Exploring diachronic collocations with DiaCollo

Bryan Jurish
jurish@bbaw.de

Universität Potsdam, Institut für Linguistik
19th June, 2017

http://kaskade.dwds.de/~jurish/diacollo2017
Overview

The Situation
- Diachronic Text Corpora
- Collocation Profiling
- Diachronic Collocation Profiling

DiaCollo
- Requests & Parameters
- Profiles, 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 2013)
  - *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”)
    - (1946–2016)
  - DDR Presseportal (“Ausreise”)
    - (1945–1993)
  - DWDS/Blogs (“Browser”)
    - (1994–2016)

- 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

“Die Bedeutung eines Wortes ist sein Gebrauch in der Sprache”
— L. Wittgenstein

“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)
The Situation: Related Work

Conventional (synchronic) Collocation Profiling

- well understood & widely accepted
  (e.g. Manning & Schütze 1999; Evert 2005)
- can’t handle (temporal) *heterogeneity*!

Diachronic Studies: Manual Corpus Partitioning

- Baker et al. (2008): 10 epochs, 1 year each
- Sagi et al. (2009): 5 epochs, ca. 100 years each
- Gulordava & Baroni (2011): 2 epochs, 10 years each
- Scharloth et al. (2013): 3400 epochs, ca. 1 week each (+smoothing)
- Kim et al. (2014): 160 epochs, 1 year each
- *Gabrielatos et al. (2012): epoch granularity* depends on *research question*!

“Latent” Distributional Approximations

- Sagi et al. (2009): LSA model w.r.t. 2000 most frequent content-bearing collocates
- Kim et al. (2014): series of vector space models à la Mikolov et al. (2013)
- *compile-time parameters, approximate counts ⇒ not viable*!
Epoch Partitioning (input)

- input corpus with documents \{A, B, \ldots, J\} over date range (1950–1999)
Epoch Partitioning (E=10)

- input corpus with documents \{A, B, \ldots, J\} over date range (1950–1999)
- partition by decade \(E = 10\)
Epoch Partitioning (E=10)

- input corpus with documents \{A, B, \ldots, J\} over date range (1950–1999)
- partition by decade \((E = 10)\)
Manual Corpus Partitioning

**Epoch Partitioning (E=10)**

<table>
<thead>
<tr>
<th>Epoch Subcorpora</th>
<th>Epoch Ranges</th>
<th>Epoch Partitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>{A, B}</td>
<td>[1950..1959]</td>
<td>A</td>
</tr>
<tr>
<td>{C, D, E}</td>
<td>[1960..1969]</td>
<td>B, C, D, E</td>
</tr>
<tr>
<td>{F}</td>
<td>[1970..1979]</td>
<td>E</td>
</tr>
</tbody>
</table>

- input corpus with documents \{A, B, \ldots, J\} over date range (1950–1999)
- partition by decade \((E = 10)\)
- collect epoch-wise subcorpora
## Manual Corpus Partitioning

### Epoch Partitioning (E=10)

<table>
<thead>
<tr>
<th>Date</th>
<th>Documents</th>
<th>Epoch Labels</th>
<th>Epoch Subcorpora</th>
<th>Epoch Ranges</th>
<th>Epoch Partitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1950</td>
<td>A B</td>
<td>e=1950</td>
<td>{A, B}</td>
<td>[1950..1959]</td>
<td></td>
</tr>
<tr>
<td>1960</td>
<td>C D E</td>
<td>e=1960</td>
<td>{C, D, E}</td>
<td>[1960..1969]</td>
<td></td>
</tr>
<tr>
<td>1990</td>
<td>I J</td>
<td>e=1990</td>
<td>{I, J}</td>
<td>[1990..1999]</td>
<td></td>
</tr>
</tbody>
</table>

- input corpus with documents \{A, B, \ldots, J\} over date range (1950–1999)
- partition by decade (\(E = 10\))
- collect epoch-wise subcorpora
- label sub-corpora (e.g. by minimum date) and analyze independently
Manual Corpus Partitioning

Epoch Partitioning (E=25)

- input corpus with documents \{A, B, \ldots, J\} over date range (1950–1999)
- partition by decade **quarter-century** \(E = 25\)
- collect epoch-wise subcorpora
- label sub-corpora (e.g. by minimum date) and analyze independently

**Problems:**
- static partitioning \(\rightsquigarrow\) labor-intensive, inflexible, & often inaccessible
- “good” epoch granularity (partition size) depends on research question

**can we generalize this?**
Manual Corpus Partitioning

**Epoch Partitioning (E=25)**

- **Epoch Labels**
- **Epoch Subcorpora**
- **Epoch Ranges**
- **Epoch Partitions**

<table>
<thead>
<tr>
<th>Date</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>1950</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>e=1950</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>{A, B, C, D, E}</td>
<td></td>
<td></td>
<td></td>
<td>[1950..1974]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1960</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1970</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1980</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>e=1975</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>{F, G, H, I, J}</td>
<td></td>
<td></td>
<td></td>
<td>[1975..1999]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1990</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **input corpus with documents \(\{A, B, \ldots, J\}\) over date range (1950–1999)**
- **partition by decade quarter-century \(E = 25\)**
- **collect epoch-wise subcorpora**
- **label sub-corpora (e.g. by minimum date) and analyze independently**
- **Problems:**
  - static partitioning ⇔ labor-intensive, inflexible, & often inaccessible
  - “good” epoch granularity (partition size) depends on research question
- **can we generalize this?**
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, 3.6K documents, 205M tokens)
  - *DDR-Presseportal* (1945–1994, 4.1M documents, 1.3G tokens)
  - *DWDS Zeitungen* (1946–2016, 10M documents, 4.7G 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
- accepts user requests as set of parameter=value pairs
- parameter passing via URL query string or HTTP POST request
- common parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>query</td>
<td>target lemma(ta), regular expression, or DDC query</td>
</tr>
<tr>
<td>date</td>
<td>target date(s), interval, or regular expression</td>
</tr>
<tr>
<td>slice</td>
<td>aggregation granularity or “0” (zero) for a global profile</td>
</tr>
<tr>
<td>groupby</td>
<td>aggregation attributes with optional restrictions</td>
</tr>
<tr>
<td>score</td>
<td>score function for collocate ranking</td>
</tr>
<tr>
<td>kbest</td>
<td>maximum number of items to return per date-slice</td>
</tr>
<tr>
<td>diff</td>
<td>score aggregation function for diff profiles</td>
</tr>
<tr>
<td>global</td>
<td>request global profile pruning (vs. default slice-local pruning)</td>
</tr>
<tr>
<td>profile</td>
<td>profile type to be computed ({native,tdf,ddc} × {unary,diff})</td>
</tr>
<tr>
<td>format</td>
<td>output format or visualization mode</td>
</tr>
</tbody>
</table>
DiaCollo: Profiles, Diffs & Indices

Profiles & Diffs

- simple request $\rightarrow$ unary profile for collocant(s)
  - filtered & projected to selected attribute(s)
  - aggregated into independent slice-wise sub-intervals
  - trimmed to $k$-best collocates for target word(s)

- diff request $\rightarrow$ comparison of two independent targets
  - highlights differences or similarities of target queries
  - can be used to compare different words
  - ...or different corpus subsets w.r.t. a given word

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
- “live” KWIC-links to underlying corpus hits $\Rightarrow$ DDC search engine
- batteries not included: corpus preprocessing, analysis, & full-text search index
  - see e.g. Jurish (2003); Geyken & Hanneforth (2006); Jurish et al. (2014), ...
Appetizer

http://kaskade.dwds.de/dstar/zeit/diacollo/?q=Krise&d=1950:*&gb=1,p%3DNE
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.
  - `-0=wbad=REGEX`: surface form blacklist regex
  - `-0=pgood=REGEX`: PoS whitelist regex
  - `-tfmin=FREQ`: minimum global term-tuple frequency
  - `-lfmin=FREQ`: minimum global lemma frequency
  - `-cfmin=FREQ`: minimum co-occurrence frequency
Basic Definitions

Corpus Data
- a corpus $\mathcal{C}$ is list of $N$ tokens $t_i$
- each token is an $n_A$-tuple of attribute values
- each token is associated with a unique non-negative integer date (year) $Y(t_i) \in \mathbb{N}$

$\mathcal{C} = t_1 t_2 \ldots t_N$
$t_i \in A_1 \times \cdots \times A_{n_A}$
$Y(t_i) \in \mathbb{N}$

Corpus Domain
- lexical domain (term vocabulary)
- temporal domain (dates)

$\mathcal{W} = \bigcup_{i=1}^{N} \{t_i\} \subseteq A_1 \times \cdots \times A_{n_A}$
$\mathcal{Y} = \bigcup_{i=1}^{N} \{Y(t_i)\} \subset \mathbb{N}$

Common Notation
- attribute projection
  - for attribute-lists
- equivalence classes

$t[j] = a_j$ for $t = \langle a_1, \ldots, a_n \rangle$
$t[J] = \langle t_{j_1}, \ldots, t_{j_{n_J}} \rangle$ for $J = \langle j_1, \ldots, j_{n_J} \rangle$
$[u]_{T/J} = \{t \in T \mid t[J] = u\} \subseteq T$
Runtime Data: Requests and Profiles

DiaCollo Request

runtime user input parameters:

- \( q \) a collocant selection expression
- \( E \in \mathbb{N} \) the target epoch size
- \( G \in \langle g_1, g_2, \ldots, g_{n_G} \rangle \) the collocate attributes to project
- \( H : \mathcal{Y} \times \mathcal{W}[G] \rightarrow \{0, 1\} \) a filter function
- \( \varphi : \mathbb{R}^4 \rightarrow \mathbb{R} \) an association score function
- \( k \in \mathbb{N} \) the maximum number of collocates per epoch

Raw Co-occurrence Frequency Profile

computation basis, for \( \mathcal{E} \subset \mathbb{N} \) a finite set of corpus epochs:

- \( r_N : \mathcal{E} \rightarrow \mathbb{N} \) the total number of corpus co-occurrences by epoch
- \( r_1 : \mathcal{E} \rightarrow \mathbb{N} \) independent collocant frequency by epoch
- \( r_2 : \mathcal{E} \times \mathcal{W}[G] \rightarrow \mathbb{N} \) independent collocate frequency by epoch
- \( r_{12} : \mathcal{E} \times \mathcal{W}[G] \rightarrow \mathbb{N} \) co-occurrence frequencies by epoch

\[ Q = \langle q, E, G, H, \varphi, k \rangle \]

\[ R_Q = \langle r_N, r_1, r_2, r_{12} \rangle \]
“co-occurrence” \( \leadsto \) moving window over \( \ell \in \mathbb{N} \) content tokens

- window never crosses selected break boundaries (e.g. sentences)

- 3-level index maps “lexical” tuple pairs to date-dependent co-frequencies for (filtered) corpus \( C = s_1 \ldots s_{n_S} \) of break-units (“sentences”) \( s_i = t_{i1} \ldots t_{in_{s_i}} \),

\[
I_{12} : \mathcal{W} \rightarrow \left( \mathcal{W} \rightarrow \left( \mathcal{Y} \rightarrow \mathbb{N} \right) \right) \\
\quad : \langle w, v, y \rangle \mapsto \sum_{i=1}^{n_S} \sum_{j=1}^{n_{s_i}} \sum_{d=-\ell}^{\ell} 1[d \neq 0 \& t_{ij} = w \& t_{i(j+d)} = v \& Y(t_{ij}) = y]
\]

- **Beware:** compile-time filters (pgood, tfmin, etc.) influence index content!
  - cfmin option prunes by co-frequency \( f(w, v, y) < f_{\text{cfmin}} \Rightarrow I_{12}(w, v, y) = 0 \)

- independent “frequencies” \( I_1(w, y), I_N(y) \) computed as true marginals:

\[
I_1 : \mathcal{W} \times \mathcal{Y} \rightarrow \mathbb{N} : \langle w, y \rangle \mapsto \sum_{v \in \mathcal{W}} I_{12}(w, v, y) \\
I_N : \mathcal{Y} \rightarrow \mathbb{N} : y \mapsto \sum_{w \in \mathcal{W}} I_1(w, y)
\]
Native Co-occurrence Relation: Context Window

Input Text: The fat cat sat on the fuzzy cat.

Content: fat cat sat fuzzy cat.

Tokens: $t_1, t_2, t_3, t_4, t_5$. 

Context Window: $\ell = 3$. 

Native Co-occurrence Relation: Context Window

Input Text: The fat cat sat on the fuzzy cat.

Input Lemma: the fat cat sit on the fuzzy cat.
Native Co-occurrence Relation: Context Window

Input Text  The fat cat sat on the fuzzy cat.
Input Lemma the fat cat sit on the fuzzy cat.
Filter the fat cat sit on the fuzzy cat.
Native Co-occurrence Relation: Context Window

Input Text  The fat cat sat on the fuzzy cat.
Input Lemma the fat cat sit on the fuzzy cat.
Filter the fat cat sit on the fuzzy cat.
Content fat cat sit fuzzy cat
Native Co-occurrence Relation: Context Window

Input Text: The fat cat sat on the fuzzy cat.
Input Lemma: the fat cat sit on the fuzzy cat.
Filter: the fat cat sit on the fuzzy cat.
Content: fat cat sit fuzzy cat
Tokens: $t_1$, $t_2$, $t_3$, $t_4$, $t_5$
Native Co-occurrence Relation: Context Window

Input Text: The fat cat sat on the fuzzy cat.

Input Lemma: the fat cat sit on the fuzzy cat.

Filter: the fat cat sit on the fuzzy cat.

Content: fat cat sit fuzzy cat

Tokens: $t_1 \langle t_2 \rangle \ t_3 \ t_4 \ t_5$

$I_{12} = \{ \langle \text{fat, cat} \rangle \mapsto 1 \}$
Native Co-occurrence Relation: Context Window

Input Text  The fat cat sat on the fuzzy cat .
Input Lemma the fat cat sit on the fuzzy cat .
Filter the fat cat sit on the fuzzy cat .
Content fat cat sit fuzzy cat
Tokens $t_1$ $t_2$ $\langle t_3 \rangle$ $t_4$ $t_5$

$j = 1$
$d = 2$

$I_{12} = \{ \langle \text{fat, cat} \rangle \mapsto 1, \langle \text{fat, sit} \rangle \mapsto 1 \}$
# Native Co-occurrence Relation: Context Window

<table>
<thead>
<tr>
<th>Input Text</th>
<th>The fat cat sat on the fuzzy cat.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Lemma</td>
<td>the fat cat sit on the fuzzy cat.</td>
</tr>
<tr>
<td>Filter</td>
<td>the fat cat sit on the fuzzy cat.</td>
</tr>
<tr>
<td>Content</td>
<td>fat cat sit fuzzy cat</td>
</tr>
<tr>
<td>Tokens</td>
<td>$t_1$ $t_2$ $t_3$ $t_4$ $t_5$</td>
</tr>
</tbody>
</table>

For $j = 1$ and $d = 3$:

$I_{12} = \{ \langle \text{fat, cat} \rangle \mapsto 1, \langle \text{fat, sit} \rangle \mapsto 1, \langle \text{fat, fuzzy} \rangle \mapsto 1 \}$
Native Co-occurrence Relation: Context Window

Input Text: The fat cat sat on the fuzzy cat.

Input Lemma: the fat cat sit on the fuzzy cat.

Filter: the fat cat sit on the fuzzy cat.

Content: fat cat sit fuzzy cat

Tokens: $\langle t_1 \rangle \quad t_2 \quad t_3 \quad t_4 \quad t_5$

$j = 2$
$d = -1$\hfill
$\sim$\hfill
$I_{12} = \left\{ \langle \text{fat, cat} \rangle \mapsto 1, \langle \text{fat, sit} \rangle \mapsto 1, \langle \text{fat, fuzzy} \rangle \mapsto 1, \langle \text{cat, fat} \rangle \mapsto 1 \right\}$
Native Co-occurrence Relation: Context Window

Input Text: The fat cat sat on the fuzzy cat.

Input Lemma: the fat cat sit on the fuzzy cat.

Filter: the fat cat sit on the fuzzy cat.

Content: fat cat sit fuzzy cat

Tokens: \( t_1 \) \( t_2 \) \( t_3 \) \( t_4 \) \( t_5 \)

\( j = 2 \) \( d = 1 \)

\[ I_{12} = \{ \langle \text{fat, cat} \rangle \mapsto 1, \langle \text{fat, sit} \rangle \mapsto 1, \langle \text{fat, fuzzy} \rangle \mapsto 1, \langle \text{cat, fat} \rangle \mapsto 1, \langle \text{cat, sit} \rangle \mapsto 1 \} \]
Native Co-occurrence Relation: Context Window

Input Text    The fat cat sat on the fuzzy cat.
Input Lemma   the fat cat sit on the fuzzy cat.
Filter        the fat cat sit on the fuzzy cat.
Content       fat cat sit fuzzy cat
Tokens        $t_1 \; t_2 \; t_3 \langle t_4 \rangle \; t_5$

$j = 2$
$d = 2$
$I_{12} = \begin{cases} 
\langle \text{fat, cat} \rangle \mapsto 1, \langle \text{fat, sit} \rangle \mapsto 1, \langle \text{fat, fuzzy} \rangle \mapsto 1, \\
\langle \text{cat, fat} \rangle \mapsto 1, \langle \text{cat, sit} \rangle \mapsto 1, \langle \text{cat, fuzzy} \rangle \mapsto 1 \end{cases}$
Native Co-occurrence Relation: Context Window

Input Text  The fat cat sat on the fuzzy cat .

Input Lemma the fat cat sit on the fuzzy cat .

Filter  the fat cat sit on the fuzzy cat →

Content  fat cat sit fuzzy cat

Tokens  $t_1$ $t_2$ $t_3$ $t_4$ $\langle t_5 \rangle$

$j=2$
$d=3$

$I_{12} = \{ \langle \text{fat, cat} \rangle \mapsto 1, \langle \text{fat, sit} \rangle \mapsto 1, \langle \text{fat, fuzzy} \rangle \mapsto 1, \langle \text{cat, fat} \rangle \mapsto 1, \langle \text{cat, sit} \rangle \mapsto 1, \langle \text{cat, fuzzy} \rangle \mapsto 1, \langle \text{cat, cat} \rangle \mapsto 1 \}$
Native Co-occurrence Relation: Context Window

Input Text
The fat cat sat on the fuzzy cat.

Input Lemma
the fat cat sit on the fuzzy cat.

Filter
the fat cat sit on the fuzzy cat.

Content
fat cat sit fuzzy cat

Tokens
⟨t₁⟩ ⟨t₂⟩ t₃ ⟨t₄⟩ ⟨t₅⟩

\[ j = 3 \]
\[ d = * \]

\[ \implies I_{12} = \{ \langle \text{fat, cat} \rangle \mapsto 1, \langle \text{fat, sit} \rangle \mapsto 1, \langle \text{fat, fuzzy} \rangle \mapsto 1, \langle \text{cat, fat} \rangle \mapsto 1, \langle \text{cat, sit} \rangle \mapsto 1, \langle \text{cat, fuzzy} \rangle \mapsto 1, \langle \text{cat, cat} \rangle \mapsto 1, \langle \text{sit, fat} \rangle \mapsto 1, \langle \text{sit, cat} \rangle \mapsto 2, \langle \text{sit, fuzzy} \rangle \mapsto 1 \} \]
Native Co-occurrence Relation: Context Window

Input Text  The fat cat sat on the fuzzy cat .
Input Lemma the fat cat sit on the fuzzy cat .
Filter the fat cat sit on the fuzzy cat .
Content fat cat sit fuzzy cat
Tokens $\langle t_1 \rangle \langle t_2 \rangle \langle t_3 \rangle \langle t_4 \rangle \langle t_5 \rangle$

$$j=4 \quad d=*$$

$I_{12} = \begin{cases} 
\langle \text{fat, cat} \rangle \mapsto 1, \langle \text{fat, sit} \rangle \mapsto 1, \langle \text{fat, fuzzy} \rangle \mapsto 1, \\
\langle \text{cat, fat} \rangle \mapsto 1, \langle \text{cat, sit} \rangle \mapsto 1, \langle \text{cat, fuzzy} \rangle \mapsto 1, \\
\langle \text{cat, cat} \rangle \mapsto 1, \langle \text{sit, fat} \rangle \mapsto 1, \langle \text{sit, cat} \rangle \mapsto 2, \\
\langle \text{sit, fuzzy} \rangle \mapsto 1, \langle \text{fuzzy, fat} \rangle \mapsto 1, \langle \text{fuzzy, cat} \rangle \mapsto 2, \\
\langle \text{fuzzy, sit} \rangle \mapsto 1
\end{cases}$
Native Co-occurrence Relation: Context Window

Input Text: The fat cat sat on the fuzzy cat.
Input Lemma: the fat cat sit on the fuzzy cat.
Filter: the fat cat sit on the fuzzy cat.
Content: fat cat sit fuzzy cat
Tokens: \( t_1 \langle t_2 \rangle \langle t_3 \rangle \langle t_4 \rangle t_5 \)

\[ I_{12} = \begin{cases} 
\langle \text{fat, cat} \rangle \mapsto 1, \langle \text{fat, sit} \rangle \mapsto 1, \langle \text{fat, fuzzy} \rangle \mapsto 1, \\
\langle \text{cat, fat} \rangle \mapsto 1, \langle \text{cat, sit} \rangle \mapsto 2, \langle \text{cat, fuzzy} \rangle \mapsto 2, \\
\langle \text{cat, cat} \rangle \mapsto 2, \langle \text{sit, fat} \rangle \mapsto 1, \langle \text{sit, cat} \rangle \mapsto 2, \\
\langle \text{sit, fuzzy} \rangle \mapsto 1, \langle \text{fuzzy, fat} \rangle \mapsto 1, \langle \text{fuzzy, cat} \rangle \mapsto 2, \\
\langle \text{fuzzy, sit} \rangle \mapsto 1 
\end{cases} \]
Native Co-occurrence Relation: Runtime

Given a user-supplied query request $Q = \langle q, E, G, H, \varphi, k \rangle$

- find collocant tuple(s) $[q]$, e.g. $[\$lemma=love] = [love]_{W/a_{lemma}}$

- raw index lookup: $\hat{I}_q: \mathcal{Y} \times \mathcal{W} \to \mathbb{N}: \langle y, v \rangle \mapsto \sum_{w \in [q]} I_{12}(w, v, y)$

- group collocates by attributes $G$: $\hat{I}_{q,G}: \mathcal{Y} \times \mathcal{W}[G] \to \mathbb{N}: \langle y, g \rangle \mapsto \sum_{v \in [g]_{W/G}} \hat{I}_q(y, v)$

- apply request filter restrictions $H$: $\hat{I}_{q,G,H} = \hat{I}_{q,G} \upharpoonright \text{ext}(H): \mathcal{Y} \times \mathcal{W}[G] \to \mathbb{N}$

- aggregate by epoch $E$: $\hat{I}_{q,G,H,E}: \mathcal{E}_E \times \mathcal{W}[G] \to \mathbb{N}: \langle e, g \rangle \mapsto \sum_{y \in [e]_E} \hat{I}_{q,G,H}(y, g)$

  where $\tilde{E}: \mathcal{Y} \to \mathbb{N}: y \mapsto E[\frac{y}{E}]$; $\mathcal{E}_E = \tilde{E}(\mathcal{Y})$; $[e]_E = \tilde{E}^{-1}(e)$

- finalize raw frequency profile $R_Q = \langle r_N, r_1, r_2, r_{12} \rangle$

  - $r_N(e) = \sum_{y \in [e]_E} I_N(y)$
  - $r_1(e) = \sum_{y \in [e]_E} \sum_{w \in [q]} I_1(w, y)$
  - $r_2(e, g) = \sum_{y \in [e]_E} \sum_{v \in [g]_{W/G}} I_1(v, y)$
  - $r_{12}(e, g) = \hat{I}_{q,G,H,E}(e, g)$

  2-pass lookup strategy required for accurate independent collocate frequencies $r_2$
TDF Co-occurrence Relation: Indexing

("term × document matrix" profile type)

- "co-occurrence" ⇝ anywhere within the selected break unit ("document")
- relatively coarse index granularity (no proximity constraints)
- for corpus partitioned into documents Doc = \{d_1, \ldots, d_{n_D}\}, store:
  - sparse term-document frequency matrix
  - date counts
  - document dates and bibliographic metadata
- occurrence date, bibliographic metadata stored as document properties
- index uses mmap() on sparse matrix PDL via PDL::CCS::Nd
  - transparent on-demand paging from disk
  - fast numerical manipulation of large N-dimensional data arrays
- optimized lookup using Harwell-Boeing offset vectors
- supports Boolean query expressions and document metadata attributes
TDF Co-occurrence Relation: Runtime

- interpret collocant query $q$ independently as:
  - set of terms $\llbracket q \rrbracket_\mathcal{W}$
  - set of documents $\llbracket q \rrbracket_{\text{Doc}}$

- index lookup with collocate grouping:
  \[
  \hat{I}_{\text{tdf}: q, G} : \mathcal{V} \times \mathcal{W}[G] \rightarrow \mathbb{N}
  \]
  \[
  (y, g) \mapsto \sum_{d \in \llbracket q \rrbracket_y} \min \left\{ \left( \sum_{w \in \llbracket q \rrbracket_\mathcal{W}} \text{tdf}(w, d) \right), \left( \sum_{v \in [g]_{\mathcal{W}/G}} \text{tdf}(v, d) \right) \right\}
  \]
  where $\llbracket q \rrbracket_y = \llbracket q \rrbracket_{\text{Doc}} \cap \text{dy}^{-1}(y)$

- candidate filtering and epoch aggregation as for native index

- final raw frequency profile $R_Q = \langle r_N, r_1, r_2, r_{12} \rangle$

  \[
  r_N(e) = \sum_{y \in \llbracket e \rrbracket_E} yf(y)
  \]
  \[
  r_1(e) = \sum_{y \in \llbracket e \rrbracket_E} \sum_{w \in \llbracket q \rrbracket_\mathcal{W}} \sum_{d \in \llbracket q \rrbracket_y} \text{tdf}(w, d)
  \]
  \[
  r_2(e, g) = \sum_{y \in \llbracket e \rrbracket_E} \sum_{v \in [g]_{\mathcal{W}/G}} \sum_{d \in \text{dy}^{-1}(y)} \text{tdf}(v, d)
  \]
  \[
  r_{12}(e, g) = \hat{I}_{\text{tdf}: q, G, H, E}(e, g)
  \]
DDC Co-occurrence Relation

- “co-occurrence” \( \rightsquigarrow \) as returned by a DDC search engine query
  - requires a running DDC search engine server for the appropriate corpus
- 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

**most flexible** back-end yet implemented, but **comparatively slow**

- generated raw frequency profile \( R_Q = \langle r_N, r_1, r_2, r_{12} \rangle \)

\[
\begin{align*}
  r_N &= \lambda_q \times \text{COUNT}(\ast \ #\text{SEP}) \ #\text{BY}[\text{date}/E] \\
  r_1 &= \lambda_q \times \text{COUNT}(\text{KEYS}(\llbracket q \& H \rrbracket \ #\text{SEP} \ #\text{BY}[G=1]) \ #\text{SEP}) \ #\text{BY}[\text{date}/E] \\
  r_2 &= \lambda_q \times \text{COUNT}(\text{KEYS}(\llbracket q \& H \rrbracket \ #\text{SEP} \ #\text{BY}[G=2]) \ #\text{SEP}) \ #\text{BY}[\text{date}/E, G=2] \\
  r_{12} &= \text{COUNT}(\llbracket q \& H \rrbracket \ #\text{SEP} \ #\text{BY}[\text{date}/E, G=2])
\end{align*}
\]

- \( \llbracket q \& H \rrbracket \) a DDC query with optional collocate restrictions
- \( \lambda_q \in \mathbb{N} \) a query-dependent scaling coefficient
- server-side pre-pruning via optional \( \#\text{FMIN} \ f_{\text{cfmin}} \) query operator
Scoring & Pruning: Basics

- \( \varphi \) maps raw frequency profiles to scalar association scores
  \[ \varphi : \mathbb{R}^4 \rightarrow \mathbb{R} \]

- Score profiles \( p_{Q,e} \) computed independently for each epoch \( e \in \mathcal{E}_E \):
  \[ p_{Q,e} : \mathcal{W}[G] \rightarrow \mathbb{R} : g \mapsto \varphi(r_N(e), r_1(e), r_2(e,g), r_{12}(e,g)) \]

- \( k \)-best pruning within each epoch:
  \[ \hat{p}_{Q,e} = p_{Q,e} \upharpoonright \text{best}_k(p_{Q,e}) \]

  - “Global” profiles prune by global corpus association score:
    \[ \hat{p}_{Q,e: \text{global}} = p_{Q,e} \upharpoonright \text{best}_k(p_{Q[0/E],e}) \]

  - Alternative: user-specified cutoff threshold

- Final diachronic profile maps epoch-labels to epoch-local profiles:
  \[ \hat{P}_Q : \mathcal{E}_E \rightarrow \mathbb{R}^{\mathcal{W}[G]} : e \mapsto \hat{p}_{Q,e} \]
Score Functions: $f$ (raw frequency)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_1$</td>
<td>target tuple (&quot;collocant&quot;) matching the user query request ($q$)</td>
</tr>
<tr>
<td>$w_2$</td>
<td>collocate tuple matching the user groupby request ($G$)</td>
</tr>
<tr>
<td>$N$</td>
<td>total number of co-occurrences in the epoch $r_N(e)$</td>
</tr>
<tr>
<td>$f_1$</td>
<td>epoch-local frequency of the collocant term: $r_1(e)$</td>
</tr>
<tr>
<td>$f_2$</td>
<td>epoch-local frequency of the collocate term: $r_2(e, w_2)$</td>
</tr>
<tr>
<td>$f_{12}$</td>
<td>epoch-local frequency of the collocation pair: $r_{12}(e, w_2)$</td>
</tr>
<tr>
<td>$\varepsilon$</td>
<td>smoothing constant, by default $\frac{1}{2}$</td>
</tr>
</tbody>
</table>

$\varphi_f(w_1, w_2) = f_{12}$

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

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_1$</td>
<td>target tuple (“collocant”) matching the user query request ($q$)</td>
</tr>
<tr>
<td>$w_2$</td>
<td>collocate tuple matching the user groupby request ($G$)</td>
</tr>
<tr>
<td>$N$</td>
<td>total number of co-occurrences in the epoch $r_N(e)$</td>
</tr>
<tr>
<td>$f_1$</td>
<td>epoch-local frequency of the collocant term: $r_1(e)$</td>
</tr>
<tr>
<td>$f_2$</td>
<td>epoch-local frequency of the collocate term: $r_2(e, w_2)$</td>
</tr>
<tr>
<td>$f_{12}$</td>
<td>epoch-local frequency of the collocation pair: $r_{12}(e, w_2)$</td>
</tr>
<tr>
<td>$\varepsilon$</td>
<td>smoothing constant, by default $\frac{1}{2}$</td>
</tr>
</tbody>
</table>

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

- better visual scaling than raw frequency
- otherwise shares raw frequency’s shortcomings
Score Functions: mi (pointwise MI × log-frequency)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_1$</td>
<td>target tuple (&quot;collocant&quot;) matching the user query request ($q$)</td>
</tr>
<tr>
<td>$w_2$</td>
<td>collocate tuple matching the user groupby request ($G$)</td>
</tr>
<tr>
<td>$N$</td>
<td>total number of co-occurrences in the epoch $r_N(e)$</td>
</tr>
<tr>
<td>$f_1$</td>
<td>epoch-local frequency of the collocant term: $r_1(e)$</td>
</tr>
<tr>
<td>$f_2$</td>
<td>epoch-local frequency of the collocate term: $r_2(e, w_2)$</td>
</tr>
<tr>
<td>$f_{12}$</td>
<td>epoch-local frequency of the collocation pair: $r_{12}(e, w_2)$</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>smoothing constant, by default $\frac{1}{2}$</td>
</tr>
</tbody>
</table>

$$\varphi_{mi}(w_1, w_2) = \log_2 \left( \frac{(f_{12} + \epsilon) \times (N + \epsilon)}{(f_1 + \epsilon) \times (f_2 + \epsilon)} \right) \times \log_2 (f_{12} + \epsilon)$$

- used by first version of Sketch Engine
- PMI gives code-length change for (optimal) joint vs. independent encodings
- PMI alone is very sensitive to low-frequency items ($\leadsto$ longer codes)
  - *post-hoc* workaround: include log-frequency coefficient
- some preference for low-frequency collocates remains
### Score Functions: \( \varphi_{ll}(w_1, w_2) \) (log-likelihood)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( w_1 )</td>
<td>target tuple (“collocant”) matching the user query request ( q )</td>
</tr>
<tr>
<td>( w_2 )</td>
<td>collocate tuple matching the user groupby request ( G )</td>
</tr>
<tr>
<td>( N )</td>
<td>total number of co-occurrences in the epoch ( r_N(e) )</td>
</tr>
<tr>
<td>( f_1 )</td>
<td>epoch-local frequency of the collocant term: ( r_1(e) )</td>
</tr>
<tr>
<td>( f_2 )</td>
<td>epoch-local frequency of the collocate term: ( r_2(e, w_2) )</td>
</tr>
<tr>
<td>( f_{12} )</td>
<td>epoch-local frequency of the collocation pair: ( r_{12}(e, w_2) )</td>
</tr>
<tr>
<td>( \varepsilon )</td>
<td>smoothing constant, by default ( \frac{1}{2} )</td>
</tr>
</tbody>
</table>

\[
\varphi_{ll}(w_1, w_2) = \text{sgn}(f_{12} | f_1, f_2) \times \log(1 + \log \lambda)
\]

\[
\log \lambda = \log \frac{L(H_0)}{L(H_1)} = f_{12} \log \frac{f_{12}N}{f_1f_2} + f_{12} \log \frac{f_{12}N}{f_1f_2y} + f_{y2} \log \frac{f_{y2}N}{f_yf_2} + f_{y2} \log \frac{f_{y2}N}{f_yf_2y}
\]

- 1-sided variant of the binomial log likelihood ratio (Dunning 1993; Evert 2008)
  - only “attracting” collocate pairs are assigned positive values
- null hypothesis \( H_0 \) filters out “uninteresting” high-frequency collocates
- very sensitive to fixed & formulaic expressions ⇔ **poor visual scaling**
  - workaround: report & scale using \( \log(1 + \log \lambda) \) rather than “pure” \( \log \lambda \)
Score Functions: \( \text{ld} \) (log-Dice coefficient)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( w_1 )</td>
<td>target tuple (&quot;collocant&quot;) matching the user query request ( (q) )</td>
</tr>
<tr>
<td>( w_2 )</td>
<td>collocate tuple matching the user groupby request ( (G) )</td>
</tr>
<tr>
<td>( N )</td>
<td>total number of co-occurrences in the epoch ( r_N(e) )</td>
</tr>
<tr>
<td>( f_1 )</td>
<td>epoch-local frequency of the collocant term: ( r_1(e) )</td>
</tr>
<tr>
<td>( f_2 )</td>
<td>epoch-local frequency of the collocate term: ( r_2(e, w_2) )</td>
</tr>
<tr>
<td>( f_{12} )</td>
<td>epoch-local frequency of the collocation pair: ( r_{12}(e, w_2) )</td>
</tr>
<tr>
<td>( \varepsilon )</td>
<td>smoothing constant, by default ( \frac{1}{2} )</td>
</tr>
</tbody>
</table>

\[
\varphi_{\text{ld}}(w_1, w_2) = 14 + \log_2 \frac{2(f_{12} + \varepsilon)}{(f_1 + \varepsilon) + (f_2 + \varepsilon)}
\]

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

\( (\text{Rychlý} \ 2008) \)
Comparison Profiles ("Diffs")

- **Idea:** compare two independently acquired queries $Q_a$ and $Q_b$
  - comparison operation (diff)

- epoch alignment (1:1, n:1, or 1:m)

- apply by epoch
  $p_{Q_a \ominus Q_b, e_{ab}} : \text{Dom}_{Q_a \ominus Q_b} / e_{ab} \rightarrow \mathbb{R} : g \mapsto p_{Q_a, e_a}(g) \ominus p_{Q_b, e_b}(g)$
  - $e_{ab} = \langle e_a, e_b \rangle \in \mathcal{E}_{a \times b}$ an aligned epoch pair
  - $\text{Dom}_{Q_a \ominus Q_b} / e_{ab} \subseteq \text{dom}(p_{Q_a, e_a}) \cup \text{dom}(p_{Q_b, e_b})$ characteristic for $\ominus$ at $e_{ab}$:
    - "pre-trimmed" operations
    - "restricted" operations

- prune and collect
  $\hat{p}_{Q_a \ominus Q_b, e_{ab}} = p_{Q_a \ominus Q_b, e_{ab}} \upharpoonright \text{best}_k(p_{Q_a \ominus Q_b, e_{ab}})$
  $\hat{P}_{Q_a \ominus Q_b} : \mathcal{E}_{a \times b} \rightarrow \mathbb{R}^{|\mathcal{G}|} : e_{ab} \mapsto \hat{p}_{Q_a \ominus Q_b, e_{ab}}$
  - companion operation $\sqcup$ (usually $= \ominus$) provides final return values
  - otherwise as for unary profiles
Diff Operations: diff (raw difference)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_a$</td>
<td>1st profile query (query, date, slice)</td>
</tr>
<tr>
<td>$Q_b$</td>
<td>2nd profile query (bquery, bdate, bslice)</td>
</tr>
<tr>
<td>$s_a$</td>
<td>1st score value operand given collocate $g$: $s_a = p_{Q_a,e_a}(g)$</td>
</tr>
<tr>
<td>$s_b$</td>
<td>2nd score value operand given collocate $g$: $s_b = p_{Q_b,e_b}(g)$</td>
</tr>
</tbody>
</table>

$s_a \ominus_{\text{diff}} s_b := s_a - s_b$

- pre-trimmed
- asymmetric
- selects collocates strongly associated only with $Q_a$
### Diff Operations: adiff (absolute difference)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_a$</td>
<td>1st profile query (query, date, slice)</td>
</tr>
<tr>
<td>$Q_b$</td>
<td>2nd profile query (bquery, bdate, bslice)</td>
</tr>
<tr>
<td>$s_a$</td>
<td>1st score value operand given collocate $g$: $s_a = pQ_a, e_a(g)$</td>
</tr>
<tr>
<td>$s_b$</td>
<td>2nd score value operand given collocate $g$: $s_b = pQ_b, e_b(g)$</td>
</tr>
</tbody>
</table>

$s_a \ominus_{\text{adiff}} s_b := |s_a - s_b|$  ;  $\ominus_{\text{adiff}} := \ominus_{\text{diff}}$

- pre-trimmed
- symmetric
- selects based on $|s_a - s_b|$, but reports raw difference $s_a - s_b$
- 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)**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_a$</td>
<td>1st profile query (query, date, slice)</td>
</tr>
<tr>
<td>$Q_b$</td>
<td>2nd profile query (bquery, bdate, bslice)</td>
</tr>
<tr>
<td>$s_a$</td>
<td>1st score value operand given collocate $g$: $s_a = p_{Q_a,e_a}(g)$</td>
</tr>
<tr>
<td>$s_b$</td>
<td>2nd score value operand given collocate $g$: $s_b = p_{Q_b,e_b}(g)$</td>
</tr>
</tbody>
</table>

$$s_a \ominus_{\text{max}} s_b := \max\{s_a, s_b\}$$

- pre-trimmed
- symmetric
- selects only stronger of the operand association scores
- potentially useful for discovering collocates deserving further investigation
Diff Operations: min (minimum)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_a$</td>
<td>1st profile query (query, date, slice)</td>
</tr>
<tr>
<td>$Q_b$</td>
<td>2nd profile query (bquery, bdate, bslice)</td>
</tr>
<tr>
<td>$s_a$</td>
<td>1st score value operand given collocate $g$: $s_a = pQ_a, e_a(g)$</td>
</tr>
<tr>
<td>$s_b$</td>
<td>2nd score value operand given collocate $g$: $s_b = pQ_b, e_b(g)$</td>
</tr>
</tbody>
</table>

$s_a \ominus_{\text{min}} s_b := \min\{s_a, s_b\}$

- 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 $\sim$ zeroes)
Diff Operations: avg (arithmetic average)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Q_a)</td>
<td>1st profile query ((\text{query}, \text{date}, \text{slice}))</td>
</tr>
<tr>
<td>(Q_b)</td>
<td>2nd profile query ((\text{bquery}, \text{bdate}, \text{bslice}))</td>
</tr>
<tr>
<td>(s_a)</td>
<td>1st score value operand given collocate (g) (s_a = pQ_a, e_a(g))</td>
</tr>
<tr>
<td>(s_b)</td>
<td>2nd score value operand given collocate (g) (s_b = pQ_b, e_b(g))</td>
</tr>
</tbody>
</table>

\[s_a \ominus_{\text{avg}} s_b := \frac{s_a + s_b}{2}\]

- 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)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_a$</td>
<td>1st profile query (query, date, slice)</td>
</tr>
<tr>
<td>$Q_b$</td>
<td>2nd profile query (bquery, bdate, bslice)</td>
</tr>
<tr>
<td>$s_a$</td>
<td>1st score value operand given collocate $g$: $s_a = p_{Q_a,e_a}(g)$</td>
</tr>
<tr>
<td>$s_b$</td>
<td>2nd score value operand given collocate $g$: $s_b = p_{Q_b,e_b}(g)$</td>
</tr>
</tbody>
</table>

- restricted
- symmetric
- selects uniformly strong associations for both $Q_a$ and $Q_b$
- to avoid singularities, actually computed as:

$$havg(s_a, s_b) := \begin{cases} 
0 & \text{if } s_a \leq 0 \text{ or } s_b \leq 0 \\
\frac{2s_as_b}{s_a+s_b} & \text{otherwise}
\end{cases}$$

$$s_a \bigtriangleup_{havg} s_b := \text{avg}(havg(s_a, s_b), \text{avg}(s_a, s_b))$$
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:*&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

2010–2016
- 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)

[source: T. Werneke]
Example 1: Selected Lemma-Clouds

1980–1989:
- Sowjetunion
- Polen
- Europa
- NATO
- Afghanistan
- AEG_Hausgeräte_GmbH
- Bonn
- Berlin
- Schmidt
- Sozialdemokratische_Partei_Deutschlands

2010–2016:
- Kiew
- European_Union
- Spanien
- Merkel
- Europa
- Griechenland
- Syrien
- Italien
- Ukraine
- Krim
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’) ⇒ 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 (⇝ *Sonntag, Innenstadt*)
  - **1990–present**: . . . voluntary (⇝ *Wohnanlage, Siedlung*)

[source: A. Geyken]
Example 2: Selected Bubble-Charts

1985–1989

1990–1994

- Innenstadt
- Sonntag
- AL
- Berlin
- Stadt

- Aktionswoche
- Innenstadt
- Wohngebiet
- autoarm
- Siedlung
- Farmsen
- City
- Wohnen
- Zone
- Modellversuch
Example 3: Revolution

well, you know...

http://kaskade.dwds.de/dstar/cta/diacollo/?q=Revolution&ds=10&f=cloud

- \(< 1770\): only ‘rotation’ sense
  - ganz, Stunde, Tag ("entire, hour, day")

- \(\geq 1770\): ‘dramatic change’
  - menschlich ("human")

- \(\geq 1790\): French Revolution
  - französisch, Frankreich ("French, France")

- \(\geq 1840\): violent political upheaval (Napoleonic era)
  - Napoleon

- \(\geq 1860\): industrial revolution
  - Industrie, industriell ("industry, industrial")

[source: L. Lemnitzer, J. Lennon, P. McCartney]
Example 3: Selected Lemma-Clouds
Example 4: 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 ⇛ berühmt, ehrlich, gelehrt, tapfer, weise, . . .
  - Frau ⇛ betrübt, lieb, schön, tugendreich, verwitwet, . . .
- differences grow less pronounced in late 18th & 19th centuries
Example 4: Selected Lemma-Clouds

1725–1749:
- lieb
- groß
- ander
- gnädig
- gemein
- weise
- gebären
- gelehrt
- ehrlich
- eigen

1825–1849:
- edel
- jung
- groß
- ander
- deutsch
- lieb
- schön
- gut
- grau
- gnädig
- grau
Example 5: 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*]
\sim 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 $\sim$ berühmt, scharfsinnig, tüchtig (“famous, astute, capable”)
- Belles Lettres $\sim$ brav, grau, rechtschaffen (“well-behaved, gray, righteous”)

Similarities (diff=min)

- groß, gelehrt, gemein, jung, alt (“great, learned, common, young, old”)
Example 5: Selected Lemma-Clouds

1700–1799
(diff=adiff)

- grau
- scharfsinnig
- berühmt
- ehrenwürdig
- gut
- rechtschaffen
- weise
- alt
- arm
- edel
- tugendhaft
- vortrefflich
- edel
- geschickt
- erfahren
- gelehrt
- jung
- vernünftig
- angesehen
- alt
- arm
- edel
- geschickt
- erfahren
- gelehrt
- jung
- vernünftig
- angesehen

1800–1899
(diff=adiff)

- grau
- scharfsinnig
- berühmt
- ehrenwürdig
- gut
- rechtschaffen
- weise
- alt
- arm
- edel
- tugendhaft
- vortrefflich
- edel
- geschickt
- erfahren
- gelehrt
- jung
- vernünftig
- angesehen
- alt
- arm
- edel
- geschickt
- erfahren
- gelehrt
- jung
- vernünftig
- angesehen
Example 6: 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** \(\leadsto\) Handeln, Gesellschaft, Öffentlichkeit, Meinung, Norm, . . .
- **Cassirer** \(\leadsto\) Anschauung, Bestimmung, Bezeichnung, Erkenntnis, Sein, . . .

Similarities (diff=havg, diff=min)

- Analyse, Ausdruck, Begriff, Beziehung, Funktion, Sinn, Sprache, . . .
Example 6: Lemma-Clouds

differences
(diff=adiff)

similarities
(diff=havg)
Example 7: 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 $\rightsquigarrow$ hierbei, hieraus, hierzu (“hereby, out of which, to which”)
    - causal/logical $\rightsquigarrow$ demnach, infolgedessen, daher (“therefore”)

  Belles Lettres
    - fixed expression drunter [und] drüber (“higgledy-piggeldy, at sixes and sevens”)
    - spatial & temporal $\rightsquigarrow$ dahinter, worauf (“behind which, upon which”)
    - concessive & adversative $\rightsquigarrow$ dawider, trotzdem (“against which, despite which”)


Example 7: Selected Lemma-Clouds

1650–1699:

1950–1999:
Example 8: 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=1&G=1
QUERY: 

\[
\text{\textquote{Getränke} gn-sub WITH } p=NN) = 2 \ (\text{\textquote{trinken} WITH } p=/VV/IP/)\]

#FMIN 1

Remarks

- uses DDC back-end for fine-grained data acquisition
- uses GermaNet thesaurus-based lexical expansion for \textquote{Getränk} ("beverage")
  \cite{Hamp1997,Lemnitzer2007,Henrich2010}
- considers only those target terms immediately preceding verb \textquote{trinken} ("to drink")
- "global" profile uses shared target-set to avoid visual clutter

Observations

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

[inspiration: C. Thomas]
Example 8: Time Series \((k = 10)\)

DiaCollo Profile

\(\text{"(Getränk|gn-sub WITH $p=NN)=2 (trinken WITH $p=/VV[IP]/)" #FMIN 1}\)

- Alkohol
- Bier
- Branntwein
- Kaffee
- Milch
- Schnaps
- Sekt
- Tee
- Sekt
- Wasser
- Wein
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.*
Thank you for listening!

http://kaskade.dwds.de/~jurish/diacollo2017
http://kaskade.dwds.de/diacollo-tutorial
http://metacpan.org/release/DiaColloDB
Addenda
# Public D* DiaCollo Instances

## Historical Corpora
- Deutsches Textarchiv (1600–1900)
- Die Grenzboten (1841–1922)
- Polytechnisches Journal (1820–1931)

## Newspaper Corpora
- ZEIT (1946–2016)

## Aggregated Corpora
- DTA+DWDS (1600–1999)
- public (+newspapers, 1600–2016)

## Non-German Corpora
- APWCF (fr, 1644–1647)
- NHESS (en, 2001–2016)

## Synchronic Corpora
- DWDS Kernkorpus (1900–1999)
- Blogs (2003–2014)
- Film Subtitles (1916–2014)

## CLARIN Corpora (non-public)
- PP Berliner Zeitung (1945–1993)
- PP Neues Deutschland (1946–1990)
- PP Neue Zeit (1945–1994)
Fiendishly Awkward Questions: Corpora

Can I use DiaCollo on my own corpus?
- sure – check out the DiaColloDB and DiaColloDB::WWW distributions on CPAN
  - cpanm is handly for batch installations
- UNIX-like environment is assumed (various flavors of Linux work great)
- KWIC-links and DDC profiles require a separate DDC index and server

What languages are supported?
- pretty much any written language ought to work: DiaCollo is language-agnostic

What corpus formats are supported?
- input data must be encoded in UTF-8
- only pre-tokenized and pre-annotated formats, e.g.
  - DDCTabs: text-dump of DDC search engine index data
  - JSON: structured JSON data conforming to DiaColloDB::Document conventions
  - TCF: CLARIN-D “Text Corpus Format” as used by WebLicht
  - TEI: basic handling for pre-tokenized TEI-like XML data (slow!)
- see DiaColloDB::Document (3pm) for an up-to-date list
Why must I tokenize and annotate my corpus myself?

- one tool ⇔ one job
- language agnosia ⇛ flexibility
- DiaCollo is *not* an all-singing+dancing, one-stop-shopping text analysis tool

* (and almost certainly never will be)
- consider CLARIN-D WebLicht for a generic corpus annotation framework

Can you annotate, index, and/or host my corpus for me?

- maybe ... we should probably talk later

Can I use DiaCollo to directly compare different corpora?

- ... on the command-line:
  - pass a list:// URL to dcdb-query.perl or dcdb-www-server.perl
  - beware the fudge and extend properties!
- ... from the dwds.de/dstar WWW GUI: only for pre-aggregated corpora
  - generic implementation: work in progress (stage 0: planning)
What is ‘DDC’, and why might I care?
- “DiaLing/DWDS Concordancer” . . . sometimes “Diabolically Defective Cruft”
- search engine used by DWDS and DTA projects at the BBAW
- required for DiaCollo KWIC-link approximations and DDC relation
- configuration & usage ⇝ BTSOTD ("beyond the scope . . . ")
How large does my corpus need to be in order to get reliable results?

- more relevant = *epoch totals* $r_N(e)$, rather than global corpus totals
  - consider increasing slice parameter $(E) \rightarrow$ reducing diachronic granularity

- “good” epoch size depends on *relative frequency* of target phenomenon
  - depends in turn on request parameters query, date, groupby $(q, H, G)$
  - see Gabrielatos et al. (2012) for a more detailed discussion

- beware *compile-time filters* and *server-side pruning*
  - indexing option `-use-all-the-data` disables filters (native, TDF)
  - `#FMIN 1` query operator disables server-side pruning (DDC)

- *corpus artefacts* are always possible
  - e.g. “Pferdebuckel” (raw), “Krise↔Tolstoj” (KWIC)

- completely subjective, non-rigorous, & informal recommendation (*YMMV*):
  - your chances are pretty good if $\min\left\{ f([q], e), f([g], e) \right\} \geq 100$
  - but also interesting results from small corpora *well below this threshold!*
Fiendishly Awkward Questions: Runtime

Can I download DiaCollo results for offline use?
- static tabular formats (Text, HTML, JSON): yes
  - use the “Raw URL” link for static tabular formats (Text, HTML)
- static canvas snapshots (bubble, cloud): yes
  - use the “Download” button in the upper right of the display canvas
- interactive GUI (bubble, cloud): yes
  - use your browser’s “Save As (Web-Page, complete)” function
- google motion charts (“gmotion”) don’t support offline use

How can I restrict the profile to immediate predecessors?
- use the DDC relation with a phrase query, e.g. "*=2 Mann" #FMIN 1
- see Example “What Makes a ‘Man’”

Why does my collocant appear as a collocate for itself?
- self-collocations are never counted for identical tokens ($d = 0$);
  cf. “Native Co-occurrence Relation”
- collocated tokens of a single type are counted twice; cf. $\text{NEAR(Krise,Krise,4)}$
  - yes, this is a wart, but it’s not the wart you probably think it is
Why does my collocate item \( g \) “disappear” in epoch \( e \)?

- It may have been eliminated by *compile-time filters* or *server-side pruning*
  - Try using the **DDC relation** with the \#FMIN 1 operator

- It may not be among the \( k \)-best collocates in epoch \( e \)
  - \( k \)-best pruning occurs **independently** for each epoch
  - Try raising \( k \text{best} \) parameter \( (k) \) and/or setting the global flag
  - Try using **groupby restrictions** \( (H) \) to select only the collocate(s) of interest

Why does the D3 date-slider (bubble, cloud) “snap” to epoch boundaries?

- DiaCollo result sets are **discrete**, cf. DiaColloDB::Profile::Multi (3pm)

- D3 format size and color are **linearly interpolated** between epochs by the GUI
  - Possible future improvement: unit granularity + moving average smoothing
Fiendishly Awkward Questions: Runtime

Why does the collocation pair \((q,g)\) appear at epoch \(e\)?

*(even though I know it doesn’t really occur until later)*

- epochs are labeled by their minimum possible element, \(\tilde{E}(y) = E\lfloor \frac{y}{E} \rfloor\)
- epoch label \(e\) represents the date interval \([e \ldots e + E - 1]\)
  - e.g. for slice \(E = 10\), epoch “1980” represents the interval 1980–1989

Why don’t the corpus KWIC links always return exactly \(f_{12}\) hits?

- DiaCollo itself does *not* create or maintain a full-text index (one tool ⇐ one job)
- retrieval of corpus hits ⇔ independent DDC server
  - DDC context query generated on-the-fly for each collocation pair
- compile-time filters ⇔ *approximate* results only
  - no equivalent DDC query expression for e.g. \(wgood, pbad, \ldots\)
- to ensure exact results, use the DDC relation with the \(#FMIN\; 1\) operator
Forensic Analysis Questions: Errors

Error: *DiaColloDB::Document::CLASS: cannot load file ...*

- your input corpus does not appear to be formatted correctly
- did you specify the correct `-dclass=CLASS` option to `dcdb-create.perl`?

Error: *No ‘query’ parameter specified!*

- your request did not include a `query (q)` parameter
- appears in WWW GUI before any request has been submitted
  - nothing to see here, move along

Error: *No data to display!*

- no index entries matched your request
- usual suspects: *compile-time filters* or *server-side pruning*
  - check parameters using `dcdb-info.perl` or WWW ‘info’ link
  - see *DiaColloDB (3pm)* for details on what the various properties mean
- try using the *DDC relation* with the `#FMIN 1` operator
Forensic Analysis Questions: Errors

Error: **You cannot submit queries from an offline data set!**
- you attempted to submit a new request to an static GUI snapshot
  - e.g. as produced a browser’s “Save As” function
- submit your request to a “live” index-wrapper instead

Error: **Variable ‘ddc_url_root’ not set: KWIC links disabled!**
- your DiaCollo index is not associated with any running DDC server
- run a DDC server process for your corpus, and set the ddcServer option

Error: **500 Internal Server Error**
- this is just an HTTP status code, **not** an error message (and not very informative)
- keep reading for some (hopefully) more useful diagnostics

Error: **ttk_process(): template error: undef error - [MESSAGE]**
- something went wrong in the WWW GUI (still not very informative)
- actual error message begins with **[MESSAGE]**
Forensic Analysis Questions: Errors

Error: ... called at FILE.pm line XYZ

- this is a stack trace of the error
- only the first line or two is likely to be informative

Error: parseQuery(): ... could not parse query: syntax error: ...

- your query \((q)\) parameter could not be parsed
- consult the “Query Syntax” section of the DiaCollo help page

Error: align(): cannot align non-trivial multi-profiles of unequal size

- you tried to compare two profiles with incompatible epoch partitions
  - \(E_{\alpha \times \beta}\) could not be defined: \(1 < |E_{E_c}| \neq |E_{E_b}| > 1\)
- see “Comparison Profiles”
Forensic Analysis Questions: Errors

Error: \textit{abstract method called}

- I probably forgot to implement something; please let me know!

Error: \textit{no ‘ddcServer’ key defined}

- you tried to use the DDC relation without declaring a DDC server
- \textbf{EITHER} edit your index header.json:
  
  \begin{verbatim}
  "ddcServer":"HOST:PORT"
  \end{verbatim}
  OR use the \texttt{-O=ddcServer=HOST:PORT} option to ddc-query.perl
  ▶ replace \texttt{HOST} and \texttt{PORT} with values appropriate for your DDC server
I think I found a big nasty stinky ugly creepy crawly bug!

- it’s entirely possible that you have, but before you pick up the bat-phone ... 
  - have you read (and tried to understand) the documentation? (RTFM)
  - have you read (and tried to understand) the error message, if any? (RTFEM)
  - have you thought about what might have gone wrong? (UYFB)
  - “Simplify, simplify” – have you tried a less complex request? (Thoreau)

- ...if none of the above help, please e-mail me a precise description of:
  - what you wanted and/or expected
  - what you tried, including full URL(s) if applicable
  - what went wrong and/or was unexpected

Disclaimer: neither the author nor the BBAW condones physical violence against users!


References


References


References


References


References


