraffe	0	1
Schwein	0	1
Motor	1	0

$m \times m$

S

	Hidden 1	Hidden 2
Hidden 1	1	0
Hidden 2	0	1

$m \times n$

V*

	Dokument 1	Dokument 2	Dokument 3
Hidden 1	1	0	0
Hidden 2	1	[illegible]	0

$n \times n$

Singularvalues

σ_1	0	…	0	0
0	σ_1	0	…	0
0	0	…	0	…
…	…		σ_{r-1}	0
0	0	…	0	σ_r

http://lsa.colorado.edu/papers/JASIS.lsi.90.pdf

Technische Hochschule Ingolstadt

Latent Semantic Analysis

A

	Dokument 1	Dokument 2	Dokument 3
Auto	1	0	0
Boot	1	0	0
Kuh	0	1	1
Esel	0	1	0
Tomate	0	0	1
Zwiebel	0	0	1
Traktor	1	0	1
Giraffe	0	1	0
Schwein	0	1	1
Motor	1	0	0

=

U

	Hidden 1	Hidden 2
Auto	1	0
Boot	1	0
Kuh	0	1
Esel	0	1
Tomate	0	0
Zwiebel	0	0
Traktor	1	0
Giraffe	0	1
Schwein	0	1
Motor	1	0

S

	Hidden 1	Hidden 2
Hidden 1	1	0
Hidden 2	0	1

V*

	Dokument 1	Dokument 2	Dokument 3
Hidden 1	1	0	0
Hidden 2	1	[illegible]	0

$$Q = \frac{q^T U_k}{diag(S_k)}$$

Remember:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2} \sqrt{\sum\limits_{i=1}^{n} B_i^2}}$$

Technische Hochschule Ingolstadt

Latent Semantic Analysis

A

	Dokument 1	Dokument 2	Dokument 3
Auto	1	0	0
Boot	1	0	0
Kuh	0	1	1
Esel	0	1	0
Tomate	0	0	1
Zwiebel	0	0	1
Traktor	1	0	1
Giraffe	0	1	0
Schwein	0	1	1
Motor	1	0	0

=

U

	Hidden 1	Hidden 2
Auto	1	0
Boot	1	0
Kuh	0	1
Esel	0	1
Tomate	0	0
Zwiebel	0	0
Traktor	1	0
Giraffe	0	1
Schwein	0	1
Motor	1	0

S

	Hidden 1	Hidden 2
Hidden 1	1	0
Hidden 2	0	1

V*

	Dokument 1	Dokument 2	Dokument 3
Hidden 1	1	0	0
Hidden 2	1	[illegible]	0

$$Q = \frac{q^T U_k}{diag(S_k)}$$

„**Latent Vector**“of all documents and the document to be classified
Distance between the latent vectors reflects the similarity of the documents

Technische Hochschule Ingolstadt

Latent Semantic Analysis

A

	Dokument 1	Dokument 2	Dokument 3
Auto	1	0	0
Boot	1	0	0
Kuh	0	1	1
Esel	0	1	0
Tomate	0	0	1
Zwiebel	0	0	1
Traktor	1	0	1
Giraffe	0	1	0
Schwein	0	1	1
Motor	1	0	0

=

U

	Hidden 1	Hidden 2
Auto	1	0
Boot	1	0
Kuh	0	1
Esel	0	1
Tomate	0	0
Zwiebel	0	0
Traktor	1	0
Giraffe	0	1
Schwein	0	1
Motor	1	0

S

	Hidden 1	Hidden 2
Hidden 1	1	0
Hidden 2	0	1

V^T

	Dokument 1	Dokument 2	Dokument 3
Hidden 1	1	0	0
Hidden 2	1	[illegible]	0

Input "Query" (i.e. a single document)

$$Q = \frac{q^T U_k}{diag(S_k)}$$

Latent Semantic Analysis

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

A

	Dokument 1	Dokument 2	Dokument 3
Auto	1	0	0
Boot	1	0	0
Kuh	0	1	1
Esel	0	1	0
Tomate	0	0	1
Zwiebel	0	0	1
Traktor	1	0	1
Giraffe	0	1	0
Schwein	0	1	1
Motor	1	0	0

U

	Hidden 1	Hidden 2
Auto	1	0
Boot	1	0
Kuh	0	1
Esel	0	1
Tomate	0	0
Zwiebel	0	0
Traktor	1	0
Giraffe	0	1
Schwein	0	1
Motor	1	0

S

	Hidden 1	Hidden 2
Hidden 1	1	0
Hidden 2	0	1

V^T

	Dokument 1	Dokument 2	Dokument 3
Hidden 1	1	0	0
Hidden 2	1	[illegible]	0

Input "Query" (i.e. a single document)

Dimension-reduced word to "topics" matrix

$$Q = \frac{q^T U_k}{diag(S_k)}$$

Latent Semantic Analysis

A =

	Dokument 1	Dokument 2	Dokument 3
Auto	1	0	0
Boot	1	0	0
Kuh	0	1	1
Esel	0	1	0
Tomate	0	0	1
Zwiebel	0	0	1
Traktor	1	0	1
Giraffe	0	1	0
Schwein	0	1	1
Motor	1	0	0

U

	Hidden 1	Hidden 2
Auto	1	0
Boot	1	0
Kuh	0	1
Esel	0	1
Tomate	0	0
Zwiebel	0	0
Traktor	1	0
Giraffe	0	1
Schwein	0	1
Motor	1	0

S

	Hidden 1	Hidden 2
Hidden 1	1	0
Hidden 2	0	1

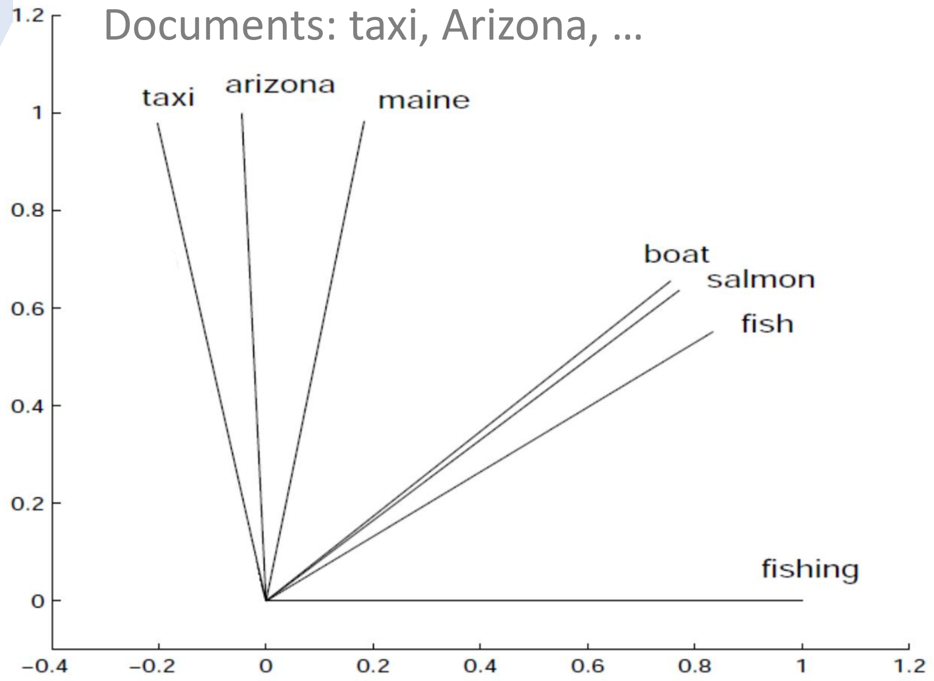
V^T

	Dokument 1	Dokument 2	Dokument 3
Hidden 1	1	0	0
Hidden 2	1	[illegible]	0

Input "Query" (i.e. a single document)

Dimensionsreduziert Wort zu „topics“ Matrix

$$Q = \frac{q^T U_k}{diag(S_k)}$$



Technische Hochschule Ingolstadt

Latent Semantic Analysis

A =

	Dokument 1	Dokument 2	Dokument 3
Auto	1	0	0
Boot	1	0	0
Kuh	0	1	1
Esel	0	1	0
Tomate	0	0	1
Zwiebel	0	0	1
Traktor	1	0	1
Giraffe	0	1	0
Schwein	0	1	1
Motor	1	0	0

U

	Hidden 1	Hidden 2
Auto	1	0
Boot	1	0
Kuh	0	1
Esel	0	1
Tomate	0	0
Zwiebel	0	0
Traktor	1	0
Giraffe	0	1
Schwein	0	1
Motor	1	0

S

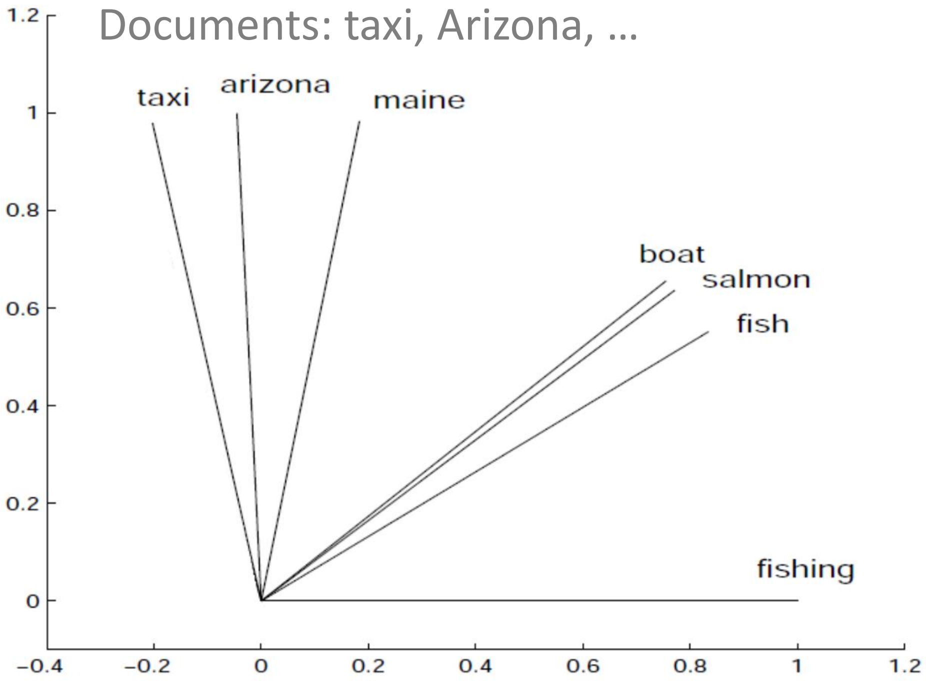
	Hidden 1	Hidden 2
Hidden 1	1	0
Hidden 2	0	1

V^T

	Dokument 1	Dokument 2	Dokument 3
Hidden 1	1	0	0
Hidden 2	1	[illegible]	0

$$Q = \frac{q^T U_k}{diag(S_k)}$$

Documents: taxi, Arizona, …



∡(fishing, fish) < ∡(fishing, taxi)
∡(fishing, boat) < ∡(fishing, taxi)

Properties

- Robust to synonyms
- Sensitive to polysemy := One word, multiple meanings, which can lead to misunderstandings and fallacies (slightly better/worse depending on the characteristic)
- Computational heavy: $O(n^2 \cdot k^3)$
  - n := Number of documents + number of features
  - k := Dimension of space

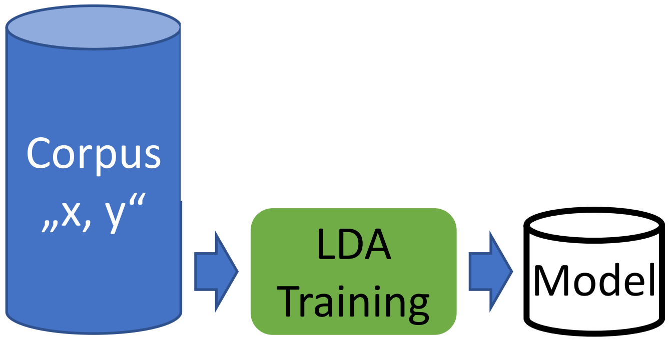
  $\Rightarrow$Lanczos process
- Non-iterative, i.e. recalculating for each document
- "K" is unknown, i.e. you have to optimize manually...

https://de.wikipedia.org/wiki/Synonym
https://de.wikipedia.org/wiki/Polysemie
https://de.wikipedia.org/wiki/Lanczos-Verfahren

Transfer Knowledge

“Learn the most your can
out of data.”

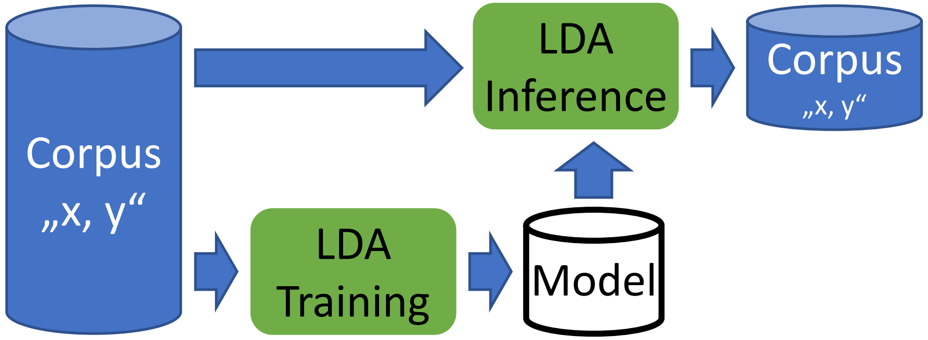
Example application



LDA($\underline{t}$) := P(d|$\underline{t}$)

„Estimate LDA Model“

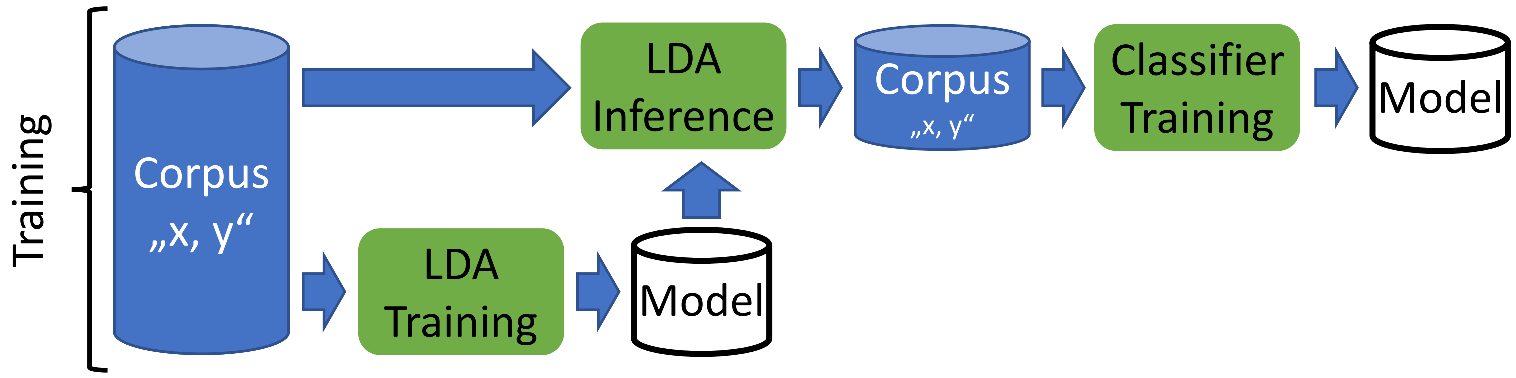
Example application



Compute P(d|t)

for the corpus

Example application

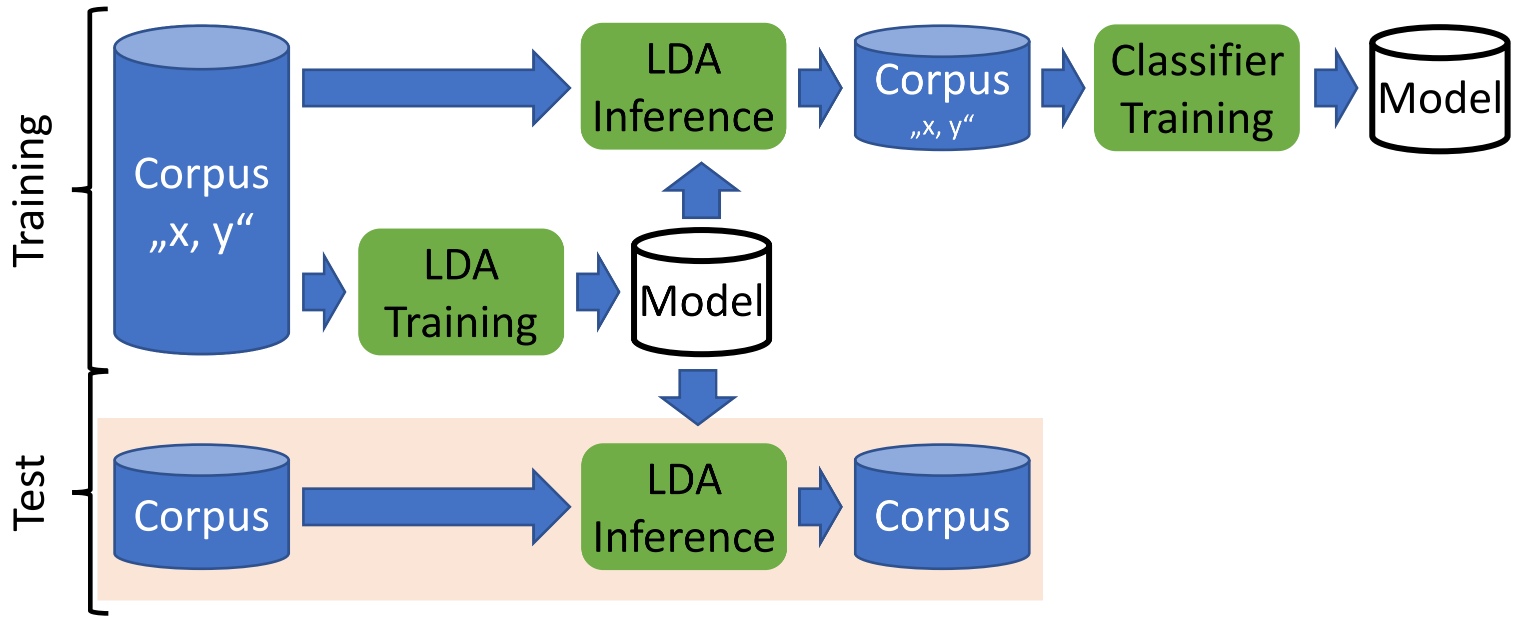


$$D(\underline{t}) = \text{argmax}_{d=\{\ldots\}}\, P(d|LDA(\underline{t}))$$

$$= \text{argmax}_{d=\{\ldots\}}\, P(LDA(\underline{t})|d)P(d)$$

„Estimate Classifier"

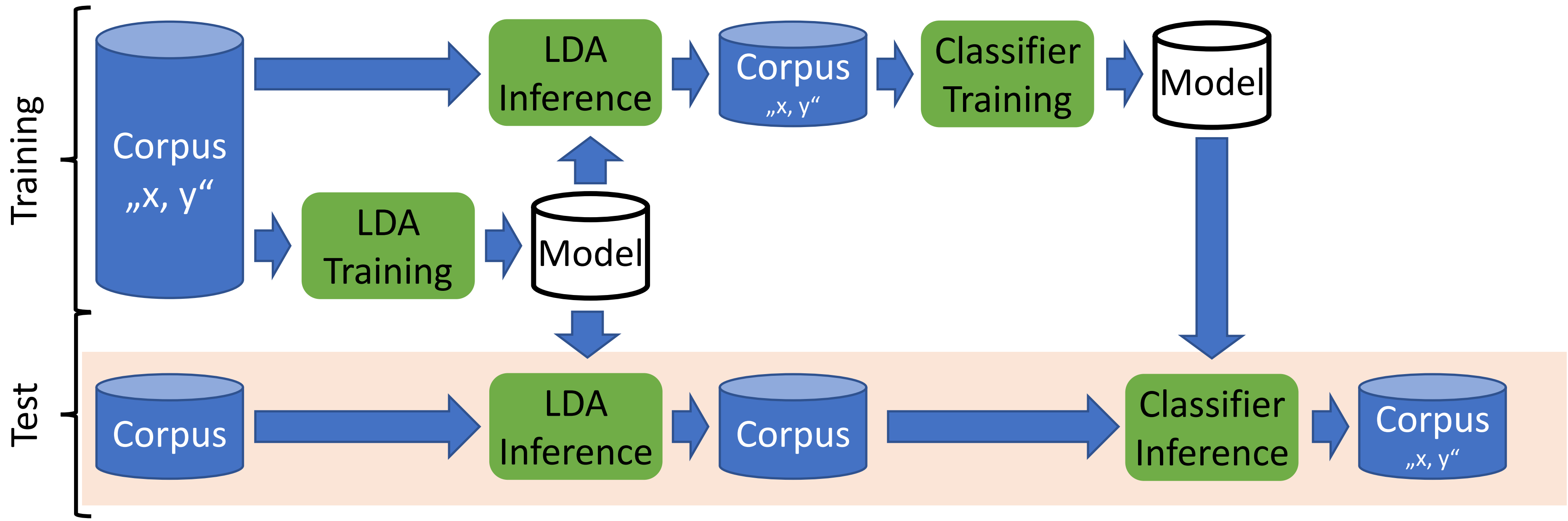
Example application



Inference:

- Compute LDA-Features

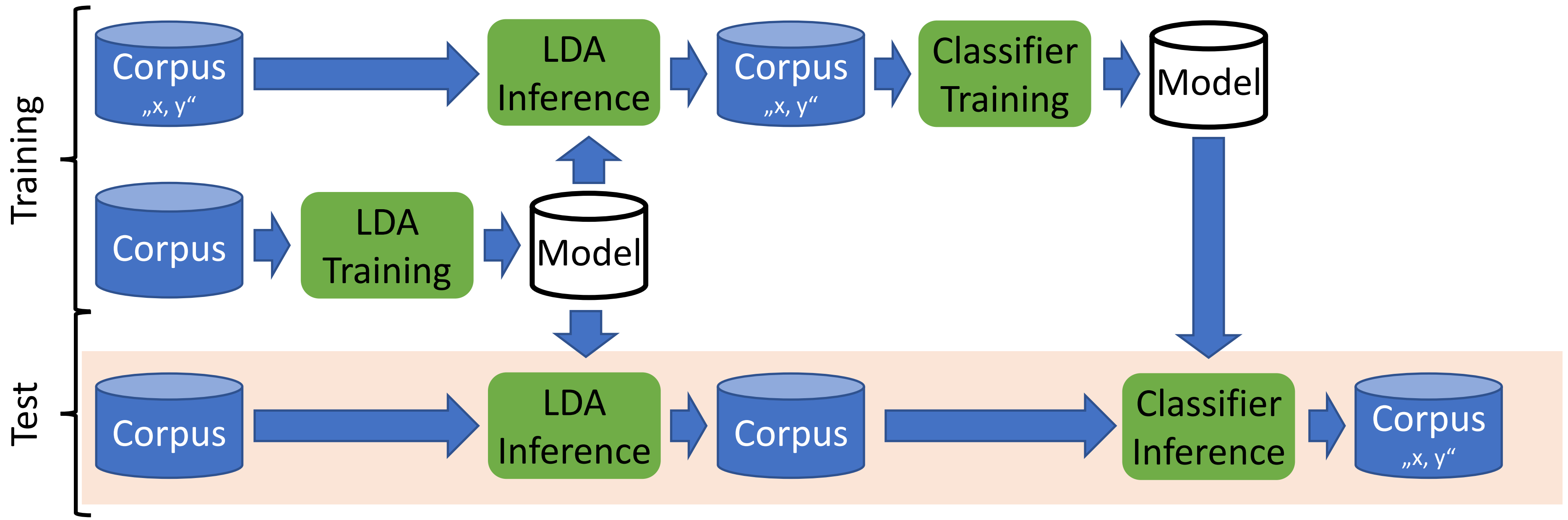
Example application



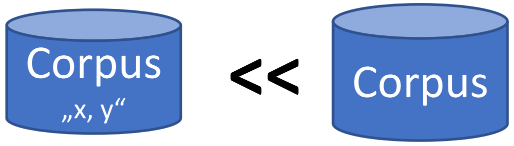
Inference:

- Compute LDA-Features
- Classification

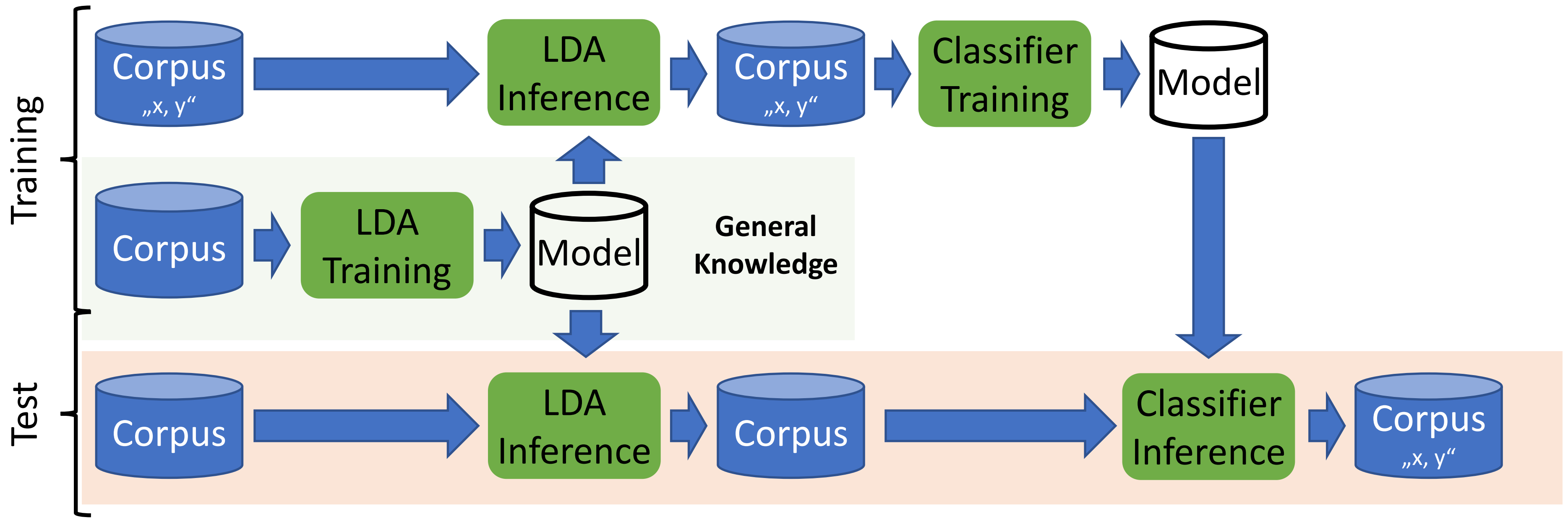
Transfer Knowledge



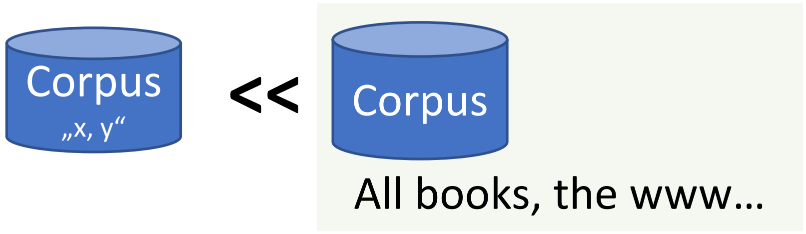
Motivation:



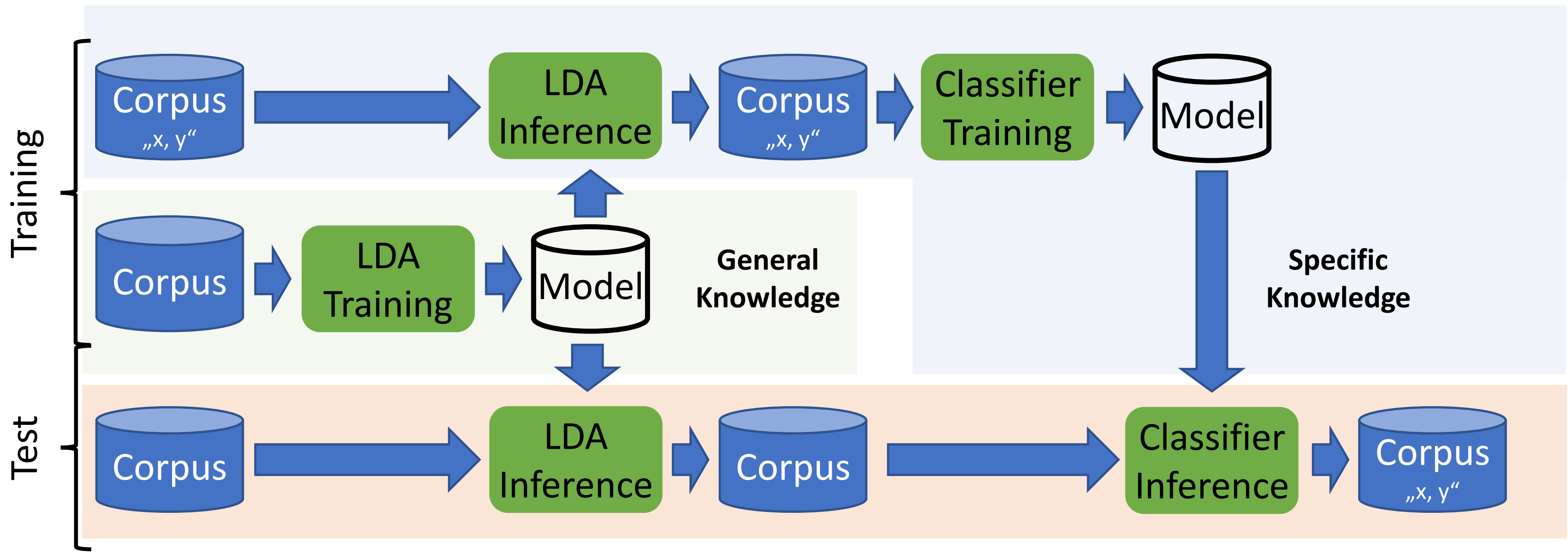
Transfer Knowledge



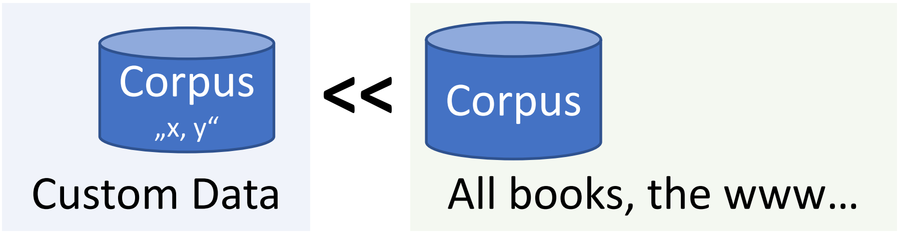
Motivation:



Transfer Knowledge

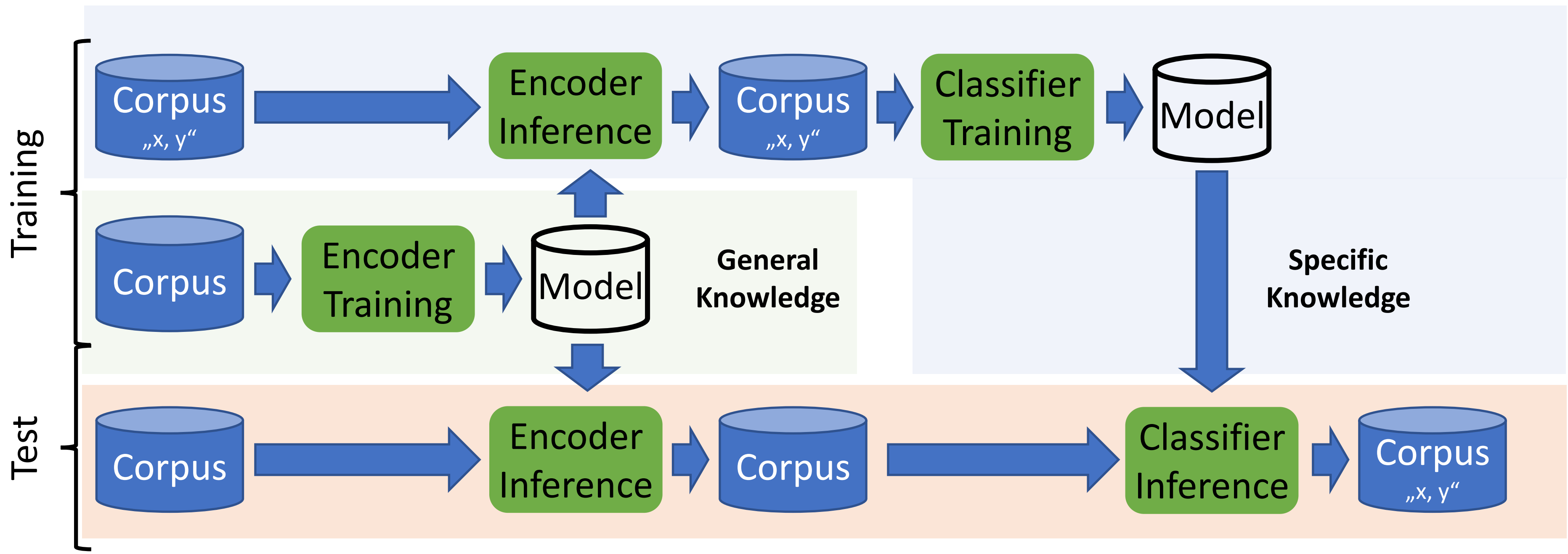


Motivation:

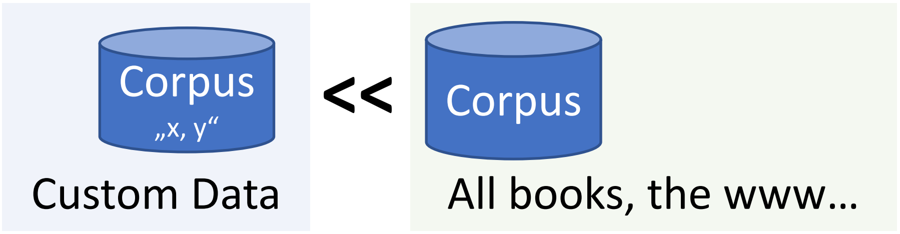


Technische Hochschule Ingolstadt

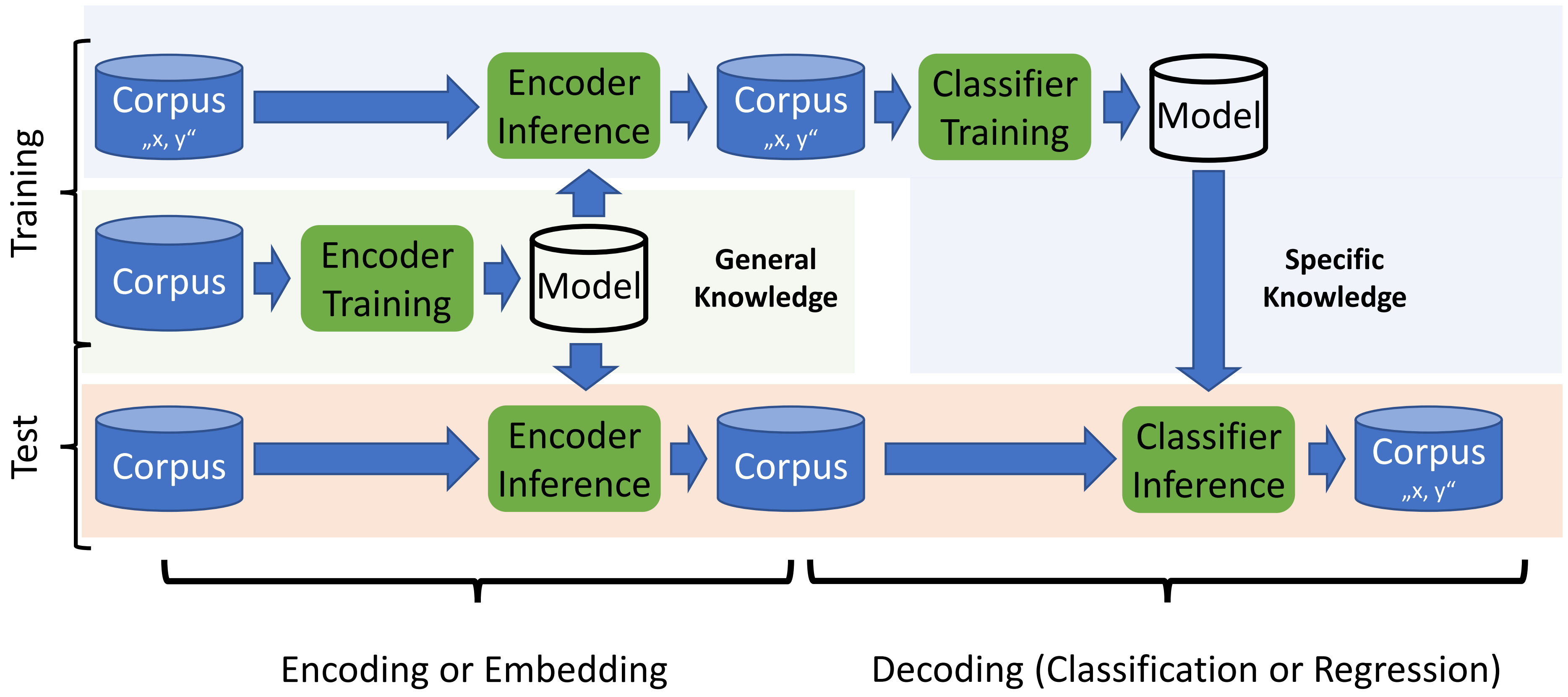
Encoder in Transfer Knowledge



Motivation:



Encoder-Decoder Architecture

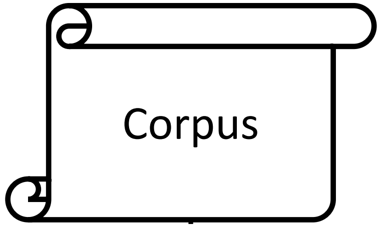


Applications

- Efficient word search
- Efficient text pre-processing
- Sprachmodellierung
- Spell checker
- Grapheme-to-Phoneme
- Spam/no-spam detection
- Sprachenerkennung
- Dokumentenerkennung
- Topic detection (unobserved)
- Sentiment Detection
- …

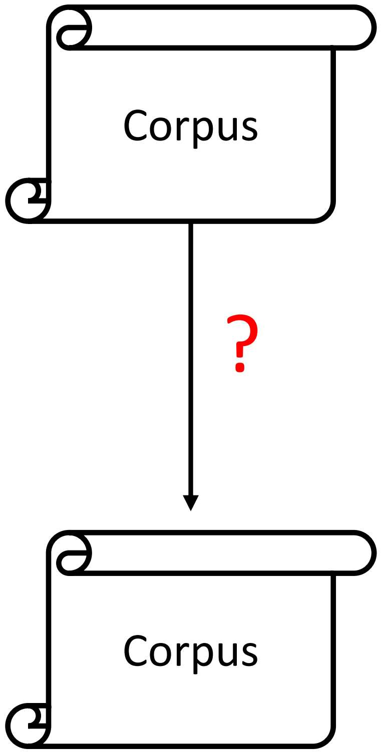
Technische Hochschule Ingolstadt

Text Preprocessing



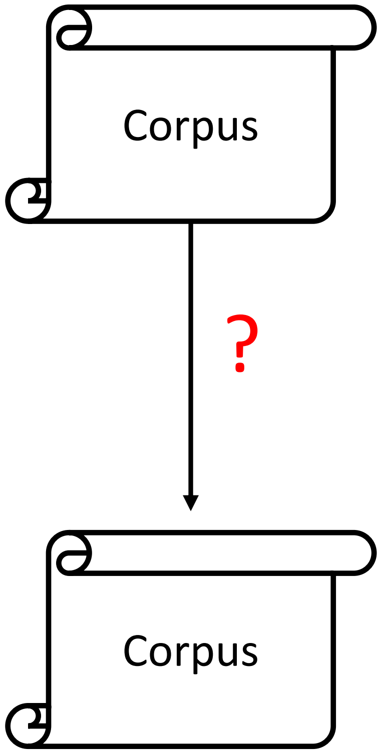
?

Text Preprocessing



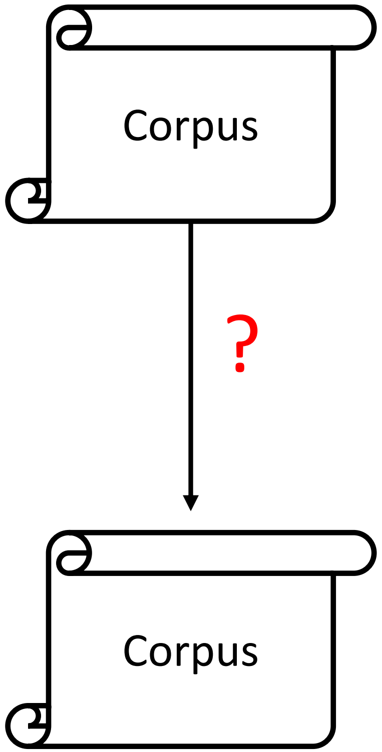
Text Preprocessing

Capitalization
Punctuation mark
Special character
Noise
Abbreviations
(Headings)
Satzsegmentierung
Wortsegmentierung
(Tokenization)
Lemma/Coat of arms



Text Preprocessing

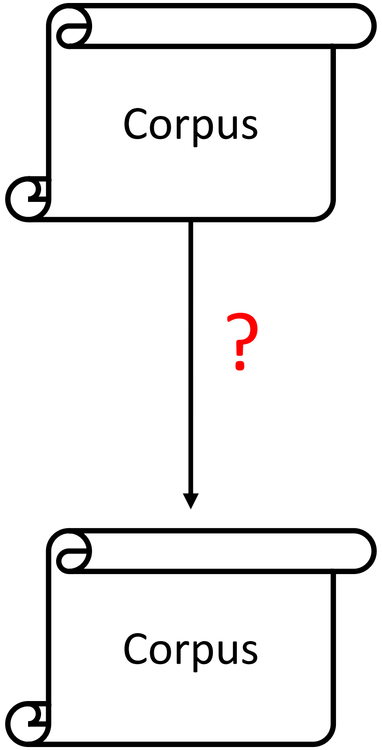
Capitalization
Punctuation mark
Special character
Noise
Abbreviations
(Headings)
Satzsegmentierung
Wortsegmentierung
(Tokenization)
Lemma/Coat of arms



Fuzzy String Match
(Levenshtein distance and WHO)
Soundex
Final State Transducer
Grammars (Formal Languages)
Decision Tree
Morphology
Max-Match Segmentation
Porter’s algorithm
Stop Words

Text Preprocessing

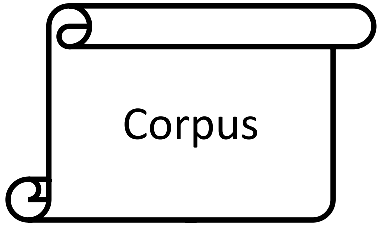
Capitalization
Punctuation mark
Special character
Noise
Abbreviations
(Headings)
Satzsegmentierung
Wortsegmentierung
(Tokenization)
Lemma/Coat of arms



Fuzzy String Match
(Levenshtein distance and WHO)
Soundex
Final State Transducer
Grammars (Formal Languages)
Decision Tree
Morphology
Max-Match Segmentation
Porter’s algorithm
Stop Words

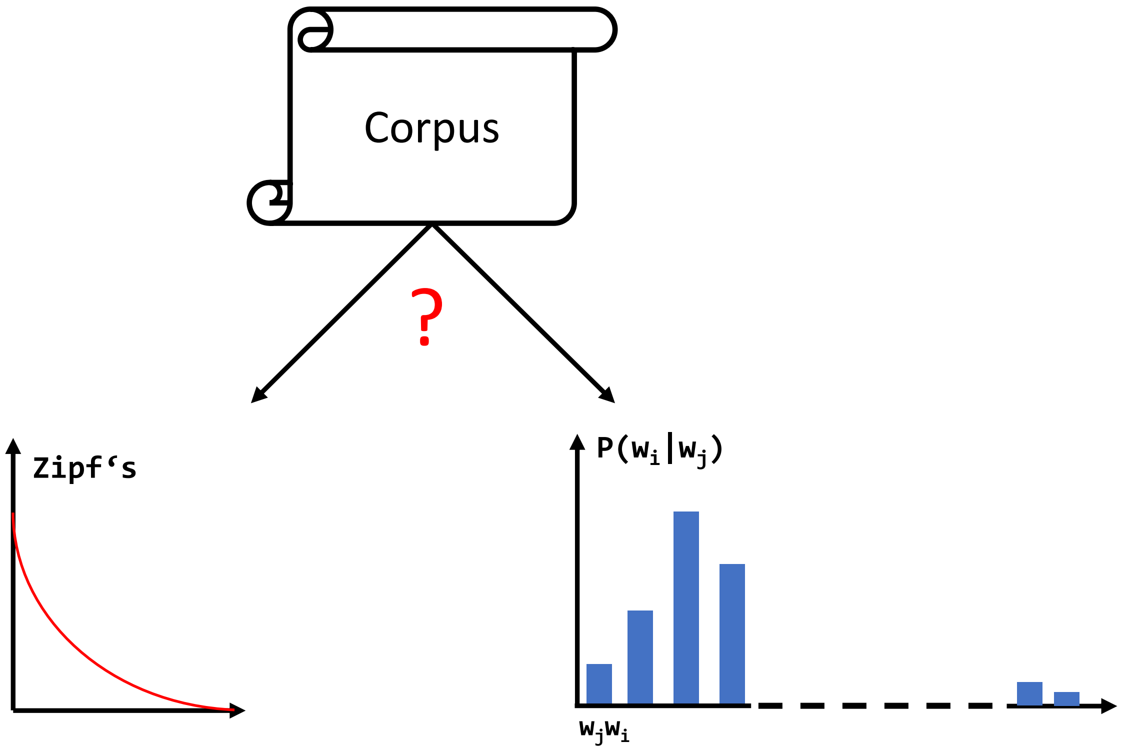
Overview:
Linguistics
Regelbasierte Textgenerierung
Translation (G2P)

Language Model



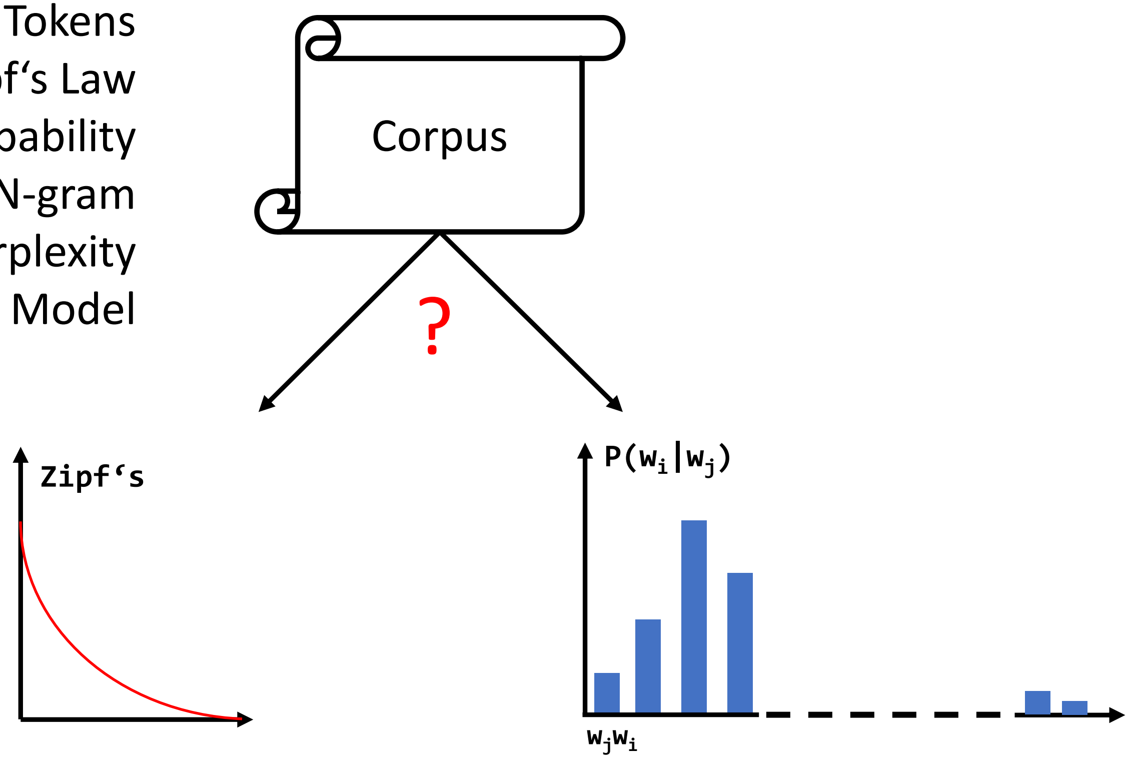
?

Language Model



Language Model

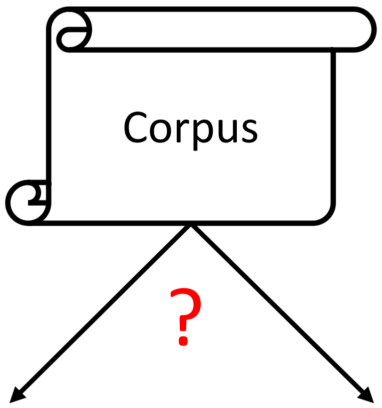
Types vs. Tokens
Zipf‘s Law
Sentence probability
Derivation N-gram
Derivation Perplexity
Shannon–Weaver Model



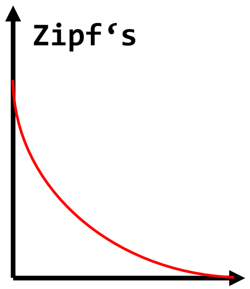
Technische Hochschule Ingolstadt

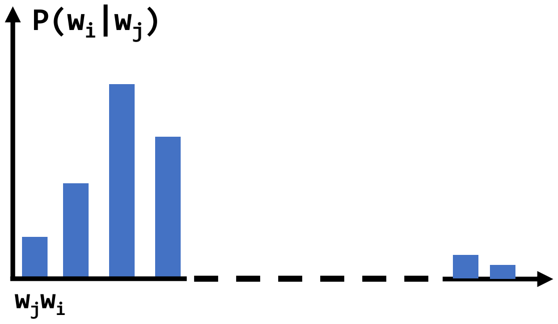
Language Model

Types vs. Tokens
Zipf‘s Law
Sentence probability
Derivation N-gram
Derivation Perplexity
Shannon–Weaver Model



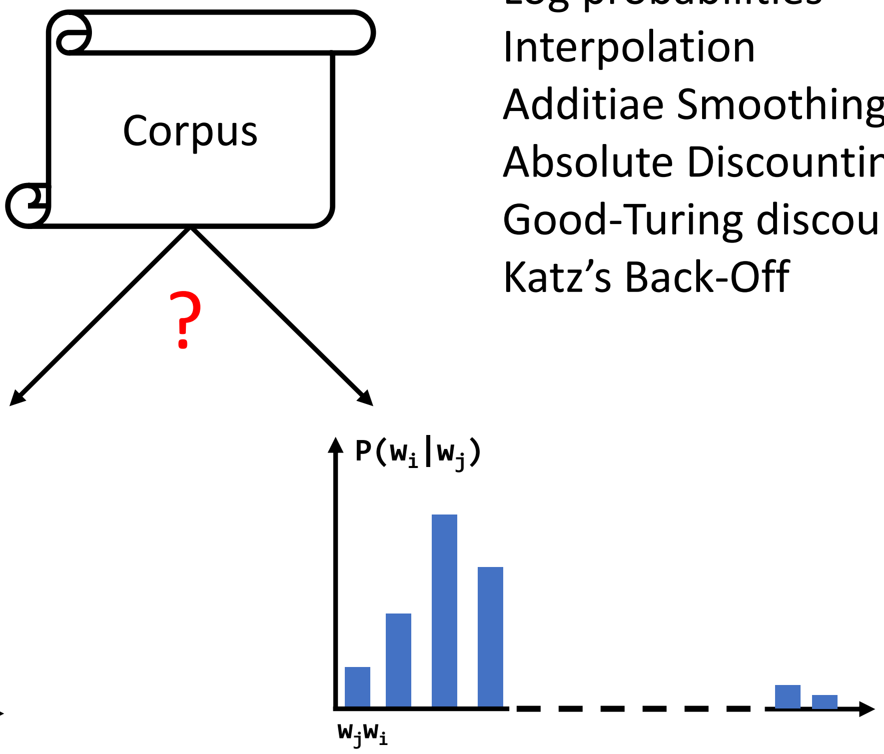
Log probabilities
Interpolation
Additive Smoothing
Absolute Discounting
Good-Turing discounting
Katz’s Back-Off

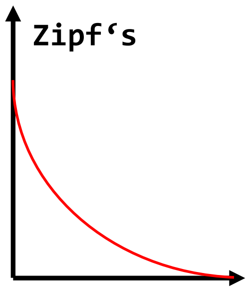




Language Model

Types vs. Tokens
Zipf's Law
Sentence probability
Derivation N-gram
Derivation Perplexity
Shannon–Weaver Model



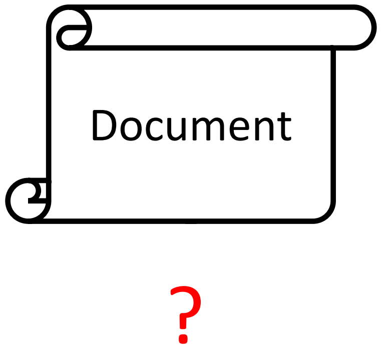


Log probabilities
Interpolation
Additiae Smoothing
Absolute Discounting
Good-Turing discounting
Katz's Back-Off

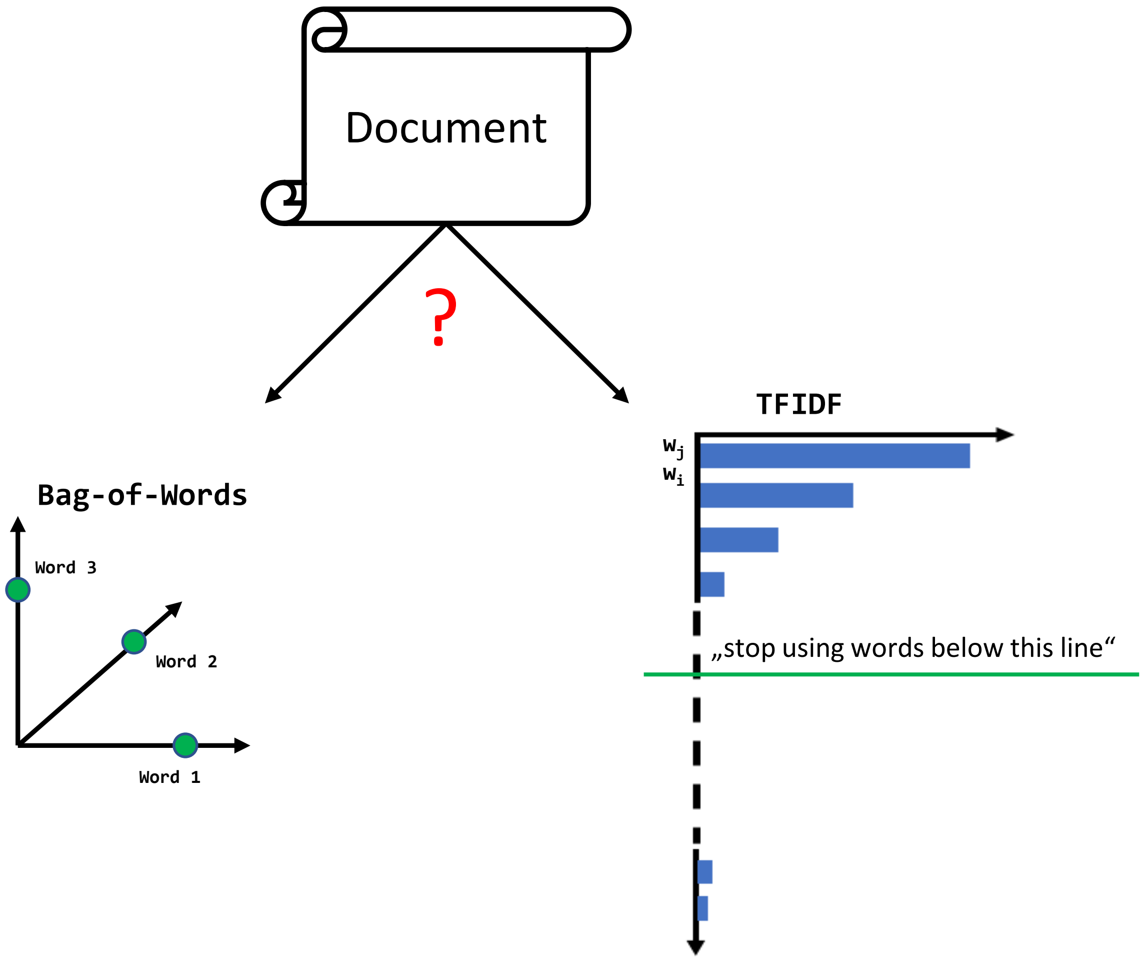
Overview:

- Kneser-Ney smoothing
- Klassenbasierte Sprachmodelle
- Language Model Adaptation
- N-gram generalization

Features

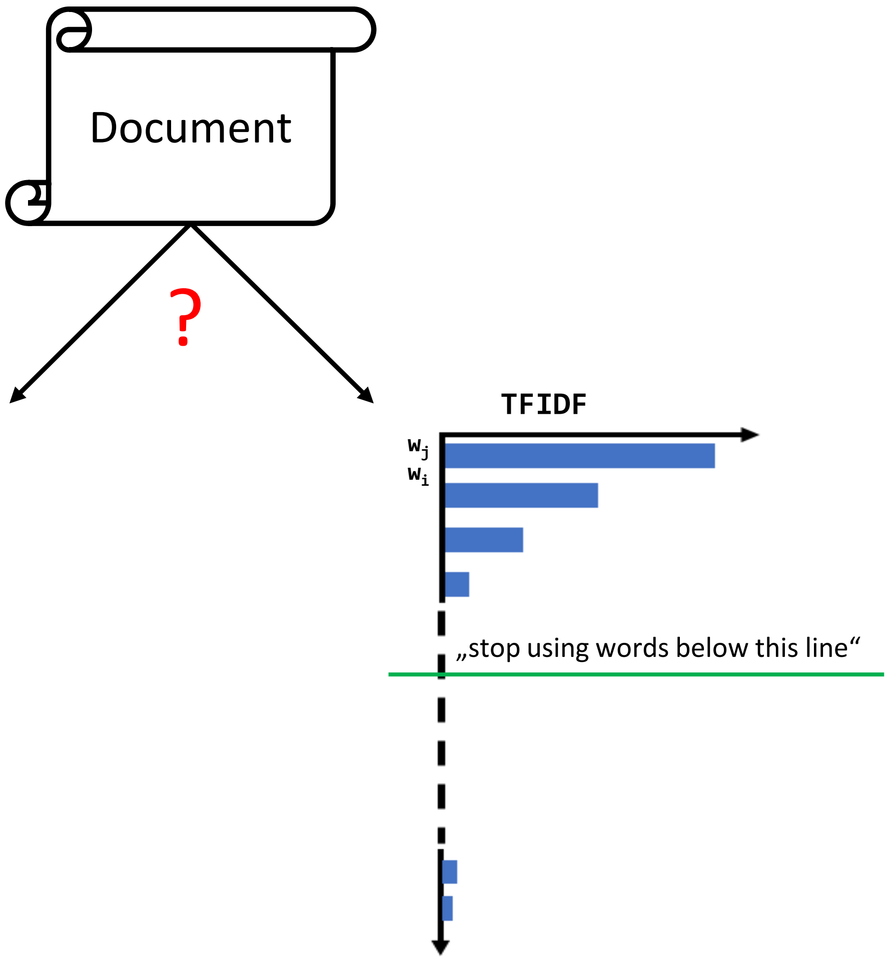


Features


Document
?
Bag-of-Words
Word 3
Word 2
Word 1
TFIDF
w_j
w_i
„stop using words below this line“

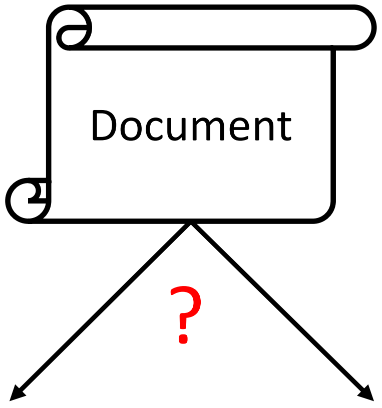
Features

"Words"
Dependent/Independent Words
Words and probabilities
Words as vectors

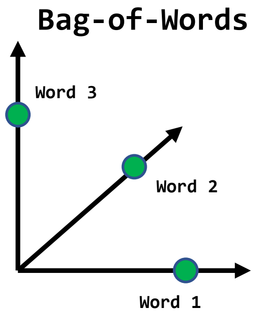


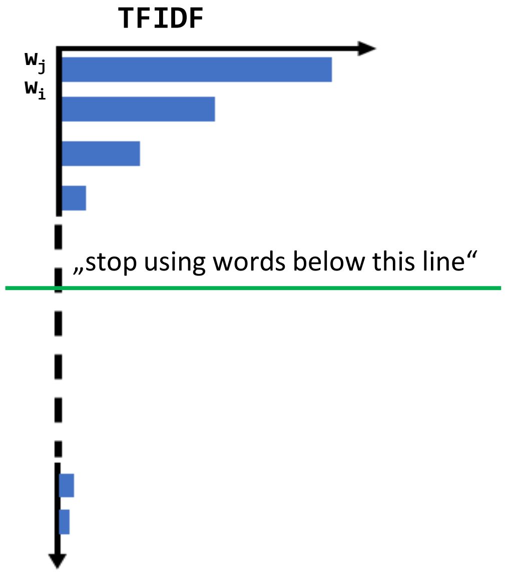
Features

"Words"
Dependent/Independent Words
Words and probabilities
Words as vectors

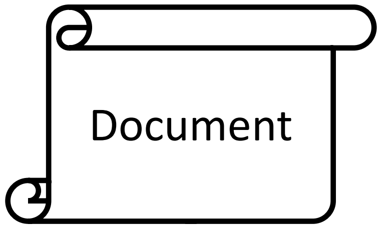


n-gram
PMI
BOW
TF
IDF





Classification



?

Classification

Document

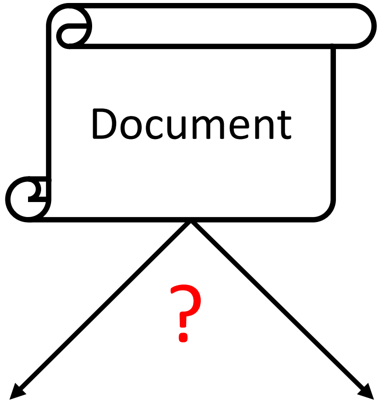
?

Ich spreche Deutsch

나는 한국어를한다

Classification

Spam/no-Spam
Language Identification
Documents Classification

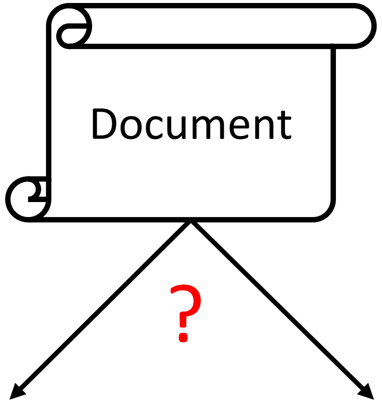


Ich spreche Deutsch

나는 한국어를한다

Classification

Spam/no-Spam
Language Identification
Documents Classification

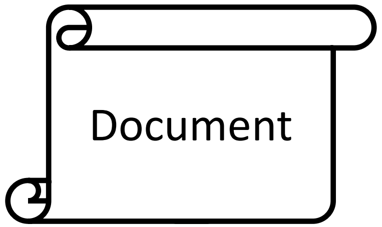


Introduction Evaluation
Naive Bayes
KNN
Distances:
- Mean Square Error
- Cosine Similarity

Ich spreche Deutsch

나는 한국어를한다

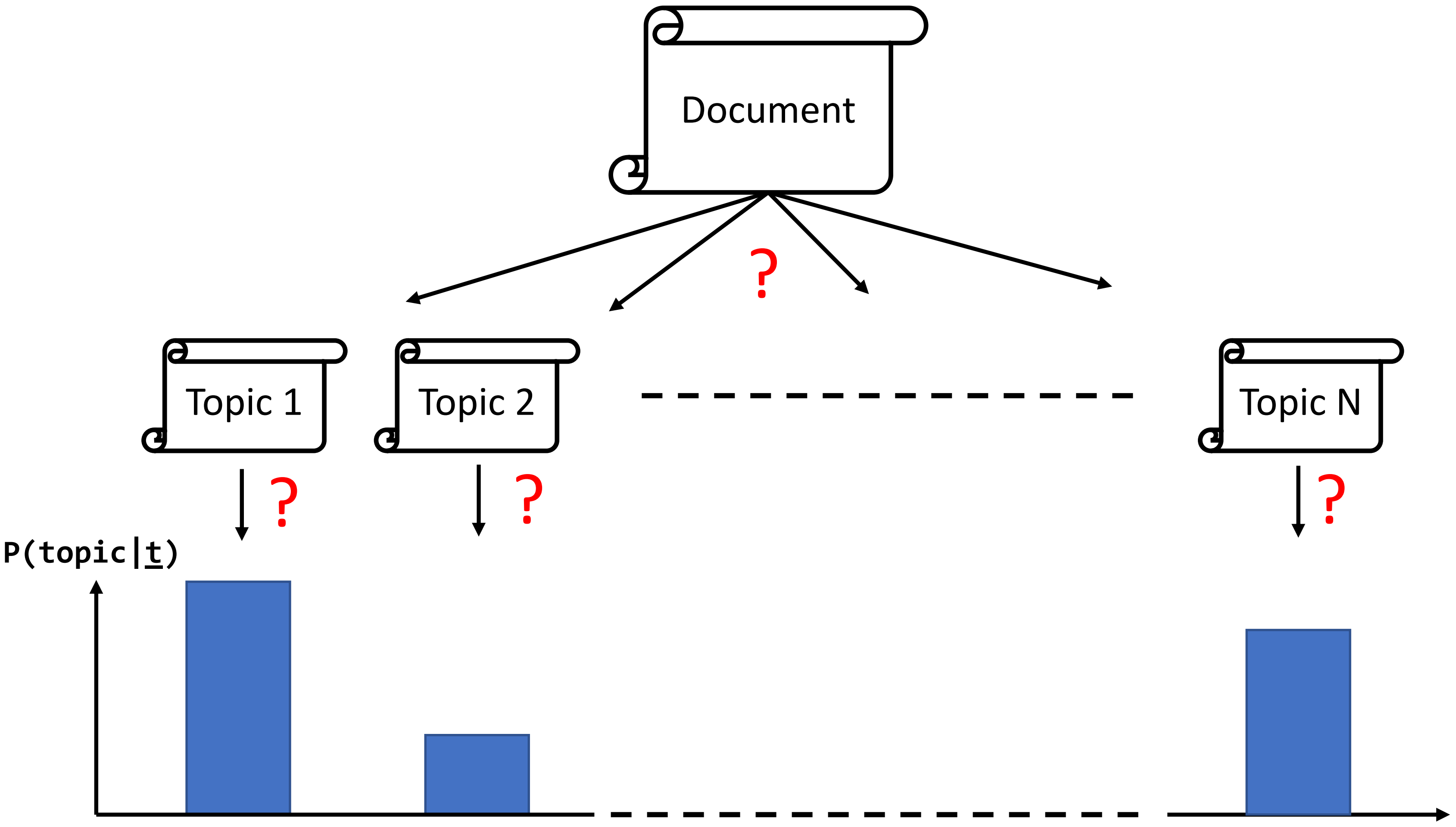
Clustering



?

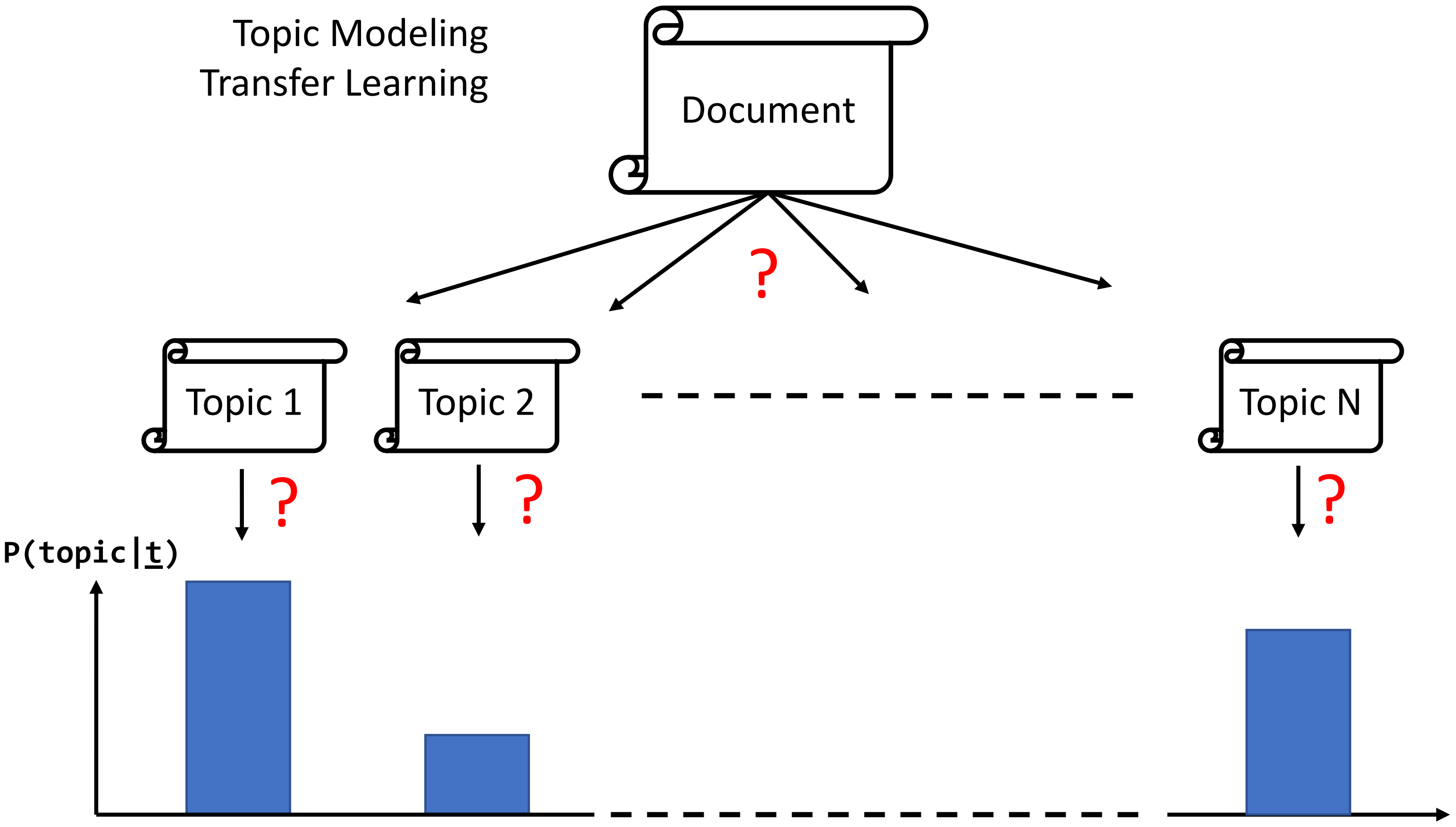
Technische Hochschule Ingolstadt

Clustering



Clustering

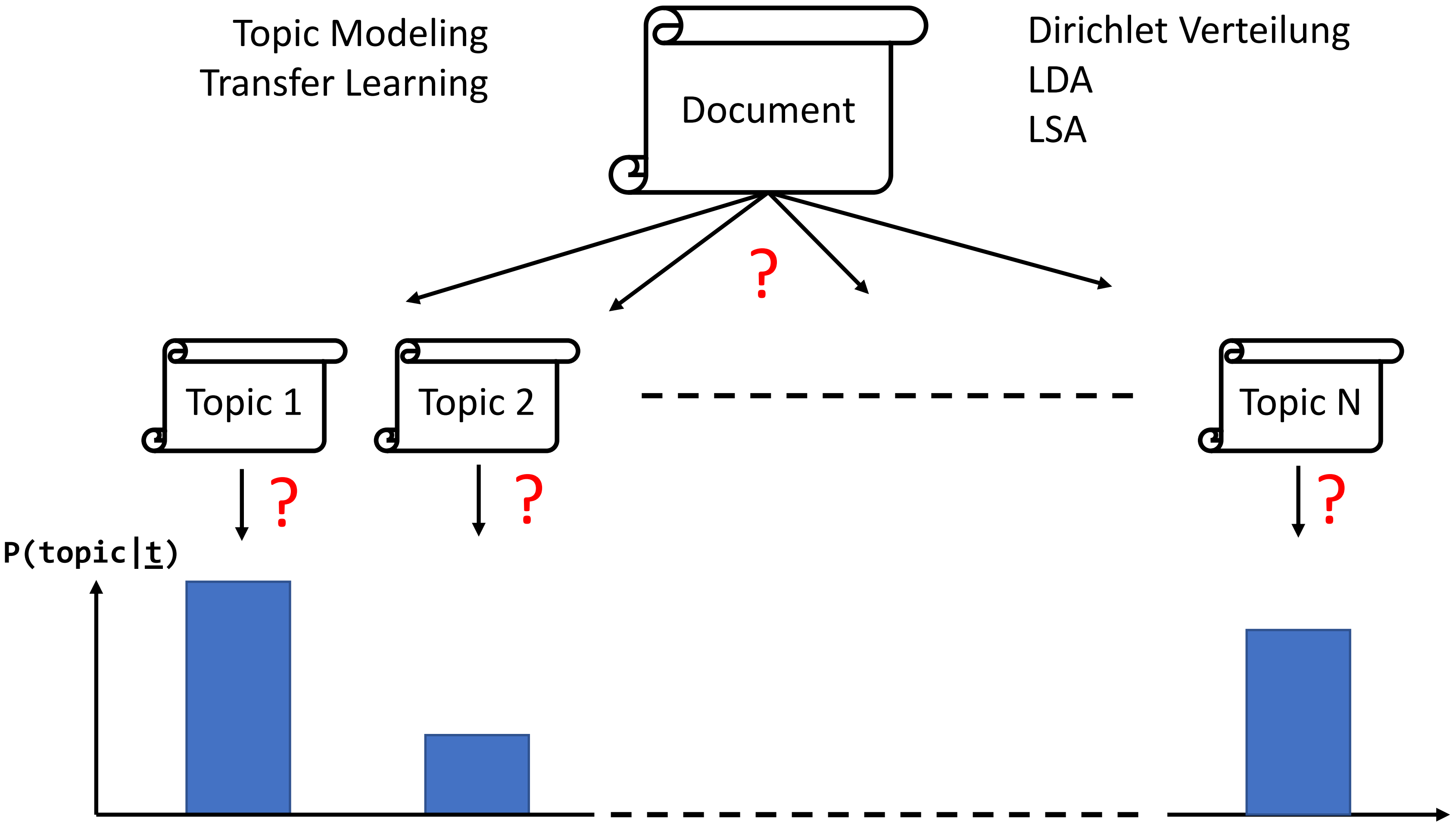
Topic Modeling
Transfer Learning



Technische Hochschule Ingolstadt

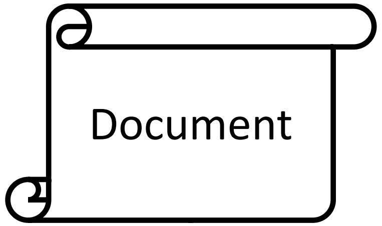
Clustering

Topic Modeling
Transfer Learning



Dirichlet Verteilung
LDA
LSA

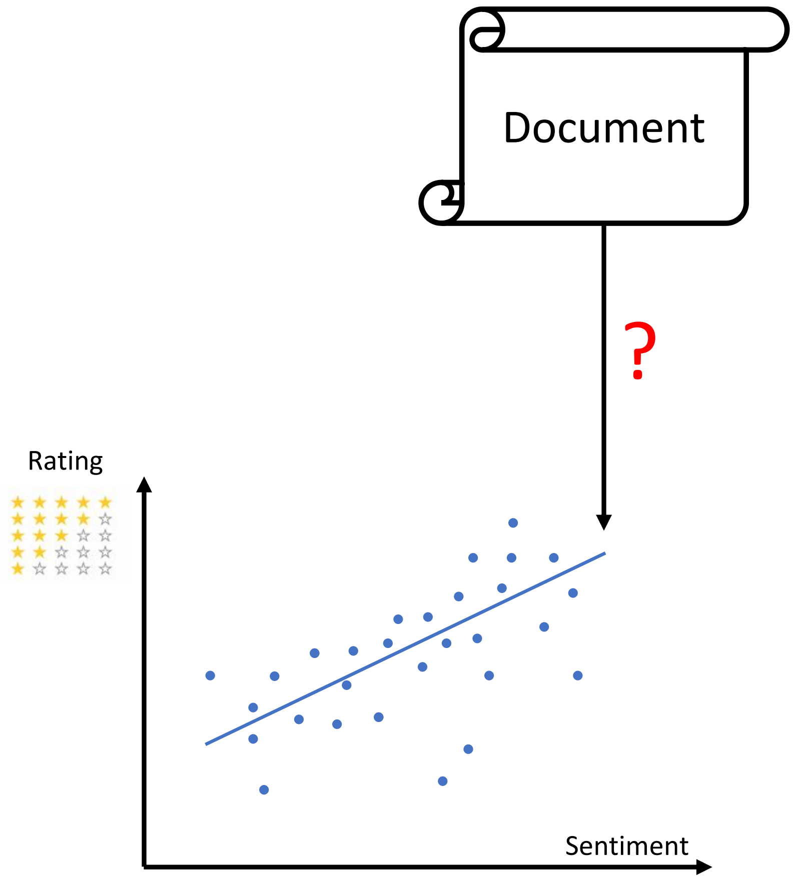
Regression



?

Technische Hochschule Ingolstadt

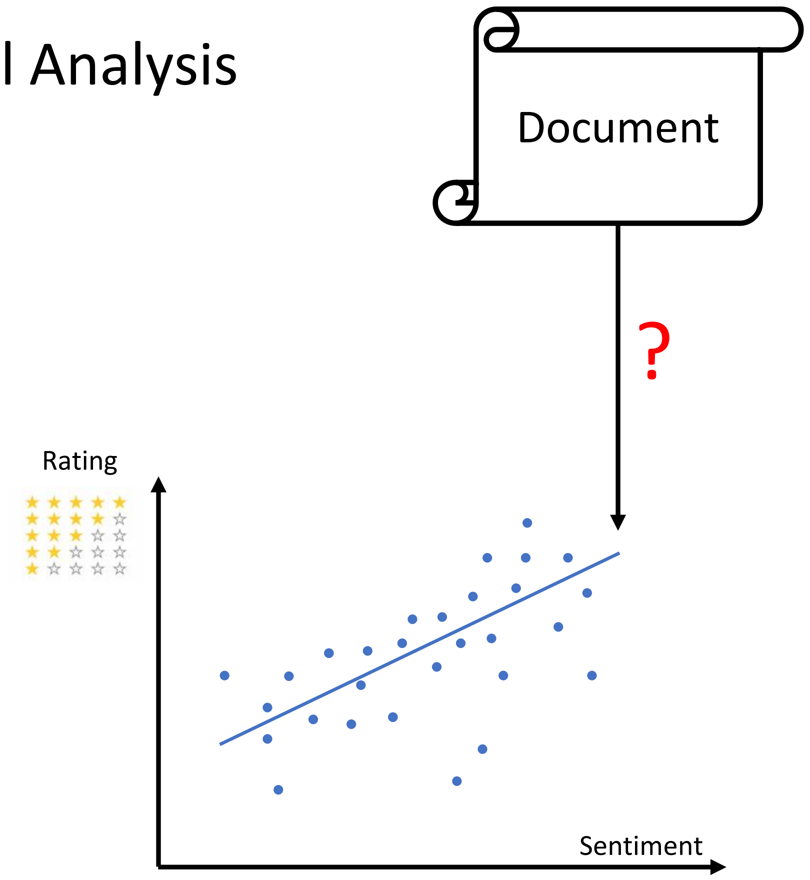
Regression



https://data-science-blog.com/blog/2017/12/20/maschinelles-lernen-klassifikation-vs-regression/

Regression

Sentimental Analysis



https://data-science-blog.com/blog/2017/12/20/maschinelles-lernen-klassifikation-vs-regression/