CO-OCCURRENCES / EMBEDDINGS

DISTRIBUTIONAL SEMANTICS – REMINDER I

Distributional semantics:

- Zellig Harris (1951): Words used in the same/similar linguistic context have similar meaning
- Alternative: „definiert man die Verteilung sprachlicher Elemente als die Summe alle Kontexte, in denen das jeweilige Element auftritt, dann können Elemente als distributionell äquivalent angesehen werden, wenn ihre Distribution gleich ist.“ (Biemann et al. 2022)

- J.R. Firth (1957): *You shall know a word by the company it keeps*

DISTRIBUTIONAL SEMANTICS – REMINDER II

- terms co-occurring with a term are its semantic features
- Calculation of significant co-occurrences
- Many significance measures, like:
  - Cosine Similarity
  - Dice Coefficient
  - Pointwise Mutual Information
  - ...

VECTOR SPACE MODEL

- Representation as points in (high-dimensional) vector space
  → feature vector
- Proximity in vector space as the degree of similarity of the entities

VECTOR SPACE MODEL – DOCUMENT-TERM-MATRIX

- Excursus Information Retrieval: word frequency in documents
- Document features: words / word frequency
  → document-term matrix
- Often: weights based on tf.idf (or similar)

VECTOR SPACE MODEL – DOCUMENT-TERM-MATRIX

Related information items

Topic	Computer	1		3			6
	Information	1	2		4		6
	Retrieval	1	2		4	5	
	Systems	1		3	4	5	6
	Users		2			5	

(a)

Item number	1	2	3	4	5	6
Author	Ash	Brown	Jones	Reynolds	Smith	David
Title	Aspects of Computerized Information Retrieval Systems	A Survey of Users of Information Retrieval	The History of Computer Systems	The State of the Art of Information Retrieval Systems	Users of New Retrieval Systems	A Study of Computerized Information Systems
Topic	Computer Information Retrieval Systems	Information Retrieval Users	Computer Systems	Information Retrieval Systems	Retrieval Systems Users	Computer Information System

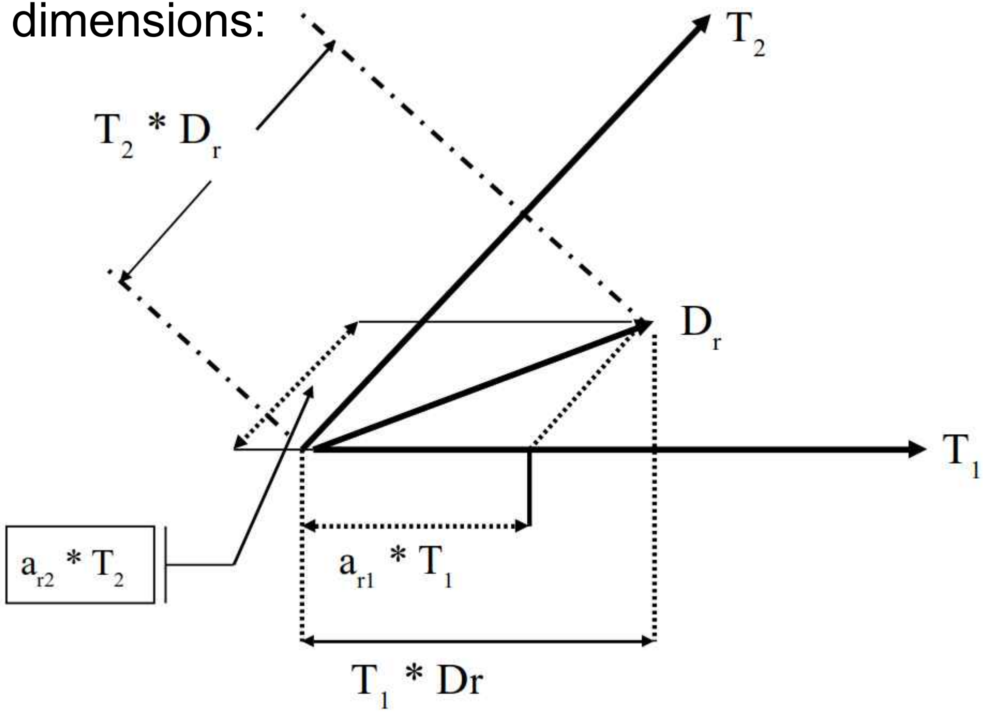
(b)

Figure 1-11 Inverted file with added item. (a) Inverted index with a new item 6. (b) Sample information items with added item 6.

Gerard Salton & Michael J. McGill (1983): Introduction to Modern Information Retrieval. McGraw-Hill College.

VECTOR SPACE MODEL – DOCUMENT-TERM-MATRIX

- Example for 2 dimensions:



VECTOR SPACE MODEL – DOCUMENT-TERM-MATRIX

- Absolute frequency:

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

Daniel Jurafsky & James H. Martin: **Speech and Language Processing**, Pearson Prentice Hall, 2024. (Draft: https://web.stanford.edu/~jurafsky/slp3/)

VECTOR SPACE MODEL – DOCUMENT-TERM-MATRIX

- Weighted via tf.idf

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	0.074	0	0.22	0.28
good	0	0	0	0
fool	0.019	0.021	0.0036	0.0083
wit	0.049	0.044	0.018	0.022

Daniel Jurafsky & James H. Martin: **Speech and Language Processing**, Pearson Prentice Hall, 2024. (Draft: https://web.stanford.edu/~jurafsky/slp3/)

VECTOR SPACE MODEL – DOCUMENT-TERM-MATRIX

- Documents as term vectors → document vector

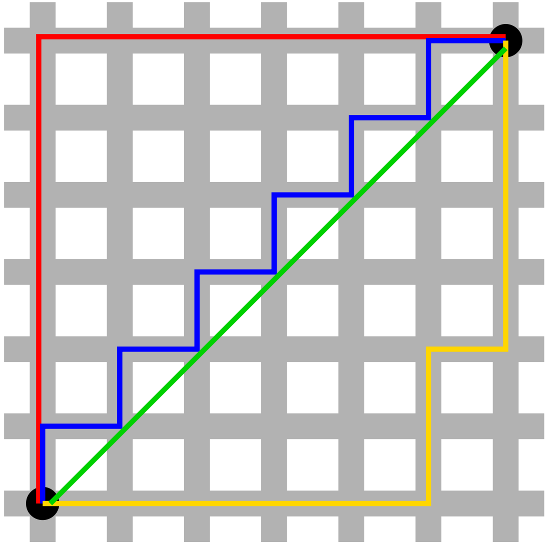
	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	0.074	0	0.22	0.28
good	0	0	0	0
fool	0.019	0.021	0.0036	0.0083
wit	0.049	0.044	0.018	0.022

- *As you like it*: (0.074, 0, 0.019, 0,049) or (1, 114, 36, 20)
- *Twelfth Night*: (0, 0, 0.021, 0.044) or (0, 80, 58, 15)
- *Julius Caesar*: (0.22, 0, 0.0036, 0.018) or (7, 62, 1, 2)
- *Henry V*: (0.28, 0, 0.0083, 0.022) or (13, 89, 4, 3)

VECTOR SPACE MODEL – MANHATTAN DISTANCE

- City Block distance / Manhattan distance s

$$s = \sum_{i=1}^{n} \left| X_i - Y_i \right|$$



Wikimedia Commons; User:Psychonaut

VECTOR SPACE MODEL – DOT PRODUCT

- Dot product

$$\sum_{i=1}^{n} X_i * Y_i$$

- Disadvantage: longer vectors → higher similarity
  → favors frequent terms (~stop words)

VECTOR SPACE MODEL – COSINE SIMILARITY

- Cosine Similarity
- Angle between two vectors

$$\cos(X,Y)=\frac{\sum_{i=1}^{n} X_i * Y_i}{\sqrt{\sum_{i=1}^{n} X_i^2}\sqrt{\sum_{i=1}^{n} Y_i^2}}$$

- based on the scalar product of two vectors X and Y
- Values: [-1, 1] or [0,1]
  - Orthogonality: 0

VECTOR SPACE MODEL – TERM-TERM-MATRIX

- Back to words → term-term-matrix / word-word-matrix
- Instead of a document term matrix, we use a term term matrix (based on the set of significant terms of a text collection and their co-occurrences)
- For vocabulary $t_1 - t_n$ (not necessarily symmetrical):

	t_1	t_2	…	t_n
t_1	-	a_{12}	…	a_{1n}
t_2	a_{21}	-	…	a_{2n}
…			…	
t_n	a_{n1}	a_{n2}	…	-

REMINDER: DISTRIBUTIONAL HYPOTHESIS

- (Harris: Words that are used in the same linguistic contexts have a similar meaning)
- Context of a word: (global) co-occurrences of a word in the corpus

  → Description by word vector in term term matrix

VECTOR SPACE MODEL – EXAMPLE I – MANHATTAN DISTANCE

	blue	boy	girl	ocean	red	white
blue	-	2	1	4	43	37
boy	2	-	47	0	1	13
girl	1	47	-	1	3	37
ocean	4	0	1	-	2	0
red	43	1	3	2	-	23
white	37	13	37	0	23	-

VECTOR SPACE MODEL – EXAMPLE I – MANHATTAN DISTANCE

- Manhattan distance:

	blue	boy	girl	ocean	red	white	
blue	-	2	1	4	43	37	
red	43	1	3	2	-	23	
Diff.	-	1	2	2	-	14	=>19

VECTOR SPACE MODEL – EXAMPLE I – MANHATTAN DISTANCE

- Difference:

	blue	boy	girl	ocean	red	white	
blue	-	2	1	4	43	37	
red	43	1	3	2	-	23	=> 19
boy	2	-	47	0	1	13	
red	43	1	3	2	-	23	=> 97
girl	1	47	-	1	3	37	
red	43	1	3	2	-	23	=> 103
ocean	4	0	1	-	2	0	
red	43	1	3	2	-	23	=> 65
red	43	1	3	2	-	23	
red	43	1	3	2	-	23	=> 0
white	37	13	37	0	23	-	
red	43	1	3	2	-	23	=> 54

VECTOR SPACE MODEL – EXAMPLE I – MANHATTAN DISTANCE

- Results:

girl	blue	foot	baby	bread	butter
boy	red	leg	child	meat	cheese
man	green	hand	mother	cake	bread
woman	grey	head	girl	cheese	sugar
mother	yellow	back	boy	milk	chocolate
child	white	side	father	toast	milk

VECTOR SPACE MODEL – EXAMPLE II – COSINE

- Example animals vs. IT vs. cake in Wikipedia:

	aardvark	...	computer	data	result	pie	sugar	...
cherry	0	...	2	8	9	442	25	...
strawberry	0	...	0	0	1	60	19	...
digital	0	...	1670	1683	85	5	4	...
information	0	...	3325	3982	378	5	13	...

Daniel Jurafsky & James H. Martin: **Speech and Language Processing**, Pearson Prentice Hall, 2024. (Draft: https://web.stanford.edu/~jurafsky/slp3/)

VECTOR SPACE MODEL – EXAMPLE II – COSINE

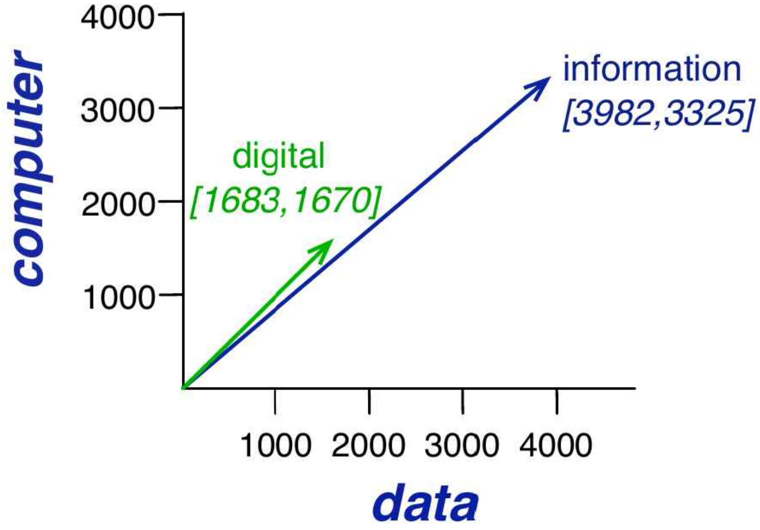
- Example „*digital*“:

	aardvark	...	computer	data	result	pie	sugar	...
cherry	0	...	2	8	9	442	25	...
strawberry	0	...	0	0	1	60	19	...
digital	0	...	1670	1683	85	5	4	...
information	0	...	3325	3982	378	5	13	...

Daniel Jurafsky & James H. Martin: **Speech and Language Processing**, Pearson Prentice Hall, 2024. (Draft: https://web.stanford.edu/~jurafsky/slp3/)

VECTOR SPACE MODEL – EXAMPLE II – COSINE

- Example *digital* vs. *information*:



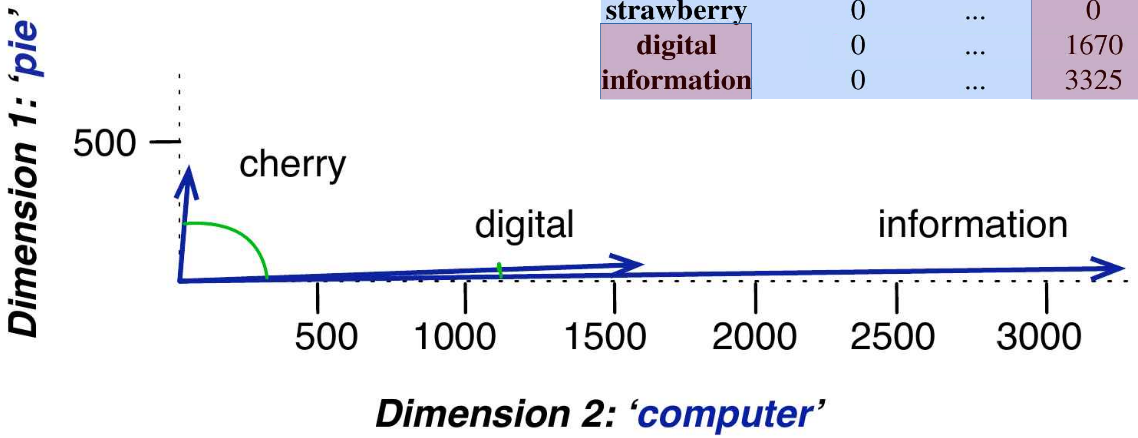
	aardvark	...	computer	data	result	pie	sugar	...
cherry	0	...	2	8	9	442	25	...
strawberry	0	...	0	0	1	60	19	...
digital	0	...	1670	1683	85	5	4	...
information	0	...	3325	3982	378	5	13	...

Daniel Jurafsky & James H. Martin: **Speech and Language Processing**, Pearson Prentice Hall, 2024. (Draft: https://web.stanford.edu/~jurafsky/slp3/)

VECTOR SPACE MODEL – EXAMPLE II – COSINE

- Cosine *digital* vs. *information* vs. *cherry*:

	aardvark	...	computer	data	result	pie	sugar	...
cherry	0	...	2	8	9	442	25	...
strawberry	0	...	0	0	1	60	19	...
digital	0	...	1670	1683	85	5	4	...
information	0	...	3325	3982	378	5	13	...



Daniel Jurafsky & James H. Martin: **Speech and Language Processing**, Pearson Prentice Hall, 2024. (Draft: https://web.stanford.edu/~jurafsky/slp3/)

VECTOR SPACE MODEL – EXAMPLE II – COSINE

- Cosine *digital* vs. *information* vs. *cherry*:

	pie	data	computer
cherry	442	8	2
digital	5	1683	1670
information	5	3982	3325

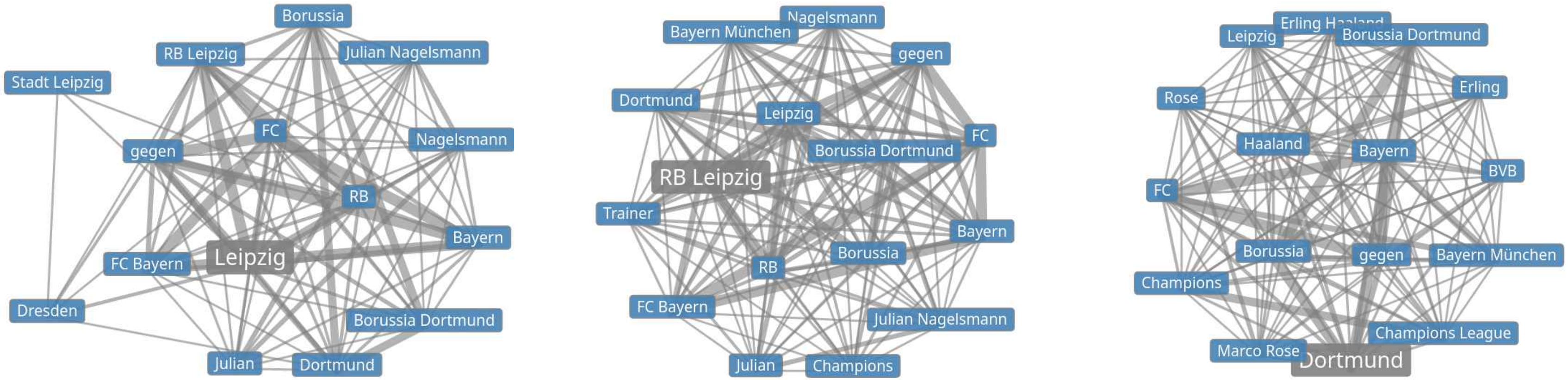
$$\cos(cherry, information) = \frac{442*5+8*3982+2*3325}{\sqrt{442^2+8^2+2^2}*\sqrt{5^2+3982^2+3325^2}} = 0{,}18$$

$$\cos(digital, information) = \frac{5*5+1683*3982+1670*3325}{\sqrt{5^2+1683^2+1670^2}*\sqrt{5^2+3982^2+3325^2}} = 0{,}996$$

Daniel Jurafsky & James H. Martin: **Speech and Language Processing**, Pearson Prentice Hall, 2024. (Draft: https://web.stanford.edu/~jurafsky/slp3/)

VECTOR SPACE MODEL – EXAMPLE III – DICE/COSINE

- Strongest word co-occurrences at *Wortschatz Leipzig*:



https://corpora.wortschatz-leipzig.de/de?corpusId=deu_news_2021

VECTOR SPACE MODEL – EXAMPLE III – DICE/COSINE

- Most similar words for *Leipzig*

Formen mit ähnlichem Satzkontext

RB Leipzig (0,50), Dortmund (0,41), Wolfsburg (0,36), Freiburg (0,36), RB (0,36), Leverkusen (0,36), Mönchengladbach (0,36), Augsburg (0,36), Hoffenheim (0,33), Eintracht Frankfurt (0,32), Köln (0,32), Bielefeld (0,32), Borussia Dortmund (0,32), Bayer Leverkusen (0,31), Mainz (0,31), Stuttgart (0,31), Union Berlin (0,30), Borussia Mönchengladbach (0,30), Bayern München (0,30), München (0,30), Bremen (0,30), Kiel (0,29), FC Köln (0,29), Frankfurt (0,29), VfL Wolfsburg (0,28), Hertha BSC (0,28), Bochum (0,28), Fürth (0,28), BVB (0,28), 1. FC Köln (0,27), Gladbach (0,27), FC Bayern München (0,26), Schalke 04 (0,26), Borussia (0,26), Bayern (0,25), Hertha (0,24), Schalke (0,24), Düsseldorf (0,23)

- Similarity measure: Cosine similarity / Dice coefficient using strongest
  - Sentence co-occurrences
  - Neighbourhood (Context window size n=1)
- To reduce complexity: only the 1000 strongest co-occurrences per word

https://corpora.wortschatz-leipzig.de/de/res?corpusId=deu_news_2021&word=Leipzig

VECTOR SPACE MODEL – EXAMPLE III – DICE/COSINE

- *Berlin*

  Hamburg (0,26), München (0,26), Köln (0,24), Düsseldorf (0,22), Dresden (0,21), Bremen (0,20), Stuttgart (0,20), Wien (0,20), Frankfurt (0,20), Dortmund (0,19), Leipzig (0,19), Hannover (0,19), Nürnberg (0,18), Deutschland (0,18), Augsburg (0,17), Nordrhein-Westfalen (0,17), London (0,16), Bonn (0,16), Rom (0,16), Brandenburg (0,16), Potsdam (0,16), Salzburg (0,16), Münster (0,16), Bayern (0,16), Baden-Württemberg (0,15), Mecklenburg-Vorpommern (0,15), Rheinland-Pfalz (0,15), Paris (0,14), Sachsen (0,14), Niedersachsen (0,14), Zürich (0,14)

- *Montag*

  Donnerstag (0,70), Mittwoch (0,70), Dienstag (0,69), Freitag (0,69), Samstag (0,51), Sonntag (0,51)

- *Bundestag*

  Landtag (0,39), Parlament (0,35), Abgeordnetenhaus (0,31), Kabinett (0,24), Bundesrat (0,23), Bundestagswahl (0,23), Senat (0,22), Gemeinderat (0,19)

- *Weihnachten*

  Ostern (0,29), Weihnachtsfest (0,24), Heiligabend (0,24), Silvester (0,21), Weihnachtszeit (0,20), Pfingsten (0,18), Feiertage (0,16), Fest (0,14), Wochenende (0,14), Herbst (0,14), Dezember (0,13), Sommer (0,13), Wochen (0,12), Jahr (0,12)

https://corpora.wortschatz-leipzig.de/de?corpusId=deu_news_2021

PROBLEM 1 – LARGE MATRICES

- Example corpus „deu_news_2021“:
  - 33.3M sentences
  - 5.5M types
  - Co-occurrences matrix T*T → 5.5M * 5.5M = 30,250G
- Example corpus „deu_mixed_2011“
  - 259M sentences
  - 37M types
  - Co-occurrences matrix T*T → 37M * 37M = 1,369,000G

PROBLEM 2 – EMPTY/SPARSE MATRICES

- Example corpus „deu_news_2021“:
  - In theory: 5.5M * 5.5M = 30,250G
- Real values
  - Context: sentence
  - Minimum co-occurrence frequency: 3
  - Minimum significance (Log-likelihood-Ratio): 6.63

    → 88.2 M
- Why? Zipf’s law...
  - ~50% of all types with frequency 1 → Consequence for term term matrix / co-occurrence matrix

https://corpora.wortschatz-leipzig.de/de?corpusId=deu_news_2021

PROBLEM 2 – EMPTY/SPARSE MATRICES

- Example:
  - *5-km-Loipe* (one sentence: „Lisa Hirner, die nach 82,5 Metern im Sprungteil auf Rang sechs gelegen war, verbesserte sich auf der 5-km-Loipe auf Platz vier.“)

    → max ~20 co-occurrences in the corpus (**before** filters or significance)

    → term vectors mostly 0 (5.5M types – 20)
- Example:
  - Stop words (like *der*)

    → Absolute frequency: 14,9M in 11,5 sentences

    → 206,543 co-cooccurences (**after** filter or significance)

    → term vector still mostly empty (5.5M Types – 206,543)

https://corpora.wortschatz-leipzig.de/de/res?corpusId=deu_news_2021&word=5-km-Loipe

PROBLEM 2 – EMPTY/SPARSE MATRICES

- Possible solution:
  - Not all words useful features for term vectors
  - Instead: most frequent n types
  - like 10K – 50K most frequent types

PROBLEM 3 – SIMILARITY IN SPARSE MATRICES

- Similarity only if some match between vectors
  - Assumption: no connection between features
  - Ignores semantic similarity between feature words (synonyms, etc.)
  - Also: morphology / tokenization
- e.g.

$$
\begin{array}{r c}
 & \begin{array}{cccc} \text{Italien} & \text{Deutschland} & \text{Violinist} & \text{Pianist} \end{array} \\
\begin{array}{r} W_1 = \text{Beethoven} \\ W_2 = \text{Paganini} \end{array} & \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}
\end{array}
$$

Source example: Chris Biemann, Gerhard Heyer & Uwe Quasthoff: Wissensrohstoff Text: Eine Einführung in das Text Mining, Springer Vieweg, 2022.

WORD EMBEDDINGS

- General: real-valued vector that represents the distributional semantics of a particular word in the embedding space.
  - Cooccurrence matrix as "sparse embeddings"
- More specific: massive reduction in dimensions (e.g. <1000)
  - „Dense Vector Embeddings“
- Various variants
  - Static word embeddings (Types)
  - Contextual word embeddings (Tokens)
  - Document Embeddings (Documents)

WORD EMBEDDINGS – WORD2VEC

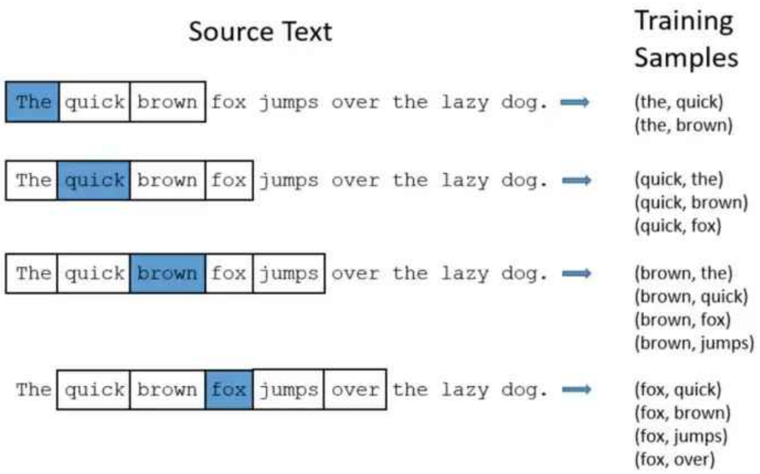
- Contexts in which words are used are learned with the help of a neural network
- Continuous Bag-of-Words architecture (CBOW)
  - uses co-occurrences to predict possible target words
- Skip-Gram architecture
  - uses the vector representation to predict co-occurrences

WORD EMBEDDINGS – SKIP-GRAMS I

- Basic idea: find likely context words for a word
- Approach:
  - Iteration over all tokens t_i in training corpus
  - Neighborhood of t_i as context (e.g. n=2 $\rightarrow$ t_{i-2}, t_{i-1}, t_{i+1}, t_{i+2})
  - Positive samples: context words found
  - Negative samples: randomly selected other words from the corpus

WORD EMBEDDINGS – SKIP-GRAMS II

- e.g. context size n=2



- Probabilistic classifier: $P(+|t_i, t_j)$ → Probability that t_j in context of t_i

Source: https://medium.com/@Aj.Cheng/word2vec-3b2cc79d674

WORD EMBEDDINGS – SKIP-GRAMS III

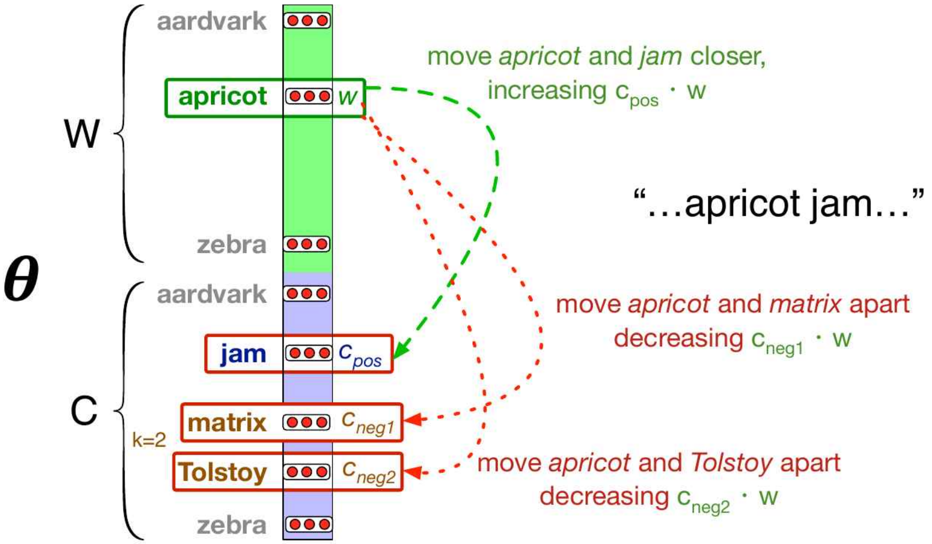
- 1. Random initialization of embeddings for all words
- 2. Incremental adjustment for the training data
    - for positive data (=word pair found in the corpus)

      → Increase the similarity of their embeddings
    - for negative data (=word pair not found in the corpus)

      → Decrease the similarity of their embeddings
- Similarity? Scalar product!
- Derivation via gradient descent: Daniel Jurafsky & James H. Martin: *Speech and Language Processing*, Pearson Prentice Hall, 2024.

WORD EMBEDDINGS – SKIP-GRAMS IV

- Example: „*a tablespoon of* apricot *jam*"
- Positive:
  - (*apricot*, *jam*)
- Negative:
  - (*apricot*, *matrix*)
  - (*apricot*, *Tolstoy*)



Source: Daniel Jurafsky & James H. Martin: *Speech and Language Processing*, Pearson Prentice Hall, 2024.

WORD EMBEDDINGS – LIBRARIES / DATA

- Many libraries / precomputed embeddings, e.g.
  - word2vec
  - FastText (embeddings for ~160 languages)
  - GloVe
  - BERT, Elmo, Flair (contextual embeddings)

WORD EMBEDDINGS – EXAMPLE I – WORTSCHATZ

angela	**sagte**	**montag**	**juni**
kanzlerin	erklärte	dienstag	september
videopodcast	betonte	mittwoch	mai
bundeskanzlerin	ergänzte	donnerstag	märz
frau	meinte	freitag	oktober
kanzleramt	sagt	montagabend	november
regierungserklrung	erläuterte	dienstagabend	april
zuzuschreiben	kommentierte	samstag	februar
staatstragend	versicherte	mittwochabend	august
sportpresse	warnte	donnerstagabend	juli
rentendebatte	unterstrich	freitagabend	dezember
europapolitik	mahnte	sonnabend	januar

Skip-Grams based on deu_news (2018?)

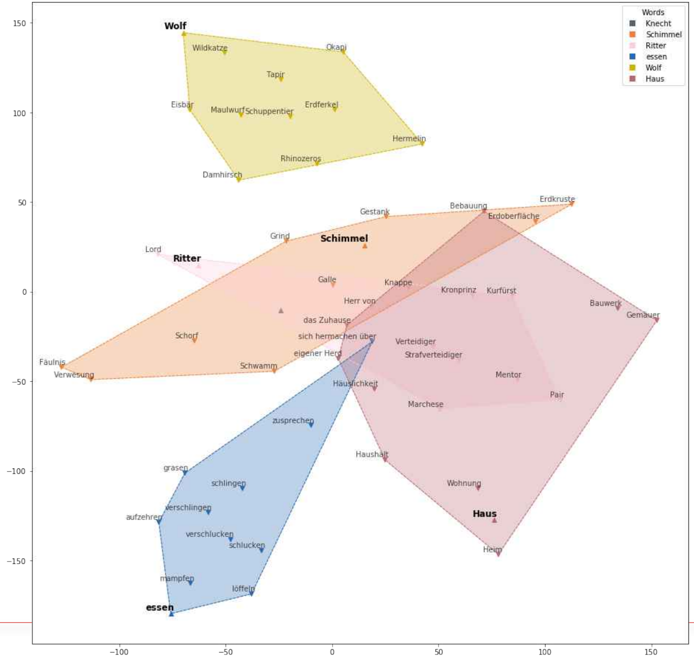
WORD EMBEDDINGS – EXAMPLE II – DORNSEIFF

- *Dornseiff - Der deutsche Wortschatz nach Sachgruppen* (various editions since 1934, Franz Dornseiff)
  - German vocabulary organized in subject groups
  - e.g. *Straße*

3.11 Waagerecht Brett, Eispanzer, Eisschicht, Flachland, Flur, Fußboden, Horizont, Kegelbahn, Plattform, Rost, Sandbank, Schneeschicht, Staubschicht, Straße, Straßenbelag, Talgrund, Terrasse, Wachsschicht, Wasserfilm, Wasserhöhe, Zeitachse
4.33 Verbinden Arm, Brücke, Durchgang, Durchstich, Gasse, Isthmus, Kanal, Landenge, Landverbindung, Meerenge, Seeweg, Straße, Tunnel, Verbindung, Verbindungsweg, Wasserweg, Weg
8.9 Straße Abzweig, Abzweigung, Anfahrt, Anfahrtsweg, Anliegerstraße, Asphaltband, Asphaltstraße, Ausweichstrecke, Autostraße, Außenring, Bundesstraße, Dorfstraße, Durchgangsstraße, Einbahnstraße, Einfallstraße, Entlastungsstraße, Fernstraße, Forstweg, Geschäftsstraße, Hauptverkehrsader, Hauptverkehrsstraße, Hauptweg, Kiesweg, Küstenstraße, Landesstraße,

WORD EMBEDDINGS – EXAMPLE II – DORNSEIFF

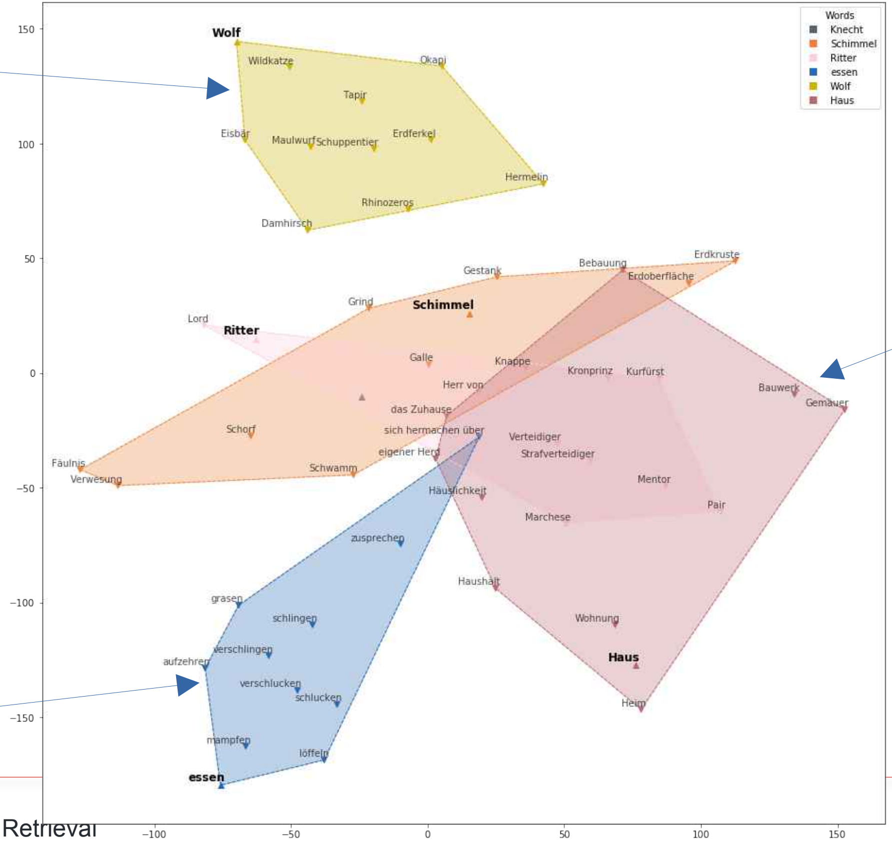
- Expansion of existing subject groups through word similarity based on fastText embeddings (*Dornseiff – Der deutsche Wortschatz nach Sachgruppen*, 9. Auflage, deGruyter, 2020)
- Here:
  - *Haus*
  - *Wolf*
  - *essen*
  - *Schimmel*
  - *Ritter*
  - *Knecht*



Quelle: Erik Körner
Layout: t-SNE (t-distributed stochastic neighbor embedding)

WORD EMBEDDINGS – EXAMPLE II – DORNSEIFF

2.8 Animals



8.1 Location (*Aufenthaltsort*)

16.5 Food, Meals

Quelle: Erik Körner
Layout: t-SNE (t-distributed stochastic neighbor embedding)

OUT-OF-VOCABULARY I

- Problems: No vectors for words that did not appear in the training data
  - e.g. languages with strong morphology (agglutinative languages such as Inuktitut)
  - Typos
- Approach *fasttext*:
  - Embeddings for character n-grams (e.g. 3-grams)
  - OOV word as the sum of the embeddings of all its n-grams
  - e.g. *Autobah**n**n:* {aut, uto, tob, oba, bah, ahn, hnn}

https://fasttext.cc

OUT-OF-VOCABULARY II

Using subword-level information is particularly interesting to build vectors for unknown words. For example, the word *gearshift* does not exist on Wikipedia but we can still query its closest existing words:

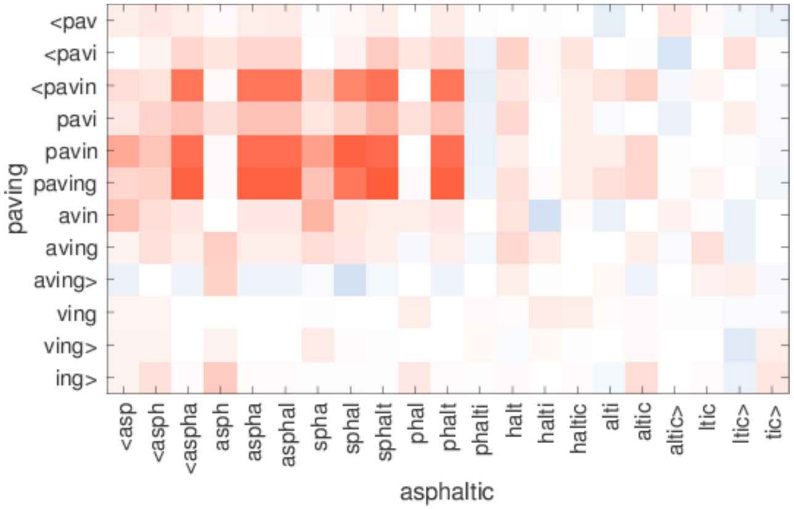
Command line   Python

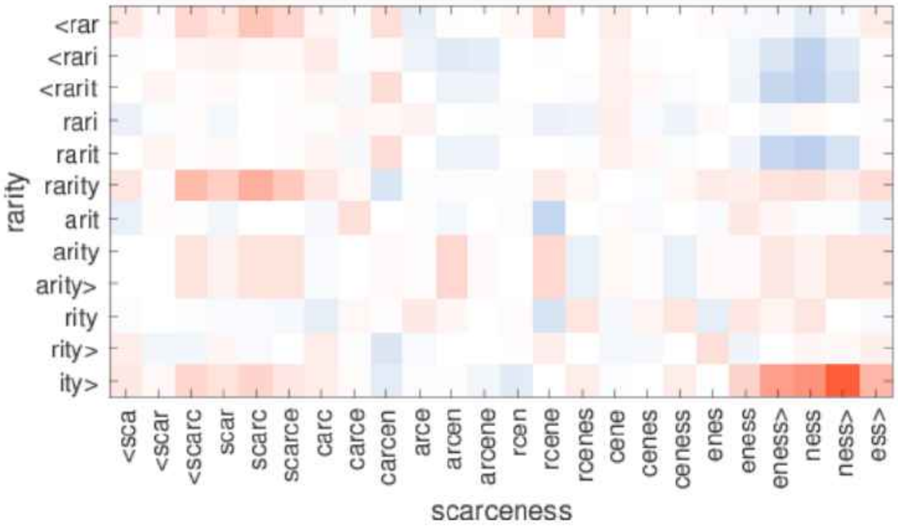
```
Query word? gearshift
gearing 0.790762
flywheels 0.779804
flywheel 0.777859
gears 0.776133
driveshafts 0.756345
driveshaft 0.755679
daisywheel 0.749998
wheelsets 0.748578
```

flywheel: Schwungrad; *driveshaft*: Gelenkwelle

https://fasttext.cc/docs/en/unsupervised-tutorial.html

OUT-OF-VOCABULARY III





Bojanowski et. al. 2017 *Enriching Word Vectors with Subword Information* https://arxiv.org/abs/1607.04606v2