Keywords und Kollokate

Statistik für Korpuslinguisten
Sommersemester 22

Philipp Heinrich

Lehrstuhl für Korpus- und Computerlinguistik
Friedrich-Alexander-Universität Erlangen-Nürnberg
philipp.heinrich@fau.de

Erlangen, 31.05.12022

Friedrich-Alexander-Universität
Philosophische Fakultät und
Fachbereich Theologie

Lehrstuhl für
Korpus- und
Computerlinguistik

Keywords

- *Keywords* sind Wörter, die in einem gegebenen Korpus überdurchschnittlich oft vorkommen – im Vgl. zur Häufigkeit in einem *Referenzkorpus*
- *Assoziationsmaße* quantifizieren den Vergleich mittels einer einzelnen reellen Zahl, basierend auf Intuition und/oder statistischen Verfahren
- *Keyness* ist ein textuelles, kein sprachliches Feature
  - gesprochene Sprache vs. geschriebene Sprache
  - soziale Medien vs. Zeitungen
  - Hochliteratur vs. Groschenromane
  - links-liberale Zeitungen vs. rechts-konservative Zeitungen
  - Grüne vs. AfD
  - ...
- Anwendung bspw. in der *Diskursanalyse*, *Indexerstellung*, ...

Keywords in CQPweb

Keyword list for whole "Corpus of German Reddit Exchanges 2010-2018 (GeRedE v1)" compared to your subcorpus "SZ" from corpus "German News (2011-2014)";
using Log Ratio (with 0.01% significance filter, adjusted LL threshold = 43.94);
items must have minimum frequency 3 in list #1 and 3 in list #2.

|< << >> Download whole list Go!

No.	Word	In whole "Corpus of German Reddit Exchanges 2010-2018 (GeRedE v1)":		In your subcorpus "SZ" from corpus "German News (2011-2014)":		+/-	Log Ratio	Log likelihood
		Frequency (absolute)	Frequency (per mill)	Frequency (absolute)	Frequency (per mill)			
1	;)	82,385	303.23	3	0.03	+	13.53	59070.99
2	Edit	96,549	355.36	4	0.03	+	13.35	69219.59
3	dapd	3	0.01	7,906	67.41	-	-12.58	18907.44
4	Englewood	3	0.01	3,069	26.17	-	-11.21	7313.93
5	SUBREDDIT	20,861	76.78	4	0.03	+	11.14	14904.49
6	Hultschiner	4	0.01	3,808	32.47	-	-11.11	9071.47
7	schonmal	19,642	72.30	4	0.03	+	11.05	14030.11
8	garnicht	14,512	53.41	3	0.03	+	11.03	10365.06
9	gibts	48,153	177.23	10	0.09	+	11.02	34393.95
10	:)	128,189	471.82	31	0.26	+	10.8	91509.08
11	nichtmal	15,653	57.61	4	0.03	+	10.72	11169.06
12	Losnummer	3	0.01	1,992	16.99	-	-10.59	4733.89
13	VOrallem	10,500	38.65	3	0.03	+	10.56	7487.68
14	schelsse	12,248	45.08	4	0.03	+	10.37	8727.32
15	Budi	9,164	33.73	3	0.03	+	10.36	6529.68
16	Gewinnklasse	7	0.03	3,952	33.70	-	-10.35	9379.09

Kollokate

- *Kollokate* eines Wortes (dem *Knoten*, oder engl. *node*) sind Wörter, die häufig in dessen Umgebung auftreten (Ko-Okkurrenz)

- Einblick in die Semantik des Wortes, vgl. Firth's (1957) *distributional hypothesis*:

  You shall know a word by the company it keeps!

- hier: Kollokation als Phänomen, das in Korpora empirisch beobachtbar ist
  - Kollokate von „Atomkraft“ im GermaParl
  - Kollokate von „Impfung“ auf Twitter

- Anwendung bspw. in der *Diskursanalyse*, *Lexikographie*, ...

- Kollokate $\neq$ Mehrworteinheiten, Idiome, Phraseologismen, ...

Kollokate von *bucket* (noun)

noun	*f*
water	183
spade	31
plastic	36
slop	14
size	41
mop	16
record	38
bucket	18
ice	22
seat	20
coal	16
density	11
brigade	10
algorithm	9
shovel	7
container	10
oats	7

verb	*f*
throw	36
fill	29
randomize	9
empty	14
tip	10
kick	12
hold	31
carry	26
put	36
chuck	7
weep	7
pour	9
douse	4
fetch	7
store	7
drop	9
pick	11

adjective	*f*
large	37
single-record	5
cold	13
galvanized	4
ten-record	3
full	20
empty	9
steaming	4
full-track	2
multi-record	2
small	21
leaky	3
bottomless	3
galvanised	3
iced	3
clean	7
wooden	6

fensterbasierte Kollokate in CQPweb

Collocation controls

Collocation based on:	Lemma (TreeTagger)	Statistic:	Log Ratio (filtered)
Collocation window *from*:	3 to the Left	Collocation window *to*:	3 to the Right
Freq(node, collocate) at least:	5	Freq(collocate) at least:	5
Filter results by:	specific collocate:	and/or tag: (none)	Submit changed parameters Go!

Extra information:

The **Log Ratio** statistic is a measurement of *how big the difference is* between the (relative) frequency of the collocate alongside the node, and its (relative) frequency in the rest of the corpus or subcorpus.

In **the current collocation analysis**, all collocates displayed have Log-likelihood of at least **14.80404**.

The use of a log-likelihood filter means that it is not necessary to set high minimum values for *Freq(node, collocate)* and *Freq(collocate)* when using Log Ratio.

There are 703 different lemma (treetagger)s in the collocation database for this query (Query "Atomkraft" returned 306 matches in 176 different texts)

[0.349 seconds - retrieved from cache]

No.	Lemma (TreeTagger)	Total no. in whole corpus	Expected collocate frequency	Observed collocate frequency	In no. of texts	Log Ratio (filtered)
1	Ausstieg	587	0.055	34	29	9.367
2	raus	245	0.023	14	8	9.346
3	aussteigen	360	0.034	18	17	9.143
4	Kohle	343	0.032	7	7	7.806
5	Nutzung	1,282	0.119	24	20	7.679
6	billig	695	0.065	7	4	6.772
7	Energieversorgung	603	0.056	5	5	6.489
8	friedlich	971	0.09	5	4	5.797
9	rein	1,520	0.142	7	5	5.635
10	zurück	2,406	0.224	5	5	4.483
11	aus	46,045	4.287	87	61	4.346
12	Risiko	2,663	0.248	5	4	4.337

textuelle Kookkurrenz

textual cooccurrence / segment-based cooccurrence

A vast deal of coolness and a peculiar degree of judgement, are requisite in catching a hat.	hat	—
A man must not be precipitate, or he runs *over* it ;	—	over
he must not rush into the opposite extreme, or he loses it altogether.	—	—
There was a fine gentle wind, and Mr. Pickwick's hat rolled sportively before it.	hat	—
The wind puffed, and Mr. Pickwick puffed, and the hat rolled *over* and *over* as merrily as a lively porpoise in a strong tide ;	hat	over

textuelle Kookkurrenz (Satzfenster)

	$w_2 \in S$	$w_2 \notin S$	
$w_1 \in S$	O_{11}	O_{12}	$= f_1$
$w_1 \notin S$	O_{21}	O_{22}	
	$= f_2$		$= N$

	over $\in S$	over $\notin S$	
hat $\in S$	1	2	$= 3$
hat $\notin S$	1	1	
	$= 2$		$= 5$

Oberflächenkookkurrenz

surface cooccurrence / distance-based cooccurrence

fensterbasiert, abgeschnitten an entsprechenden Grenzen

A vast deal of coolness and a peculiar degree of judgement, are ¡requisite in catching a **hat**¡. A man must not be precipitate, or he runs over it ; he must not rush into the opposite extreme, or he loses it altogether. [. . .] There was a fine gentle ¡wind, and Mr. Pickwick's **hat** *rolled* sportively before it¡. The wind puffed, and Mr. ¡Pickwick puffed, and the **hat** *rolled* over and over¡ as merrily as a lively porpoise in a strong tide ; and on it might have *rolled*, far beyond Mr. Pickwick's reach, had not its course been providentially stopped, just as that gentleman was on the point of resigning it to its fate.

Oberflächenkookkurrenz (L4, R4)

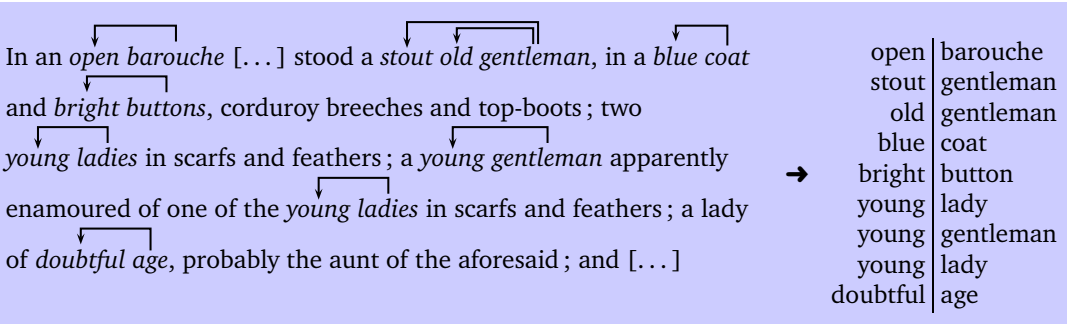
	w_2	$\neg w_2$	
$near(w_1)$	O_{11}	O_{12}	$\approx k \cdot f_1$
$\neg\, near(w_1)$	O_{21}	O_{22}	
	$= f_2$	$= N - f_1$	

	roll	¬roll	
$near$(hat)	2	18	= 20
$\neg\, near$(hat)	1	87	
	= 3	= 108	

syntaktische Kookkurrenz

syntactic cooccurrence / relational cooccurrence

Ausnutzung syntaktischer Strukturen



syntaktische Kookkurrenz

	$*\|w_2$	$*\|\neg w_2$	
$w_1\|*$	O_{11}	O_{12}	$= f_1$
$\neg w_1\|*$	O_{21}	O_{22}	
	$= f_2$		$= N$

	$*$|gent.	$*\|\neg$gent.	
young$\|*$	1	2	$= 3$
$\neg$young$\|*$	2	4	
	$= 3$		$= 9$

Kontingenztabelle (beobachtete Häufigkeiten)

	word	other words	
corpus_1	$O := O_{11}$	O_{12}	$= R_1$
corpus_2	O_{21}	O_{22}	$= R_2$
	$= C_1$	$= C_2$	$= N$

Indifferenztabelle (erwartete Häufigkeiten *bei Unabhängigkeit*)

	word	other words	
corpus_1	$E := E_{11} = \frac{R_1 C_1}{N}$	$E_{12} = \frac{R_1 C_2}{N}$	$= R_1$
corpus_2	$E_{21} = \frac{R_2 C_1}{N}$	$E_{22} = \frac{R_2 C_2}{N}$	$= R_2$
	$= C_1$	$= C_2$	$= N$

Assoziationsmaße

Quantifikation der Abweichung:

	word	other words	
corpus_1	O vs. E	O_{12} vs. E_{12}	$= R_1$
corpus_2	O_{21} vs. E_{21}	O_{22} vs. E_{22}	$= R_2$
	$= C_1$	$= C_2$	$= N$

- log-ratio $= \log \frac{O_{11}/R_1}{O_{21}/R_2}$
- MI $= \log_2 O/E$
- t-score $= \frac{O-E}{\sqrt{O}}$
- $LLR = 2 \sum_{ij} O_{ij} \log \frac{O_{ij}}{E_{ij}}$
- $\chi^2 = \sum_{ij} \frac{(O_{ij}-E_{ij})^2}{E_{ij}}$
- ...

Software

- Berechnung der Assoziationsmaße unkompliziert
  - R: einfache Datensatzmanipulation
  - Python: association-measures
  - CLI (Perl): UCS toolkit
- korrektes und effizientes Zählen am besten nach Korpusindexierung in CWB
  - R: PolmineR
  - Python: cwb-ccc
  - GUI (PHP): CQPweb