DiaCollo

Bryan Jurish
jurish@bbaw.de

University of Birmingham
28th June, 2016

Overview

The Situation

- Diachronic Text Corpora
- Collocation Profiling
- Diachronic Collocation Profiling

DiaCollo

- Requests & Parameters
- Profile, Diffs & Indices

Gory Details

- Corpus Indexing
- Co-occurrence Relations
- Scoring & Comparison Functions

Examples

Summary & Conclusion

The Situation: Diachronic Text Corpora

- heterogeneous text collections, especially with respect to ***date of origin***
  - other partitionings potentially relevant too, e.g. by author, text class, etc.
- increasing number available for linguistic & humanities research, e.g.
  - *Deutsches Textarchiv (DTA)* *(Geyken et al. 2011)*
  - *Referenzkorpus Altdeutsch (DDD)* *(Richling 2011)*
  - Corpus of Historical American English (COHA) *(Davies 2012)*
- . . . but even putatively "synchronic" corpora have a temporal extension, e.g.
  - DWDS/ZEIT ("Kohl") *(1946–2015)*
  - DDR Presseportal ("Ausreise") *(1945–1993)*
  - DWDS/Blogs ("Browser") *(1994–2014)*
- should expose temporal effects of e.g. ***semantic shift***, ***discourse trends***
- problematic for conventional natural language processing tools
  - implicit assumptions of **homogeneity**

The Situation: Collocation Profiling

> *"You shall know a word by the company it keeps"*
> — J. R. Firth

Basic Idea

(Church & Hanks 1990; Manning & Schütze 1999; Evert 2005)

- **lookup** all candidate collocates (w_2) occurring with the target term (w_1)
- **rank** candidates by association score
  - "chance" co-occurrences with high-frequency items must be **filtered out**!
  - statistical methods require **large data sample**

What for?

- computational lexicography *(Kilgarriff & Tugwell 2002; Didakowski & Geyken 2013)*
- neologism detection *(Kilgarriff et al. 2015)*
- distributional semantics *(Schütze 1992; Sahlgren 2006)*
- "text mining" / "distant reading" *(Heyer et al. 2006; Moretti 2013)*

Diachronic Collocation Profiling

The Problem: (temporal) heterogeneity

- conventional collocation extractors assume **corpus homogeneity**
- co-occurrence frequencies are computed only for **word-pairs** (w_1, w_2)
- influence of **occurrence date** (and other document properties) is irrevocably lost

A Solution (sketch)

- represent terms as n-tuples of independent attributes, **including occurrence date**
  - alternative: "document" level co-occurrences over sparse TDF matrix
- partition corpus **on-the-fly** into **user-specified intervals** ("date slices", "epochs")
- collect independent slice-wise profiles into final result set

Advantages

- full support for diachronic axis
- variable query-level granularity
- flexible attribute selection
- multiple association scores

Drawbacks

- sparse data requires larger corpora
- computationally expensive
- large index size
- no syntactic relations (yet)

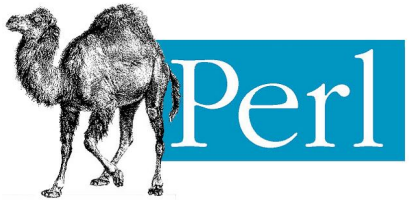
DiaCollo: Overview

General Background

- developed to aid CLARIN historians in analyzing discourse topic trends
- successfully applied to mid-sized and large corpora, including:
  - J. G. Dingler's *Polytechnisches Journal* (1820–1931, 19K documents, 35M tokens)
  - *Deutsches Textarchiv* (1600–1900, 2.6K documents, 173M tokens)
  - *DDR-Presseportal* (1946–1993, 3M documents, 942M tokens)
  - *DWDS Zeitungen* (1946–2015, 10M documents, 4.3G tokens)

Implementation

- Perl API, command-line, & RESTful DDC/D* **web-service plugin** + GUI
- fast native indices over n-tuple inventories, equivalence classes, etc.
- **scalable** even in a high-load environment
  - no persistent server process is required
  - native index access via direct file I/O or `mmap()` system call



- various output & visualization formats, e.g. TSV, JSON , HTML, d3-cloud

DiaCollo: Requests & Parameters

- request-oriented RESTful service *(Fielding 2000)*
- accepts user requests as set of *parameter=value* pairs
- parameter passing via URL query string or HTTP POST request
- common parameters:

Parameter	Description
`query`	target lemma(ta), regular expression, or DDC query
`date`	target date(s), interval, or regular expression
`slice`	aggregation granularity or "0" (zero) for a global profile
`groupby`	aggregation attributes with optional restrictions
`score`	score function for collocate ranking
`kbest`	maximum number of items to return per date-slice
`diff`	score aggregation function for diff profiles
`global`	request global profile pruning (vs. default slice-local pruning)
`profile`	profile type to be computed ({native,tdf,ddc} × {unary,diff})
`format`	output format or visualization mode

DiaCollo: Profiles, Diffs & Indices

Profiles & Diffs

- simple request → unary **profile** for target term(s) *(profile, query)*
  - **filtered** & **projected** to selected attribute(s) *(groupby)*
  - **trimmed** to k-best collocates for target word(s) *(score, kbest, global)*
  - **aggregated** into independent slice-wise sub-intervals *(date, slice)*
- diff request → **comparison** of two independent targets *(profile, bquery, ...)*
  - highlights **differences** or **similarities** of target queries *(diff)*
  - can be used to compare different words *(query ≠ bquery)*
    ...or different corpus subsets w.r.t. a given word *(e.g. date ≠ bdate)*

Indices & Attributes

- compile-time filtering of native indices: frequency threshholds, PoS-tags
- default index attributes: *Lemma (l)*, *Pos (p)*
- finer-grained queries possible with TDF or DDC back-ends
- **batteries not included**: corpus preprocessing, analysis, & full-text search index
  - see e.g. Jurish (2003); Geyken & Hanneforth (2006); Jurish et al. (2014), ...

Gory Details

Corpus Indexing

Input Corpus

- abstract input class `DiaColloDB::Document`
  - currently supported sub-classes: DDCTabs, JSON, TCF, TEI
- input corpus must be **pre-tokenized** and **pre-annotated**
  - user-defined token-attribute selection
  - D* project uses attributes `Lemma` and `PoS` ("part-of-speech")
- may include user-defined **break markers**
  - e.g. clause-, sentence-, page-, and/or paragraph-boundaries

Content Filtering

- not all corpus types are "interesting"
  - e.g. closed classes, *hapax legomena*, etc.
- Regular expression & frequency filters used to pre-prune corpus, e.g.
  - `-O wbad=`*REGEX* : surface form blacklist regex
  - `-O pgood=`*REGEX* : PoS whitelist regex
  - `-tfmin=`*FREQ* : minimum global term-tuple frequency
  - `-lfmin=`*FREQ* : minimum global lemma frequency

Native Co-occurrence Relation

("collocations" profile type)

- "co-occurrence" $\rightsquigarrow$ moving window over $d_{\max}$ content tokens
- window never crosses selected break boundaries
- for corpus $C = s_1 \ldots s_{n_C}$ of break-units ("sentences") $s_i = x_{i1} \ldots x_{in_{s_i}}$

$$\mathrm{f}_{12}(w, v) = \sum_{i=1}^{n_C} \sum_{j=1}^{n_{s_i}} \sum_{d=-d_{\max}}^{d_{\max}} \mathbb{1}[d \neq 0 \;\&\; x_{ij} = w \;\&\; x_{i(j+d)} = v]$$

- independent "frequencies" $\mathrm{f}_1(w)$, N computed as marginals:

$$\mathrm{f}_1(w) = \sum_{v \in \mathcal{X}} \mathrm{f}_{12}(w, v)$$
$$N = \sum_{w \in \mathcal{X}} \mathrm{f}_1(w)$$

- date component distinguishes index tuples $x_{ij} \in \mathcal{X} \subseteq (\mathcal{A}^{n_A} \times \mathrm{Date})$
- 2-level index maps "lexical" tuples (-date) to date-dependent frequencies

$$I_{12} : \mathcal{A}^{n_A} \rightarrow (\mathrm{Date} \rightarrow \mathbb{N})$$

- attribute- and epoch-wise aggregation performed ***on-the-fly*** at runtime
- 2-pass lookup strategy required for accurate collocate frequencies f_2

TDF Co-occurrence Relation

("term × document matrix" profile type)

- "co-occurrence" $\rightsquigarrow$ anywhere within the selected break unit ("document")
- for corpus $C = d_1 \ldots d_{n_D}$ of "documents" $d_i = t_{i1} \ldots t_{in_{d_i}}$ with $\mathrm{tdf}(t, d)$ the frequency of term $t \in \mathcal{A}^{n_A}$ in document d:

$$\mathrm{f}_{12}(w, v) = \sum_{i=1}^{n_D} \min\{\mathrm{tdf}(w, d_i), \mathrm{tdf}(v, d_i)\}$$

- occurrence date, bibliographic metadata stored as ***document properties***
- index uses `mmap()` on sparse matrix PDL via PDL::CCS::Nd
- optimized lookup using Harwell-Boeing offset vectors
- coarse index granularity (no proximity constraints)
- supports Boolean query expressions and document metadata attributes

DDC Co-occurrence Relation

("ddc" profile type)

- "co-occurrence" $\rightsquigarrow$ as returned by a ***DDC*** query Q for slice interval I and grouping attributes G:

$$\mathrm{f}_{12}(W,V) = \texttt{COUNT(}Q\texttt{ \#SEP \#BY[date/}I\texttt{,}G\texttt{=2])}$$
$$\mathrm{f}_{1}(W) = \texttt{COUNT(KEYS(}Q\texttt{ \#BY[date/}I\texttt{,}G\texttt{=1]) \#SEP) \#BY[date/}I\texttt{,}G\texttt{=1]}$$
$$\mathrm{f}_{2}(V) = \texttt{COUNT(KEYS(}Q\texttt{ \#BY[date/}I\texttt{,}G\texttt{=2]) \#SEP) \#BY[date/}I\texttt{,}G\texttt{=2]}$$

- query subscripts ("match-IDs") identify collocant (=1) and collocates (=2)
- supports full range of the DDC query language, including:
  - user-specified break collections (e.g. sentence, file, paragraph)
  - break- and token-level Boolean query expressions
  - phrase- and proximity-queries
  - bibliographic metadata filters
  - server-side term expansion pipelines
- requires a running DDC server for the appropriate corpus
- most flexible back-end yet implemented
- comparatively slow (computationally expensive, resource-hungry)

Scoring Functions: Common Definitions

Variable	Description
w_1	target tuple ("collocant") matching the user `query` request
w_2	collocate tuple matching the user `groupby` request
N	total number of co-occurrences in the profile relation
f_{12}	frequency of the collocation pair: $\mathrm{f}_{12}(w_1, w_2)$
f_1	total frequency of the query term in the selected profile type: $\mathrm{f}_1(w_1)$
f_2	total frequency of the collocate term the selected profile type: $\mathrm{f}_2(w_2)$
ε	smoothing constant, by default $\frac{1}{2}$

- slice-local profiles $p_{s,y}$

$$p_{s,y} : \mathcal{G} \to \mathbb{R} : w_2 \mapsto \mathrm{score}_s(w_1, w_2)$$

- trimmed by default to k-best (`kbest`) collocates for independently by slice

$$\hat{p}_{s,y} = p_{s,y} \restriction \arg\max^{(k)}_{w_2} p_{s,y}(w_2)$$

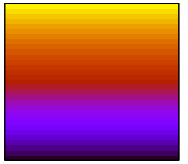
- "global" multi-profiles use a shared restriction set for all slices:

$$p_{s,*}(w_2) = \sum_{y \in Y} p_{s,y}(w_2)$$
$$\hat{p}_{s,y} = p_{s,y} \restriction \arg\max^{(k)}_{w_2} p_{s,*}(w_2)$$

Scoring Functions: f (raw frequency)

Variable	Description
w_1	target tuple ("collocant") matching the user `query` request
w_2	collocate tuple matching the user `groupby` request
N	total number of co-occurrences in the profile relation
f_{12}	frequency of the collocation pair: $\mathrm{f}_{12}(w_1, w_2)$
f_1	total frequency of the query term in the selected profile type: $\mathrm{f}_1(w_1)$
f_2	total frequency of the collocate term the selected profile type: $\mathrm{f}_2(w_2)$
ε	smoothing constant, by default $\frac{1}{2}$

$$\mathrm{score}_{\mathrm{f}}(w_1, w_2) = f_{12}$$

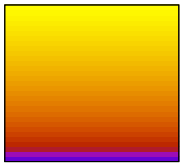


- immediately interpretable, but not very robust
- Zipf distribution leads to "lopsided" visualizations
- values may not comparable across slices (e.g. for non-balanced corpora)
- many false positives with high-frequency collocates
- **not** generally a good measure of collocate affinity

Scoring Functions: lf (log frequency)

Variable	Description
w_1	target tuple ("collocant") matching the user `query` request
w_2	collocate tuple matching the user `groupby` request
N	total number of co-occurrences in the profile relation
f_{12}	frequency of the collocation pair: $\mathrm{f}_{12}(w_1, w_2)$
f_1	total frequency of the query term in the selected profile type: $\mathrm{f}_1(w_1)$
f_2	total frequency of the collocate term the selected profile type: $\mathrm{f}_2(w_2)$
ε	smoothing constant, by default $\frac{1}{2}$

$$\mathrm{score}_{\mathrm{lf}}(w_1, w_2) = \log_2(f_{12} + \varepsilon)$$

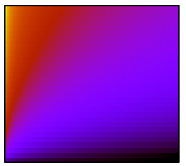


- better visual scaling than raw frequency
- otherwise shares raw frequency's shortcomings

Scoring Functions: mi (pointwise MI × log-frequency)

Variable	Description
w_1	target tuple ("collocant") matching the user `query` request
w_2	collocate tuple matching the user `groupby` request
N	total number of co-occurrences in the profile relation
f_{12}	frequency of the collocation pair: $\mathrm{f}_{12}(w_1, w_2)$
f_1	total frequency of the query term in the selected profile type: $\mathrm{f}_1(w_1)$
f_2	total frequency of the collocate term the selected profile type: $\mathrm{f}_2(w_2)$
ε	smoothing constant, by default $\frac{1}{2}$

$$\mathrm{score}_{\mathrm{mi}}(w_1, w_2) = \log_2 \frac{(f_{12}+\varepsilon)\times(N+\varepsilon)}{(f_1+\varepsilon)\times(f_2+\varepsilon)} \times \log_2(f_{12}+\varepsilon)$$

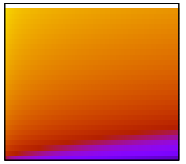


- used by first version of Sketch Engine *(Kilgarriff et al. 2004)*
- PMI gives code-length change for (optimal) joint vs. independent encodings
- PMI alone is very sensitive to low-frequency items ($\rightsquigarrow$ longer codes)
  - *post-hoc* workaround: include log-frequency coefficient
- some preference for low-frequency collocates remains

Scoring Functions: II (log-likelihood)

Variable	Description
w_1	target tuple ("collocant") matching the user `query` request
w_2	collocate tuple matching the user `groupby` request
N	total number of co-occurrences in the profile relation
f_{12}	frequency of the collocation pair: $\mathrm{f}_{12}(w_1, w_2)$
f_1	total frequency of the query term in the selected profile type: $\mathrm{f}_1(w_1)$
f_2	total frequency of the collocate term the selected profile type: $\mathrm{f}_2(w_2)$
ε	smoothing constant, by default $\frac{1}{2}$

$$\mathrm{score}_{\mathrm{ll}}(w_1, w_2) = \mathrm{sgn}(f_{12}|f_1, f_2) \times \log(1 + \log \lambda)$$

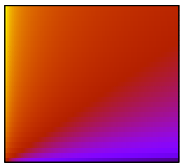


- 1-sided variant of the binomial log likelihood ratio *(Dunning 1993; Evert 2008)*
  - only "attracting" collocate pairs are assigned positive values
- null hypothesis filters out "uninteresting" high-frequency collocates
- very sensitive to fixed & formulaic expressions $\rightsquigarrow$ ***poor visual scaling***
  - workaround: report & scale using $\log(1 + \log \lambda)$ rather than "pure" $\log \lambda$

Scoring Functions: ld (log-Dice coefficient)

Variable	Description
w_1	target tuple ("collocant") matching the user `query` request
w_2	collocate tuple matching the user `groupby` request
N	total number of co-occurrences in the profile relation
f_{12}	frequency of the collocation pair: $\mathrm{f}_{12}(w_1, w_2)$
f_1	total frequency of the query term in the selected profile type: $\mathrm{f}_1(w_1)$
f_2	total frequency of the collocate term the selected profile type: $\mathrm{f}_2(w_2)$
ε	smoothing constant, by default $\frac{1}{2}$

$$\mathrm{score}_{\mathrm{ld}}(w_1, w_2) = 14 + \log_2 \frac{2(f_{12}+\varepsilon)}{(f_1+\varepsilon)+(f_2+\varepsilon)}$$



- "lexicographer-friendly" association score *(Rychlý 2008)*
- less susceptible to low-frequency outliers than PMI × log-frequency product
- good filtering of "uninteresting" high-frequency collocates
- "intuitive" visual scaling (consistent with human perceptual givens)
- default score used by DiaCollo

Diff Operations: Common Definitions

Variable	Description
q_{a}	1st profile query (`query`, `date`, `slice`)
q_{b}	2nd profile query (`bquery`, `bdate`, `bslice`)
p_{a}	1st profile function $\mathrm{profile}(q_{\mathrm{a}}) : \mathcal{G} \to \mathbb{R} : w_2 \mapsto \mathrm{score}_{\mathrm{a}}(w_{1\mathrm{a}}, w_2)$
p_{b}	2nd profile function $\mathrm{profile}(q_{\mathrm{b}}) : \mathcal{G} \to \mathbb{R} : w_2 \mapsto \mathrm{score}_{\mathrm{b}}(w_{1\mathrm{b}}, w_2)$
s_{a}	1st score value operand given collocate w_2: $s_{\mathrm{a}} = p_{\mathrm{a}}(w_2)$
s_{b}	2nd score value operand given collocate w_2: $s_{\mathrm{b}} = p_{\mathrm{b}}(w_2)$

- comparison scores diff_d computed for independent slice profiles p_{a}, p_{b}:

$$\mathrm{diff}_d(p_{\mathrm{a}}, p_{\mathrm{b}}) : \mathcal{G} \to \mathbb{R} : w_2 \mapsto p_{\mathrm{a}}(w_2) \ominus_d p_{\mathrm{b}}(w_2)$$

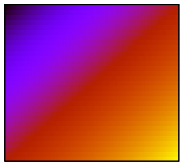
- various diff operations d act on only selected domain subsets:
  - ***pre-trimmed*** operations $\mathrm{dom}(\hat{p}_{\mathrm{a}}) \cup \mathrm{dom}(\hat{p}_{\mathrm{b}})$
  - ***restricted*** operations $\mathrm{dom}(p_{\mathrm{a}}) \cap \mathrm{dom}(p_{\mathrm{b}})$
  - ***untrimmed*** operations $\mathrm{dom}(p_{\mathrm{a}}) \cup \mathrm{dom}(p_{\mathrm{b}})$
- k-best collocates are selected by maximum diff score:

$$p_{\mathrm{a}\ominus_d\mathrm{b}} : \mathcal{G} \to \mathbb{R} : w_2 \mapsto \mathrm{diff}_d(p_{\mathrm{a}}, p_{\mathrm{b}})$$

Diff Operations: diff (raw difference)

Variable	Description
q_a	1st profile query (`query`, `date`, `slice`)
q_b	2nd profile query (`bquery`, `bdate`, `bslice`)
p_a	1st profile function $\mathrm{profile}(q_\mathrm{a}) : \mathcal{G} \to \mathbb{R} : w_2 \mapsto \mathrm{score}_\mathrm{a}(w_{1\mathrm{a}}, w_2)$
p_b	2nd profile function $\mathrm{profile}(q_\mathrm{b}) : \mathcal{G} \to \mathbb{R} : w_2 \mapsto \mathrm{score}_\mathrm{b}(w_{1\mathrm{b}}, w_2)$
s_a	1st score value operand given collocate w_2: $s_\mathrm{a} = p_\mathrm{a}(w_2)$
s_b	2nd score value operand given collocate w_2: $s_\mathrm{b} = p_\mathrm{b}(w_2)$

$$s_\mathrm{a} \ominus_\mathrm{diff} s_\mathrm{b} := s_\mathrm{a} - s_\mathrm{b}$$

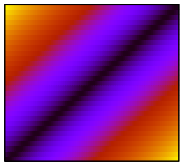


- pre-trimmed
- asymmetric
- selects collocates strongly associated only with q_a

Diff Operations: adiff (absolute difference)

Variable	Description
q_a	1st profile query (`query`, `date`, `slice`)
q_b	2nd profile query (`bquery`, `bdate`, `bslice`)
p_a	1st profile function $\mathrm{profile}(q_\mathrm{a}) : \mathcal{G} \to \mathbb{R} : w_2 \mapsto \mathrm{score}_\mathrm{a}(w_{1\mathrm{a}}, w_2)$
p_b	2nd profile function $\mathrm{profile}(q_\mathrm{b}) : \mathcal{G} \to \mathbb{R} : w_2 \mapsto \mathrm{score}_\mathrm{b}(w_{1\mathrm{b}}, w_2)$
s_a	1st score value operand given collocate w_2: $s_\mathrm{a} = p_\mathrm{a}(w_2)$
s_b	2nd score value operand given collocate w_2: $s_\mathrm{b} = p_\mathrm{b}(w_2)$

$$s_\mathrm{a} \ominus_\mathrm{adiff} s_\mathrm{b} :\approx |s_\mathrm{a} - s_\mathrm{b}|$$

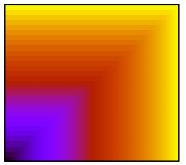


- pre-trimmed
- symmetric
- selects based on $|s_\mathrm{a} - s_\mathrm{b}|$, but reports raw difference $s_\mathrm{a} - s_\mathrm{b}$
- returns most extreme differences among strong collocates of q_a and q_b
- sign of returned score indicates association preference for q_a $(+)$ or q_b $(-)$

Diff Operations: max (maximum)

Variable	Description
q_{a}	1st profile query (`query`, `date`, `slice`)
q_{b}	2nd profile query (`bquery`, `bdate`, `bslice`)
p_{a}	1st profile function $\mathrm{profile}(q_{\mathrm{a}}) : \mathcal{G} \to \mathbb{R} : w_2 \mapsto \mathrm{score}_{\mathrm{a}}(w_{1\mathrm{a}}, w_2)$
p_{b}	2nd profile function $\mathrm{profile}(q_{\mathrm{b}}) : \mathcal{G} \to \mathbb{R} : w_2 \mapsto \mathrm{score}_{\mathrm{b}}(w_{1\mathrm{b}}, w_2)$
s_{a}	1st score value operand given collocate w_2: $s_{\mathrm{a}} = p_{\mathrm{a}}(w_2)$
s_{b}	2nd score value operand given collocate w_2: $s_{\mathrm{b}} = p_{\mathrm{b}}(w_2)$

$$s_{\mathrm{a}} \ominus_{\max} s_{\mathrm{b}} := \max\{s_{\mathrm{a}}, s_{\mathrm{b}}\}$$

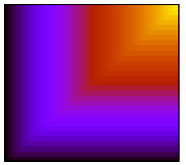


- pre-trimmed
- symmetric
- selects only stronger of the operand association scores
- potentially useful for discovering collocates deserving further investigation

Diff Operations: min (minimum)

Variable	Description
q_{a}	1st profile query (`query`, `date`, `slice`)
q_{b}	2nd profile query (`bquery`, `bdate`, `bslice`)
p_{a}	1st profile function $\mathrm{profile}(q_{\mathrm{a}}) : \mathcal{G} \to \mathbb{R} : w_2 \mapsto \mathrm{score}_{\mathrm{a}}(w_{1\mathrm{a}}, w_2)$
p_{b}	2nd profile function $\mathrm{profile}(q_{\mathrm{b}}) : \mathcal{G} \to \mathbb{R} : w_2 \mapsto \mathrm{score}_{\mathrm{b}}(w_{1\mathrm{b}}, w_2)$
s_{a}	1st score value operand given collocate w_2: $s_{\mathrm{a}} = p_{\mathrm{a}}(w_2)$
s_{b}	2nd score value operand given collocate w_2: $s_{\mathrm{b}} = p_{\mathrm{b}}(w_2)$

$$s_{\mathrm{a}} \ominus_{\min} s_{\mathrm{b}} := \min\{s_{\mathrm{a}}, s_{\mathrm{b}}\}$$

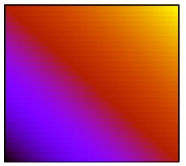


- restricted
- symmetric
- selects only weaker of the operand association scores
- high scores indicate similar strong association preferences
- very sensitive to sparse data problems (missing data $\rightsquigarrow$ zeroes)

Diff Operations: avg (arithmetic average)

Variable	Description
q_a	1st profile query (`query`, `date`, `slice`)
q_b	2nd profile query (`bquery`, `bdate`, `bslice`)
p_a	1st profile function $\mathrm{profile}(q_\mathrm{a}) : \mathcal{G} \to \mathbb{R} : w_2 \mapsto \mathrm{score}_\mathrm{a}(w_{1\mathrm{a}}, w_2)$
p_b	2nd profile function $\mathrm{profile}(q_\mathrm{b}) : \mathcal{G} \to \mathbb{R} : w_2 \mapsto \mathrm{score}_\mathrm{b}(w_{1\mathrm{b}}, w_2)$
s_a	1st score value operand given collocate w_2: $s_\mathrm{a} = p_\mathrm{a}(w_2)$
s_b	2nd score value operand given collocate w_2: $s_\mathrm{b} = p_\mathrm{b}(w_2)$

$$s_\mathrm{a} \ominus_\mathrm{avg} s_\mathrm{b} := \frac{s_\mathrm{a} + s_\mathrm{b}}{2}$$

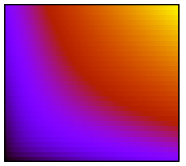


- restricted
- symmetric
- selects strong associations for either q_a or q_b, preferring shared associations
- ***not*** very sensitive to non-uniform operand values
  - high scores do not necessarily indicate similar collocation behavior

Diff Operations: havg (harmonic average)

Variable	Description
q_a	1st profile query (`query`, `date`, `slice`)
q_b	2nd profile query (`bquery`, `bdate`, `bslice`)
p_a	1st profile function $\mathrm{profile}(q_\mathrm{a}) : \mathcal{G} \to \mathbb{R} : w_2 \mapsto \mathrm{score}_\mathrm{a}(w_{1\mathrm{a}}, w_2)$
p_b	2nd profile function $\mathrm{profile}(q_\mathrm{b}) : \mathcal{G} \to \mathbb{R} : w_2 \mapsto \mathrm{score}_\mathrm{b}(w_{1\mathrm{b}}, w_2)$
s_a	1st score value operand given collocate w_2: $s_\mathrm{a} = p_\mathrm{a}(w_2)$
s_b	2nd score value operand given collocate w_2: $s_\mathrm{b} = p_\mathrm{b}(w_2)$

$$s_\mathrm{a} \ominus_\mathrm{havg} s_\mathrm{b} :\approx \frac{2 s_\mathrm{a} s_\mathrm{b}}{s_\mathrm{a} + s_\mathrm{b}}$$



- restricted
- symmetric
- selects uniformly strong associations for both q_a and q_b
- to avoid singularities, actually computed as:

$$\begin{aligned} \mathrm{havg}(s_\mathrm{a}, s_\mathrm{b}) &:= \begin{cases} 0 & \text{if } s_\mathrm{a} \leq 0 \text{ or } s_\mathrm{b} \leq 0 \\ \frac{2 s_\mathrm{a} s_\mathrm{b}}{s_\mathrm{a} + s_\mathrm{b}} & \text{otherwise} \end{cases} \\ s_\mathrm{a} \ominus_\mathrm{havg} s_\mathrm{b} &:= \mathrm{avg}(\mathrm{havg}(s_\mathrm{a}, s_\mathrm{b}), \mathrm{avg}(s_\mathrm{a}, s_\mathrm{b})) \end{aligned}$$

Examples

Example 1: Newsworthy Crises

'Krise' in DIE ZEIT (west) and Neues Deutschland (east)

http://kaskade.dwds.de/dstar/zeit/diacollo/?q=Krise&d=1950:2015&gb=l,p%3DNE

1950–1959
- Berlin blockade aftermath

1960–1969
- anti-government protests & strikes in France

1970–1979
- Nixon & Brandt resignations; Iranian revolution

1980–1989
- *Solidarność* in Poland; Soviet war in Afghanistan; Schmidt coalition collapses

1990–1999
- wars in ex-Yugoslavia, Kosovo & Chechnya; financial crises in Asia & Mexico

2000–2009
- global financial crisis

2010–2014
- civil wars in Syria & the Ukraine; Greek bankruptcy

Compare:
- *Krise*: DDR-PP *Neues Deutschland*: 3-year slices, proper name collocates (NE)
- *Krise*: DDR-PP *Neues Deutschland*: 5-year slices, common noun collocates (NN)

Example 1: Selected Lemma-Clouds

1980–1989:

2010–2014:

Example 2: Lexicography

'autofrei' (automobile-free)

http://kaskade.dwds.de/dstar/zeitungen/diacollo/?q=autofrei&ds=5&f=bub

Lexicography & Collocations

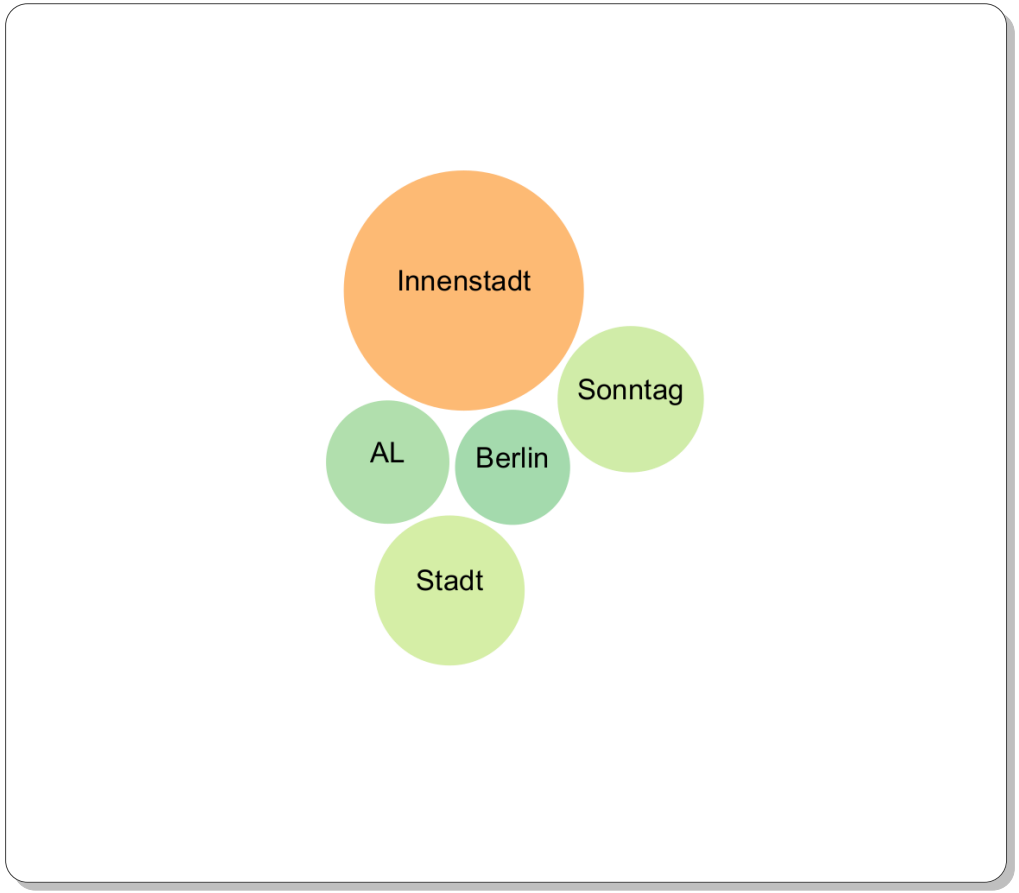
- collocation preferences correlate strongly with word meanings
- new senses ('neosemantemes') $\Rightarrow$ new collocates
  - *Maus* ("mouse"): rodent *vs.* input device
  - *Ampel* ("traffic light"): traffic signal *vs.* political coalition

The case of *autofrei* ("automobile-free")

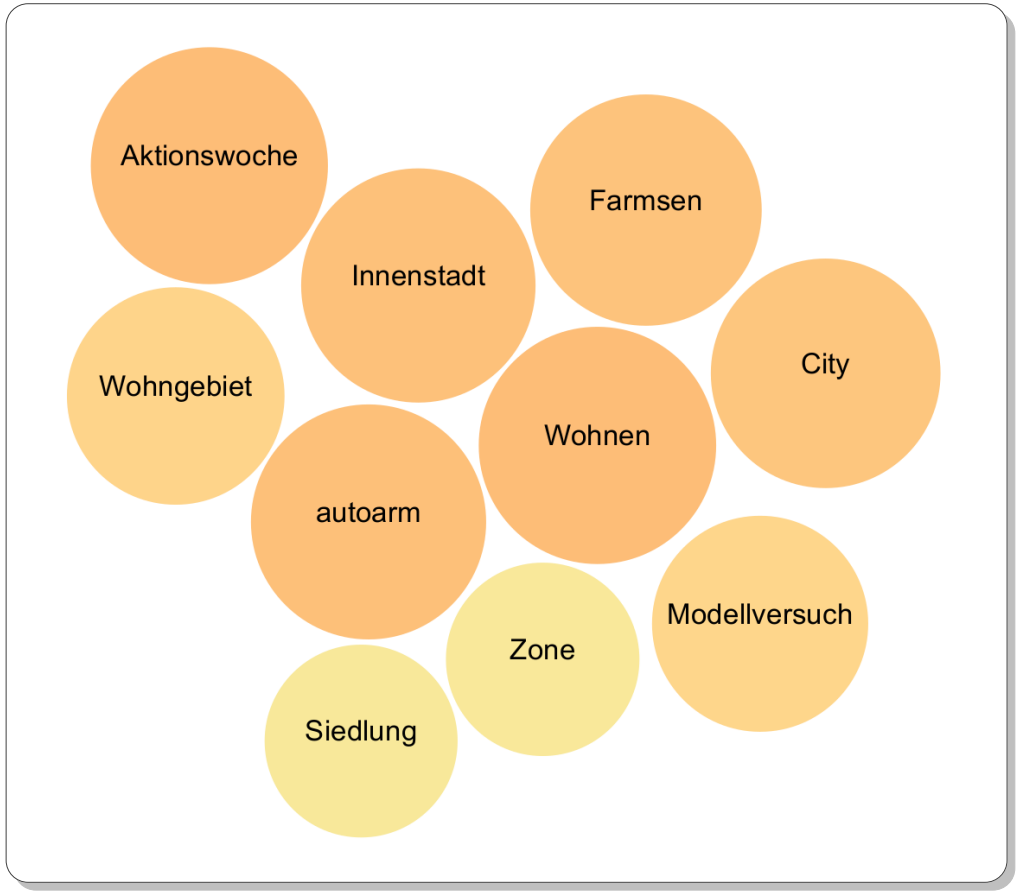
- Duden: *keinen Autoverkehr aufweisend* ("lacking automobile traffic")
- DWDS corpora reveal ***two sub-senses***:
  - **1970–1989**: ... by ordinance ($\rightsquigarrow$ *Sonntag*, *Innenstadt*)
  - **1990–present**: ... voluntary ($\rightsquigarrow$ *Wohnanlage*, *Siedlung*)

Example 2: Selected Bubble-Charts

1985–1989



1990–1994



Example 3: Gender & Cultural Bias

'Mann' vs. 'Frau' in the Deutsches Textarchiv (1600–1900)

http://kaskade.dwds.de/dstar/dta/diacollo/?q=Mann&bq=Frau&d=1600:1899&ds=25&gb=l,p%3DADJA&f=cld&p=d2

Disclaimer

- historical corpus data can reveal persistent cultural biases
- linked collocation data does not reflect the opinions of the author or the BBAW!

Observations

- biological fact: ***schwangere*** *Frau* (only appears 1675–1724)
- fixed & formulaic expressions very prominent
  - ***gnädige*** *Frau* (masculine variant: *gnädiger Herr*)
  - *Frau X* ***geborene*** *Y* (birth- vs. married surname)
  - *der* ***gemeine*** *Mann* (masculine generic)
- pretty much exclusively cultural bias:
  - ***Mann*** $\rightsquigarrow$ *berühmt, ehrlich, gelehrt, tapfer, weise, ...*
  - ***Frau*** $\rightsquigarrow$ *betrübt, lieb, schön, tugendreich, verwitwet, ...*
- differences grow less pronounced in late 18^{th} & 19^{th} centuries

Example 3: Selected Lemma-Clouds

1725–1749:

1825–1849:

Example 4: What Makes a 'Man'?

'[ADJA] Mann' in the Deutsches Textarchiv (1600–2000)

http://kaskade.dwds.de/dstar/dta/diacollo/?profile=diff-ddc&k=25&f=cloud ...

```
QUERY: "*=2 Mann" #has[textClass,Wissenschaft*]
~QUERY: "*=2 Mann" #has[textClass,Belletristik*]
GROUPBY: l,p=ADJA
```

Remarks

- 'diff' profile provides direct comparison of genres ***science*** vs. ***belles lettres***
- uses DDC back-end for fine-grained data acquisition

Differences (`diff=adiff`)

- ***Science*** $\rightsquigarrow$ *berühmt, scharfsinnig, tüchtig* ("famous, astute, capable")
- ***Belles Lettres*** $\rightsquigarrow$ *brav, grau, rechtschaffen* ("well-behaved, gray, righteous")

Similarities (`diff=min`)

- *groß, gelehrt, gemein, jung, alt* ("great, learned, common, young, old")

Example 4: Selected Lemma-Clouds

1700–1799

(`diff=adiff`)

1800–1899

(`diff=adiff`)

Example 5: Genealogy of Terminology

Habermas vs. Cassirer in the DWDS Kernkorpus

http://kaskade.dwds.de/dstar/kern/diacollo/?ds=0&bds=0&k=20&p=diff-tdf&f=cld&diff=adiff

QUERY: * #has[author,/Habermas/]
~QUERY: * #has[author,/Cassirer/]
GROUPBY: l,p=NN

Remarks

- uses TDF (term × document) matrix back-end for bibliographic meta-data queries
- sets `slice=0` parameter to acquire date-independent profiles
- `groupby` clause selects only common noun lemmata (STTS tag `NN`)
- modest sample size (Habermas: 516k tokens, Cassirer: 130k tokens)
- Habermas himself openly acknowledges Cassirer's influence

Differences (`diff=adiff`)

- ***Habermas*** $\rightsquigarrow$ *Handeln, Gesellschaft, Öffentlichkeit, Meinung, Norm, . . .*
- ***Cassirer*** $\rightsquigarrow$ *Anschauung, Bestimmung, Bezeichnung, Erkenntnis, Sein, . . .*

Similarities (`diff=havg`, `diff=min`)

- *Analyse, Ausdruck, Begriff, Beziehung, Funktion, Sinn, Sprache, . . .*

Example 5: Lemma-Clouds

differences

(`diff=adiff`)

similarities

(`diff=havg`)

Example 6: Pronominal Adverbs by Genre

'[PAV]' in aggregated DTA+DWDS (1600–2000)

http://kaskade.dwds.de/dstar/dta+dwds/diacollo/?p=diff-ddc&k=50&f=cld&G=1 ...

QUERY: $p=PAV=2 #has[textClass,**Wissenschaft***]

∼QUERY: $p=PAV=2 #has[textClass,**Belletristik***]

Remarks

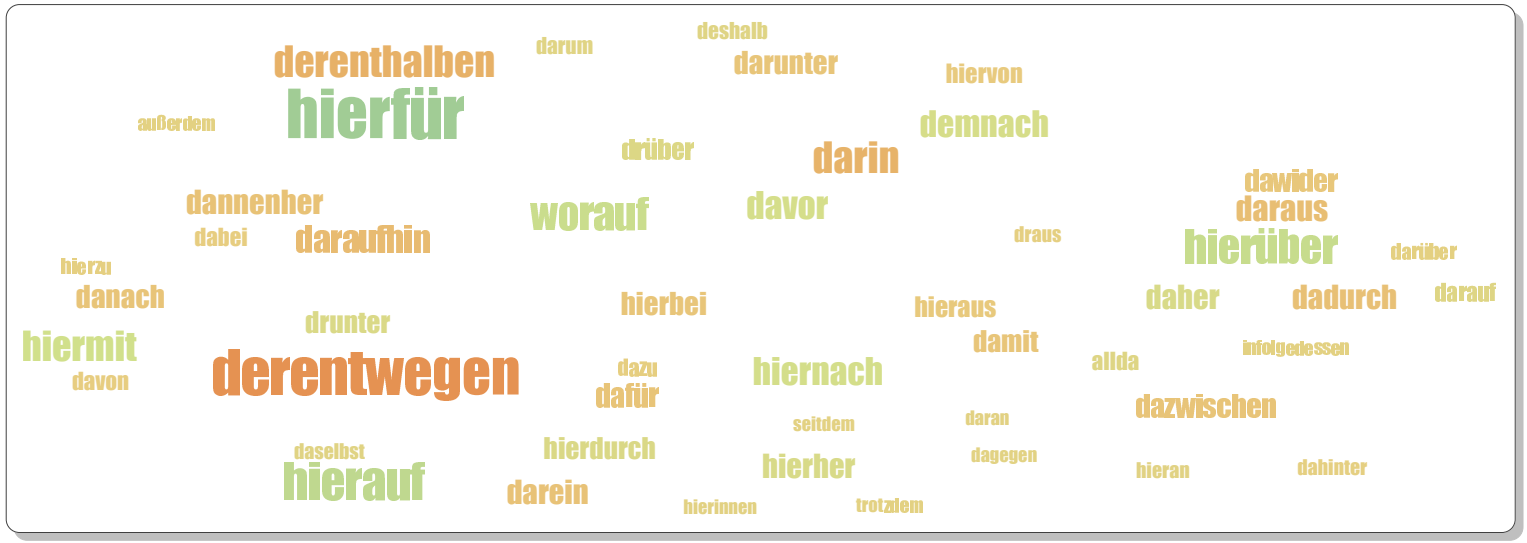
- 'diff' profile provides direct comparison of genres ***science*** vs. ***belles lettres***
- uses DDC back-end for querying functional category

Observations

- divergent: differences grow more pronounced over time
- **Science**
  - *hier-* anaphorics ⇝ *hierbei, hieraus, hierzu* ("hereby, out of which, to which")
  - causal/logical ⇝ *demnach, infolgedessen, daher* ("therefore")
- **Belles Lettres**
  - fixed expression *drunter [und] drüber* ("higgledy-piggeldy, at sixes and sevens")
  - spatial & temporal ⇝ *dahinter, worauf* ("behind which, upon which")
  - concessive & adversative ⇝ *dawider, trotzdem* ("against which, despite which")

Example 6: Selected Lemma-Clouds

1650–1699:



1950–1999:

Example 7: 400 Years of Potables

'[GETRÄNK] trinken' in aggregated DTA+DWDS (1600–2000)

http://kaskade.dwds.de/dstar/dta+dwds/diacollo/?d=1600%3A1999&ds=50&k=20&p=ddc&f=cld&g=l&G=1

QUERY: "(**Getränk**|gn-sub WITH $p=NN)=2 (**trinken** WITH $p=/VV[IP]/)" #FMIN 1

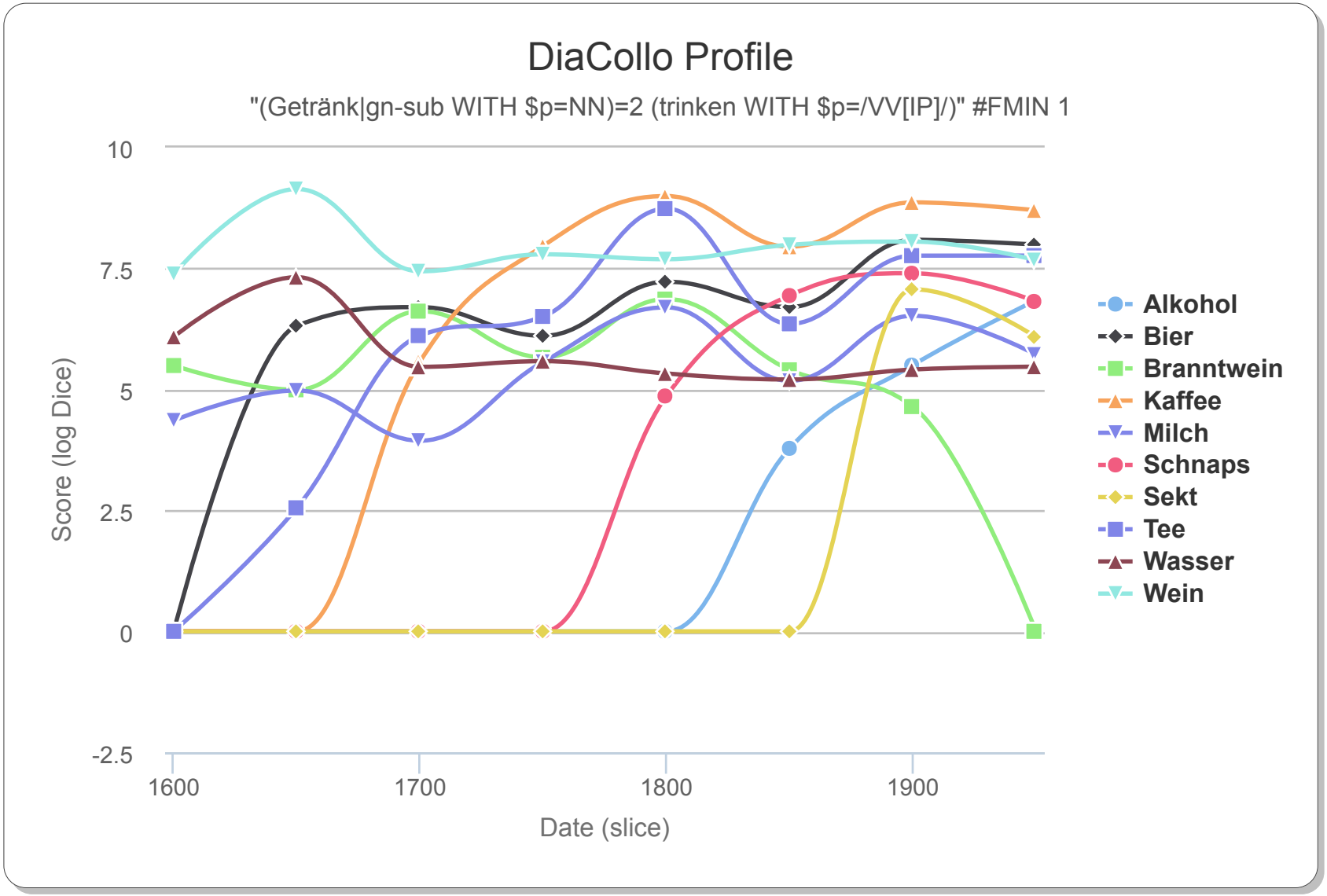
Remarks

- uses DDC back-end for fine-grained data acquisition
- uses GermaNet thesaurus-based lexical expansion for ***Getränk*** ("beverage")
- considers only those target terms immediately preceding verb ***trinken*** ("to drink")
- "global" profile uses shared target-set to avoid visual clutter

Observations

- near-constants: *Bier, Milch, Wasser, Wein* ("beer, milk, water, wine")
- 1650–1750: *Tee, Kaffee, Schokolade* ("tea, coffee, chocolate") appear
- 1800–1900: *Schnaps* displaces *Branntwein*; *Champagner* appears
- 1850–1900: *Alkohol* ("alcohol") as category of beverages
- 1900–2000: *Kognak, Saft, Sekt, Whisky* ("cognac, juice, sparkling wine, whisky")

Example 7: Time Series $(k = 10)$



Summary & Conclusion

Diachronic Collocation Profiling

- diachronic text corpora ⇝ *semantic shift, discourse trends*
- conventional tools ⇝ *implicit assumptions of homogeneity*
- diachronic profiling ⇝ *date-dependent lexemes*

DiaCollo

- on-the-fly corpus partitioning ⇝ *arbitrary query granularity*
- DDC/D* integration ⇝ *fine-grained queries, corpus KWIC links*
- RESTful web service ⇝ *external API, online visualization*

Applications

- exploration & discovery ⇝ *large source collections*
- analysis & investigation ⇝ *data acquisition for hypothesis testing*
- evaluation & assessment ⇝ *historical semantics, history of concepts, &c.*

— The End —

Thank you for listening!

http://kaskade.dwds.de/diacollo

http://metacpan.org/release/DiaColloDB

http://clarin-d.de/de/kollokationsanalyse-in-diachroner-perspektive