Semantische Ähnlichkeit

Katja Markert, mit einigen Folien von Dan Jurafsky

Institut für Computerlinguistik
Uni Heidelberg
markert@cl.uni-heidelberg.de

January 7, 2020

Bisher und heute

1. Bisher: Lexika und Wortnetze
2. Bisher: Semantische Relationen
3. Jetzt: Wortähnlichkeit
4. Jetzt: ganz kurz mit Lexikonmethoden
5. Jetzt: ausführlicher mit distributionellen Methoden (**keine manuellen Ressourcen nötig**)

Übersicht

Übersicht

Semantische Wortähnlichkeit: die Aufgabe

Gruppiere folgende Worte nach (semantischer) Ähnlichkeit:

Apfel
Banane
Mann
Grapefruit
Frau
Baby
Wassermelone
Kind
Traube

Semantische Wortähnlichkeit: die Aufgabe

Gruppiere folgende Worte nach (semantischer) Ähnlichkeit:

Apfel	Apfel
Banane	Banane
Mann	Grapefruit
Grapefruit	Wassermelone
Frau	Traube
Baby	Mann
Wassermelone	Frau
Kind	Baby
Traube	Kind

Semantische Wortähnlichkeit: die Aufgabe

Gruppiere folgende Worte nach (semantischer) Ähnlichkeit:

Apfel	Apfel
Banane	Banane
Mann	Grapefruit
Grapefruit	Wassermelone
Frau	Traube
Baby	Mann
Wassermelone	Frau
Kind	Baby
Traube	Kind

Tue dies automatisch!

Lexika o. distributionell!

Wortähnlichkeit

- **Synonymie:** Binäre Relation, WordNet synsets
- **Ähnlichkeit:** Graduell, schwächer, nicht direkt in WordNet inkodiert
- **Ähnlichkeit:** eigentlich zwischen senses, nicht Worten
  - $bank_1$ ähnlich zu $fund_3$
  - $bank_2$ ähnlich zu $slope_5$
- Lexikalische Methoden meist auf senses, distributionelle auf Wortebene

Warum Wortähnlichkeit?

NLP Aufgaben:

- Frage-Antwort-Systeme
- Generierung
- Entdeckung von Plagiaten
- Automatische Benotung (essay grading)

Aber auch: Typische Aufgabe in Sprachtests (TOEFL), Modelle des Sprachlernens, historische Linguistik etc.

Beispiel: Plagiatsentdeckung

MAINFRAMES

Mainframes are primarily referred to large computers with rapid, advanced processing capabilities that can execute and perform tasks equivalent to many Personal Computers (PCs) machines networked together. It is characterized with high quantity Random Access Memory (RAM), very large secondary storage devices, and high-speed processors to cater for the needs of the computers under its service.

Consisting of advanced components, mainframes have the capability of running multiple large applications required by many and most enterprises and organizations. This is one of its advantages. Mainframes are also suitable to cater for those applications (programs) or files that are of very high

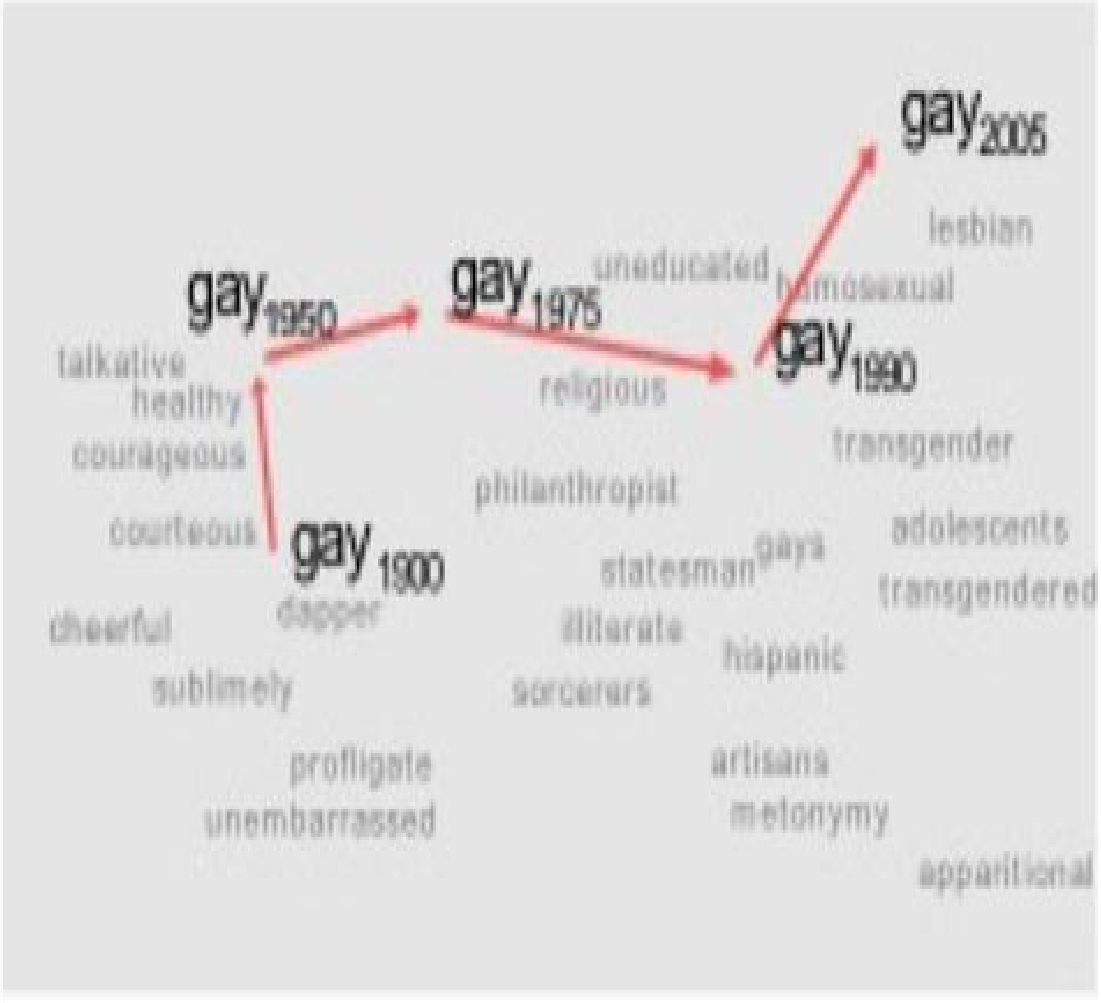
MAINFRAMES

Mainframes usually are referred those computers with fast, advanced processing capabilities that could perform by itself tasks that may require a lot of Personal Computers (PC) Machines. Usually mainframes would have lots of RAMs, very large secondary storage devices, and very fast processors to cater for the needs of those computers under its service.

Due to the advanced components mainframes have, these computers have the capability of running multiple large applications required by most enterprises, which is one of its advantage. Mainframes are also suitable to cater for those applications or files that are of very large demand

Beispiel: Historische Semantik

Kulkarni, Al-Rfou, Perozzi, Skiena 2015



Ähnlichkeit vs. Relationiertheit

- Ähnlichkeit: Auto, Fahrrad, Traktor
- Relationiertheit: Auto— Benzin, Auto — Lenkrad, Auto— Fahrer, Auto — lenken, Auto — verschmutzend

Verwandte Unterscheidung:

- **Paradigmatische Relationen:** austauschbare Einheiten in einem Kontext
- **Syntagmatische Relationen:** Worte, die (häufig) in einer Äusserung aufeinander folgen

Zwei Arten von Algorithmen

Für Ähnlichkeitsbestimmung:

Lexikonbasiert

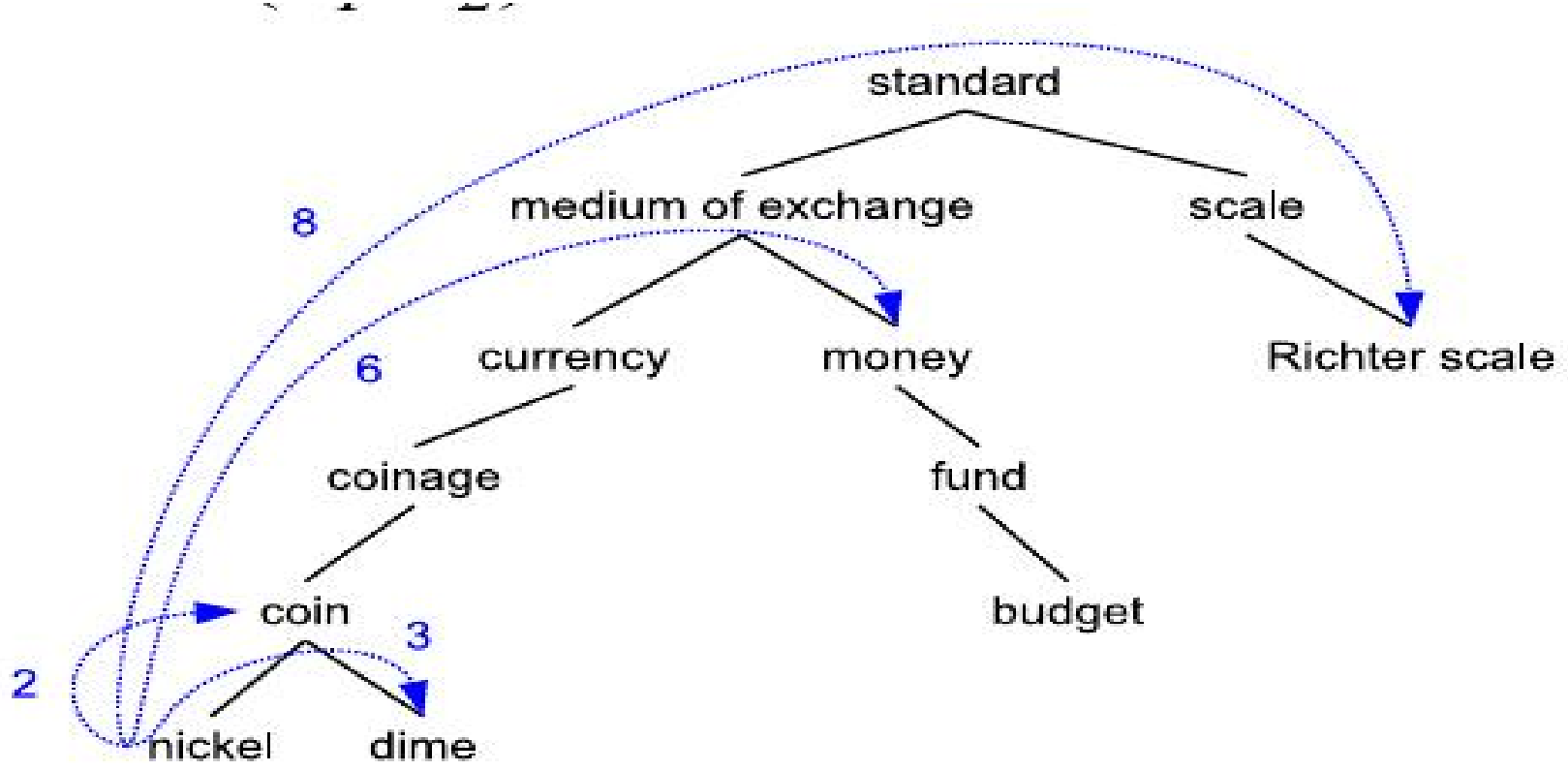
- Benutzung der Hierarchie (heute)
- Benutzung der glosses/Definitionen (wird nicht besprochen)

Distributionell

- “You shall know a word by the company it keeps” (Firth)
- Vektorbasiert: lineare Algebra

Übersicht

WordNethierarchie: Pfadmethode



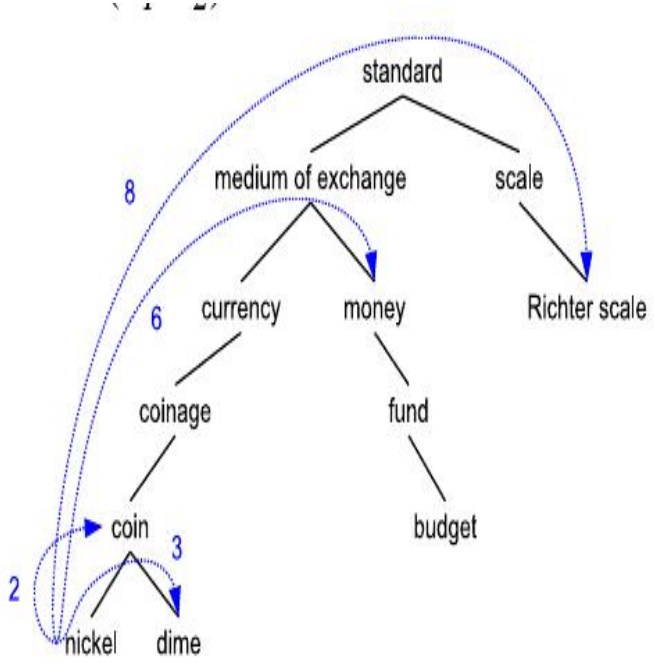
$pathlen(c_1, c_2) = 1 +$ Anzahl der Kanten des kürzesten Pfades

$simpath(c_1, c_2) = \frac{1}{pathlen(c_1, c_2)}$

$wordsim(w_1, w_2) = \max_{c_1 \in senses(w_1), c_2 \in senses(w_2)} simpath(c_1, c_2)$

Zwischen 0 bis 1

Pfadmethode: Beispiel



simpath(nickel,coin) = 1/2 = .5
simpath(fund,budget) = 1/2 = .5
simpath(nickel,currency) = 1/4 = .25
simpath(nickel,money) = 1/6 = .17
simpath(coinage,Richter scale) = 1/6 = .17

Probleme

- Nimmt an, dass alle Kanten gleiche “Distanz” ausdrücken
- Unabhängig von Platz in Hierarchie
- Es gibt Adaptierungen (Leacock und Chodorov, Wu und Palmer), die versuchen, dies durch Normalisierungen zu vermeiden.
- Besser: Gemeinsamkeiten und Unterschiede der beiden Konzepte benutzen
- Besser: Wenn man weit hoch in die Hierarchie “muss”, dann ist die Verbindung schwächer
- Besser: Information Content Metriken (siehe Folien im Anhang)

Zwischenfazit

- Wortähnlichkeit kann durch WordNethierarchie berechnet werden
- Hierarchie: Einfache Pfadlängenberechnung leidet unter unterschiedlicher Körnigkeit in verschiedenen Hierarchieteilen
- Hierarchie: Information content kann dies beheben, ist aber korpusabhängig (für P(C) Berechnung)
- Algorithmen eigentlich für sense-Ähnlichkeit: muss meist durch Maximierung über alle senses zweier Worte auf Worte übertragen werden
- **Brauchen sehr viel manuelle Vorbereitung: Wordneterstellung**

Übersicht

Vektorraummodell

You shall know a word by the company it keeps (Firth,57).

- Wortbedeutung $\approx$ Kontext.
  - Ich trinke *Maruinho*.
  - Der *Maruinho* ist sehr stark.
  - *Maruinho* ist aus Weizen gebrannt.
- Worte sind ähnlich, wenn sie in ähnlichen Kontexten vorkommen
- Worte werden als Vektoren in einem hochdimensionalen Vektorraum eingebettet (**embeddings**).

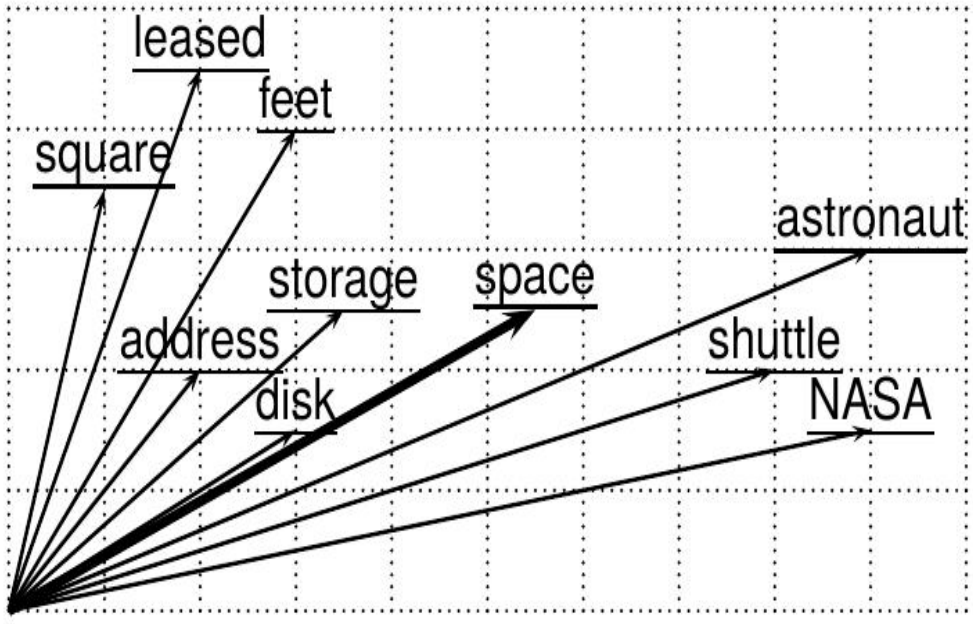
$$\mathit{Embedding} : f : V \rightarrow \mathbb{R}^n$$

- Worte sind sich ähnlich, wenn sie im Vektorraum nah beieinander liegen.

Vorteile von Worten als Vektoren

- Alle Methoden aus der linearen Algebra stehen uns zur Verfügung
- Input von Methoden aus dem Maschinellen Lernen brauchen Zahlen

Vektorraummodell: vereinfacht



- Ist *space* näher an *NASA* oder an *square*?
- Welches Wort ist *space* am ähnlichsten?
- Wie konstruieren wir die Vektoren und wie messen wir Ähnlichkeit?

Beispiel: Vektorraummodell

And this is the last thing the anthropologist who tries to understand symbols can afford to do. he must start with the explanations and commentaries which his informants themselves offer about their **symbols. these must *first* be examined in** the contexts in which they are usually employed, where they occur naturally, although subsequent generalizing discussion helps the anthropologist to improve his very **superficial initial standing. *learning* the meaning of** symbols is part of the **anthropologist's practical semantics: *discovering* the meaning of** words, noticing when their use is appropriate and when it is not. all this requires imagination, patience, considerable linguistic skill, but above all a rigorous respect for the facts. **these must come *first*; fantasy can come** later ...

Beispiel: Vektorraummodell

And this is the last thing the anthropologist who tries to understand symbols can afford to do. he must start with the explanations and commentaries which his informants themselves offer about their **symbols. these must *first* be examined in** the contexts in which they are usually employed, where they occur naturally, although subsequent generalizing discussion helps the anthropologist to improve his very **superficial initial standing. *learning* the meaning of** symbols is part of the **anthropologist's practical semantics: *discovering* the meaning of** words, noticing when their use is appropriate and when it is not. all this requires imagination, patience, considerable linguistic skill, but above all a rigorous respect for the facts. **these must come *first*; fantasy can come** later ...

	these	meaning	the	practical	come
first					
learning					
discovering					

Beispiel: Vektorraummodell

And this is the last thing the anthropologist who tries to understand symbols can afford to do. he must start with the explanations and commentaries which his informants themselves offer about their **symbols. these must *first* be examined in** the contexts in which they are usually employed, where they occur naturally, although subsequent generalizing discussion helps the anthropologist to improve his very **superficial initial standing. *learning* the meaning of** symbols is part of the **anthropologist's practical semantics: *discovering* the meaning of** words, noticing when their use is appropriate and when it is not. all this requires imagination, patience, considerable linguistic skill, but above all a rigorous respect for the facts. **these must come *first*; fantasy can come** later ...

	these	meaning	the	practical	come
first	2	0	0	0	2
learning					
discovering					

Beispiel: Vektorraummodell

And this is the last thing the anthropologist who tries to understand symbols can afford to do. he must start with the explanations and commentaries which his informants themselves offer about their **symbols. these must *first* be examined in** the contexts in which they are usually employed, where they occur naturally, although subsequent generalizing discussion helps the anthropologist to improve his very **superficial initial standing. *learning* the meaning of** symbols is part of the **anthropologist's practical semantics: *discovering* the meaning of** words, noticing when their use is appropriate and when it is not. all this requires imagination, patience, considerable linguistic skill, but above all a rigorous respect for the facts. **these must come *first*; fantasy can come** later ...

	these	meaning	the	practical	come
first	2	0	0	0	2
learning	0	1	1	0	0
discovering					

Beispiel: Vektorraummodell

And this is the last thing the anthropologist who tries to understand symbols can afford to do. he must start with the explanations and commentaries which his informants themselves offer about their **symbols. these must *first* be examined in** the contexts in which they are usually employed, where they occur naturally, although subsequent generalizing discussion helps the anthropologist to improve his very **superficial initial standing. *learning* the meaning of** symbols is part of the **anthropologist's practical semantics: *discovering* the meaning of** words, noticing when their use is appropriate and when it is not. all this requires imagination, patience, considerable linguistic skill, but above all a rigorous respect for the facts. **these must come *first*; fantasy can come** later ...

	these	meaning	the	practical	come
first	2	0	0	0	2
learning	0	1	1	0	0
discovering	0	1	1	1	0

Beispiel: Vektorraummodell

And this is the last thing the anthropologist who tries to understand symbols can afford to do. he must start with the explanations and commentaries which his informants themselves offer about their **symbols. these must *first* be examined in** the contexts in which they are usually employed, where they occur naturally, although subsequent generalizing discussion helps the anthropologist to improve his very **superficial initial standing. *learning* the meaning of** symbols is part of the **anthropologist's practical semantics: *discovering* the meaning of** words, noticing when their use is appropriate and when it is not. all this requires imagination, patience, considerable linguistic skill, but above all a rigorous respect for the facts. **these must come *first*; fantasy can come** later ...

	these	meaning	the	practical	come
first	2	0	0	0	2
learning	0	1	1	0	0
discovering	0	1	1	1	0

Wort-Wort matrix

Eigenschaften

- **Zielworte** in Reihen, **Kontextworte** in Spalten
- Jede Reihe ist ein **Ko-okkurrenzvektor** für das Zielwort.
- Jede Dimension ein Maß für Assoziation mit einem Kontextwort.
- Assoziationsmaß beruht auf Kookkurrenz zwischen Zielwort und Kontextwort in einem Korpus (meist in einem Wortfenster)
- Kookkurrenzvektor $\approx$ hochdimensionale Zusammenfassung des Verhaltens des Zielwortes
- Im Beispiel: unterschiedliche Zielworte und Kontextworte
- In echt: meist das Vokabular im ganzen als Ziel und Kontextworte. Matrixdimension $|V| \times |V|$
- Wegen Zipfschem Gesetz: die Matrix ist “sparse”.

Parameter

Schritte zur Konstruktion eines Vektorraummodels:

1. Wähle **Kontextworte/Merkmale** c_i, z.B. die 10,000 häufigsten Worte im Korpus.
2. Bestimme **Fenstergröße**. Z. B. 10 Worte (5 links und 5 rechts).
3. Für jedes Zielwort w, bestimme **Kookkurrenzvektor** mit Hilfe von Assoziationmaß z.B. Kookkurrenzhäufigkeit mit Kontextword c_i
4. Berechne mit **Ähnlichkeitsmaß**, ob Vektoren nah beiander liegen

Parameter 1: Kontextworte

- Benutze nur die häufigsten 10K bis 50K Wörter. Warum?
- Evtl. eliminiere Funktionswörter. Warum?
- Wörter oder Lemmas?
- Benutze (grammatische Funktion, Wort) Paare als Merkmale. Warum?
  - Dimension *subject-of learn*. Wie oft ist Zielwort *student* Subjekt von *learn*?
  - Dimension *object-of learn*. Wie oft ist Zielwort *student* Objekt von *learn*?

Parameter 1: Kontextworte mit grammatischer Funktion

- Wort-Wort-Matrix
- Hier aus Stefan Evert, 2012: Kontextwörter in syntaktischer Relation zum Zielwort: Obj-V

Vektoren aus Kookkurenzwerten

	get w_1	see w_2	use w_3	hear w_4	eat w_5	kill w_6
knife	51	20	84	0	3	0
cat	52	58	4	4	6	26
???	115	83	10	42	33	17
boat	59	39	23	4	0	0
cup	98	14	6	2	1	0
pig	12	17	3	2	9	27
banana	11	2	2	0	18	0

Parameter 2: Fenstergröße

(Fenstergröße natürlich irrelevant, wenn wir grammatische Funktionen wählen)

Häufigste Nachbarn von *car* und *dog* (BNC)

2-Wort Fenster

car	dog
van	cat
vehicle	fox
truck	fox
motorcyle	pet
driver	rabbit
motor	pig

Tendenz: paradigmatische Assoziationen

Parameter 2: Fenstergröße

30-wort Fenster

car	dog
drive	kennel
park	puppy
bonnet	pet
windscreen	bitch
headlight	rottweiler

Tendenz: Syntagmatische Assoziationen (nicht immer)

Parameter 3: Assoziationsmaße

Mit Zielwort w_j und Kontextwort c_i

Assoc	$assoc(w_j, c_i)$
$assoc_{bin}$	1, wenn mindestens einmal w_j mit c_i vorkommt, 0 sonst
$assoc_{freq}$	$f(w_j, c_i)$
$assoc_{cond}$	$p(w_j \mid c_i)$
$assoc_{pmi}$	$\log \frac{p(w_j, c_i)}{p(w_j) \cdot p(c_i)}$
...	...

Für ein Wort w und Kontextworte $c_1, \ldots c_n$ können wir nun ein embedding konstruieren mit dem gewähltem Assoziationsmaß konstruieren:

$$\vec{w} = (assoc(w, c_1), assoc(w, c_2), assoc(w, c_3) \ldots, assoc(w, c_n))$$

Parameter 3: Pointwise mutual information als Assoziationsmaß

- **Pointwise mutual information.** Vorteile? Nachteile?

$$PMI(w_j, c_i) = log_2 \frac{p(w_j, c_i)}{p(w_j) \times p(c_i)}$$

- Setze $PMI(w_j, c_i) = 0$, wenn $f(w_j, c_i) = 0$
- Wahrscheinlichkeiten werden durch Frequenzen geschätzt
- Verbesserung: **Positive pointwise mutual information**. Warum?

$$PPMI(w_j, c_i) = max(PMI(w_j, c_i), 0)$$

.

- Weitere Verbesserung: **Shifted positive pointwise mutual information**

$$SPPMI_k(w_j, c_i) = max(PMI(w_j, c_i) - \log_2 k, 0)$$

.

Parameter 4: Möglichkeit I: Euklidische Metrik

Euklidische Metrik für zwei Vektoren $\vec{v}, \vec{w} \in \mathbb{R}^n$

$$d_2(\vec{v}, \vec{w}) := \sqrt{\sum_{i=1}^{n} (v_i - w_i)^2}$$

Dies entspricht der geometrischen Interpretation des "Luftlinienabstands" im $\mathbb{R}^2$ (oder $\mathbb{R}^3$):

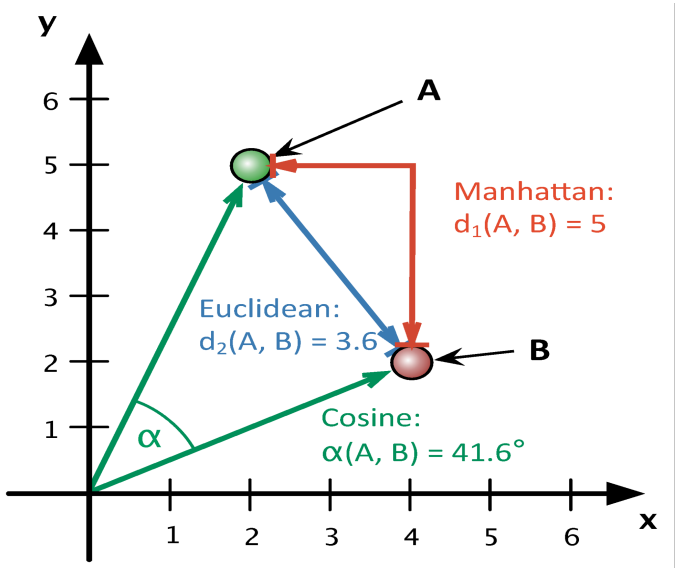


Bild von `http://dh2016.adho.org/static/data/290.html`

Parameter 4: Möglichkeit 2: Kosinusähnlichkeit aus Skalarprodukt

Das **Skalarprodukt** zweier Vektoren im $\mathbb{R}^n$ ist definiert als eine Abbildung $\cdot : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ mit $\vec{v} \cdot \vec{u} := \sum_{i=1}^{n} v_i \cdot u_i$

Notation: oft auch geschrieben als $\langle \vec{v}, \vec{u} \rangle$.

Beispiel im $\mathbb{R}^3$:

$$(1,-2,1)\cdot(3,4,-1) = 1\cdot 3 + (-2)\cdot 4 + 1\cdot(-1) = 3 + (-8) + (-1) = -6$$

Parameter 4: Eigenschaften von Skalarprodukt

$\vec{v} \cdot \vec{u} := \sum_{i=1}^{n} v_i \cdot u_i$

- Skalarprodukt groß, wenn hohe Werte in denselben Dimensionen
- Skalarprodukt klein, wenn Nullen in verschiedenen Dimensionen
- Symmetrisch
- Also gutes Ähnlichkeitsmaß
- Abhängigkeit von Vektorlänge!

Normalisiertes Skalarprodukt = cosine

Normalisiertes Skalarprodukt = Kosinus des Winkels zwischen zwei (Nicht-Null-)Vektoren $\vec{x}, \vec{y} \in \mathbb{R}^n$

$$cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}||\vec{y}|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}}$$

- Abstrahiert von Länge der Vektoren (warum und wie?)
- Werte zwischen -1 und +1. Für PPMI und frequenzbasierte Vektoren: zwischen 0 und 1
- Am höchsten (=1) , wenn Winkel zwischen Vektoren am kleinsten: Vektoren zeigen genau in die gleiche Richtung (sind parallel)
- Am niedrigsten (=-1), wenn in gegensätzliche Richtung
- Null, wenn orthogonal

Parameter 4: Geometrische Interpretation des Skalarprodukts

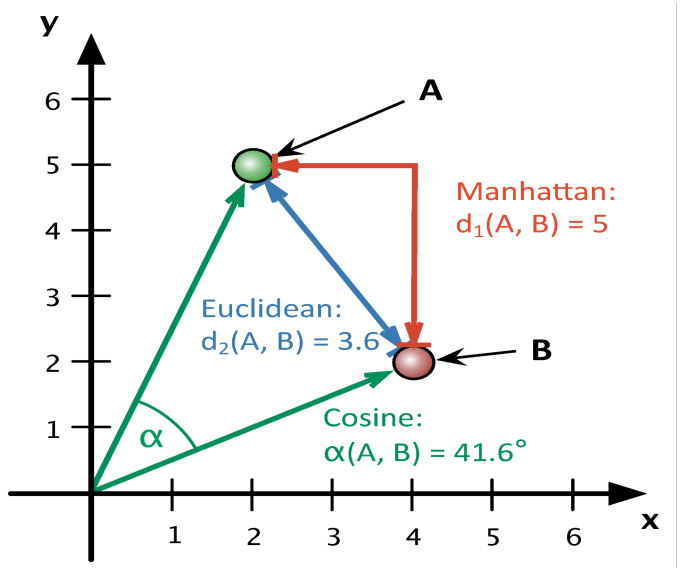


Bild von `http://dh2016.adho.org/static/data/290.html`

Übersicht

Beispiel 1: Assoziationen

Korpus BNC, Zielwort w, $V = \{species, computer, animal\}$, rel_{win10} (window 5 to the right and 5 to left of w), $A_{freq} = assoc_{freq}$

$\vec{cat} = (A_{freq}(cat, species), A_{freq}(cat, computer), A_{freq}(cat, animal)) = (59, 5, 304)$

$\vec{carnivore} = (A_{freq}(carnivore, species), A_{freq}(carnivore, computer), A_{freq}(carnivore, animal)$
$(21, 1, 21)$

$\vec{feline} = (A_{freq}(feline, species), A_{freq}(feline, computer), A_{freq}(feline, animal)) = (2, 0, 5)$

$\vec{airport} = (A_{freq}(airport, species), A_{freq}(airport, computer), A_{freq}(airport, animal)) = (4, 12, 2)$

Beispiel 1: Embeddingsmatrix

Solange wir für alle Vektordarstellungen das gleiche Vokabular, das gleiche Assoziationsmaß sowie die gleiche Relation/Fenstergröße nehmen, sind die embeddings verschiedener Wörter vergleichbar (da in den gleichen Vektorraum eingebettet).

Matrixschreibweise (mit Vokabular in Spalten):

	species	*computer*	*animal*
cat	59	5	304
carnivore	21	1	21
feline	2	0	5
airport	4	12	2

Hierbei immer implizit: Assoziationsmaß sowie Relation!

Beispiel 1: Beobachtungen

	species	*computer*	*animal*
cat	59	5	304
carnivore	21	1	21
feline	2	0	5
airport	4	12	2

- Man sieht die ähnlicheren Wortpaare sind in den gleichen Vokabulardimensionen “stark” bzw schwach
- Wenn man unbedacht vorgeht, dann könnte die Ähnlichkeit fälschlicherweise von der Worthäufigkeit (= Vektorlänge) abhängen!
- Die Matrix ist im allgemeinen hochdimensional und “sparse”

Beispiel 1: Beobachtungen

Die gleiche Matrix mit $assoc_{bin}$:

	species	*computer*	*animal*
cat	1	1	1
carnivore	1	1	1
feline	1	0	1
airport	1	1	1

Beispiel 1: Euklidische Distanz d_2

	species	*computer*	*animal*
cat	59	5	304
carnivore	21	1	21
feline	2	0	5
airport	4	12	2

$d_2(cat, carnivore) = \sqrt{(59-21)^2 + (5-1)^2 + (304-21)^2} = 285$ (gerundet)

$d_2(cat, feline) = \sqrt{(59-2)^2 + (5-0)^2 + (304-5)^2} = 304$ (gerundet)

$d_2(cat, airport) = \sqrt{(59-4)^2 + (5-12)^2 + (304-2)^2} = 307$ (gerundet)

Ist dies, was wir wollen? Wo liegt das Problem?

Beispiel 1: Probleme mit euklidischer Distanz

- Abhängigkeit von Vektorlänge = Worthäufigkeit
- Distanz deswegen auch nicht nach oben beschränkt
- Distanz anstatt Ähnlichkeit $\rightarrow$ Umwandlung in Ähnlichkeit z.B. mit $sim(v, w) = 1 - d_2(v, w)$ $\rightarrow$ Negative Ähnlichkeiten

Besser: Direkte Ähnlichkeitsmaße, die nicht längenabhängig sind.

- Eine Möglichkeit: Normiere Vektoren zuerst (siehe Übungsaufgabe)
- Zweite Möglichkeit: Kosinusähnlichkeit

Beispiel 1: Kosinusähnlichkeit

$$cos(\vec{x},\vec{y}) = \frac{\vec{x}\cdot\vec{y}}{|\vec{x}||\vec{y}|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

	species	*computer*	*animal*
cat	59	5	304
carnivore	21	1	21
feline	2	0	5
airport	4	12	2

$$cos_{sim}(cat, carnivore) = \frac{59\cdot21+5\cdot1+304\cdot21}{\sqrt{59^2+5^2+304^2}\cdot\sqrt{21^2+1^2+21^2}} = \frac{7628}{\sqrt{95922}\sqrt{883}} = 0.828$$

$$cos_{sim}(cat, feline) = \frac{59\cdot2+5\cdot0+304\cdot5}{\sqrt{59^2+5^2+304^2}\cdot\sqrt{2^2+0^2+5^2}} = 0.98$$

$$cos_{sim}(cat, airport) = \frac{59\cdot4+5\cdot12+304\cdot2}{\sqrt{59^2+5^2+304^2}\cdot\sqrt{4^2+12^2+2^2}} = 0.227$$

Beispiel 2: Grammatische Funktion

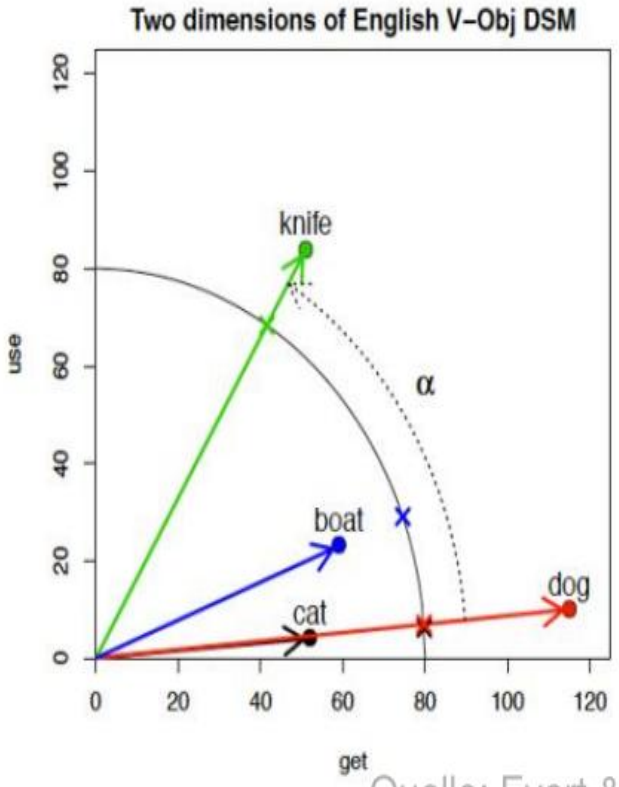
- Wort-Wort-Matrix
- Hier aus Stefan Evert, 2012: Kontextwörter in syntaktischer Relation zum Zielwort: Obj-V

Vektoren aus Kookkurenzwerten

	get w_1	see w_2	use w_3	hear w_4	eat w_5	kill w_6
knife	51	20	84	0	3	0
cat	52	58	4	4	6	26
???	115	83	10	42	33	17
boat	59	39	23	4	0	0
cup	98	14	6	2	1	0
pig	12	17	3	2	9	27
banana	11	2	2	0	18	0

Beispiel 2: Grammatische Funktion

- Geometrische Interpretation distributioneller Ähnlichkeit
- Richtung wichtiger als Ort im Raum
- Winkel zwischen Wortvektoren als Ähnlichkeitsmaß



Quelle: Evert & Lenci, 2009

Beispiel 3: PPMI

Term-Term-Matrix mit Frequenzen (aus Jurafsky und Martin, Edition 3)

	computer	data	pinch	result	sugar	
apricot	0	0	1	0	1	2
pineapple	0	0	1	0	1	2
digital	2	1	0	1	0	4
information	1	6	0	4	0	11
	3	7	2	5	2	19

- Die Randhäufigkeiten entsprechen nicht den Unigramfrequenzen der Wörter (Warum nicht?)
- Die Gesamthäufigkeit N der Beobachtungen (hier 19) entspricht im Normalfall nicht der Korpusgröße (Warum nicht?)

Beispiel 3: PPMI

Term-Term-Matrix mit Frequenzen:

	computer	data	pinch	result	sugar	
apricot	0	0	1	0	1	2
pineapple	0	0	1	0	1	2
digital	2	1	0	1	0	4
information	1	6	0	4	0	11
	3	7	2	5	2	19

$$ppmi(information, data) = max(\log_2 \frac{\frac{6}{19}}{\frac{11}{19} \cdot \frac{7}{19}}, 0) = 0.57$$

$$ppmi(information, computer) = max(\log_2 \frac{\frac{1}{19}}{\frac{11}{19} \cdot \frac{3}{19}}, 0) = max(\log_2 \frac{19}{33}, 0) = 0$$

$$ppmi(apricot, computer) = max(\log_2 \frac{\frac{0}{19}}{\frac{2}{19} \cdot \frac{3}{19}}, 0) = max(\log_2 0, 0) = 0$$

$$ppmi(apricot, pinch) = max(\log_2 \frac{\frac{1}{19}}{\frac{2}{19} \cdot \frac{2}{19}}, 0) = 2.25$$

Beispiel 3: PPMI

Term-Term-Matrix mit Frequenzen:

	computer	data	pinch	result	sugar	
apricot	0	0	1	0	1	2
pineapple	0	0	1	0	1	2
digital	2	1	0	1	0	4
information	1	6	0	4	0	11
	3	7	2	5	2	19

$$ppmi(information, data) = max(\log_2 \frac{\frac{6}{19}}{\frac{11}{19} \cdot \frac{7}{19}}, 0) = 0.57$$

$$ppmi(information, computer) = max(\log_2 \frac{\frac{1}{19}}{\frac{11}{19} \cdot \frac{3}{19}}, 0) = max(\log_2 \frac{19}{33}, 0) = 0$$

$$ppmi(apricot, computer) = max(\log_2 \frac{\frac{0}{19}}{\frac{2}{19} \cdot \frac{3}{19}}, 0) = max(\log_2 0, 0) = 0$$

$$ppmi(apricot, pinch) = max(\log_2 \frac{\frac{1}{19}}{\frac{2}{19} \cdot \frac{2}{19}}, 0) = 2.25$$

Beispiel 3: PPMI

Term-Term-Matrix mit Frequenzen:

	computer	data	pinch	result	sugar	
apricot	0	0	1	0	1	2
pineapple	0	0	1	0	1	2
digital	2	1	0	1	0	4
information	1	6	0	4	0	11
	3	7	2	5	2	19

$$ppmi(information, data) = max(\log_2 \frac{\frac{6}{19}}{\frac{11}{19} \cdot \frac{7}{19}}, 0) = 0.57$$

$$ppmi(information, computer) = max(\log_2 \frac{\frac{1}{19}}{\frac{11}{19} \cdot \frac{3}{19}}, 0) = max(\log_2 \frac{19}{33}, 0) = 0$$

$$ppmi(apricot, computer) = max(\log_2 \frac{\frac{0}{19}}{\frac{2}{19} \cdot \frac{3}{19}}, 0) = max(\log_2 0, 0) = 0$$

$$ppmi(apricot, pinch) = max(\log_2 \frac{\frac{1}{19}}{\frac{2}{19} \cdot \frac{2}{19}}, 0) = 2.25$$

Beispiel 3: PPMI

Term-Term-Matrix mit Frequenzen:

	computer	data	pinch	result	sugar	
apricot	0	0	1	0	1	2
pineapple	0	0	1	0	1	2
digital	2	1	0	1	0	4
information	1	6	0	4	0	11
	3	7	2	5	2	19

$$ppmi(information, data) = max(\log_2 \frac{\frac{6}{19}}{\frac{11}{19} \cdot \frac{7}{19}}, 0) = 0.57$$

$$ppmi(information, computer) = max(\log_2 \frac{\frac{1}{19}}{\frac{11}{19} \cdot \frac{3}{19}}, 0) = max(\log_2 \frac{19}{33}, 0) = 0$$

$$ppmi(apricot, computer) = max(\log_2 \frac{\frac{0}{19}}{\frac{2}{19} \cdot \frac{3}{19}}, 0) = max(\log_2 0, 0) = 0$$

$$ppmi(apricot, pinch) = max(\log_2 \frac{\frac{1}{19}}{\frac{2}{19} \cdot \frac{2}{19}}, 0) = 2.25$$

Beispiel 3: PPMI

Term-Term-Matrix mit Frequenzen:

	computer	data	pinch	result	sugar	
apricot	0	0	1	0	1	2
pineapple	0	0	1	0	1	2
digital	2	1	0	1	0	4
information	1	6	0	4	0	11
	3	7	2	5	2	19

$$ppmi(information, data) = max(\log_2 \frac{\frac{6}{19}}{\frac{11}{19} \cdot \frac{7}{19}}, 0) = 0.57$$

$$ppmi(information, computer) = max(\log_2 \frac{\frac{1}{19}}{\frac{11}{19} \cdot \frac{3}{19}}, 0) = max(\log_2 \frac{19}{33}, 0) = 0$$

$$ppmi(apricot, computer) = max(\log_2 \frac{\frac{0}{19}}{\frac{2}{19} \cdot \frac{3}{19}}, 0) = max(\log_2 0, 0) = 0$$

$$ppmi(apricot, pinch) = max(\log_2 \frac{\frac{1}{19}}{\frac{2}{19} \cdot \frac{2}{19}}, 0) = 2.25$$

Beispiel 3: PPMI

Term-Term-Matrix mit PPMI

	computer	data	pinch	result	sugar
apricot	0	0	2.25	0	2.25
pineapple	0	0	2.25	0	2.25
digital	1.66	0	0	0	0
information	0	0.57	0	0.47	0

Ein Problem: PPMI überschätzt seltene Kontextwörter (siehe *pinch*).
Wie löst man das? (Smoothing siehe Jurafsky und Martin, Kapitel 6)

Zwischenfazit

- Vektormodelle sind eine einfache Möglichkeit, Wortähnlichkeiten zu berechnen. Benötigen keine manuell kreiierten Ressourcen.
- Setzen der Parameter: Fenstergröße, Vokabular, Ähnlichkeitsmass, Assoziationsmass
- Oft gute Settings: 10K bis 50K häufigste Worte in 10-Wort Fenster, (S)PPMI, Kosinusähnlichkeit

Ausblick

Probleme durch sparse Vektoren

- Overfitting durch zu viele Parameter mit geringen “counts”
- Zu viele fälschlich unterschiedliche Dimensionen:
  - *Sputnik* kommt mit *Kosmonaut* vor
  - *Apollo17* kommt mit *Astronaut* vor
  - *Kosmonaut* und *Astronaut* sind unterschiedliche Dimensionen. Damit lässt sich die “Ahnlichkeit zwischen *Sputnik* und *Apollo17* schwer fassen.
- Daher weiterführend (Kapitel 6.7): dichte Wortvektoren durch Dimensionsreduzierung oder Embeddings mit neuronalen Netzen

Übersicht

Datensätze

Menschliche Ähnlichkeitsnormen. Zum Beispiel: WordSim353.
`http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353/`

Wort1	Wort2	Rating
tiger	cat	7.35
tiger	tiger	10.00
drug	abuse	6.85
bread	butter	6.19
cup	coffee	6.58
cup	object	3.69
king	cabbage	0.23
king	queen	8.58
king	rook	5.92

Was fällt auf?

State-of-the Art (SPPMI Modelle mit Matrixfaktorisierung, unüberwacht): 0.69 Korrelation. (Levy und Goldberg, 2014)

Datensätze

SimLex 999: Auch Adjektive und Verben. Nur Ähnlichkeit, nicht Relationiertheit.

`https://www.cl.cam.ac.uk/~fh295/simlex.html`

bread	cheese	1.95
cup	spoon	2.02
cup	jar	5.13
vanish	disappear	9.8

Performanz eines word2vec oder glove Vektormodels: um die 0.40

- Schwierigeres Datenset
- Menschliches Agreement: 0.78
- Bestes derzeitiges Modell (benutzt Vektormodelle/embeddings plus lexikalische Datenbanken plus **überwachtes** Lernen!): 0.76
- Update zur Performanz siehe Webseite!

TOEFL-Test: Substituierbarkeit im Kontext

TOEFL: Test of English as a foreign language

Substituierbarkeit im Kontext:
*You will find the office at the main **intersection***

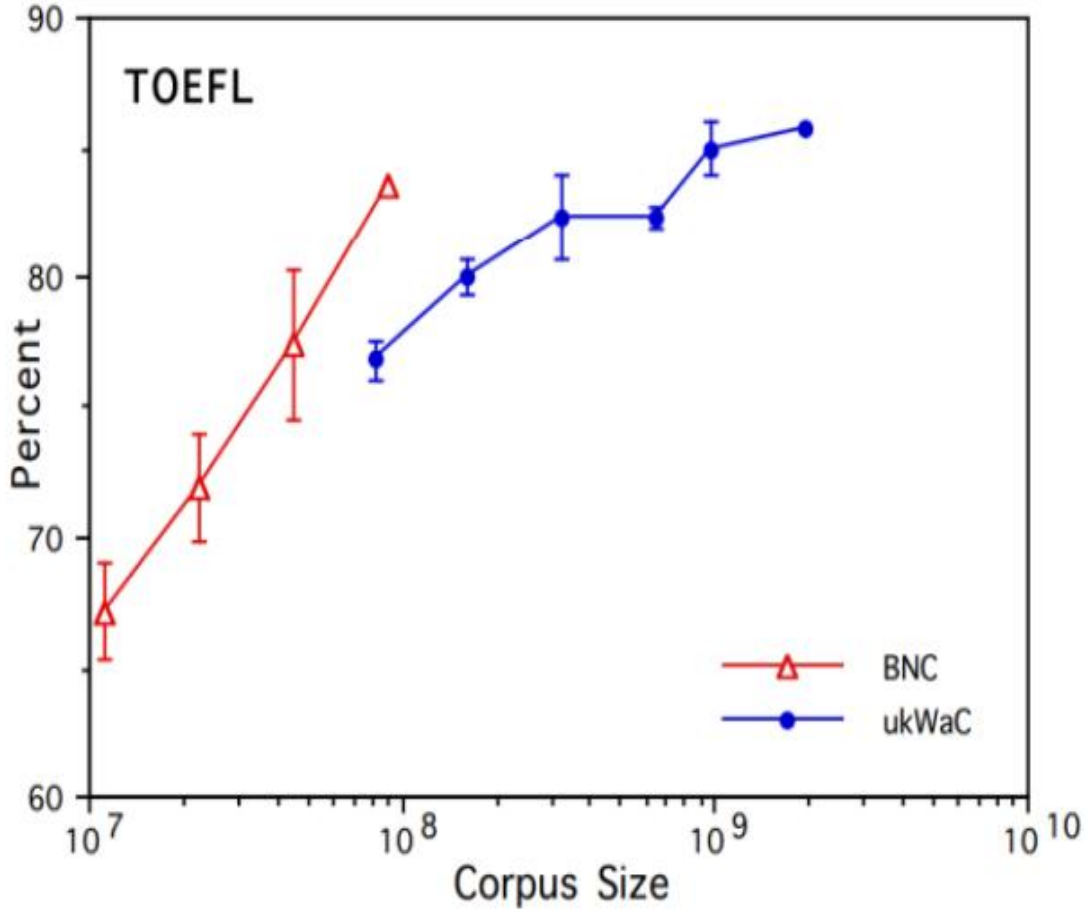
1. place
2. crossroads
3. roundabout
4. building

TOEFL Resultate

- Durchschnitt nicht-Englisch-Muttersprachler: 64%
- WordNet-basiert [Jarmasz&Spakowicz 2003]
- Thesaurusbasiertes Ähnlichkeitsmaß Lin: Nomen: 85%, insgesamt 22%
- Korpusbasiert [Bullinaria & Levy 2012]
  - Korpus: ukWaC (Web-Korpus, 1,8 Milliarden Wörter)
  - PPMI, Cosine
  - alle Wörter: 86%

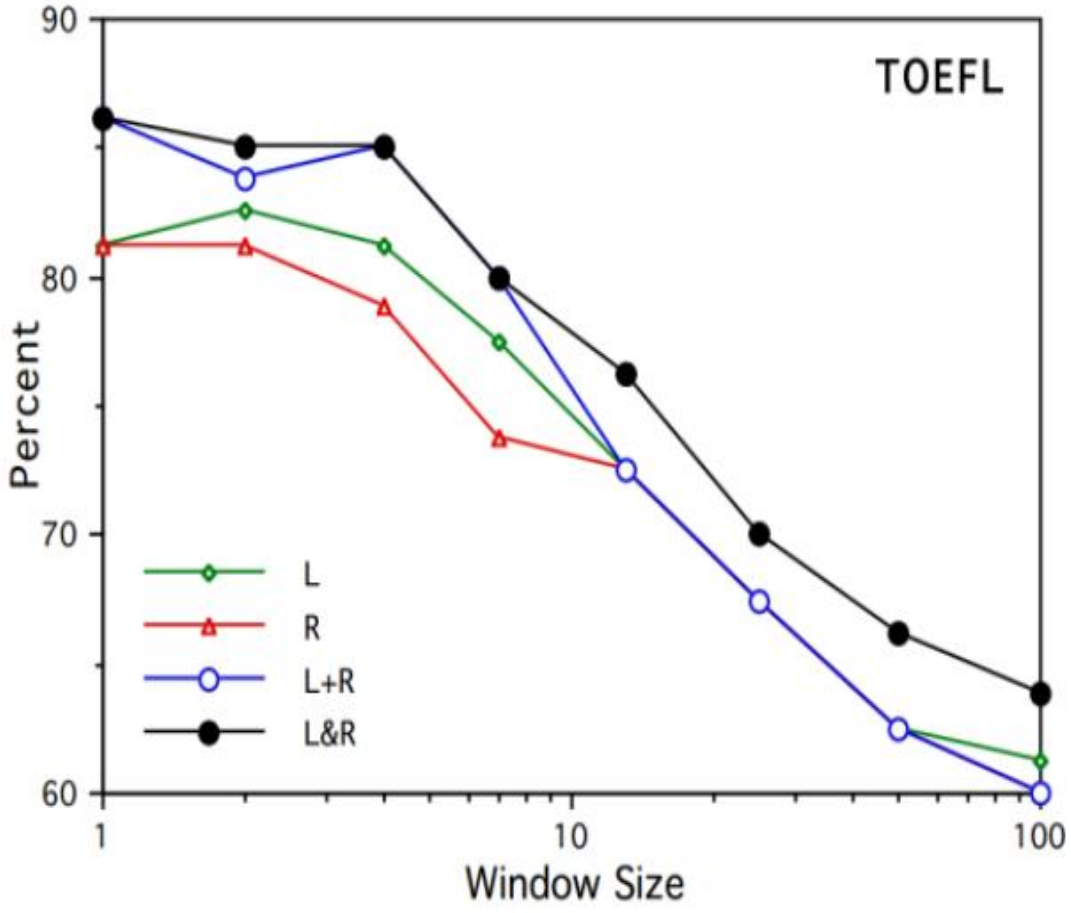
TOEFL Evaluation

Aus Bullinaria und Levy, 2012



TOEFL Evaluation

Aus Bullinaria und Levy, 2012



Beurteilung

- Beste Resultate werden erst erzielt, wenn man aus “sparse” Vektoren dichte macht (via Latent semantic analysis oder bei neural word embeddings)
- Vorteil: funktioniert ohne manuelle Thesauruserstellung
- Nachteil: Vermischung von Relationen: relatedness, Ähnlichkeit etc.
- Nachteil: Sense-Unterscheidung wird ignoriert

Übersicht

Referenzen: Ähnlichkeitsberechnung mit Thesaurus

- In NLTK implementiert:
  `http://www.nltk.org/howto/wordnet.html` beschreibt, wie man WordNet in NLTK verwendet
- WordNet::Similarity
  - `http://wn-similarity.sourceforge.net/`
  - Web Demo: `http://ws4jdemo.appspot.com`
- Noch inkludiert in Jurafsky und Martin (2nd edition), Kapitel 20.6

Referenzen: Distributionelle Methoden

- * Jurafsky und Martin, 3rd edition, Kapitel 6. Dies entält auch ein weiteres vollständig durchgerechnetes Beispiel für PPMI.
- Toolkits:
  - DISCO `http://www.linguatools.de/disco/disco_en.html`
  - Neuronale Netze word2vec: `https://code.google.com/archive/p/word2vec/`
  - Glove: `http://nlp.stanford.edu/projects/glove/`

Referenzen: Mathematischer Hintergrund

- Vorlesung Embeddings: https://www.cl.uni-heidelberg.de/courses/ss19/emb/material/, insbesondere https://www.cl.uni-heidelberg.de/courses/ss19/emb/material/vectors.pdf und https://www.cl.uni-heidelberg.de/courses/ss19/emb/material/metrics_sim.pdf
- Gerd Fischer: Lineare Algebra. Eine Einführung für Studienanfänger
- Serie von Videos von 3Blue1Brown. Startet hier: https://www.youtube.com/watch?v=fNk_zzaMoSs&list=PLZHQObOWTQDPD3MizzM2xVFitg

Referenzen: Distributionelle Methoden weiterführend

- Levy, O., & Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In Advances in neural information processing systems (pp. 2177-2185).
- Mikolov, T., Chen, K., Corrado, G.,& Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).
- Bullinaria, J. A., & Levy, J. P. (2012). Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and SVD. Behavior research methods, 44(3), 890-907.
- Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

Referenzen: Evaluation, Anwendungen

- Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin (2002): Placing Search in Context: The Concept Revisited, ACM Transactions on Information Systems, 20(1):116-131, January 2002
- Felix Hill, Roi Reichart and Anna Korhonen (2015): SimLex-999: Evaluating Semantic Models with (Genuine) Similarity Estimation. 2014. Computational Linguistics. 2015
- Kulkarni, V., Al-Rfou, R., Perozzi, B., Skiena, S. (2015). Statistically significant detection of linguistic change. In Proceedings of the 24th International Conference on World Wide Web (pp. 625-635).

Übersicht

Anhang

Information Content für die Berechnung von Wortähnlichkeiten in WordNet

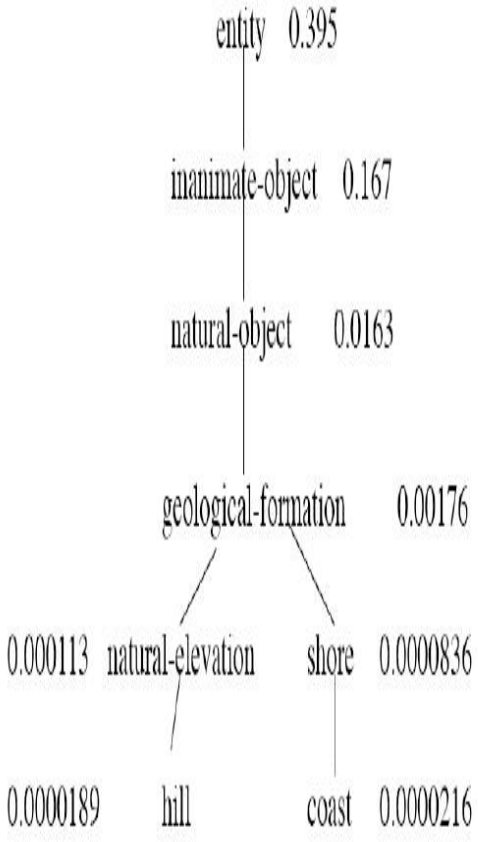
Information content Metriken

Wahrscheinlichkeit eines Konzepts

P(c): Wahrscheinlichkeit, dass ein zufällig ausgewähltes Wort in einem Korpus dem Konzept *c* angehört. Zufallsvariable per Konzept.

- Für jedes Konzept *c*: ein Wort ist entweder Element des Konzepts mit Wahrscheinlichkeit $P(c)$ oder nicht $(1 - P(c))$
- $P(\mathit{Wurzelknoten}) = 1$
- Je niedriger in de Hierarchie, desto niedriger $P(c)$

Berechnung von $P(c)$



- Gehe durch Korpus: jede Instanz von *hill* erhhöht Häufigkeit von *hill, natural elevation, ...*
- $P(c) = \frac{\sum_{w \in words(c)} \#w}{N}$
- N Korpuslänge, $words(c)$ alle Worte in allen Kinderknoten von c

Grafik aus Lin(1998): An information-theoretic definition of similarity. ICML 1998

Information content: Definitionen

Information content

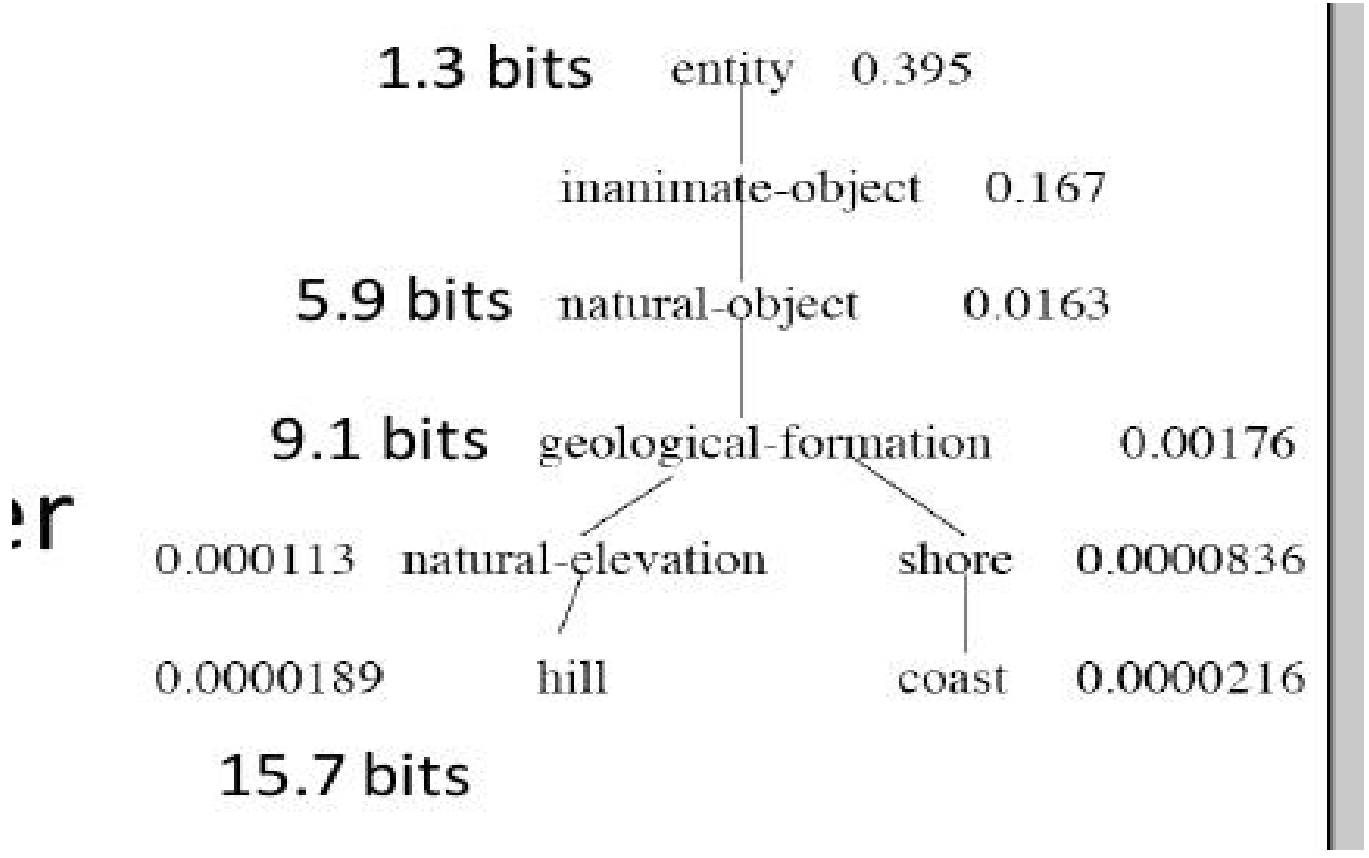
Information content wird mit $-log_2 p(x)$ gemessen und die Einheit ist Bit. Für $p(x) = 0$, ist Information content als 0 definiert

Eigenschaften:

- Additiv für unabhängige Ereignisse (da Wahrscheinlichkeit in dem Fall multiplikativ)
- Je wahrscheinlicher, desto geringer die Information

Information content am Beispiel

$$IC(c) = -log_2 P(c)$$



Information content für Wortähnlichkeit

P. Resnik (1999). Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. JAIR 11, 95-130.

$$sim_{resnik}(c_1, c_2) = -log_2 P(LCS(c_1, c_2)$$

- LCS: least common subsumer von c_1, c_2
- Wortähnlichkeit hängt von Gemeinsamkeiten ab
- Vermeidet dadurch Probleme, die beim reinen Kantenzählen entstanden
- Es gibt Erweiterungen wie Lin(1998): An information-theoretic definition of similarity. ICML 1998

Zwischenfazit

- Wortähnlichkeit kann durch WordNethierarchie berechnet werden
- Hierarchie: Einfache Pfadlängenberechnung leidet unter unterschiedlicher Körnigkeit in verschiedenen Hierarchieteilen
- Hierarchie: Information content kann dies beheben, ist aber korpusabhängig (für P(C) Berechnung)
- Algorithmen eigentlich für sense-Ähnlichkeit: muss meist durch Maximierung über alle senses zweier Worte auf Worte übertragen werden
- **Brauchen sehr viel manuelle Vorbereitung: Wordneterstellung**