CorpusEye Manual

Eckhard Bick

University of Southern Denmark & GrammarSoft ApS

eckhard.bick@gmail.com

1 General

CorpusEye (https://corp.visl.dk) is a complex graphical user interface for grammatically-linguistically annotated corpora. The system supports searches at all linguistic levels, providing concordances and various statistics involving absolute or relative frequencies. Regular expressions can be used throughout and sequential searches allow the creation of subcorpora on the fly. Search results can be compared across corpora or grouped in bar charts according to meta-data attributes such as author, year or title. Vector-based semantic similarities can be visualized using 2-dimensional scatter plots with user-defined axes.

This manual describes all important aspects of *CorpusEye*, progressing from basic to complex issues. Sections 9 and 10 provide usage examples, the former for teaching, the latter for research.

1.1 History

CorpusEye was originally conceived during the VISL project at the University of Southern Denmark (SDU) as a teaching and research tool for Constraint Grammar-annotated corpora. The first version was programmed in 2004 using the IMS'[1] original CQP[2] engine and launched for the project's six core languages – Danish, English, German, French, Spanish and Portuguese (Bick 2005). *CorpusEye* has continuously added new corpora and new functionality, often driven by specific research projects or parser development. It was completely restructured in 2023 and 2024, when SDU left the project and *GrammarSoft ApS* took over all development, hosting and maintenance. Not least the graphical visualisation tools, e.g. bar charts and scatter plots, were added in this period.

1.2 Scope and speed

CorpusEye is primarily a linguistic tool rather than a bibliographical one. In this vein, searches are sentence-based and guided by linguistic categories such as lemma, inflection and morphosyntax. Search hits will be shown , as “quoted excerpts”, with limited context and, where necessary for GDPR[3] reasons, anonymously or in random order. For most corpora, semantic annotation has been added to facilitate sociolinguistic research, focusing on topics such as hate speech or stereotypes.

Search speed depends on both corpus size, search pattern complexity and use load (number of simultaneous users). For small or medium-sized corpora, most searches will take 1-3 seconds. However, in spite of various hardware and software optimizations, as well as parallel processing,

[1] Institut für Maschinelle Sprachverarbeitung: https://www.ims.uni-stuttgart.de/-
[2] Corpus Query Processor
[3] General Data Protection Regulation: https://gdpr-info.eu/

large corpora and/or complex searches may make for considerably longer response times. In these cases, the site will start loading a concordance page for page while continuing to process the search.

If *CorpusEye* is used for teaching, please avoid multiple simultaneous searches by the whole class. Preferably, use the site for demonstration only and have students do exercises synchronically, at home.

1.3 Licensing

The current *CorpusEye* is being maintained and developed by *GrammarSoft.* Now without public support, the site needs to be economically viable and has started to sell institutional user licenses, as well as annotation and integration services for user-provided corpora. For the time being, individual users still enjoy free access to all tools and existing open corpora.

2 Corpora

2.1 Languages

Currently, *CorpusEye* covers the main Germanic and Romance languages. The largest corpora, and arguably the best annotations, are available for Danish, German, English, Portuguese and Esperanto. Medium-level corpora are available for Spanish, French and Italian on the Romance side, and for Dutch, Swedish and Norwegian in the Germanic camp. In addition, there are experimental corpora for further Nordic languages (Icelandic, Faroese).

2.2 Corpus types

Across the language board, the site contains a large variety of corpora. Some are classical, balanced and curated corpora from professional providers, such as the English *BNC* or the Danish *Korpus 90*. Others are large, domain-oriented corpora, the two main types being news corpora and literary corpora. Depending on meta data, both allow diachronic or cross-author comparisons. For copyright reasons, literary corpora typically cover classical literature up to the 1920-1940ies. Literary corpora are available for Danish, Portuguese, Esperanto and Norwegian. Examples of news corpora are the Portuguese *Público* and *Folha,* the Danish *Information* or the Spanish *Camtie* corpus. Television news are represented for English, with the *UCLA-CSA* corpus.

Another, mixed-genre type of corpus is based on crawled internet data. These are available for almost all *CorpusEye* languages and contain a broad mixture of both curated and more spontaneous, personal texts, such as blogs and fora. An important source of crawled data in *CorpusEye* are the *Leipzig Corpora Collection*[4]. Dedicated, extensive blog corpora exist for English and Portuguese.

The social network genre is represented through the XPEROHS corpora for German and Danish, containing Twitter and Facebook data crawled between 2018 and 2023. Taken together, this bilingual corpus is by far the largest corpus on *CorpusEye.* A similar corpus for Portuguese is the *Netlang* hate-speech corpus. Another corpus with more spontaneous written data is the *ENRON E-mail corpus.*

[4] https://wortschatz.uni-leipzig.de/en

At the other end of the spectrum, *Wikipedia* dumps provide curated texts for all languages, a distinctive trait being the large spread of topics and vocabulary.

CorpusEye also offers speech corpora, mostly for Portuguese, such as the *C-ORAL* and *NURC* corpora, both with spontaneous, conversational speech. For many languages, there are corpora of parliamentary debates/speeches, a genre situated somewhere between speech and curated text. Given its multi-lingual nature, the *Europarl*[5] corpus provides comparable data across most of the *CorpusEye* languages.

Finally, there are small, specialized project corpora compiled for specific research, for instance historical or dialectal corpora *(COLONIA* for Portuguese), a Shakespeare corpus *(KEMPE),* a Danish archaeology corpus *(Skalk),* teaching media or student essays. Project corpora have been annotated and integrated for individual users or user groups and may be password-protected.

2.3 Sub-corpora

Large *CorpusEye* corpora may have a division into sub-corpora, typically based on a year-by-year split. This serves a double purpose: First, the split makes it easy to target certain time periods and to carry out diachronic comparisons. Second, using a subcorpus rather than the whole corpus saves server load and is a good way to experiment with complex (and hence, slow) search patterns before applying them to the corpus in its entirety.

Corpora that can be unfolded into sub-corpora have a blue, clickable text under the corpus name, saying ‘show sub-corpora’. For Danish, this is the case for the *Information* newspaper corpus (years 1996-2008) and for the *Twitter* social media corpus (years 2017-2023).

2.4 Annotation

All corpora have been annotated with *GrammarSoft’s* Constraint Grammar-parsers at various linguistic levels. For the most part, the automatic annotation has been left *as-is*. Therefore, category errors and other errors may be present to a certain degree, and since lower level errors (e.g. word class) may each propagate into several higher-level errors (syntax, semantics), error frequency rises with the complexity of the annotation level. As a rule of thumb across languages and genres, depending on text quality and the spelling errors, CG annotation will be 99% correct for POS (part-of-speech) and 96% correct for syntactic function.

For a few corpora, annotation has been checked manually. Notably, this is the case for the Danish *Arboretum* and the Portuguese *Floresta* treebanks.

2.4.1 Annotation levels

The following CG annotation fields can be searched, either using the menus and category fields in the graphical interface *(refine)* or the CQP query language (directly in the query field). Below, category abbreviations for the latter are given in […] brackets.

Category	**CQP**	**Comments**
Wordform	[word=”x”]	The actual token in the text. This will usually be a single word or a punctuation item, but may be a multi-word

[5] https://www.statmt.org/europarl/

		expression (MWE) in the case of names or function words[6].
Lemma	[lex=”x”]	The lexeme or, inflectionally speaking, dictionary base form. i.e. ‘eat’ for *eat, eats, ate, eaten, eating*
Part of speech (POS)	[pos=”x”]	The word class: N (noun), V (verb), ADJ (adjective), ADV (adverb), PRP (preposition), DET (determiner), PERS (personal pronoun), INDP (independent pronoun), ART (article), NUM (numeral), KS (subordinating conjunction), KC (co-ordinating conjunction), IN (interjection)
Inflection	[morph=”x”]	Inflection category values, covering e.g. gender (M/F), number (S/P), case (NOM/ACC/DAT/GEN) and definiteness (DEF/IDF) for nouns and person-number (1S, 3P ...), tense (PR, IMPF, FUT ...), mood (IND, IMP ...), diathesis (ACT|PAS) and finity (VFIN, INF, PCP ...) for verbs
Syntactic function	[func=”x”]	clause and group level functions, e.g. ‘SUBJ>’ (subject), ‘<ACC’ (direct object) or ‘<ADVL’ (adverbial) for the former and ‘>N’ (prenominal) or ‘>A’ (pre-adject) for the latter, with arrows pointing towards the dependency head.
Semantic role	[role=”x”]	The semantic function, such as AG (agent), PAT (patient), TH (theme), LOC (location) or INS (instrument). Semantic roles are generally encoded on group heads, but raised from prepositions to their (nominal) arguments.
Semantic class	[sem=”x”]	This is a lexical, rather than a functional, category, fully implemented for nouns and named entities (NE)[7]. Distinctions follow the ‘semantic prototype’ concept. There are 200-250 categories for nouns, organized in a shallow hierarchy, e.g. <tool> and <tool-shoot> or <H> (human) and <Hprof> (profession term).
Framenet class	extra=[.*fn:x.*]	This is the lexical semantic class for verbs, e.g. *fn:eat, fn:buy.* It is searched for as part of the *extra* field and needs reg.ex. wild cards (.*) if used outside the graphical search mask, e.g. _“.*fn:eat.*”_. The framenet class projects semantic roles onto its syntactic dependents.
Extra	extra=[.*x.*]	This field contains secondary tags without their own, separate search field. Examples are pronoun subclasses, as well as sentiment or domain markers (e.g. *Q\+*[8], *Q\-*, *D:med*). Since there may be other tags in the extra field, reg.ex. wild cards are needed around the search tag if used without the graphical search mask.
Dependency links	n->m	Dependency links point from dependent to head and are visible with mouse-over in the concordance. Links can be used in searches by opening the *Dep Head* mask in the graphical search interface, allowing category constraints on the dependency head (as well as on the *self* dependent in the upper mask)

[6] Spaces in MWEs are written as such in the search, but will be shown as ‘_’ (underscores) in the concordance. Internally, CG uses ‘=’.

[7] Semantic class is also available for function adverbs and – for some languages, notably Danish, German and Esperanto – adjectives. However, these are not placed in a separate field in the CQP search structure, but must be searched as part of the *extra* field.

[8] Sentiment markers (Q+, Q-) need a backslash before the ‘+’ og ‘-’ sign when used in the extra field.

2.4.2 Constraint Grammar

Constraint Grammar (CG) is both a parsing formalism and an annotation convention. For parsing and disambiguation, it implements a rule-based strategy manipulating cohorts of tag string readings assigned to tokens in running text.

CG follows a reductionist strategy that will never remove the last reading, with an in-built robustness not achievable with phrase structure grammars (PSGs). Rule conditions can be drawn from any other word (and its tags) in the sentence context and beyond. Rules can add, remove, select or substitute readings and tags, as well as delete, insert or move entire token cohorts, using sophisticated regular expressions, grammar-defined sets and variables, as well as tag or set unification. Rules are run in batches, from safest to most heuristic, further enhancing robustness.

Native CG annotation separates tag fields through spaces, and competing readings as separate cohort lines. For structural annotation, dependency links and named relations are used between numbered tokens.

Constraint Grammar was introduced in the 1990ies (Karlsson et al., 1994) and has since been used both academically and commercially, for a wide range of languages, including many under-resourced languages. The formalism has evolved over the years, with several different compiler implementations, the current standard being the VISL CG3 formalism (Bick, 2023). For an overview of languages, projects and tools, see: https://edu.visl.dk/visl2/constraint_grammar.html.

2.5 Adding a corpus

In order to add a new, user-provided corpus to *CorpusEye*, the corpus needs to undergo certain steps.

- **Format**: First of all, the corpus needs to be compiled into text format, either .txt or .xml. This means that PDFs have to be converted and text images run through OCR (optical character recognition). Both steps, but especially OCR, are bound to introduce errors and may need manual or semi-manual correction.
- **Grammatical annotation**: Second, the corpus needs to be CG-annotated. For *CorpusEye* languages, such annotation can be provided by GrammarSoft, but for new languages the may have to be done by the user before submitting the corpus. Depending on genre and modality of the corpus (speech corpora, dialect, jargon, learner corpora, social media etc.), existing CG grammars may have to be adapted to achieve a reasonable performance.
- **Meta-annotation**: If desired, meta information (e.g. bibliographical info, time stamps) needs to be added, or – if present – preserved, in standardized fields for the search engine.
- Pseudonymization: Depending on the copyright situation, the corpus needs to be processed for anonymization, or rather, pseudonymization, where person-references are stripped from id fields, and names and certain number strings in the text replaces with (linguistically equivalent) dummies.

- **Data structure**: Native CG format has to be converted into a CQP-readable database.
- Configuration: Depending on the available annotation, *CorpusEye's* configuration files need to be adapted. Where relevant, new category type must be added and unused ones removed.

3 Help files and documentation

CorpusEye provides some documentation and help files:

- *help* (file): Searching for clauses rather than words, how to formulate a CQP search, how to use regular expressions, frequency lists, multi-word expressions (MWE), mouse-over, pop-up tag windows, how to reference beginning-of-sentence
- *use cases* (PowerPoint presentation): 2024 Presentation of *CorpusEye* tools with examples from various research fields
- *tag list* (file/table): An alphabetical list of primary and secondary CG tags, with short definitions and language break-down
- *Guided tour* (Flash film): Tour of an early version of *CorpusEye*, covers basic and graphical searches, concordances and frequency statistics, but not metadata grouping or semantic scatter plots
- *How-to* (file): Short list of relevant concepts
- *Copyright* (file): By-language list of corpora, with genre, size and source information
- *Exercises* (file): Pedagogical classroom exercises for getting acquainted with *CorpusEye*, focusing on language awareness, with examples and tasks covering the VISL core languages.
- *Publications* (link): List of relevant publications about CG applications and CG grammars or corpora for various languages

4 Search interface

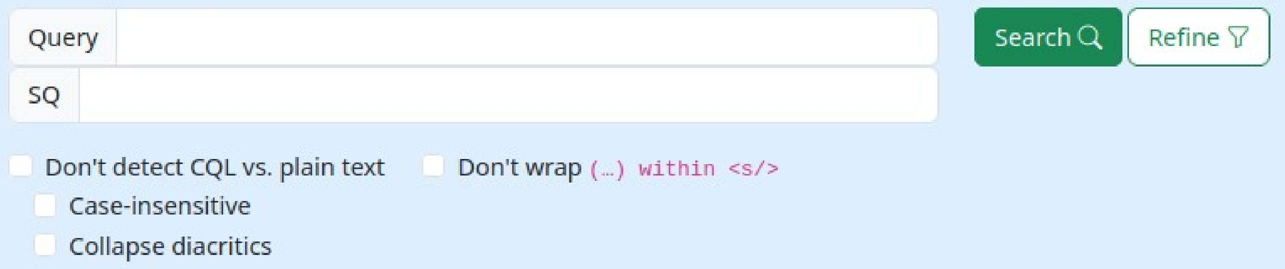
The entry page of *CorpusEye* (corp.visl.dk) offers, for all languages, a choice between the up-to-date Manatee interface and the older, classical CQP interface with fewer options.

Once a language has been selected, the site displays a full list of corpora available for that language, as well as a search field for basic text searches. Choose one or more corpora before running any search.

Information on the size (in million words) type and annotation level of a corpus is provided in parentheses after the corpus name. The symbols used all have mouse-over definitions (red=password, green=histogram and metadata grouping, yellow=semantics). The blue i-link leads to a file with source and copyright information.

4.1 Basic text searches

The input field for a basic text search is marked ‘Query’. It is situated at the top of the individual language entry pages, over the list of corpora.



□ Don't detect CQL vs. plain text □ Don't wrap `(…) within <s/>`
□ Case-insensitive
□ Collapse diacritics

Possible input are individual words *(hus)*, a chain of words *(et hus*, *har gjort)* or sentence fragments. Unlike a Google search, a string of words has to match adjacent words literatim. It does not mean OR or AND. Note that in basic text searches, text is assumed to target only entire words. Therefore, *‘hus’* does not match *‘husly’*. Regular expressions (see section 4.4) can be used to circumvent this:

For OR searches, use the pipe symbol ‘|’ and parentheses to create lists of words: *(hus|hytte|tårn)*

For searching parts of words, use the wild card ‘.*’ (zero or any number of characters). For instance, *‘hus.*’* will match *hus, husly, huse, husar, husarrest* etc.

For AND searches, i.e. sentences where two words must occur, but without necessarily being adjacent, use sequential search (SQ) as a hack (see section 4.6.3). Enter one word into the ‘Query’ field and another in the ‘SQ’ field. It is the second (SQ) word that will be focused/centered in the resulting concordance.

The SQ field also accepts (and detects) formal CQP searches, so-call ‘*CQP speak*’. In these, each [….] refers to one word, with one or more annotation categories (see 2.4.1) inside, e.g.:

[word="hus" & func!="SUBJ>"] (instance of ‘hus’ not functioning as a left-placed subject)

Here, ‘&’ means AND, ‘|’ means OR and ‘!’ (before the equal sign) means NOT.

In order to access the graphical search interface (see section 4.6), with category menus, click the ‘Refine’ button in the right upper corner.

Multi-word expressions (MWE)

CorpusEye follows the tokenization (word-by-word segmentation) performed by the Constraint Grammar-parsers used to annotate its corpora. This means that word tokens may sometimes be mulit-word expressions (MWEs) and contain spaces. Examples are names like *Bill_Clinton* or *Den_Danske_Bank*, but also some complex function words like *instead_of (*da: *i_stedet_for*, pt: *em_vez_de)* that constitute close syntact-semantic units with no meaningful internal analysis.

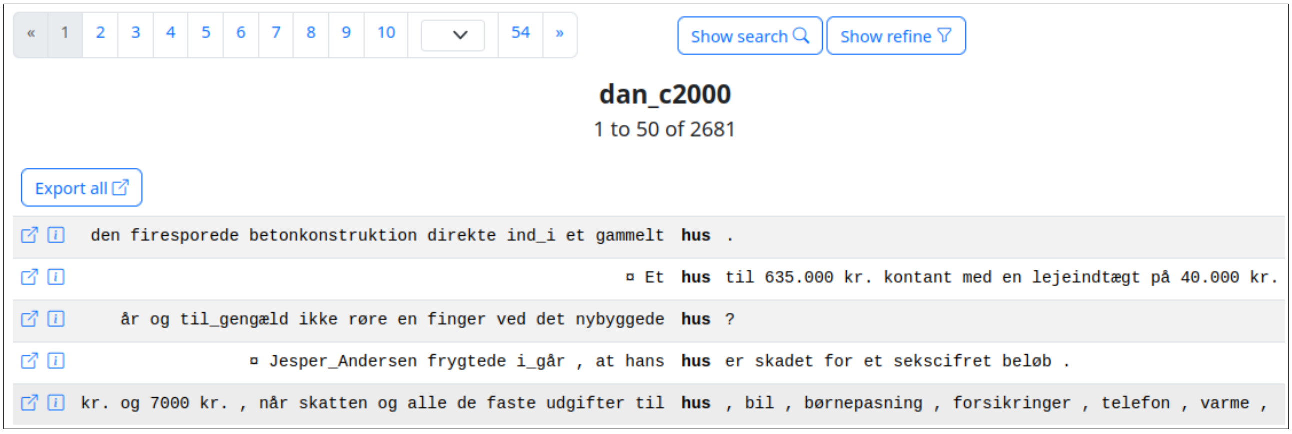
In the concordances, MWE’s are shown with ‘_’ instead of spaces, and mouse-over analysis windows will show the tagging of the whole MWE, rather than its parts.

Because the text search field interprets spaces as token/word boundaries, MWEs cannot be searched for this way, as there would be no match in the corpus for the individual parts. Instead, cqp format, e.g. [word=”instead of”] or the graphical search interface (Refine) must be used, where MWEs are searched for as single units (with spaces!) in the word and lemma (base) fields. Conversely, MWEs will not be recognized in cqp-speak or in the graphical search interface if the parts get one word box each.

4.2 Concordances

After pressing the ‘Search’ button, a concordance of matching sentences from the selected corpus (or corpora) will be shown. For each hit, the matching word(s) will be shown in center position, in bold face, with truncated sentence context on either side. Currently context is set to a fixed maximum 15 tokens or 60 characters (whichever is smaller) on either side.

Concordances are shown in pages, with the default page size set to 50 lines. One advantage of page-by-page segmentation is that *CorpusEye* can start to load, and show, the first page, while still working on the rest internally, making for a short response time and better user experience. Use the top or bottom page bars for page navigation.



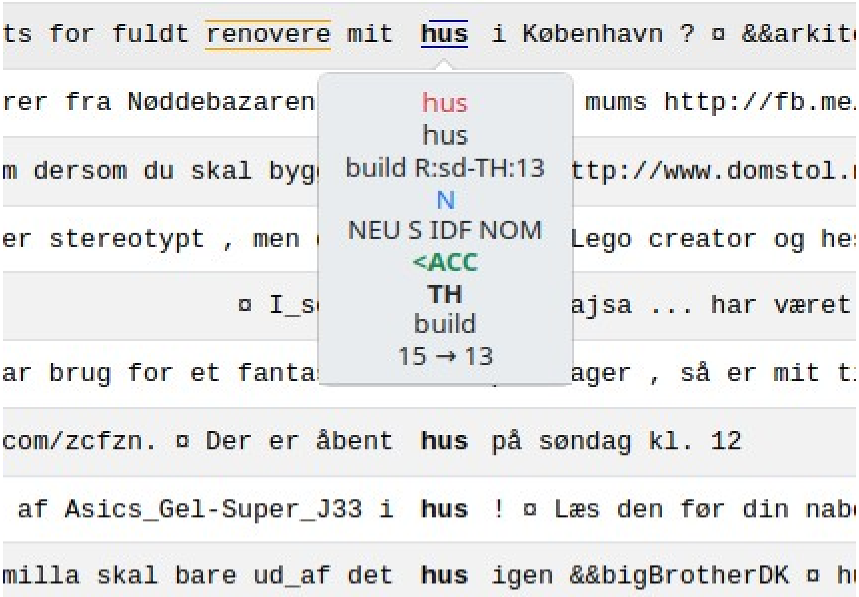
dan_c2000

1 to 50 of 2681

```
      den firesporede betonkonstruktion direkte ind_i et gammelt  hus  .
                                                            ¤ Et  hus  til 635.000 kr. kontant med en lejeindtægt på 40.000 kr.
        år og til_gengæld ikke røre en finger ved det nybyggede  hus  ?
                                ¤ Jesper_Andersen frygtede i_går , at hans  hus  er skadet for et sekscifret beløb .
 kr. og 7000 kr. , når skatten og alle de faste udgifter til  hus  , bil , børnepasning , forsikringer , telefon , varme ,
```

In the example, for the Danish *Korpus 2000,* 2681 hits were found for the word ‘hus’. The first 50 are shown, the rest can be reached by going from page to page, or by jumping to a specific page (1-54).

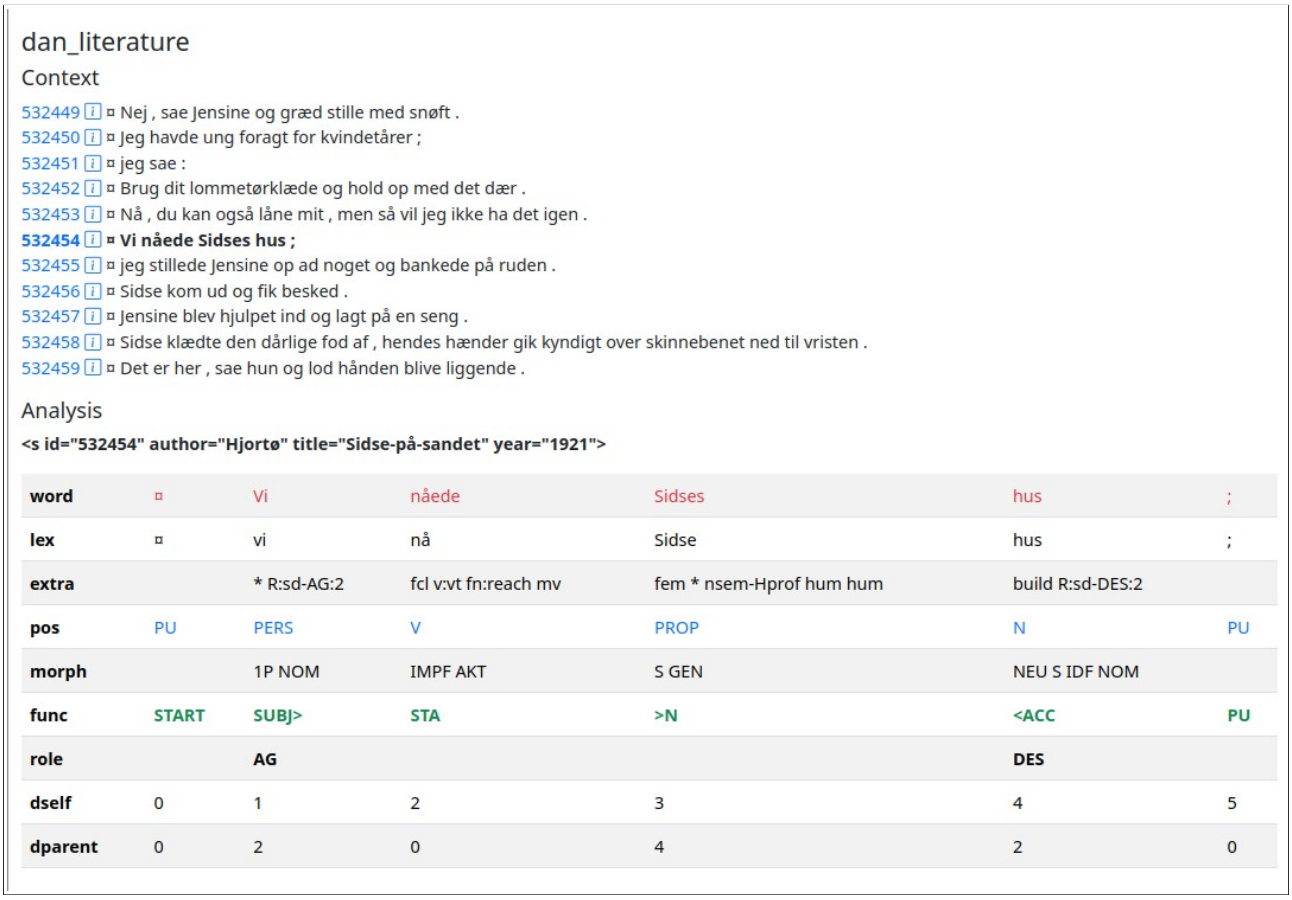
Touching a word with the mouse will highlight the dependency link to its head and show a pop-up frame with the available tagging, in the order given in section 2.4.1. Word forms are in red, POS in blue, syntactic function in green and semantic roles in bold face.’



```
ts for fuldt renovere mit  hus  i København ? ¤ &&arkit
rer fra Nøddebazaren                 mums http://fb.me.
m dersom du skal byg                ttp://www.domstol.
er stereotypt , men                 Lego creator og he
              ¤ I_s                 ajsa ... har været
ar brug for et fanta                ager , så er mit t
com/zcfzn. ¤ Der er åbent  hus  på søndag kl. 12
 af Asics_Gel-Super_J33 i  hus  ! ¤ Læs den før din nab
milla skal bare ud_af det  hus  igen &&bigBrotherDK ¤ h
```

hus
hus
build R:sd-TH:13
N
NEU S IDF NOM
<ACC
TH
build
15 → 13

Further information and context can be found by clicking the i-button left of a concordance sentence. This will open a browser window with a detailed analysis and ±5 sentences of context. Between context and analysis, a line is shown with available meta information.



dan_literature

Context

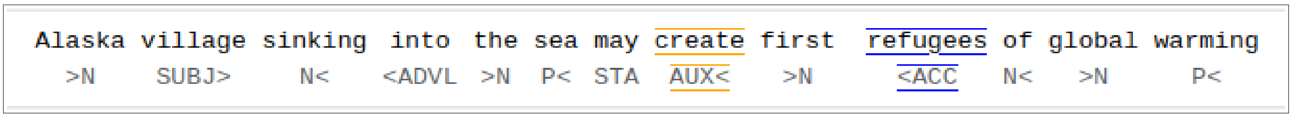
532449 ⓘ ¤ Nej , sae Jensine og græd stille med snøft .
532450 ⓘ ¤ Jeg havde ung foragt for kvindetårer ;
532451 ⓘ ¤ jeg sae :
532452 ⓘ ¤ Brug dit lommetørklæde og hold op med det dær .
532453 ⓘ ¤ Nå , du kan også låne mit , men så vil jeg ikke ha det igen .
532454 ⓘ ¤ Vi nåede Sidses hus ;
532455 ⓘ ¤ jeg stillede Jensine op ad noget og bankede på ruden .
532456 ⓘ ¤ Sidse kom ud og fik besked .
532457 ⓘ ¤ Jensine blev hjulpet ind og lagt på en seng .
532458 ⓘ ¤ Sidse klædte den dårlige fod af , hendes hænder gik kyndigt over skinnebenet ned til vristen .
532459 ⓘ ¤ Det er her , sae hun og lod hånden blive liggende .

Analysis

<s id="532454" author="Hjortø" title="Sidse-på-sandet" year="1921">

word	¤	Vi	nåede	Sidses	hus	;
lex	¤	vi	nå	Sidse	hus	;
extra		* R:sd-AG:2	fcl v:vt fn:reach mv	fem * nsem-Hprof hum hum	build R:sd-DES:2	
pos	PU	PERS	V	PROP	N	PU
morph		1P NOM	IMPF AKT	S GEN	NEU S IDF NOM	
func	START	SUBJ>	STA	>N	<ACC	PU
role		**AG**			**DES**	
dself	0	1	2	3	4	5
dparent	0	2	0	4	2	0

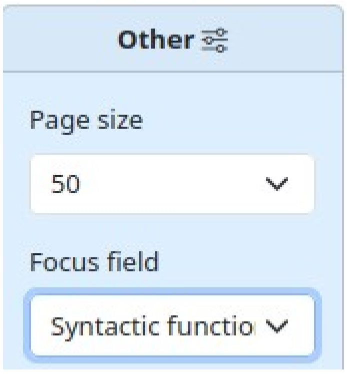
As a default, a concordance line will show ordinary, running text, i.e. a sequence of word forms (tokens). However, for visualization of syntactic or semantic patterns, other fields can be shown as well (one type at a time), as a so-called ‘focus field’, for instance syntactic function:



Alaska	village	sinking	into	the	sea	may	create	first	refugees	of	global	warming
>N	SUBJ>	N<	<ADVL	>N	P<	STA	AUX<	>N	<ACC	N<	>N	P<

(Alaska village sinking into the sea may create first **refugees** of global warming)

You can choose or change page size and focus field in a separate menu at the bottom of the statistics sidebar.



Other

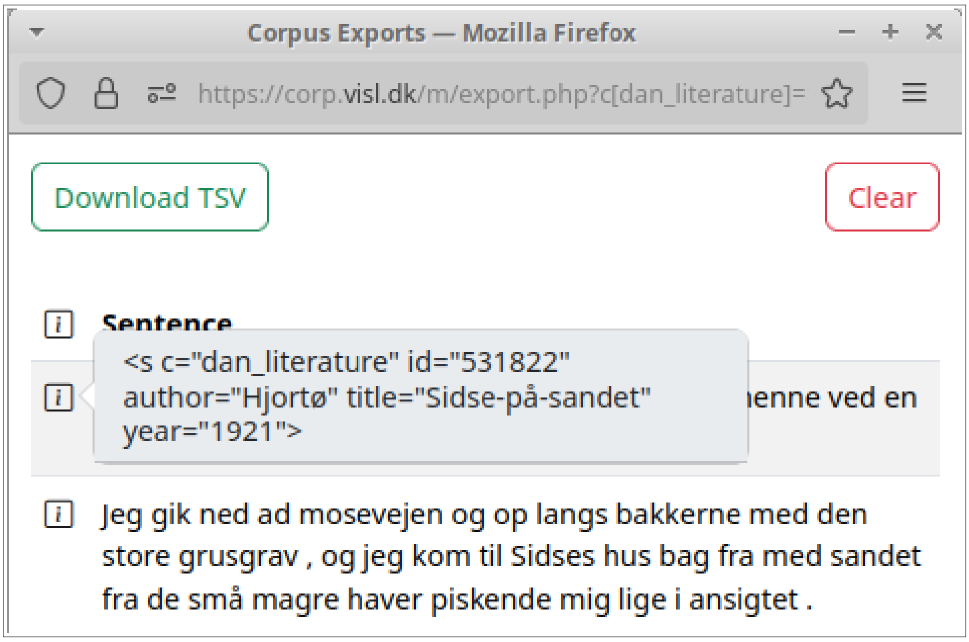
Page size: 50

Focus field: Syntactic functio

4.3 Excerpts

The leftmost button (box and arrow) on each concordance line is used to export the sentence (not just the concordance!) in plain text (.txt) to a separate window for safekeeping. Further, later clicks on other lines will add them to the existing window, allowing you to collect them for teaching or research.

Like the concordance itself, the export window has i-buttons that will display available meta-information.


Corpus Exports — Mozilla Firefox
https://corp.visl.dk/m/export.php?c[dan_literature]=
Download TSV
Clear
Sentence

<s c="dan_literature" id="531822" author="Hjortø" title="Sidse-på-sandet" year="1921">

ιenne ved en

Jeg gik ned ad mosevejen og op langs bakkerne med den store grusgrav , og jeg kom til Sidses hus bag fra med sandet fra de små magre haver piskende mig lige i ansigtet .

The ‘Export all’-button, in the left upper corner of the concordance, will export all examples on a given concordance page (as a default, 50). The content of the export window may be downloaded as a spreadsheet table, using the green ‘Download TSV’ button.

4.4 Regular expressions

Regular expressions can be used instead of ordinary text anywhere in the *CorpusEye* interface. At the simplest level, you can use a dot (.) as a dummy character, in combination with optionality and repetition operators, and:

- .? = zero or one character (optionality), e.g. "øve.?" (øve, øver, øvet, øves), or - with an optional 'l' - "l?øve.?" (øve ... + løve, løver, løves)
- .* = zero or more characters (greedy, i.e. longest possible match), e.g. "hus.*" (hus, huse, husene, husar, husarrest)
- .+ = one or more characters (greedy, i.e. longest possible match), e.g. "hus.+" (huse, husene etc., *ikke* hus), or "oh+" (oh, ohh, ohhh, ...)
- .*? = zero or more (non-greedy, i.e. shortest possible match)
- .+? = one or more (non-greedy, i.e. shortest possible match)

Note that these are the same operators that are used in the menu-based part of the interface *(refine search)* as optionality and repetition operators for entire search fields.

Another useful regex feature are *sets* of characters. Sets are enclosed in angular brackets [], and ranges can be defined with a hyphen [c1-c2]. Negated sets start with a caret [^...]. Ordinary sets [] are sets of characters. For complex 'sets' of expressions, round brackets are used, and set items are separated by the '|'-operator (logical *OR*).

- ordinary set: [aeiou] (vowels)
- range set: [a-z] (alphabet), also combined with individual letters [a-zæøå] or another range [a-zA-Z0-9]
- negated set: [^aeiou], [^a-z], [^a-zA-ZæøåÆØÅ]
- complex sets: (wine|beer|milk|tea|coffee), also with dummies and character sets, (.*i[sz]e|.*ate) (organise, organize, validate), equivalent to (.*(i[sz]e|ate)) or (.*(i[sz]|at)e)

A number of protected symbols lets you define start and end of a string or non-printing characters like tabs and line breaks. Also, som sets are pre-defined as protected symbols:

- \t = tab
- \n = line break (newline)
- \s = space character, tab or newline (plus windows line feed \f and carriage return \r)
- \w = word character (i.e. an alphanumeric letter [a-zA-Z0-9])
- \d = digit (i.e. a number [0-9])

The \w symbol is a short cut to circumvent letter sets. However, it is a little unsafe for non-English languages, since accented letters are not necessarily included, depending on system settings. For Danish, for instance, [a-zA-ZæøåÆØÅ] is safest. Another solution are 'inverted' symbols, in upper case, which negate the original set symbol. Thus, '\S' means anything *but* a space character, so it will include accented letters and æøå as well (together with parentheses and the like ...).

- ^ = start of string(-line), used in initial position, e.g. /^anti/ (starting with 'anti-', not 'Chianti'.
- $ = end of string(-line), used in final position, e.g. /anti$/ ('Chianti', not 'anti-')

Start and end markers are presupposed in this interface, so you won't need them. In fact, if you want *anti* to match both *Chianti* and *antidote*, you have to use ".*anti.*".

At the most advanced level, regular expressions allow variables. In fact, any bracketed part () of a regular expressions is regarded as a variable and can be referred to later in the same expression. Variables are named as back-slashed numbers, counting round brackets from left to right as \1, \2, \3 etc. However, cqp does not allow this particular feature. You can still use it in the old interface (rectangular flags), on running text corpora. \s([gc][aeiou])[^]+(\1\w+)+\s finds you "gaelic" alliteration rhymes in an English corpus. \s\w(\w+)-\w\1\s, surrounded by single spaces, finds you lots of "willy-nilly"-constructions as well as some "cut-out" and "four-hour" cases. \s[a-z]([a-z]+))\-[a-z]\1\s does the same, but avoids soccer results.

Start of sentence: In the newer corpora, each sentence has a start-marker (¤) which is regarded as a "word-form", and can be used to look for sentence-initial item, i.e. finite verbs in inverted Danish sentences.

4.5 CQP searches

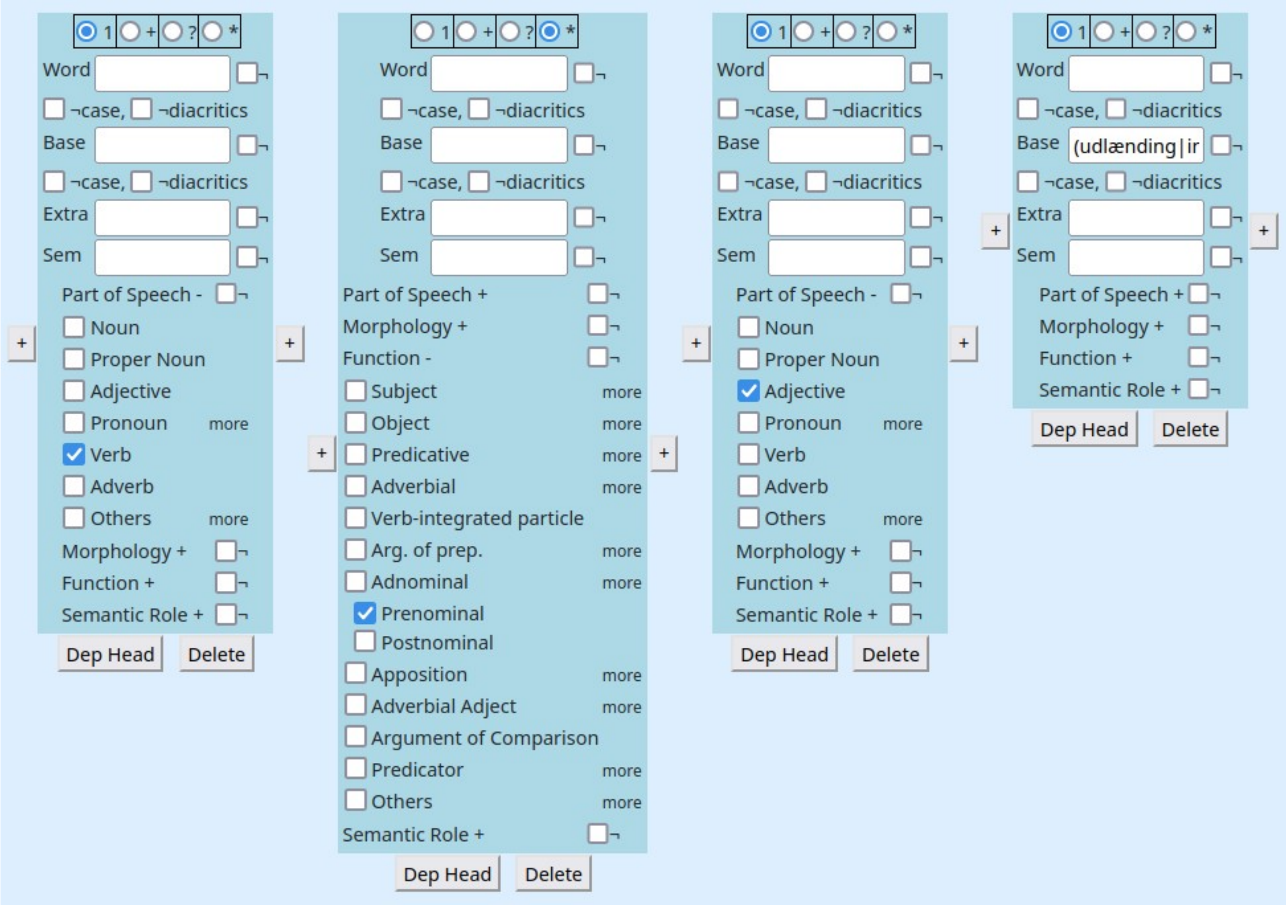
Instead of normal text searches, or menu-based "refine" searches, you can type cqp-searches directly on the search line. Don't worry, you needn't - the interface will translate your menu choices into cqp -, but if you are familiar with cqp, this may be a fast option. By the way, cqp-translations of text or menu searches are shown at the top of every concordance page, so you can experiment using cut & paste, changing parts of an existing cqp-expression rather than writing one from scratch. The basic syntax for a cqp-search is the following:

- text words are written in quotes
- attributes of fields are written in angular brackets. Each token gets one (square) bracket.
    - word form: [word="children"]
    - lexeme: [lex="child"]
    - part of speech: [pos="N"] (noun)
    - morphology/inflexion: [morph=".*P.*"] (plural)
    - syntactic function: [syn="SUBJ>"] (subject left of verb
    - secondary tags: [extra=".*Hprof.*"] (human professional)
- fields applying to the same word token are joined in the same bracket, and are linked with a boolean operator. For clarity, fields may be enclosed in ordinary brackets (). Fields can be negated:
    - '&'-operator (AND): [lex="child" & func=".*ACC.*"], same as [(lex="child") & (func=".*ACC.*")].
    - '|'-operator (OR): [lex="child" | lex="daughter" | lex="son"].
    - '!'-operator (NEGATION): [lex="child" & ! func=".*ACC.*"].
    - An empty [] matches any token.

Both text words and attributes allow regular expressions (cf.). Thus, [word="child.*"] will give you all word starting with *child*. Note that you will have to fill in .* dummies, if you are looking for only one of several tags in a field, as in [morph=".*P.*"] and [extra=".*Hprof.*"]. The morph field has more than one tags for most languages, and there is always the risk of there being more than one secondary tag. And, though rarely, syntactic function may be marked as ambiguous or unresolved. Only the word and lex fields are safely 1-item fields.

4.6 Graphical search interface (GUI): Refine

The ‘Refine’ button opens a graphical search interface, where each blue options box represents a word to be searched for. You can add or remove word boxes using the ‘+’ button on the box sides or the ‘Delete’ button in the box footer.



● 1 ○ + ○ ? ○ *
Word ☐¬
☐ ¬case, ☐ ¬diacritics
Base ☐¬
☐ ¬case, ☐ ¬diacritics
Extra ☐¬
Sem ☐¬
Part of Speech - ☐¬
☐ Noun
☐ Proper Noun
☐ Adjective
☐ Pronoun more
☑ Verb
☐ Adverb
☐ Others more
Morphology + ☐¬
Function + ☐¬
Semantic Role + ☐¬

○ 1 ○ + ○ ? ● *
Word ☐¬
☐ ¬case, ☐ ¬diacritics
Base ☐¬
☐ ¬case, ☐ ¬diacritics
Extra ☐¬
Sem ☐¬
Part of Speech + ☐¬
Morphology + ☐¬
Function - ☐¬
☐ Subject more
☐ Object more
☐ Predicative more
☐ Adverbial more
☐ Verb-integrated particle
☐ Arg. of prep. more
☐ Adnominal more
☑ Prenominal
☐ Postnominal
☐ Apposition more
☐ Adverbial Adject more
☐ Argument of Comparison
☐ Predicator more
☐ Others more
Semantic Role + ☐¬

● 1 ○ + ○ ? ○ *
Word ☐¬
☐ ¬case, ☐ ¬diacritics
Base ☐¬
☐ ¬case, ☐ ¬diacritics
Extra ☐¬
Sem ☐¬
Part of Speech - ☐¬
☐ Noun
☐ Proper Noun
☑ Adjective
☐ Pronoun more
☐ Verb
☐ Adverb
☐ Others more
Morphology + ☐¬
Function + ☐¬
Semantic Role + ☐¬

● 1 ○ + ○ ? ○ *
Word ☐¬
☐ ¬case, ☐ ¬diacritics
Base (udlænding|ir ☐¬
☐ ¬case, ☐ ¬diacritics
Extra ☐¬
Sem ☐¬
Part of Speech + ☐¬
Morphology + ☐¬
Function + ☐¬
Semantic Role + ☐¬

The top bar in each box specifies if the box/word is optional or reiterated. The standard is ‘1’, i.e. exactly one word that matches all search patterns in this word box. ‘+’ means 1 or more, ‘?’ means none or more (i.e. optional), and ‘*’ for none, one or several. The latter is useful, for instance, for chains of (pre-nominal) attributes, like ‘the *big black hairy* dog’, or for allowing other words between a verb and its object.

There are two types of search field – text fields and tick-off fields. Text fields allow regular expressions, and – if relevant – case- or diacritic-insensitive search. Tick-off fields with a ‘+’ c(e.g POS or morphology) can be unfolded into “menus” by clicking the plus sign. Re-collapsing the menu is done by clicking the minus sign that replaces the plus sign in the unfolded menu. Some menu categories (e.g. the ‘pronoun’ POS category) have a sub-menu of their own, which can be (un)folded using the ‘more’ button next to the category.

Many fields can be negated by clicking the ‘¬’-sign (a dash with a tail) next to their tick boxes.

Finally, in the footer of each search box, there is a button for dependency searches “Dep Head”. Clicking it will open a light blue version of the original search box underneath. In this new search box, all search conditions apply to the dependency head token of the word in question. For subject and objects, for instance, the ‘Dep Head’ box will refer to the clause’s main verb. Once you have opened a ‘Dep Head’ box, you will see a ‘Sibling’ button in the new footer below. This can be used to put search constraints on a sibling (sister) dependent of the dependency head. For instance, departing from an object, the subject or an adverbial will be a sibling.

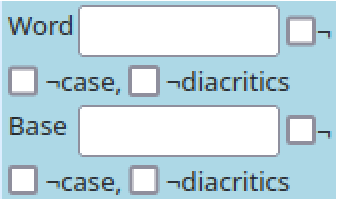
Clicking on an unfolded ‘Dep Head’- or ‘Sibling’-button will close the respective sub-box.

4.6.1 Search fields

The individual search fields in a search box match linguistic categories, from low-level lexical categories (higher up) to higher-level functional categories (further down). In CQP speak, i.e. when writing in the query field rather than using the menus, each word box is a square bracket with one or more (&) category-value pairs, e.g. [pos=”N” & syn =”SUBJ>”].

After making your choices in the graphical search boxes, use the ‘Search’ button at the bottom to send off the finished search command. If you are modifying a previous search graphically, you must remember to “translate” the graphical search choices into CQP speak by mouse-clicking into the ‘Query’-field below the blue word boxes. Only then it is safe to hit the adjacent ‘Search’ button. If you wish, you can also modify a previous query directly in the CQP expression in the ‘Query’ field.

Word (Wordform) and **Base (Lemma)**: The two text fields at the top are for the word itself, rather than its grammar. ‘Word’ is the full, potentially inflected, wordform as such, while ‘Base’ is the lemma, i.e. the uninflected dictionary form of the word, e.g. *eat* for *eat, eats, eating, eaten, ate.*



For most research, the lemma (base) is more useful, as it introduces a minimal level of abstraction and lead to more hits, and – therefore – a higher level of significance in a statistical analysis. Also, if the parser has identified and normalized spelling errors or other lexical variation, the lemma is the more robust search alternative.

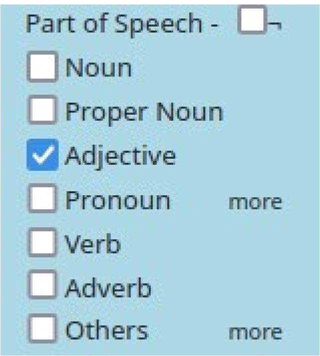
On the other hand, if your interest is explicitly said variation, e.g. in a diachronic or cross-corpus genre perspective, you would either want to search for specific wordforms, or to search for a lemma and then sort the result in terms of wordforms.

Another reason to compare wordforms with lemmas could be lexeme-linked grammatical restrictions – for instance, a lack of plural inflection for mass nouns.

There are some special cases for the word and lemma fields:

- **sentence start:** sometimes, it is necessary to make sure that a searched-for word is the first word in the sentence (for instance, in research on imperatives or topicalizations). *CorpusEye* therefore inserts a special ‘¤’ token as the first “word” in each sentence. The token’s POS is punctuation (PU), and its lemma equals the wordform (¤).
- **emoticons and emojis:** these appear as-is in the word field, but are assigned groupable semantic lemmas, e.g. *emo-happy* or *emo-angry* for emoticons. Emojis have further subdivisions, e.g. *emo-happy-Smiling-Face-With-Smiling-Eyes*, so the best way to search for an emoticon/emoji lemma is with an added catch-all (.*), e.g.: ‘*emo-happy.**’. The part-of-speech is adverb (ADV).

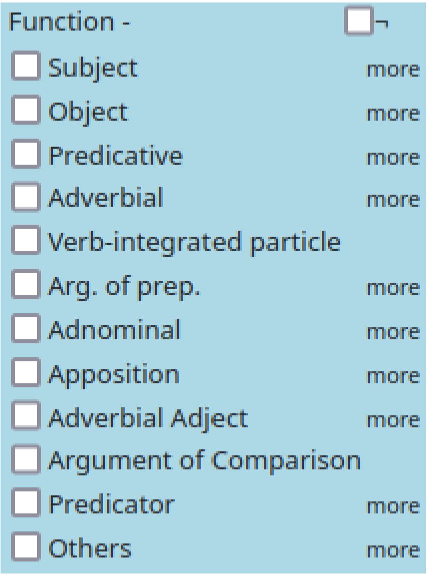
Part-of-speech (POS): This is a clickable menu containing the main inflecting word classes (noun, adjective, verb and pronoun) as well as ‘adverb’ and a sub-menu ‘Others’ (conjunctions, preposition, numeral, article, interjection, prefix and punctuation). Note that proper nouns are classified separately from common nouns and that pronouns are subdivided into ‘independent pronoun’ (uninflecting, noun-like, e.g. *‘nothing’, ‘everybody’*), determiner pronoun (adjectival and – in many languages - inflecting, e.g. *‘hver(t)’, ‘nogle’, ‘al(t)’, ‘alle’*) and personal pronoun (person- and gender-marked).



Part of Speech - □¬
☐ Noun
☐ Proper Noun
☑ Adjective
☐ Pronoun more
☐ Verb
☐ Adverb
☐ Others more

For CQP speak in the query field, upper case letter abbreviations are used, e.g. N=noun, PRP=preposition.

Morphology: This menu is dependent on a POS choice, as different parts-of-speech may have different inflection categories. Therefore, the ‘morphology’ menu can only be unfolded, once a POS has been chosen. For Danish nouns, for instance, the ‘morphology’ menu contains the gender, number, case and definiteness categories, each of which has a ‘more’-button leading to a list of values for the category in question, e.g. S (singular) and P (plural) for the category of ‘number’.



Function - □¬
☐ Subject more
☐ Object more
☐ Predicative more
☐ Adverbial more
☐ Verb-integrated particle
☐ Arg. of prep. more
☐ Adnominal more
☐ Apposition more
☐ Adverbial Adject more
☐ Argument of Comparison
☐ Predicator more
☐ Others more

Syntactic function: The third collapsible menu contains clause- and group-level functions. The former comprise subject, objects, and free or bound predicatives and adverbials, as well as the verbal functions within the predicator (e.g *main verb, auxiliary*). Objects are subdivided motivated by case, into accusative (direct) object, dative (indirect) object, genitive object (for German) and prepositional (oblique) object, and the Germanic languages have a distinction between full (real) subjects and provisional or formal subjects.

At the group level, a minimal system of dependency attachment functions is used, distinguishing between (pre- or post-) adnominals, adverbial adjects and arguments of prepositions and comparison particles. The ‘Other’ category, when unfolded by clicking ‘more’, contains structural functions (subordinator, co-ordinator), communicative functions (vocative, question, command), as well as topic and focus. For many functions, a distinction is made between dependency attachment direction, so you can search, for instance, for post-positioned subjects (“right” subjects, after the verb). When using the query field rather than the menus, these attachment directions are written as arrows, e.g.:

SUBJ> (left subject), <ACC (right accusative object), >N (prenominal), P< (argument of preposition).

For a complete list of categories, with abbreviations and definitions, see: https://edu.visl.dk/tagset_cg_general.pdf . Language-specific explanations and examples are available in VISL's respective language sections (edu.visl.dk), e.g.:

Danish: https://edu.visl.dk/visl/da/info/dansymbolcg.html

Spanish: https://edu.visl.dk/visl/es/info/symbolsopen.es.html

Portuguese: https://edu.visl.dk/visl/pt/info/symbolset-manual.html

Semantic role: Semantic (or thematic) roles represent a deeper layer of function than syntactic functions. For instance, the subject of the verb "build" gets an agent role (AG) in the active sentence, but takes over the object's result role (RES) in the passive sentence. In a full framenet analysis, semantic roles are linked to, and projected by, certain verb frame classes.

The main clause level categories are agent (AG), cognizer (COG), speaker (SP), patient (PAT), Donor (DON), Recipient (REC), Beneficiary (BEN), Experiencer (EXP) and theme (TH). Adverbial roles cover space and time roles, as well as instrument (INS). The space menu contains location (LOC), origin (ORI), destination/goal (DES) and path (PATH). Time categories are "metaphorical clones" of the space categories, i.e. temporal location (LOC-TMP) and temporal goal (DES-TMP).

For a complete list of roles, with explanations and examples, see the VISL documentation:

https://edu.visl.dk/~eckhard/pdf/semantic_roles_manual.pdf

Semantic class (Sem): The sem and extra fields are text fields rather than menus because of the very large number of possible categories. You can mouse-over inspect words in a concordance, or click the 'i' (info) button before a concordance line to see what categories are used.

Semantic class is about semantic form. It is a lexeme (dictionary) category rather than a function. For instance, town names like Berlin, Washington and Ankara have the semantic class of "town" <Ltown> and belong to the super-category of "civitas" <civ>, together with country names. In terms of semantic function however, towns and countries can not only fulfill place roles (e.g. location or destination) – they can also function as agents or speaker, going to war or making announcements.

Constraint Grammar organizes semantic class categories in shallow ontologies. Most *CorpusEye* languages have implemented a full noun ontology with ~250 categories. When searching for a human noun, you can use a regular expression to make a hypernym category *H[a-z]** that will lump all human subcategories, e.g. *Hprof* (profession), *Hfam* (family term), *Hnat* (nationality term), *Hideo* (follower), *Htit* (title) etc. Other hypernym categories are *food.** or *drink.**, lumping subcategories like *food-h* (human-produced food), *food-c* (countable food term) or *drink-alco* (alcoholic drink).

For proper nouns CG- and pattern-driven named-entity modules (NER) are used for classification, the main categories being *hum* (human), *org* (organization), *inst* (institution with a fixed location), *civ* (places that can act), *top* (natural places), *occ* (event), *brand* etc.

Some major *CorpusEye* languages (en, da, de, pt, eo) also offer Framenet annotation, implying a semantic class tag for verbs. These are prefixed with *'fn:'*, e.g. *fn:buy*, *fn:sell*, *fn:run*, *fn:eat* etc.

Most languages also have some semantic tagging of adjectives, e.g. *jh* for ‘human adjective’ or *jnat* (nationality adjective). But only Danish, German and Esperanto have full systems with hundreds of categories, such as *jpsych* (psychological feature), *jemo* (emotion), *jtime*, *jsize*, *jcol* (colour) etc. Categories will often containing information about domains, e.g. jmed (medical), or which type of noun a given adjective can combine with, e.g. jfood, jdrink, janat (anatomical).

For a full list of semantic classes, see:

nouns and NER: https://edu.visl.dk/semantic_prototypes_overview.pdf

adjectives: https://edu.visl.dk/semantic_prototypes_adj.pdf

verb frames: https://framenet.dk/verbal_prototypes.pdf

Extra field: This is the field for everything else …. containing secondary tags mapped from lexica or created during the parsing process. The <...> brackets are used for the original CG tags, but not part of the *CorpusEye* search fields.

- syntactic-semantic subclasses of pronouns: <rel> (relative), <interr> (interrogative), <quant[0-3]?> (quantifier pronouns), <artd> (definite article), <arti> (indefinite article), <setop> (set operator adverbs, e.g. negation), <aloc> (locative adverbs), <adir> (direction adverbs), <atemp> (temporal adverbs) etc.; <foc> (adverb with a focusing function)
- valency tags for verbs: <vt> (monotransitive), <vdt> (ditransitive), <vk.*> (copula), <vtk.*> (trans-objective), <vi> (intransitive), <ve> (“ergative”), <xxx^vp> (binds pp’s with the preposition ‘xxx’, <xxx^vtp> (same plus a direct object), <x> (auxiliary with infinitive)
- valency tags for other word classes: <+xxx> (binds a pp with the preposition ‘xxx’, <+INF> (binds an infinitive clause), <+FS> (binds a finite clause)
- domains: <D:med> (medicine), <D:sp> (sport), <D:food>
- diachronicity: <D:A> (archaic), <D:neo> (neologism), <Rare>
- sentiment: <Q+> (positive). <Q-> (negative), <Q0> (neutral)
- language variation and spelling: <error> <R:xxx> (corrected word form), <ORI:xxx> (original word form), <foreign> (foreign and loan words), <sms> (sms and social media jargon), <olddansk> (old Danish)[9]
- compounding and derivation, e.g. <N:indvandring~s+stop> or <F:anti+bakteriel>. These tags start with the part-of-speech of the first part, followed by a colon. ‘+’ delimits roots from each other and from prefixes and suffixes (Danish and German). Fuge letters, i.e. letters added at the contact line in a compound, but not part of the preceding lemma, are split off with a ‘~’ (tilde). For Esperanto, a distinction is made between ‘+’ (between roots) and ‘%’ (before/after a prefix or a suffix), and inflectional endings are set off with a ‘|’ (bar).

4.6.2 Emoticons and emojis

Emoticons (emotion-carrying punctuation strings, e.g. :), :(, :-)) and emojis (character pictograms, e.g. 😮, 🙄) are an important part of not least social media corpora. *CorpusEye* categorizes

[9] Note that the annotation convention for orthographical variation is to keep the original wordform (or at least to store it in an <ORI:....> tag), but to normalize the lemma (and everything else) to modern standard spelling.

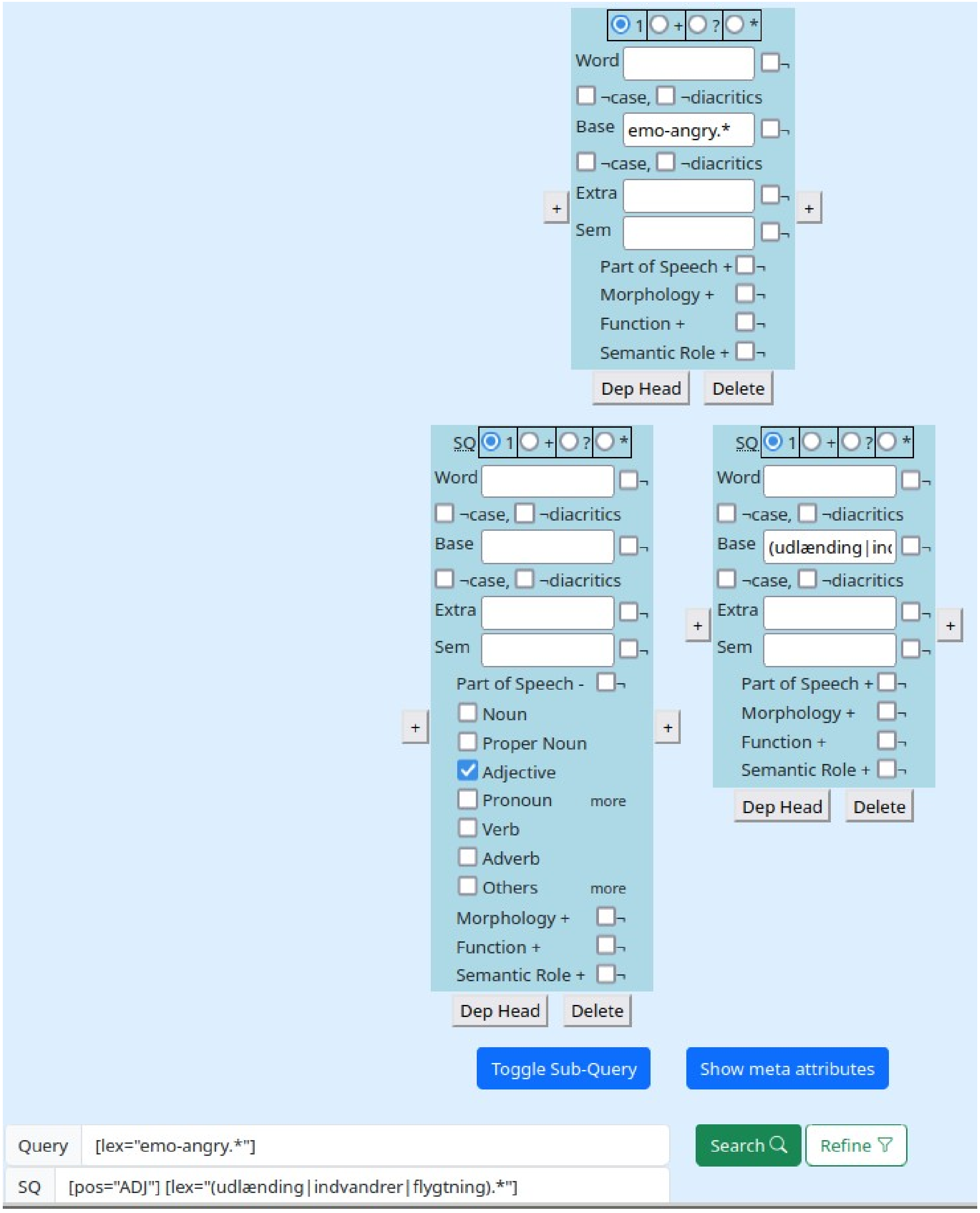
emoticons and emojis as adverbs and stores their type (function) as a baseform, e.g. *‘emo-surprise-With-Open-Mouth’* (😮).

CorpusEye uses 9 hypernym classes of emoticons and emojis: emo-happy.*, emo-love.*. emo-laugh.*, emo-sad.*, emo-angry*, emo-horror.*, emo-surprise.*, emo-skeptical.*, emo-wink.*

Other classes are Emo-symbol and Emo-gesture, the former with a whole battery of pictograms for things, animals, places etc.

4.6.3 Sequential searches: Sub-corpora on the fly

A sequential search or sub-query (SQ search) can be described as one search being run on the result of another. The technique is useful for pre-filtering, i.e. creating a subcorpus on the fly. For instance, the first search could ask for a domain tag, a positive sentiment, an emoji skepticism marker or simply a list of topic-defining lemmata (e.g. *foreigner, immigrant,* refugee). The second, sequential search can then look for certain linguistic patterns in this subcorpus, e.g. associated adjectives or imperatives.



Base emo-angry.*

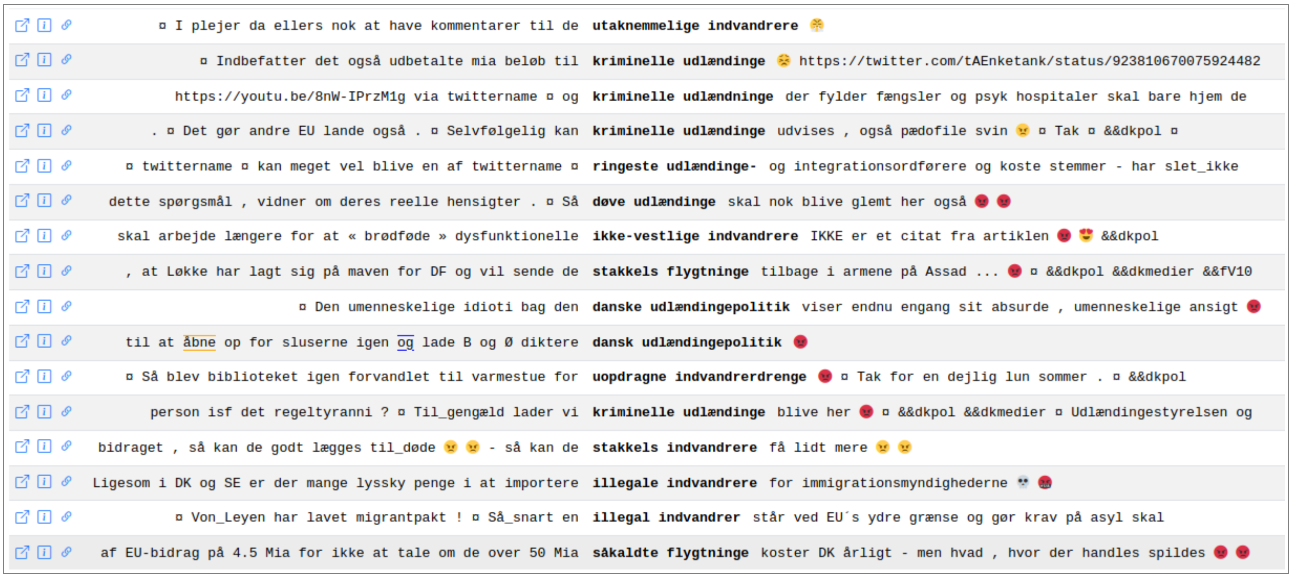
☑ Adjective

Base (udlænding|in

Query [lex="emo-angry.*"]

SQ [pos="ADJ"] [lex="(udlænding|indvandrer|flygtning).*"]

The resulting concordance will center on the hits from the final, second search, leaving the result from the first search present, but unmarked. You can easily expand a normal search into an SQ search by adding a sub-query search pattern in the SQ field, under the ‘Query’ field. In the graphical interface, press the blue ‘Toggle Sub-Query’ button to open a second row of word boxes. As with the first (upper) search, any choices in the SQ boxes will be converted and shown as CQP text fields by mouse-clicking into the ‘Query’ or ‘SQ’ fields. Do not forget to do this before pressing the ‘Search’ button, otherwise you may be running an old or incomplete search pattern.



¤ I plejer da ellers nok at have kommentarer til de	**utaknemmelige indvandrere**	😤
¤ Indbefatter det også udbetalte mia beløb til	**kriminelle udlændinge**	😣 https://twitter.com/tAEnketank/status/923810670075924482
https://youtu.be/8nW-IPrzM1g via twittername ¤ og	**kriminelle udlændninge**	der fylder fængsler og psyk hospitaler skal bare hjem de
. ¤ Det gør andre EU lande også . ¤ Selvfølgelig kan	**kriminelle udlændinge**	udvises , også pædofile svin 😠 ¤ Tak ¤ &&dkpol ¤
¤ twittername ¤ kan meget vel blive en af twittername ¤	**ringeste udlændinge-**	og integrationsordførere og koste stemmer - har slet_ikke
dette spørgsmål , vidner om deres reelle hensigter . ¤ Så	**døve udlændinge**	skal nok blive glemt her også 😡 😡
skal arbejde længere for at « brødføde » dysfunktionelle	**ikke-vestlige indvandrere**	IKKE er et citat fra artiklen 😡 😍 &&dkpol
, at Løkke har lagt sig på maven for DF og vil sende de	**stakkels flygtninge**	tilbage i armene på Assad ... 😡 ¤ &&dkpol &&dkmedier &&fV10
¤ Den umenneskelige idioti bag den	**danske udlændingepolitik**	viser endnu engang sit absurde , umenneskelige ansigt 😡
til at åbne op for sluserne igen og lade B og Ø diktere	**dansk udlændingepolitik**	😡
¤ Så blev biblioteket igen forvandlet til varmestue for	**uopdragne indvandrerdrenge**	😡 ¤ Tak for en dejlig lun sommer . ¤ &&dkpol
person isf det regeltyranni ? ¤ Til_gengæld lader vi	**kriminelle udlændinge**	blive her 😡 ¤ &&dkpol &&dkmedier ¤ Udlændingestyrelsen og
bidraget , så kan de godt lægges til_døde 😠 😠 - så kan de	**stakkels indvandrere**	få lidt mere 😠 😠
Ligesom i DK og SE er der mange lyssky penge i at importere	**illegale indvandrere**	for immigrationsmyndighederne 💀 🤬
¤ Von_Leyen har lavet migrantpakt ! ¤ Så_snart en	**illegal indvandrer**	står ved EU´s ydre grænse og gør krav på asyl skal
af EU-bidrag på 4.5 Mia for ikke at tale om de over 50 Mia	**såkaldte flygtninge**	koster DK årligt - men hvad , hvor der handles spildes 😡 😡

Another, simple way of exploiting the SQ option is for finding sentences with two independent features, where the two hits do not have to be adjacent or dependency-linked.

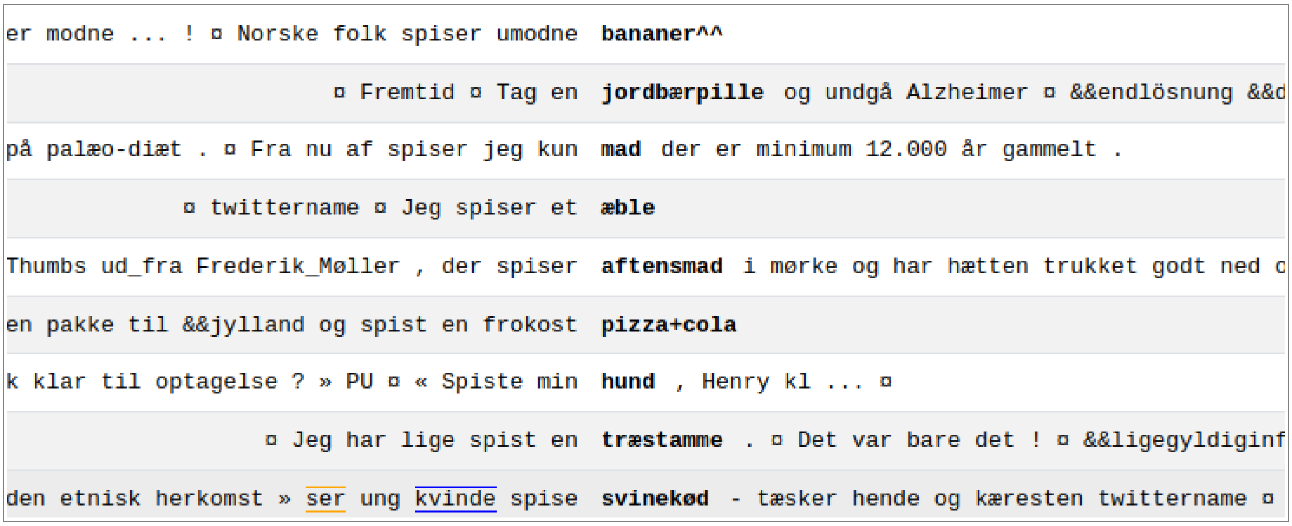
4.6.4 Structural searches: Dependency

CorpusEye supports dependency searches in dependency-annotated corpora. More specifically, you can address dependent-head and sibling-sibling (sister) relations through ‘h_...’ and ‘s_...’ fields, respectively. ‘h_pos’, for instance, means part-of-speech of the head, and ‘s_func’ the function of a sibling.

Dependency searches can be used to examine selection restrictions (or other relations) independently of distance in the sentence. Thus, we can look for direct objects of eating verbs in the Danish Twitter corpus:

[pos=”N” & func=”<ACC” & h_pos=”V” & h_sem=”fn:eat”]

In the resulting concordance, dependency links can be made visible by touching a word with mouse-over – this will highlight the word itself in blue and its dependency head in orange.



er modne ... ! ¤ Norske folk spiser umodne	**bananer^^**	
¤ Fremtid ¤ Tag en	**jordbærpille**	og undgå Alzheimer ¤ &&endlösnung &&d
på palæo-diæt . ¤ Fra nu af spiser jeg kun	**mad**	der er minimum 12.000 år gammelt .
¤ twittername ¤ Jeg spiser et	**æble**	
Thumbs ud_fra Frederik_Møller , der spiser	**aftensmad**	i mørke og har hætten trukket godt ned o
en pakke til &&jylland og spist en frokost	**pizza+cola**	
k klar til optagelse ? » PU ¤ « Spiste min	**hund**	, Henry kl ... ¤
¤ Jeg har lige spist en	**træstamme**	. ¤ Det var bare det ! ¤ &&ligegyldiginf
den etnisk herkomst » ser ung kvinde spise	**svinekød**	- tæsker hende og kæresten twittername ¤

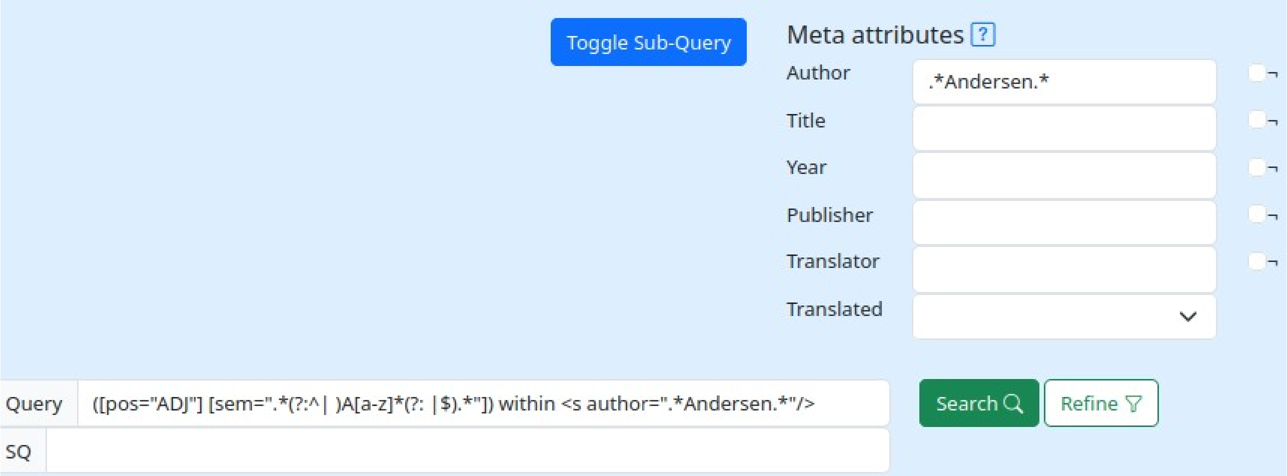
4.6.5 Meta attributes

Given that a corpus has a systematic use of meta attributes, these can be part of a search mask, too. For instance, literary corpora (present for da, pt, eo, no) commonly use author, title and publication year, and the same could be used for newspaper articles or blogs. Social media corpora often need to be anonymized, but may still contain time stamps or topics (hashtags). And even in a monolingual corpus, it may be necessary to flag the original language of translated texts – or speeches, as would be the case for the Europarl corpus of European parliamentary debates.

Meta attributes are part of the <s ...> tag linked to each sentence in a corpus. In CQP format, these can be searched by enclosing the rest of the search mask in round parenthesis and adding ‘within <s meta=”xxx”>’. For instance, for exploring which animal attributes the Danish author H. C. Andersen used in his tales, the meta restriction would be ‘author=...’:

([pos="ADJ"] [sem=".*(?:^|)A[a-z]*(?: |$).*"]) within <s author=".*Andersen.*"/>

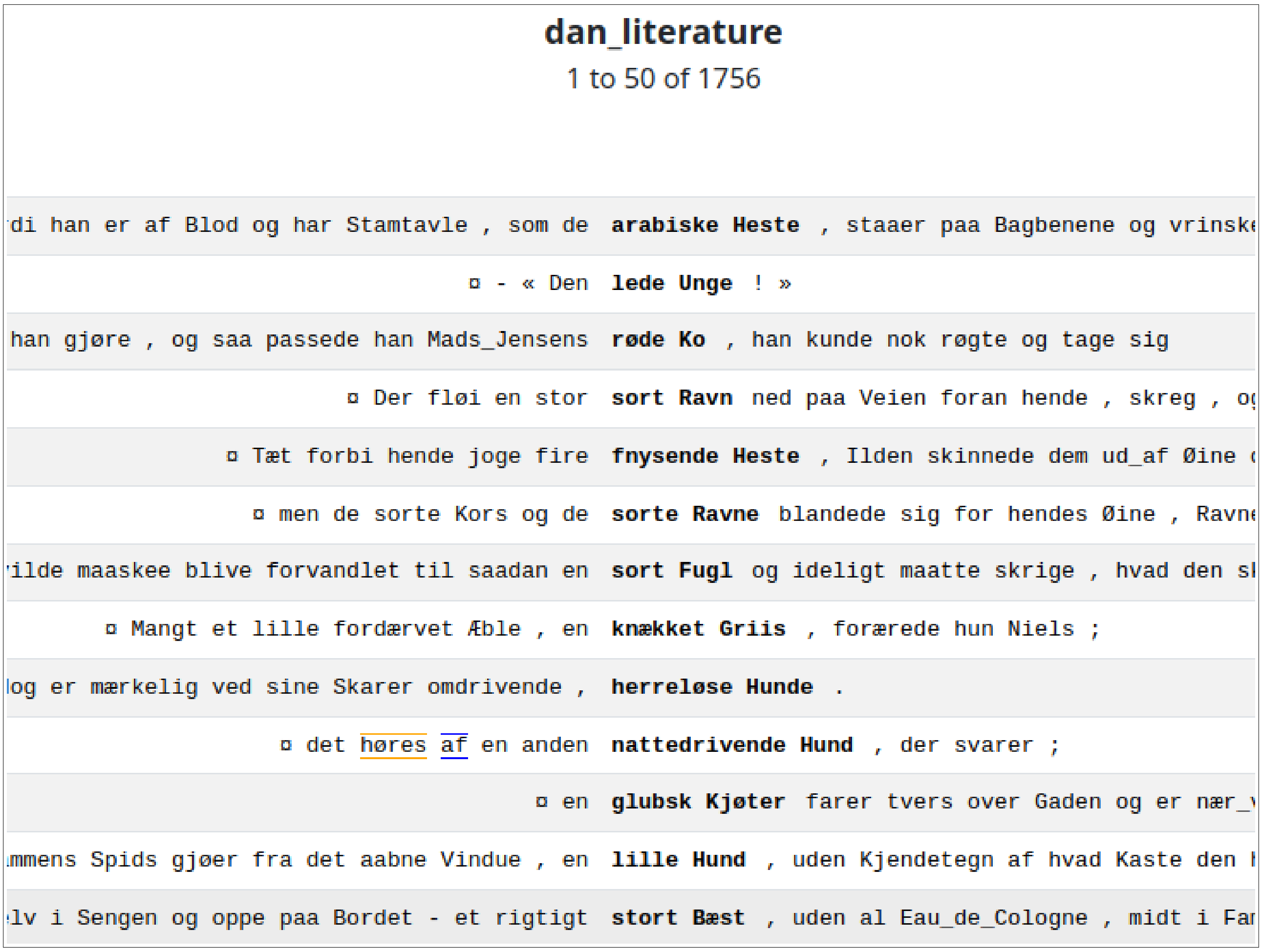
In the graphical search interface, meta attributes can be accessed at the bottom, next to the ‘Sub-query’ button, just above the ‘Search’ button:



Author: .*Andersen.*
Title:
Year:
Publisher:
Translator:
Translated:

Query: ([pos="ADJ"] [sem=".*(?:^|)A[a-z]*(?: |$).*"]) within <s author=".*Andersen.*"/>
SQ:

The resulting concordance will look like this:



dan_literature

1 to 50 of 1756

di han er af Blod og har Stamtavle , som de	**arabiske Heste**	, staaer paa Bagbenene og vrinske
¤ - « Den	**lede Unge**	! »
han gjøre , og saa passede han Mads_Jensens	**røde Ko**	, han kunde nok røgte og tage sig
¤ Der fløi en stor	**sort Ravn**	ned paa Veien foran hende , skreg , o
¤ Tæt forbi hende joge fire	**fnysende Heste**	, Ilden skinnede dem ud_af Øine
¤ men de sorte Kors og de	**sorte Ravne**	blandede sig for hendes Øine , Ravne
ilde maaskee blive forvandlet til saadan en	**sort Fugl**	og ideligt maatte skrige , hvad den s
¤ Mangt et lille fordærvet Æble , en	**knækket Griis**	, forærede hun Niels ;
og er mærkelig ved sine Skarer omdrivende ,	**herreløse Hunde**	.
¤ det høres af en anden	**nattedrivende Hund**	, der svarer ;
¤ en	**glubsk Kjøter**	farer tvers over Gaden og er nær_
mmens Spids gjøer fra det aabne Vindue , en	**lille Hund**	, uden Kjendetegn af hvad Kaste den
lv i Sengen og oppe paa Bordet - et rigtigt	**stort Bæst**	, uden al Eau_de_Cologne , midt i Fa

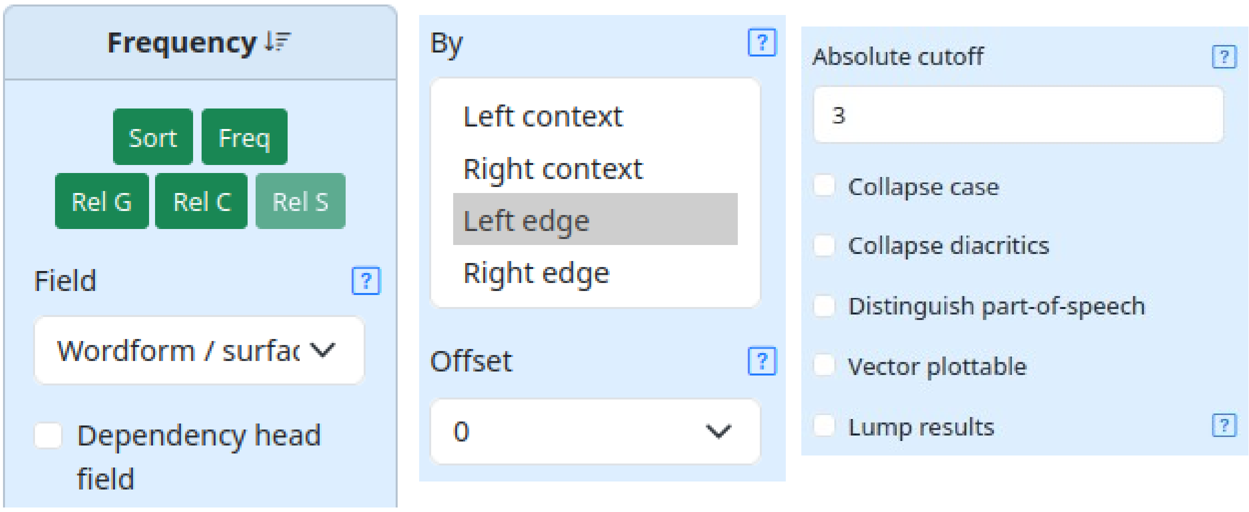
Frequency-sorting this as an n-gram list will show that Andersen liked to talk about “little” (cute) animals, and also what he associated with non-cute animals, e.g. ‘stygge muldvarp’ (mean/ugly mole), ‘vilde svaner’ (wild swans).

Meta attributes can be used to generate bar charts counting/grouping search results by author, title, year, language etc. For this, use the ‘Group by’ section in the statistics side bare (cp. section 6).

5 Statistics

5.1 Frequency lists

The left side bar on *CorpusEye’s* concordance pages is used for statistical analysis. Words in a given position relative to the search can be sorted not only alphabetically *(Sort)*, but also for frequency *(Freq)*. The default is to count and frequency-sort the words at the left edge of the (bold-faced) hit string, but this can be changed to right edge, left context or right context. You can specify positions inside the string or a more distant context by setting the *offset* value, counting right (+) or left (minus) from a standard edge or context position.



Field: Wordform / surfac

□ Dependency head field

Offset: 0

Absolute cutoff: 3

□ Collapse case

□ Collapse diacritics

□ Distinguish part-of-speech

□ Vector plottable

□ Lump results

The list header contains the corpus name and the overall number of hits in this corpus. If several corpora were searched at the same time, individual frequency list will be shown for each. The list itself shows the individual frequencies of the words encountered at sort position. Different metrics are provided in five columns. From right to left, n*um* is the absolute number of occurrences, *freq/conc* the percentage out of all hits (i.e. the concordance), and *freq/corp* is the in-corpus frequency normalized to 1:100,000,000. Finally, weighted or relative frequencies are also shown (*freq/norm)*, relative to either the corpus in question *(C:freq/norm)* or the global/combined data set for this language *(G:freq/norm)* .

Frequency cut-off (Absolute cut-off)

Large, un-curated corpora, especially internet and social media-corpora, contain a lot of non-dictionary words with a very low individual frequency. These may be true, but rare words, but may just as well be spelling or analysis errors, or even corpus compilation artefacts. In an absolute frequency list, these so-called *hapaxes* will automatically "disappear" at the bottom of the list, but if you use relative frequencies to prioritize words that are typical for the search context, hapaxes may appear higher up in the list exactly *because* they are rare in the language as such (Rel G) or at least in the corpus as a whole (Rel C). If this "dilutes" your frequency list and makes it less readable and informative, try using the frequency cut-off field. Here, you can specify for a minimum of occurrences for the words to be included in the list. Setting the value to 3 means that word with only 1 or 2 occurrences (in context) will be excluded from the list. For a 10-page frequency list (i.e. 500 lines with a page size of 50), it is probably safe to use thresholds as high as 5 or 10[10].

Collapse case or diacritics

These options are meant to reduce the number of lines in a frequency list or visualization chart and make them more readable, removing distinctions merely caused by spelling errors or spelling variation. Note that 'collapse case' will lump different case variants under the most frequent form, which may not necessarily be all lower-case. The same is true for 'collapse diacritics', which will

[10] According to Zipf's law, there is an inverse relation between rank and frequency. Typically, the lower half of a frequency list will be single occurrences, and the next 25% above that, words occurring twice. Only the upper 12.5% will be words occurring 4 times or more.

normally mean that words with missing accents will be grouped under the standard spelling (with an accent) rather than vice versa.

Distinguish part-of-speech

This button adds POS when sorting for wordform or lemma, reducing ambiguity in the resulting frequency list. For instance, the English word *'top'*, for instance, can be either a noun, an adjective or a verb, and these cases will be split up and show up in different places in the frequency list if the POS box is checked.

Of course, if you are sorting a position within the search mask, and if this position already had a POS condition in your search, nothing is gained, as there will only be one POS represented in the concordance for this word.

Also note that sorting for relative frequency doesn't work with the POS checkbox on, because the background databases only contain information on words or lemmas, not for word+POS or lemma+POS. So using this option is only relevant for simple, absolute frequence sorting ('Freq'-button).

Page size

Like long concordances, long frequency lists will be shown page by page. Page size is 50 lines by default, but can be set to a different value in the 'Page size'-menu at the bottom of the statistics sidebar. You can navigate between pages using the page bars a the top or bottom. Use the arrows for a page-by-page inspection, or jump[11] to a specific page using a number field or – if shown – a number menu.

TSV export

Frequency lists may be downloaded as a spreadsheet table, using the green 'Download TSV' button in the upper right corner. Note that the download is not page-by-page, but for the entire list, which may be useful for lexicography and for the systematic inspection of longer lists using external sorting or filtering tools.

5.1.1 Sorting fields

As a default, wordforms are used for sorting, but the 'Field' menu allows the user to sort for any other field, i.e. lemma, POS, morphology, syntactic function, semantic class or semantic role. Further sort options are the same fields, but for the word's dependency head instead.

Sorting fields and sorting positions that were not literatim part of the search pattern are more informative than those that were. Thus, sorting the search *[word="house"]* for wordform & left edge will not yield a list, but a single line. Conversely, wordform or lemma sorting makes sense for context positions, or if the search string contained higher-level categories (e.g. *[sem="food"])* will yield a meaningful list of words that are food items. In this case, not least for highly inflecting languages, lemma sorting is preferable to wordform sorting, and will provide a more "dictionary-like" lexical break-down of hits or contexts.

[11] Jumping forward to a higher page number can be useful if sections of the corpus are assumed to differ from each other in principled ways, as may be the case with a corpus covering different periods, topics or authors. In these cases, pages at the other end of the corpus are likely to contain different examples and a different lexical spread.

5.1.2 Relative frequencies

Relative frequencies are used for weighting. Instead of absolute frequencies, a relative measure is computed and used for sorting instead. Here, words that do occur in the search string concordance, but are rare in the overall corpus *(C:freq/norm)* or in the language as such *(G:freq/norm)*, will get higher values than words that have a similar absolute frequency in the concordance, but are frequent words to begin with, i.e. in other, general contexts, too. Ranking for relative frequency will therefore move typical or interesting hits towards the top of the list. For instance, when searching for immigrant and foreigner nouns with a determining adjective to the left, and then sorting the adjectives (i.e. the left search edge) for lemma, nationality adjectives, but also the word *criminal* will rank higher in a relative-frequency sorting than in a neutral (absolute) frequency sorting.

In mathematical terms, relative frequencies of the *G:freq/norm* type are computed by dividing the actual frequency by a standard "lexical" frequency taken from a multi-genre mix of background corpora (itself normalized to 1:100,000). The precise metric is the square of the local frequency (*freq/conc*) divided by overall lexical frequency, times 100,000. A word with an in-concordance frequency of 1%, and a norm frequency of 1:10,000 will receive a ranking value of 1. The same goes for a 10% word with a 1:100 norm frequency or an 0.1% word with a 1:1,000,000 norm frequency. If a word w1 in a concordance list has a *freq/norm* value 900 times higher than another word w2 in the same list, this can either mean that the w1 is 30 times more frequent than w2 in the search context (because local frequencies are squared), or that the standard frequency of w1 is 900 times lower than that of w2. Standard frequencies are set at a minimal value of 1:10,000,000 for rare words. In order to compensate for spelling errors and individual proper nouns, words with only one or two occurrences are frequency-punished with a factor 0.01 and 0.02 respectively.

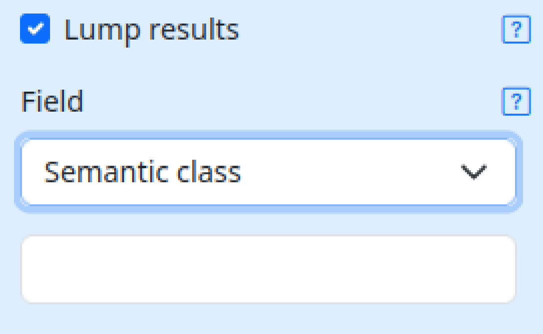
5.1.3 Multiple corpora

You can choose to search more than one corpus at a time. Just tick off what you need in the list of corpora.

In multi-corpus mode, CorpusEye will show concordance hits for one corpus after another. Frequency counts will be show in parallel, first the sum result to the left, then the individual corpora. A word-click in one of the frequency lists will lead you back to a concordance for that specific corpus only.

The multi-corpus option is meant to compare different types of corpora, not as a catch-all search for the whole corpus set for that language. So preferably, don’t choose more than 2-4 corpora at a time, especially with larger corpora, to avoid server strain and processing lag, and to avoid a too-crowded output page.

5.1.4 Category lumping


☑ Lump results

Field

Semantic class

Sometimes, a sorting field will yield more distinctions than what is intended or desirable for clarity. To constrain the number of types in the frequency lists, field values can be lumped. For ignoring orthographical variation in wordform or lemma lists, a simple method is to collapse upper/lower case and/or diacritics. Both choices are available as tick boxes in the statistics sidebar, under the *offset* menu.

Another case, however, are syntactic and semantic categories. Not least for the latter, more coarse-grained distinctions are sometimes desirable. This can be achieved by providing a list or "sortables", specifying what is to appear in the frequency list, and how to lump categories into "super-categories".

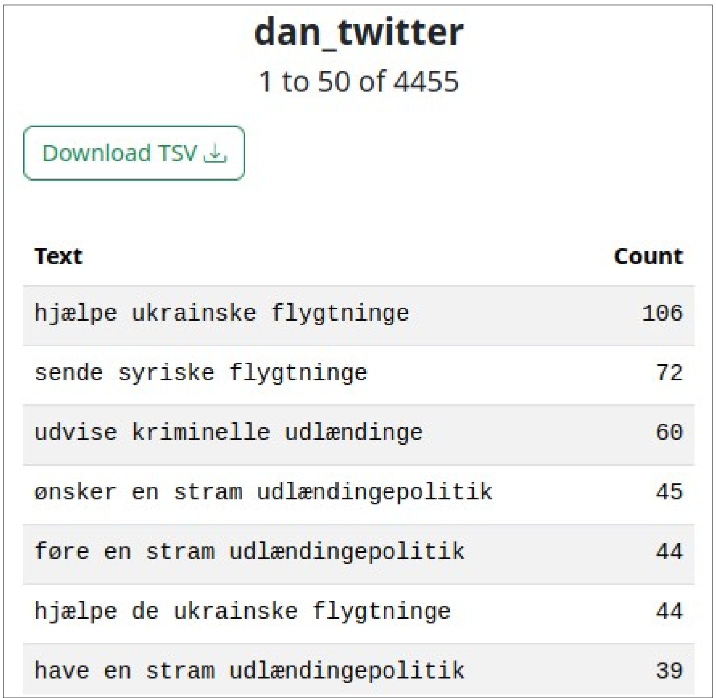
You can enter a semicolon-separated list of lumped "sortables" under the 'Field'-menu. Due to how their shallow hierarchy is implemented in category names, semantic class categories can easily be lumped into super-categories by using regular expressions, e.g.:

H.;A.*;L.*;tool.*;food.**

where *H.** lumps *H, Hprof, Hfam, Hideo, Hnat, Hetn, Hage, Hsick* etc., and *tool* lumps *tool-mus, tool-shoot, tool-l, tool-s, tool-light* etc.

As a welcome side-effect, category lumping also handles unresolved ambiguity. For instance, rather than distinguishing between A, B, C and ambiguous A B, A C, B C and A B C, a lumping list of A;B;C will use only the first occurrence of a tag in a tag line, meaning that e.g. A C will count as A, and not open a separate ambiguous category line of A C.

5.2 N-grams


dan_twitter

1 to 50 of 4455

Text	Count
hjælpe ukrainske flygtninge	106
sende syriske flygtninge	72
udvise kriminelle udlændinge	60
ønsker en stram udlændingepolitik	45
føre en stram udlændingepolitik	44
hjælpe de ukrainske flygtninge	44
have en stram udlændingepolitik	39

N-grams are usually defined as word chains of length n, i.e. bigrams, trigram, four-grams etc. In *CorpusEye* n-grams are the result of multi-word searches, for instance a noun with an adjective to the left plus possibly a governing verb like "love" or "hate".

Inspecting a frequency-sorted list of n-grams is a quick method of gaining an overview over topics, lexicon and fixed expressions used in a corpus in a certain type of context. As always, we suggest a “zooming technique”, going back from this (quantitative) bird’s eye view to a more qualitative inspection of individual examples by choosing one or other interesting line in the frequency list to get a concordance for the n-gram in question, checking for context and variation. Inspection is also a hatch against false results, since it will reveal analysis errors and non-literal usage.

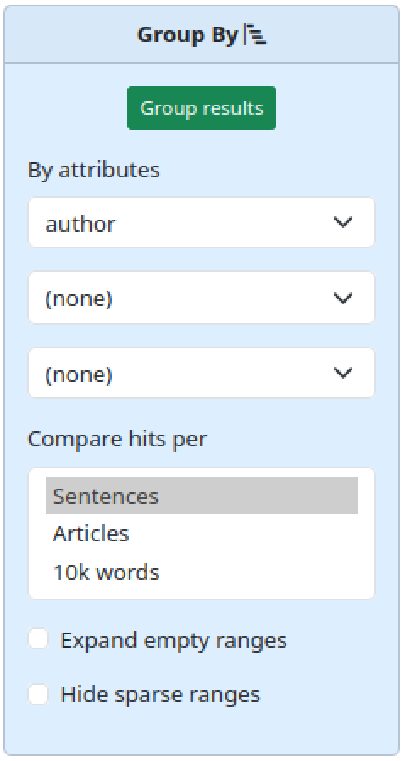
Since CorpusEye search fields can be optional or repeated, the result are not necessarily fixed-length n-grams in the classical sense. For the search below, for instance, the V-X-ADJ-N chain may contain zero, one or multiple instances (*) of a prenominal modifier (>N):

[pos="V"] [func=".*>N.*"]* [pos="ADJ"] [lex="(udlænding|indvandrer|flygtning).*"]

So the resulting n-gram frequency list will contain both “hjælpe ukrainske flygtninger” (3 words) and “hjælpe de ukrainske flygtninge” (4 words).

6 Bar charts: “Group by”

6.1 Grouping meta attributes



Group By

Group results

By attributes

author

(none)

(none)

Compare hits per

Sentences

Articles

10k words

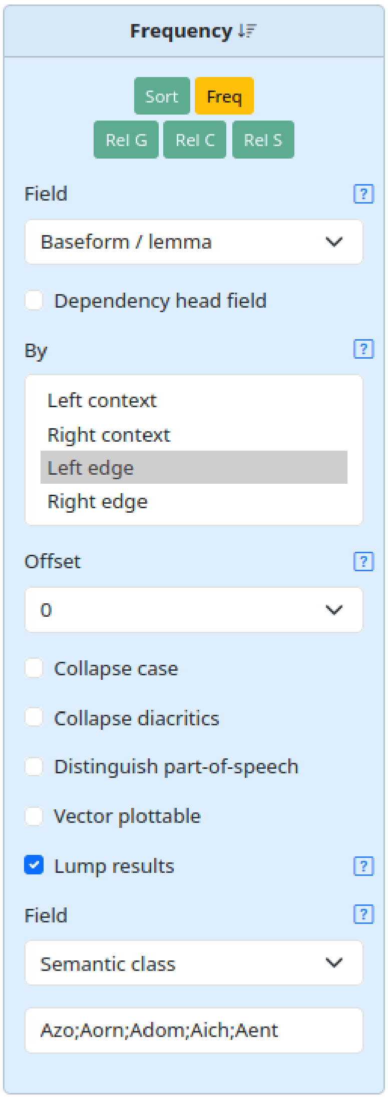
□ Expand empty ranges

□ Hide sparse ranges

For a corpus with meta annotation, any search result can be grouped by, for instance, author, year or title, if that information is available for the corpus in question. Depending on whether you do ‘group by’ directly from concordance view or only after generation of a frequency list, the result will be a simple histogram or a stacked bar chart. In the latter, each bar is color-stacked for the top results from the frequency list. In the histogram, all results are summed up and visualized as one-coloured columns.

Let’s say we have made a search [sem=”A[a-z]*”] in the Danish literature corpus and looking at the resulting concordance. If we now use the ‘Group by’ option with the meta attribute in the first menu set to “author”, we will get a bar chart visualization of the prevalence of animals in the various authors’ work, showing that some authors are much more into animals than others.

Note that up to 3 meta attributes can be combined for a more fine-grained comparison. Try also to group results by “title” rather than



Frequency

Sort Freq

Rel G Rel C Rel S

Field

Baseform / lemma

□ Dependency head field

By

Left context

Right context

Left edge

Right edge

Offset

0

□ Collapse case

□ Collapse diacritics

□ Distinguish part-of-speech

□ Vector plottable

☑ Lump results

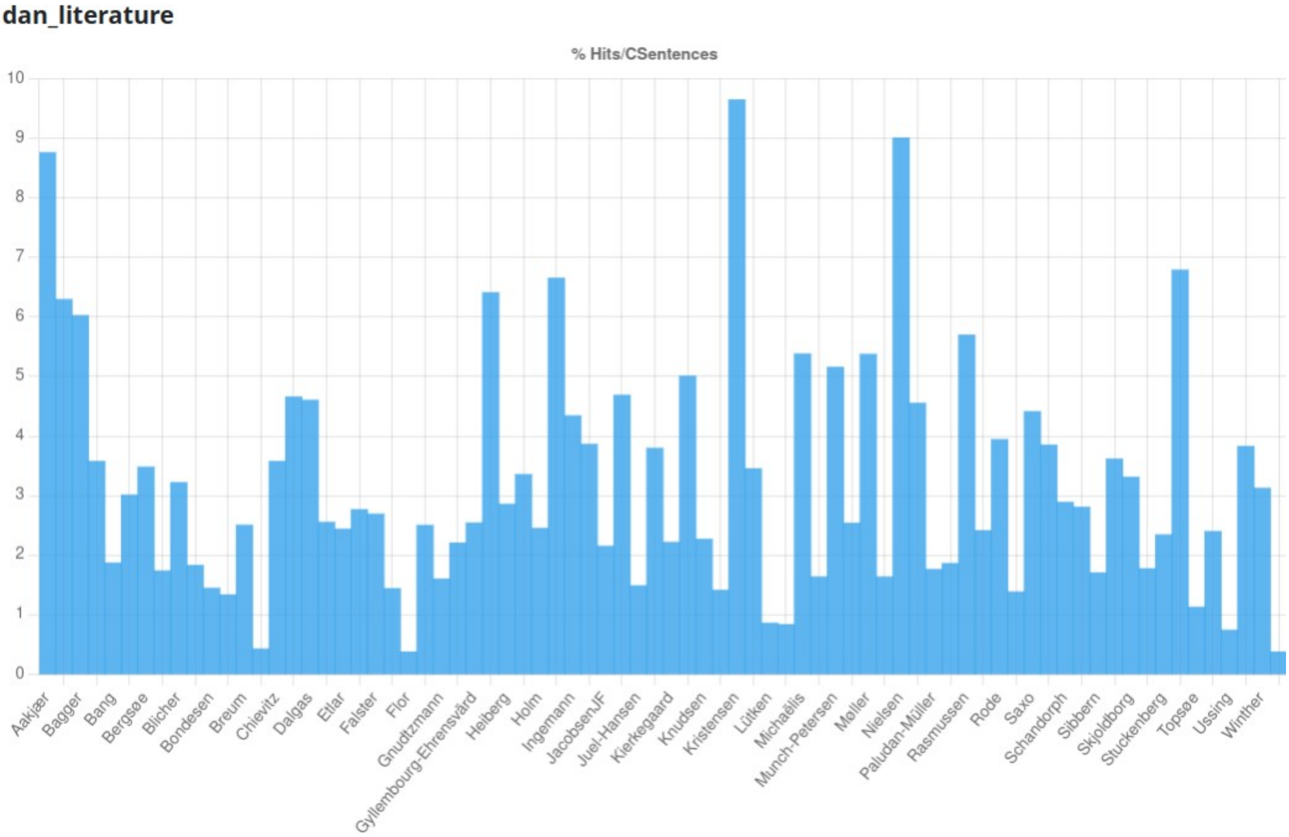
Field

Semantic class

Azo;Aorn;Adom;Aich;Aent

“author”, which will give a much longer chart (if necessary, use the horizontal scroll bar). If there are (too) many columns in the bar chart, there will not be space enough to provide a feature value (e.g. author name) below the column. However, you can still get this information by mouse-over-touching the column in question.

Hit statistics can be set to either ‘per sentence’ or ‘per 10,000 words’. The former is the default, but for cross-corpus comparisons, the latter may be more neutral (independent of sentence length).



Time histograms

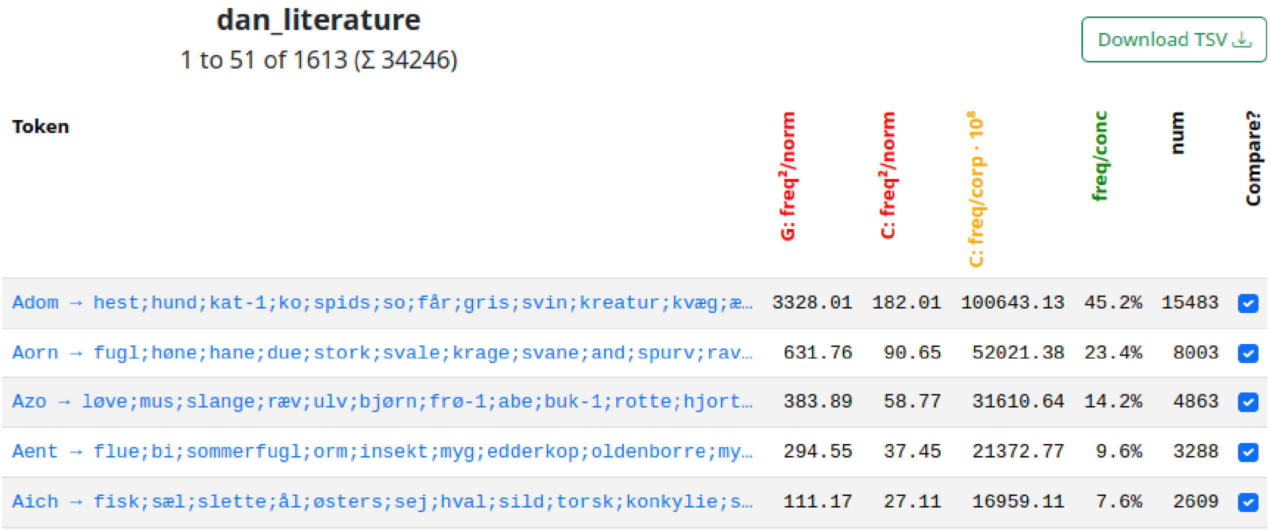
For corpora with a timeline, e.g. newspaper corpora or social media posts, the sorting sidebar will show a separate ‘histogram’ section, with various granularity choices for a time/date break-down (year, month, day etc.). Such time histograms are equivalent, in terms of chart type, with group-by-year charts.

For bothe time histograms and group-by-year charts, there may be empty date ranges or periods with very few data, e.g. depending on corpus harvesting in the case of a social media corpus. In these cases, if you want to ignore outliers that may be due to sparse data, choose ‘Hide sparse ranges’. If you want an evenly spaced timeline, choose ‘Expand empty ranges’.

6.2 Stacked bar charts

Next, let us go back to the original concordance view of the animal search example and create a frequency-list for “left edge” and “lemma”, using the ‘Lump results’ option for the semantic classes field (sem), with 5 groups: Azo (land animals), Aorn (birds), Adom (domestic animals), Aich (fish) and Aent (insects). The resulting list shows, as could be expected, that land animals are more frequently mentioned than birds, and that the trailing categories are insects and fish, in that order.

In the last column of the frequency list (‘Compare’), you will find check boxes for which items to include in the stacked bar chart, the maximum allowed number being 10. As a default, the system pre-selects the top 10 lines (or all lines, if there are fewer than 10).



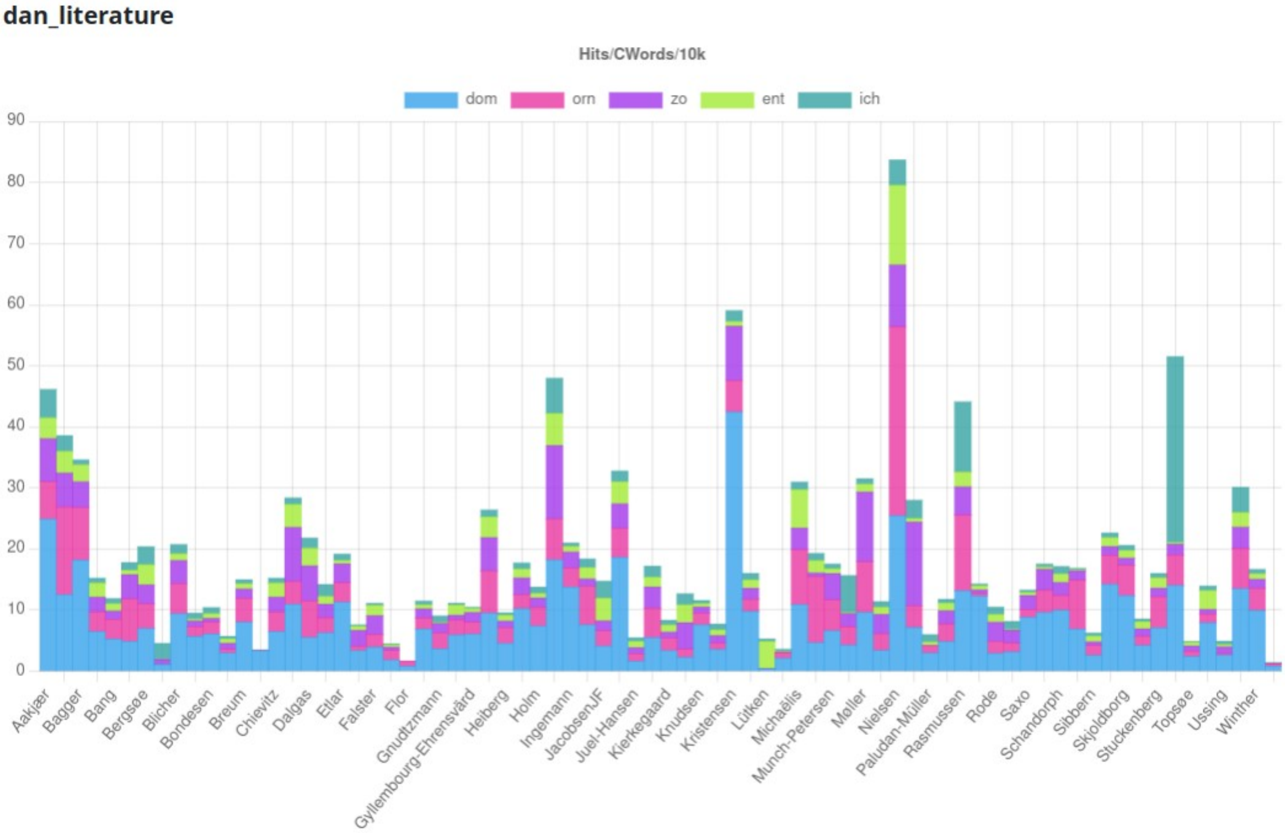
dan_literature

1 to 51 of 1613 (Σ 34246)

Token	G: freq²/norm	C: freq²/norm	C: freq/corp · 10^8	freq/conc	num	Compare?
Adom → hest;hund;kat-1;ko;spids;so;får;gris;svin;kreatur;kvæg;æ…	3328.01	182.01	100643.13	45.2%	15483	☑
Aorn → fugl;høne;hane;due;stork;svale;krage;svane;and;spurv;rav…	631.76	90.65	52021.38	23.4%	8003	☑
Azo → løve;mus;slange;ræv;ulv;bjørn;frø-1;abe;buk-1;rotte;hjort…	383.89	58.77	31610.64	14.2%	4863	☑
Aent → flue;bi;sommerfugl;orm;insekt;myg;edderkop;oldenborre;my…	294.55	37.45	21372.77	9.6%	3288	☑
Aich → fisk;sæl;slette;ål;østers;sej;hval;sild;torsk;konkylie;s…	111.17	27.11	16959.11	7.6%	2609	☑

We can now group these results by author and this time, we will get a *stacked* bar chart with colors representing the top items of the frequency list. The chart shows that some authors are more fond of birds or fish than of land animals. If we group by the meta attribute of ‘title’ rather than ‘author’, we get much bigger differences, with a strong focus on one animal group or other. You can also try not to use the ‘lump categories’ feature and work with a lemma frequency list instead, which will tell you which individual animals, bird or fish are being written about by who (or when).

Mouse-over-touching a stacked column will not only reveal the value of the grouped meta attribute, but also provide frequency details for the stacked and coloured sorting categories.



Below the bar chart you will find a detailed break-down for each author (year, title etc.), with both absolute frequencies (Hits/CWords/10k), and relative frequencies per sentence or per article. '% Sentences' is the percentage of sentences that had hit, while '%Hits/CSentences' is the (somewhat higher) percentage of hits/sentences (as there may be more than one hit in a given sentence). Finally, for comparison, the size of the author, title or year subcorpus is provided in three different metrics – words, sentences and articles (texts).

Clicking on an author (year, title etc.), in the leftmost column in this table, will lead to a concordance with hits only for this particular author in isolation, for closer inspection or further statistics. This feature is one of the main strengths of *CorpusEye* as a research and corpus inspection tool – it will always allow you to go seamlessly back and forth between quantitative and qualitative methods, between statistics and inspection.

For further examples of stacked bar charts, see section 10.2 (language change) and 10.3 (distant reading).

7 Corpus inspection: Quantitative-qualitative

In corpus linguistics, neither qualitative nor quantitative methods can stand alone. On its own, qualitative inspection is inadequate for larger corpora and risks missing out on important patterns, while quantitative analysis lacks the depth and context for safely interpreting its results.

We therefore recommend going back and forth between the two in a kind of hermeneutic circle – starting out with a not too-specific search for something interesting we think we may find, and getting a statistical overview of the pattern or lexical feature in the corpus in question. We can then inspect individual items from a frequency list or bar chart and examine their context in a more targeted concordance. Next, we may decide to refine the original search pattern in order to sharpen or change our focus, to accommodate for any insights gained, and to prepare for a new round of statistics.

CorpusEye supports this iterative “zooming” method (Kalwa, 2019) as a matter of course, throughout the interface, by allowing you, on the one hand, to do statistics on any concordance result, and, on the other hand, to seamlessly click-navigate from any item in a frequency list or bar chart back to concordance inspection, focusing on that item.

In addition, search result pages will keep a copy of the search mask open at the bottom of the page, so you can easily modify and adapt your search based on current results, using either the CQP input field or the graphical ‘Refine’ search. Also at the bottom of the page, you can change or expand your choice of corpora. This can be used for corpus comparison, but also to harmlessly scale more experimental searches and quantitative analysis: Start with a small corpus that will allow fast results even for complex searches, modify the search based on preliminary quantitative and qualitative results, only then to run the “mature” search on a larger corpus.

8 Semantic vectors: Scatter plots

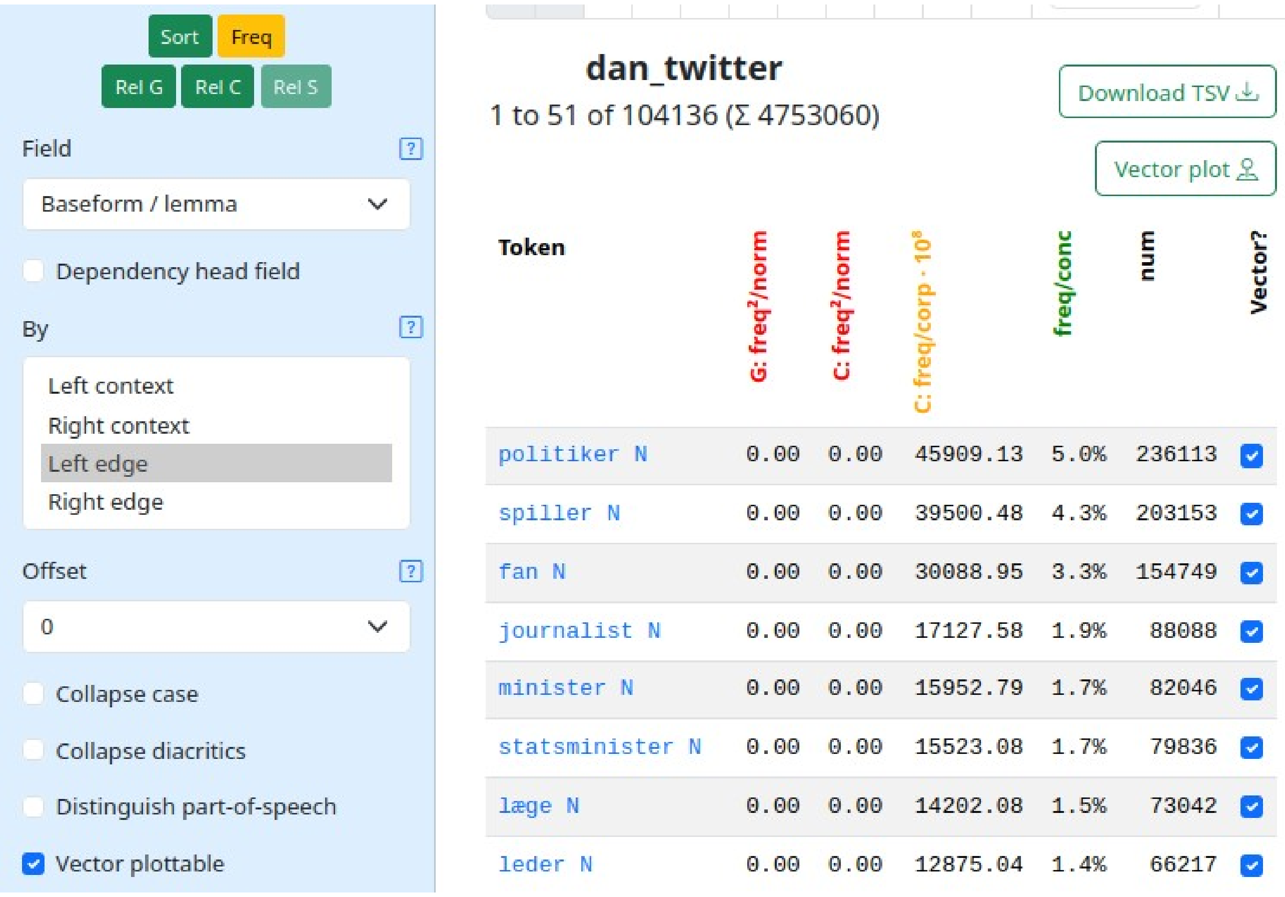
As a relatively new option, *CorpusEye* supports the measurement of semantic similarity and makes it possible to scatter-plot search results in a two-dimensional semantic space with user-defined x- and y-axis.

To do so, *CorpusEye* maintains a semantic vector database[12] for all “dictionary words” (lemma-POS combinations) of sufficient frequency in each corpus enabled for this tool. Semantic word vectors are machine-learned word embeddings in a high-dimensional contextual space, with coordinates computed from co-occurring words, where the position of a word approximates its meaning, distribution and similarity to other words[13].

In order to prepare for a semantic scatter plot, click ‘vector-enabled’ in the statistics sidebar and create a frequency list. The field menu will be ignored and automatically set to ‘lemma’, since the tool uses lemma-POS combinations in its database. For instance, to explore gender stereotypes linked to professions, run a search for the noun-semantic class <Hprof> (profession). In CQP speak this would be: [sem=”Hprof”]. For the Danish Twitter corpus, after frequency-sorting the resulting concordance, the screen will look like this:

[12] The databases are created using word2vec and tensor flow software for modelling, applying the skipgram method during training.

[13] This statistical approach to semantics is very different from the discrete and prototypical semantic classes used in the CG annotation of the *CorpusEye* corpora. The method’s strength is measurability, coverage and self-adaptability to genre, diachronics etc. However, for the individual word, classification and similarities may be misleading because of issues like semantic ambiguity, sparse data, negation, quantification and truth/likelihood/doubt markers in the context. Optimally, results should therefore be checked by other methods and qualitative inspection.



Field

Baseform / lemma

□ Dependency head field

By

Left context
Right context
Left edge
Right edge

Offset

0

□ Collapse case

□ Collapse diacritics

□ Distinguish part-of-speech

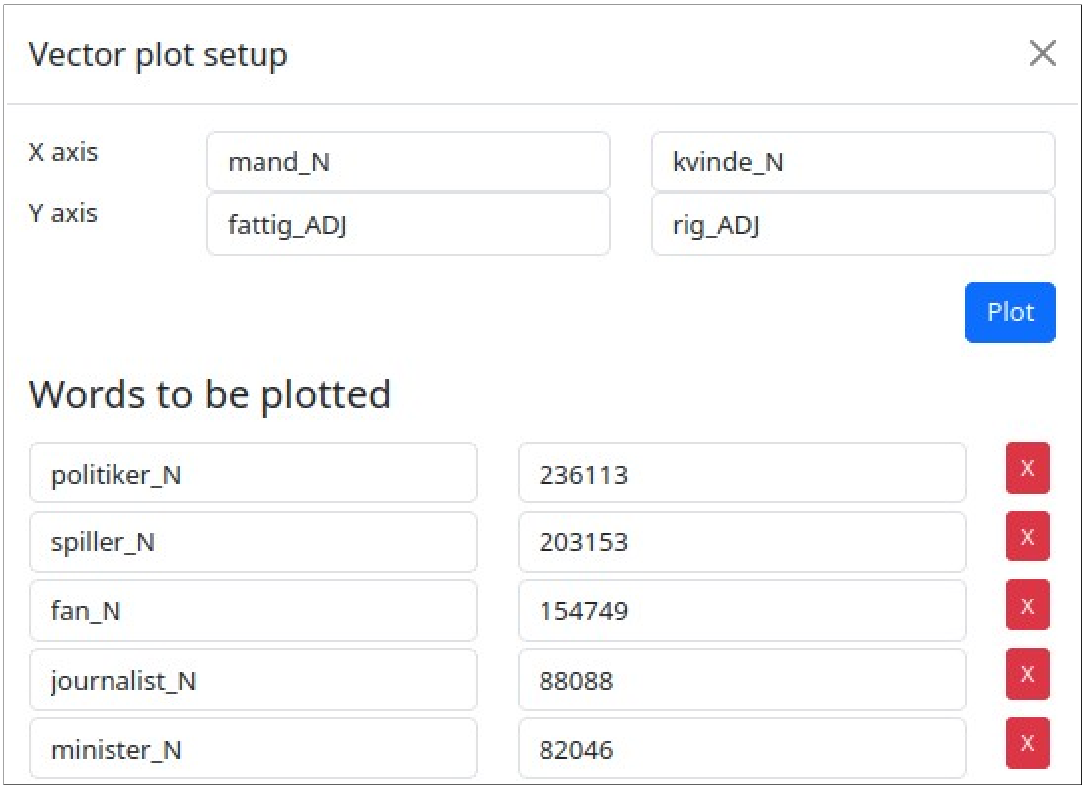
☑ Vector plottable

dan_twitter

1 to 51 of 104136 (Σ 4753060)

Token	G: freq²/norm	C: freq²/norm	C: freq/corp · 10^8	freq/conc	num	Vector?
politiker N	0.00	0.00	45909.13	5.0%	236113	☑
spiller N	0.00	0.00	39500.48	4.3%	203153	☑
fan N	0.00	0.00	30088.95	3.3%	154749	☑
journalist N	0.00	0.00	17127.58	1.9%	88088	☑
minister N	0.00	0.00	15952.79	1.7%	82046	☑
statsminister N	0.00	0.00	15523.08	1.7%	79836	☑
læge N	0.00	0.00	14202.08	1.5%	73042	☑
leder N	0.00	0.00	12875.04	1.4%	66217	☑

The next step is to click on the ‘Vector plot’ button in the upper right corner. This will open a configuration window for the upcoming scatter plot:



Vector plot setup

X axis	mand_N	kvinde_N
Y axis	fattig_ADJ	rig_ADJ

Words to be plotted

politiker_N	236113	X
spiller_N	203153	X
fan_N	154749	X
journalist_N	88088	X
minister_N	82046	X

The configuration pop-up will also allow you to (un-)select[14] which words to include in the scatter plot. This is useful for making the scatter plot more readable and more relevant by limiting the number of words included and by removing outliers and problematic cases, e.g. involving ambiguity or possible analysis errors.

More importantly, the pop-up also makes it possible to define the axes for the semantic similarity evaluation of your search results. Typically, but not necessarily, two words of opposing polarity are chosen for each axis, creating a 2-dimensional semantic space. Examples are:

- *mand_N – kvinde_N* [man – woman] or *macho_ADJ – feminin_ADJ*
- *ung_ADJ – gammel_ADJ* [young -old]

It is also possible to define complex vectors as the end points of an axis, by using semicolon-separated lemma lists:

- *forurening_N; fossil_ADJ – vedvarende_ADJ; grøn_ADJ; miljøvenlig_ADJ* [polution; fossil – renewable; green; environment-friendly]

You can also create a sentiment axis using emojis:

- *emo-happy.*_ADV – emo-angry.*_ADV; emo-sad.*_ADV*

A word's coordinates in the semantic scatter plot will then be calculated by comparing its own vector to the vectors of the four axis end points, in terms of "contextual" similarity (word embeddings similarity). A similarity (SIM) value of 1 means exact synonyms, while 0 means no correlation. For words with mutually exclusive contexts, this value can also become negative.

Using *man-woman* for the x-axis and *young-old* for the y-axis, a word's actual coordinates will be computed as follows:

x = SIM(*man,word* -1) – SIM(*woman,word* -1)

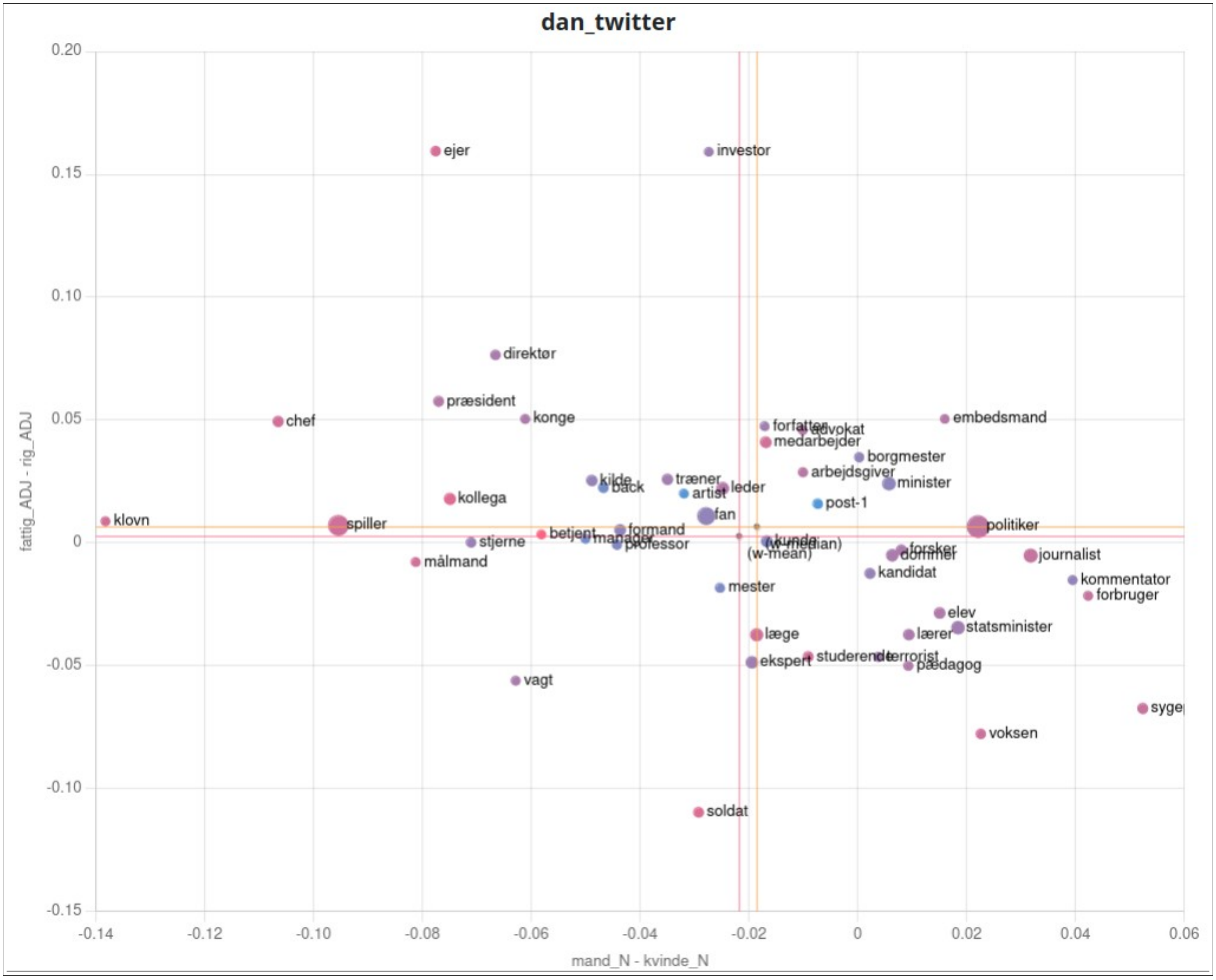
y = SIM(*young,word* -1) – SIM(*old,word* -1)

Applying these axes to the frequency list of professions, we get a scatter-plot with well-paid occupations at the top *(investor, director, president, boss, king)* and underpaid jobs at the bottom *(soldier, guard, nurse).* On the x-axis, we find jobs with high female participation to the right *(nurse),* and male-dominated professions to the left *(clown, player, goal-keeper, boss).*

Note that coordinates in the scatter plot do not necessarily correlate with actual medium salaries for a given profession, or the factual percentage of women working in it. Rather, vectors should be interpreted as linguistic manifestations of a language community's stereotypes.

Note also that the x- and y-zero-line are not necessarily centered, as the set of target words may be skewed toward one or other end of an axis. In the example, the vertical zero-line is situated asymmetrically to the right, leaving more space to male professions than to female ones, one interpretation being that 'profession' as a whole, as a concept, is more strongly associated with men than with women.

[14] This is also possible using the check-boxes next to the frequency list in the previous screen.



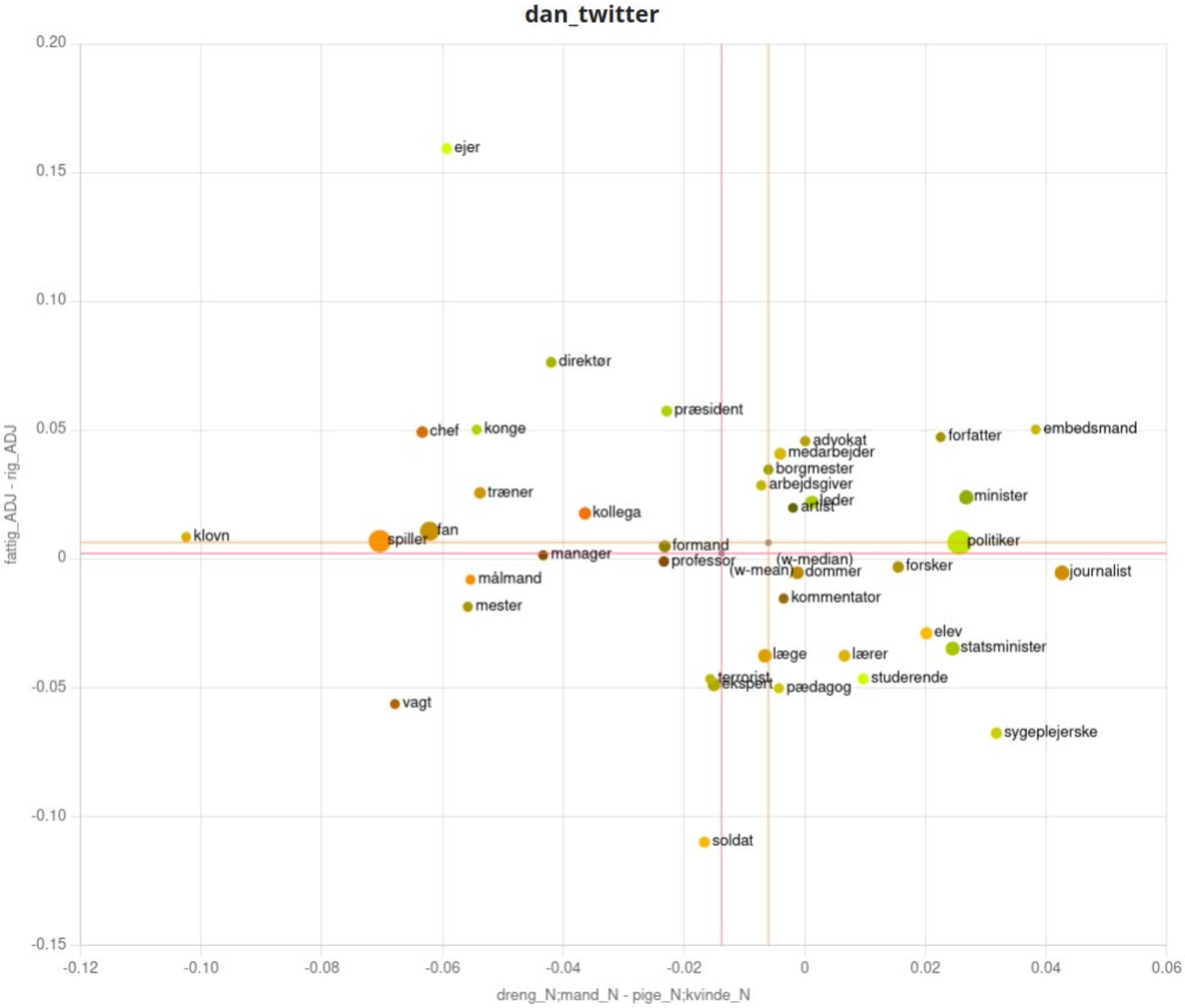
Mean and median

In order to provide alternative "ground zeroes", and to allow more relative interpretations, *CorpusEye* provides a purple line cross for the word cohort's mean coordinates, and an orange line cross for the coordinates of its median member. The cross point of the latter will always be a real word, while the former is a vector average not necessarily equal to that of a real word.

Dot size and colour

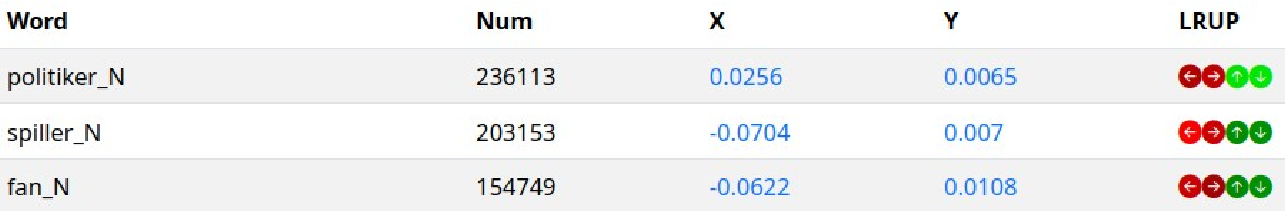
Dot sizes in the semantic scatter plots represent absolute frequency, so common words have large dots, and rare words have small dots. More specifically, it is the area, rather than the diameter, of a dot/circle that correlates with frequency.

Since dot coordinates are computed by subtracting to similarity values from each other, the absolute similarity with the words defining an axis is not visible. If a given word is related to both ends of an axis, even substantial vector similarities will cancel each other out. Therefore, in order to give an idea about absolute similarities, a colour coding scheme is used for the dots. The actual RGB color values are computed with a red component for the x-axis and a green component for the y-axis (and blue set to zero). If a word has a similarity of 0.4 or higher with either end of an axis, the respective axis colour component is set to 255 (maximum in the RGB system), and if the highest correlation with either end is zero or negative, the axis colour component is set to 0. Thus, if a word correlates highly with one of the ends of both axes, its dot colour will be yellow (red = 255, green = 255, blue = 0), if it correlates with nothing, its dot will be black (red = 0, green = 0, blue = 0).



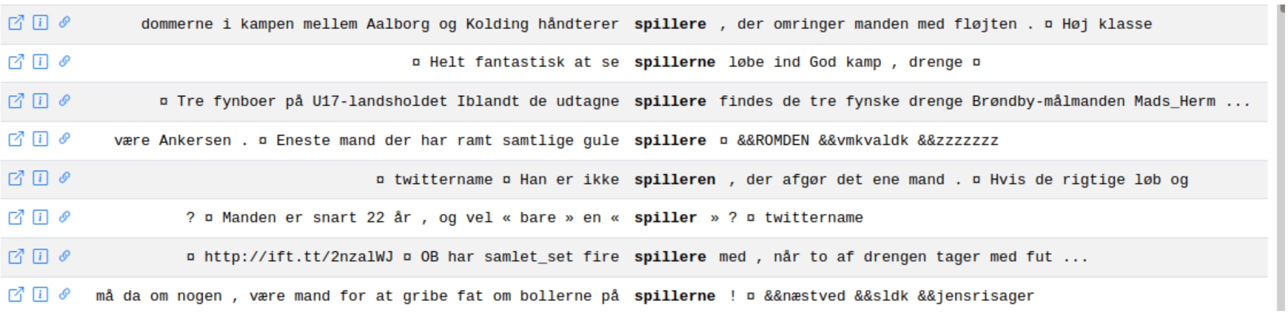
In the example plot of professions, red is the male-femal x-axis, green the poor-rich y-axis. Both 'spiller' (player) and 'politiker' (politician) are neutral on the y-axis, but the latter is stark green, the former orange-red. This means that politician (incl. the likewise green 'minister' and 'statsminister') do not strongly correlate with gender, even though the ultimate x-coordinate is on the female side. 'Politician' does, however, correlate strongly with 'rich' and 'poor', so the word's neutral placement on the y-axis means either that the two cancel each other out, or – more likely – that the strong association is a contextual one rather than true similarity, i.e. politician speakin about wealth and the poor. 'Spiller' (player), on the other hand, is orange-red because it does strongly correlate with gender, so the fact that the word is placed on the male side of the plot is significant. The same is true for the slightly less male 'chef' (boss) and 'målmand' (keeper). Note that the "richest" term, 'ejer' (owner) has a green dot, while the "poorest", 'soldat' (soldier) is orange. This means that ownership has a true with wealth and that its "malehood" is based on a weak correlation (i.e. absence of a visible red component), while soldiers are slightly, but safely male and that, for this word, being poor is only a relative rather than an absolute association (absence of a visible green component).

You can find the **exact values** for frequency as well as x- and y-coordinates in a table underneath the scatter plot, sorted by absolute frequency. The arrow column to the right (LRUP) colour-codes each word's association strength (vector similarity) with the four axis endpoints, left-right for the x-axis, and up-down for the y-axis. Clear, bright colours, i.e. light red and light green, mean a strong association, darker colours mean a weaker association, with brownish shades or black meaning almost or completely unrelated.



Word	Num	X	Y	LRUP
politiker_N	236113	0.0256	0.0065	←→↑↓
spiller_N	203153	-0.0704	0.007	←→↑↓
fan_N	154749	-0.0622	0.0108	←→↑↓

Note that the x- and y-coordinates are clickable. Like elsewhere in *CorpusEye*, it is possible to switch back and forth between quantitative-statistical tools and qualitative inspection/analysis. In this case, the click link will lead to a concordance, from the same corpus, containing matches with *both* the list word itself (1. column) *and* either or any word having been used to define the axis in question. For instance, clicking on the x-axis coordinate next to *spiller* vil display sentences with *spiller* co-occurring with one or more instances of *mand*, *kvinde*, *dreng* or *pige:*



dommerne i kampen mellem Aalborg og Kolding håndterer	**spillere**	, der omringer manden med fløjten . ¤ Høj klasse
¤ Helt fantastisk at se	**spillerne**	løbe ind God kamp , drenge ¤
¤ Tre fynboer på U17-landsholdet Iblandt de udtagne	**spillere**	findes de tre fynske drenge Brøndby-målmanden Mads_Herm ...
være Ankersen . ¤ Eneste mand der har ramt samtlige gule	**spillere**	¤ &&ROMDEN &&vmkvaldk &&zzzzzzz
¤ twittername ¤ Han er ikke	**spilleren**	, der afgør det ene mand . ¤ Hvis de rigtige løb og
? ¤ Manden er snart 22 år , og vel « bare » en «	**spiller**	» ? ¤ twittername
¤ http://ift.tt/2nzalWJ ¤ OB har samlet_set fire	**spillere**	med , når to af drengen tager med fut ...
må da om nogen , være mand for at gribe fat om bollerne på	**spillerne**	! ¤ &&næstved &&sldk &&jensrisager

Polarity

Note that the semantic axes need not be bipolar. If you just want to measure similarity with a given word or concept and can't think of an antonym, put it at plus (right/upper) end of an axis and use a completely unrelated "grounding word" at the other (left/lower) end, e.g. 'nil_N' or 'thing_N'. This vil move the zero point for that axis far to the left (x) or bottom (y) of the chart and provide more "unipolar" measurements. An example is the Danish compound *'husdyr'* (domestic animal), which does not have a "wild" counterpart. So here, for the other end of the axis, you can either define a complex vector using actual words for wild animals *(*e.g. *lion_N;tiger_N;wolf_N)* or make the axis unipolar using a neutral term.

Pitfalls

Semantic scatter-plots are nice to provide a general overview over a set of results, but there may be unexpected and counter-intuitive positions for one or other individual word, and semantic vector similarities should be seen as a point of departure for further, qualitative study rather than absolute truths.

- One possible problem is that antonyms may actually share a lot of context, or even the same context with only a negation or a quantifier as a difference. So if word A is used a lot together with word X, and B with ‘not X’ (rather than with X’ antonym Y), then A and B may end up with similar vectors.

- Two words that should intuitively belong to different ends of an axis, may end up with similar vectors because they are culturally or otherwise linked and therefor co-occurring in many sentences. For instance, on a ‘domestic animal’ scale, the Danish word for fox, *‘ræv’* may end up close to *‘høne’* (hen), because of the Danish idiom of ‘foxes asked to guard chicken’ *(ræven der vogter høns)*, used as a metaphor of placing responsibility on people who will exploit it to their own advantage.
- Semantically ambiguous words will get mixed vectors, because the embeddings use lemma+POS. Part-of-speech does resolve part of the ambiguity problem, but not wordclass-internal ambiguity. Even in semantically disambiguated corpora, semantic class is a separate tag. In principle, semantic “sub-lemmata” could be introduced, but that would be a complex change and make the interface less intuitive in other ways, not least by not being able to safely use dictionary entry forms in the lemma fields.

One way of handling these pitfalls, especially in a pedagogical or illustrative setting, is simply to remove such problematic words from the scatter plot using the check-boxes in the configuration pop-up. Another solution, for publication, would be to provide specific footnotes for the identified problematic words.

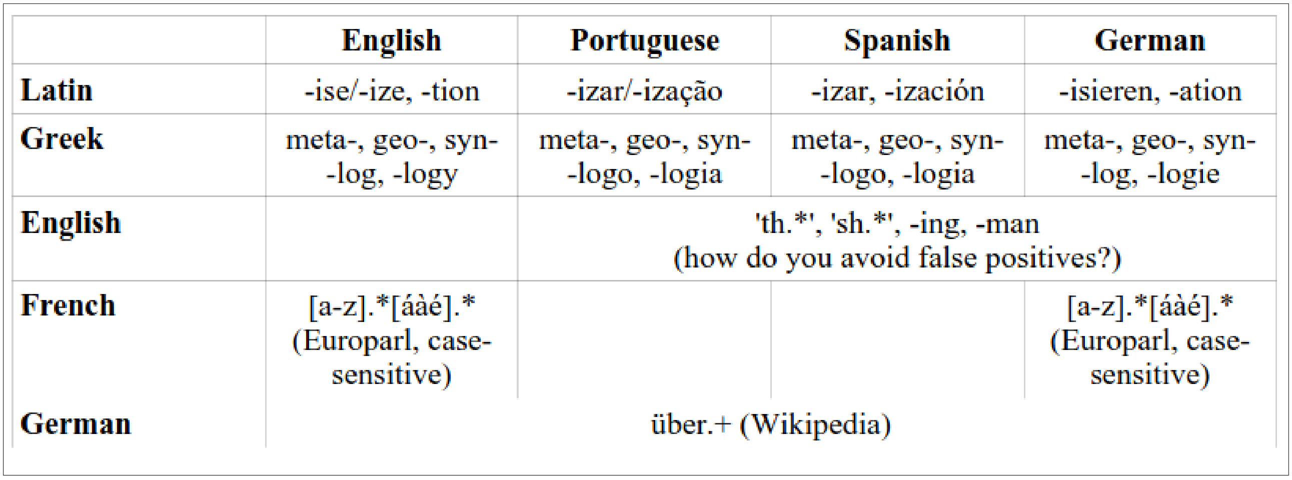
For further examples of semantic scatter plots, see section 10.4 (hate speech), 10.5 (gender studies) and 10.6 (brand profiling).

9 Use cases: Teaching

9.1 Language awareness exercises

A corpus interface can be a great tool for teaching language awareness, both at the university entrance and secondary school levels. The examples below are meant to provide a point of departure for designing one’s own exercises. For lexical variation exercises and others, where it makes sense, perform a frequency statistics once you have a concordance!

- **Spelling variation:** check alternative spellings, common errors and deprecated spellings for frequency and context. For instance,
  - how common is the unofficial double s in the inflected forms of ‘virus’ in Danish? [lex=”virus+e[nrt]”]
  - How common is the – once wrong – spelling ‘Spontanität’ in German, as opposed to ‘Spontaneität’? [lex=”Spontane?ität]
  - Differences between Brazilian and European Portuguese: Aids, aids, Sida, sida - what is “normal” in Portuguese - and where? Frequency?
- **Loan words:** Every language uses loan words from other languages, and often the words have been in the language so long that only an etymological dictionary will tell where they really came from. Compile lists of loanwords by using certain prefixes or suffixes. Can you find words with mixed "international" elements? Use regular expressions, e.g. '.*isation'!



	English	Portuguese	Spanish	German
Latin	-ise/-ize, -tion	-izar/-ização	-izar, -ización	-isieren, -ation
Greek	meta-, geo-, syn- -log, -logy	meta-, geo-, syn- -logo, -logia	meta-, geo-, syn- -logo, -logia	meta-, geo-, syn- -log, -logie
English		'th.*', 'sh.*', -ing, -man (how do you avoid false positives?)		
French	[a-z].*[áàé].* (Europarl, case-sensitive)			[a-z].*[áàé].* (Europarl, case-sensitive)
German	über.+ (Wikipedia)			

- **Grammatical variation:**
  - How do languages assign grammatical gender? Can gender change over time? For instance, look at the distribution of *‘a personagem’* (female) and *‘o personagem’* (male) in Portuguese. Use ‘Refine’ to compose 2 word boxes, one for the article and one for the noun!
  - Where does the German participle prefix *‘ge-’* go in English loan verbs, and does it combine with a German -t or an English -ed? Check the frequency of e.g. *gesaved* - *gesavet, downgeloaded* - *downgeloadet, outgesourced* - *outgesourcet* - *outgesourct, eingelogt* - *eingelogd* - *eingelogged* – *eingelogget*
  - English loan nouns do often have two plural variants – an English -s, and a Danish zero-morpheme (i.e. nothing) or -er. Check the distribution of *password/passwords, hooligans/hooliganer, print/prints, podcast/podcasts* etc. How do you make sure that the forms without -s actually are plurals? Answer: Add a word box to the left asking for a plural determiner pronoun or adjective, possibly also asking for adnominal (prenominal) syntactic function. Do a frequency break-down! Inspect some results!
- **Synonyms:**
  - For a language of your choice, check the usage (what, where, when, by who?) of different words for a handheld phone, e.g. *mobile (phone), cell( )phone, cellular phone, smartphone, handy, pocket phone …*
  - Loan words may face competition from newly created expressions or loan translations, and it may take years before one or the other has "won". Check the following:
    - German: downloaden – herunterladen
    - Danish: homepage – hjemmeside, website – websted
- **New words: abbreviations:** Not all new words are loan words or compounds. A special way to create a new word is to turn an abbreviation into a regular word - itself then subject to variation, inflection and compounding.
  - Danish: Find out whether upper or lower case is preferred, and if derivation is possible: *SMS/sms, DVD/dvd/Dvd, CD-ROM/cd-rom/CD-rom/CD-Rom*

  - Various languages: Check verbal forms: [A-Z][A-Z]+'?+ing (English), [A-Z][A-Z]+'?+en (German), [A-Z][A-Z]+'?+ar (Spanish/Portuguese), [A-Z][A-Z]+'?+ede (Danish/Swedish)

- **Swear words:** Abusive language (4-letter words etc.) are a sociolinguistic treasure grove. Find out about the unwritten rules of this rather under-researched area of language teaching. Use chat, Twitter or blog corpora if possible!
  - English: find compounds with 'shit-' (shit[a-z]+) and 'fuck' (fuck[a-z]+). Is there a semantic difference as to which nouns these two prefixes can attach to?
    - 'fucking' is also used on its own, as either an adjective or an adverb. Explore the syntactic restrictions for the use of 'fucking' as an adverb!
  - Danish: Is there a grammatical difference between compound starting in 'skide-', 'møg-' and 'pisse-' on the one hand, and 'lorte-' on the other hand?
    - Compare with 'super-' and 'kæmpe-'! Find other augmentative prefixes!
    - Find syntactic rules for the use of 'sgu' in Danish sentences! Which word-classes are allowed directly left and right of 'sgu'?
- **Language and biological gender:** In corpus-linguistic terms, gender/sex-dependent differences of context and usage are among the easier topics to check:
  - Danish: Determine which nouns most frequently occur to the right of 'his' and 'her' (English) or 'hans' and 'hendes' (Danish). Use the statistics buttons 'freq' and 'rel' with “right context”. Use the Shakespeare corpus KEMPE and the ENRON email corpus for English, or Korpus 90 and 2000 for Danish.
  - Brazilian Portuguese: Find out which nouns typically occur left of dele and dela, respectively (Folha corpus). Note that the parser has split dele into de <sam-> and ele <-sam>.
  - Spanish, Portuguese and other Romance languages: Compile a list of typical adjectives used next to 'hombre' and 'mujer'. Use the Wikipedia corpus and "refine search" to ask for an adjective right of 'hombre'/'mujer', or use cqp-speak in the normal interface: [word="mujer"] [pos="ADJ"]. Sort the resulting concordance by "right edge" and relative frequency ("rel" button).
  - Danish: Find compounds with 'mand.+' og 'kvinde.+', maybe 'dreng.+' and 'pige.+', and compare the results statistically! Use [a-zæøå]{4,} instead of .+ to rule out simple inflected forms.
- **Complex words**
  - Find the longest Portuguese/German/Danish/English/French/Spanish word! Dots can be used to indicate the number of letters ('....'), but a smarter solution is an expression like [a-zæøåäöüé]{20,30}, meaning a word between 20 and 30 letters. Don't (!) use .+ after many dots - it's very hard on the server. Can you beat the Danish 'Menneskerettighedskonventionen' (Human rights convention) or

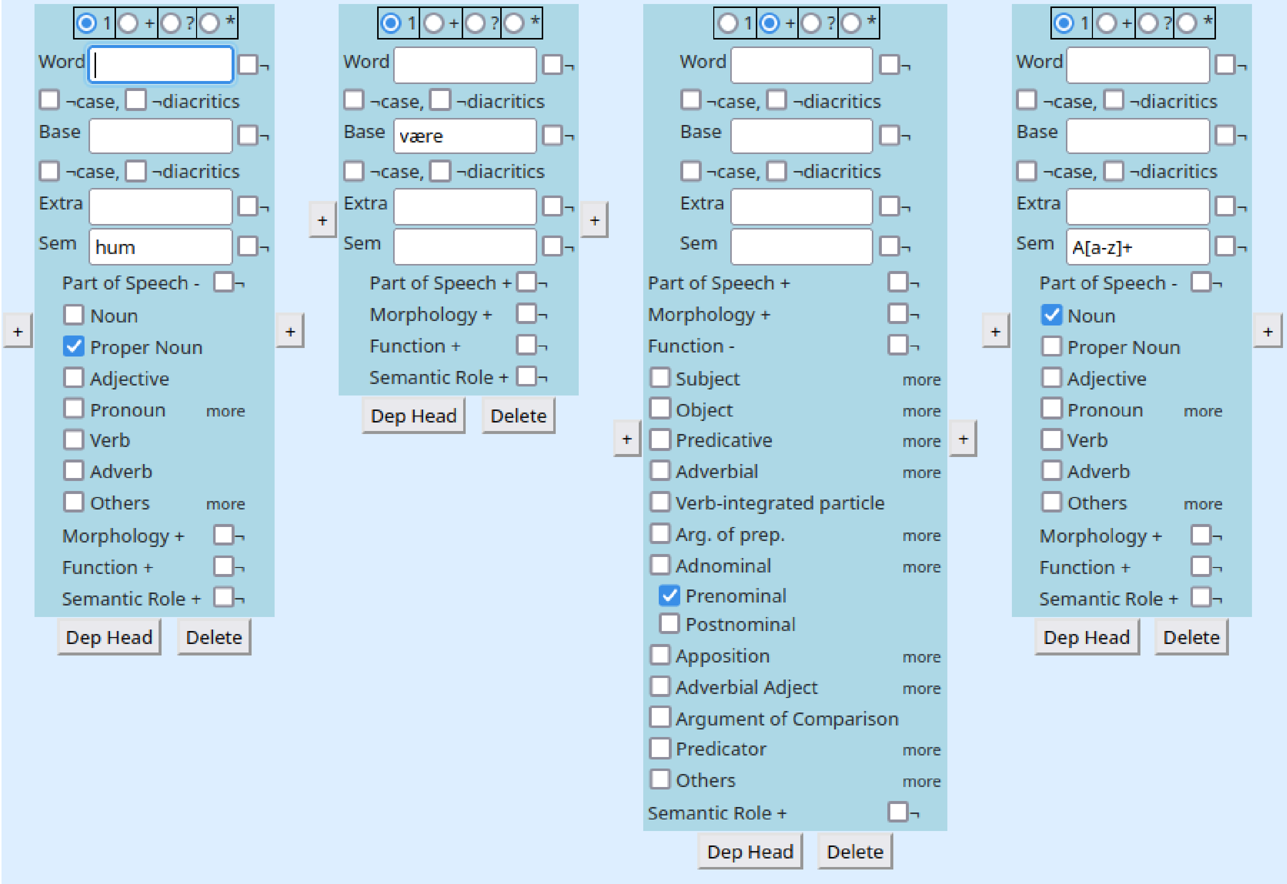
'Tændstiketiketsamlersammenslutningen' (Matchbox label collector society), from Korpus 2000?

  - Portuguese: Find words ending in -ódromo! (any Brazilian/Portuguese differences?)
  - Find as many verbs as possible ending in -geben (German) or -sætte (Danish). Use "left-edge" sorting with the frequency button ("freq").
  - For English, find phrasal constructions with 'give', 'stand' etc., i.e. 'give' followed by an adverb or a preposition.

9.2 Finding linguistic examples

With its many different corpora and languages, and its robust grammatical annotation, CorpusEye can be a useful tool for finding examples of linguistic phenomena to be used for teaching or language exploration.

- **Word order:** Many rules restrict the order of words in a sentence, and word class is - apart from meaning - a key factor in formulating these rules. Use the statistical tools in Corpuseye: "left edge" vs. "right edge", and frequency listings ("freq")
  - Portuguese: Find out if object pronouns are more common right or left of (a) finite verbs, (b) infinitives, using the Brazilian Folha corpus and the "freq" statistics. What is the typical left context for the VFIN-PRON and PRON-VFIN cases, respectively? Are the results different on the European Público98 corpus?
  - Various languages: Find subjects to the right of their verbs! Are certain verbs more likely to occur in these constructions than others?
  - English: Find out which word class is allowed between an infinitive marker ('to') and its infinitive! Use "refine search" with 3 fields, and search by "left context, offset=1".
  - English and Danish: Compile a list of adverbs that can be placed inside a prepositional phrase (i.e. between a preposition and its argument)
  - Choose a language and find out where in a chain of several adjectives that language places nationality adjectives (Italian, Russian etc.). If the corpus has a tag <jnat>, use it in the sem field (without the < > brackets. If not, create a list in the baseform field (in round parentheses, and '|' between the words)
- **Semantic topics:**
  - Find profession nouns, using different methods (suffix, context, special tags: <Hprof>)
  - High-level task: Find time adverbs and other time expressions and classify them!
  - Find examples of metaphorical attribution of animal features to humans, by looking for 'human name' + 'to be' + 1 or more prenominals + 'animal noun', e.g. for the Danish Twitter corpus: [pos="PROP" & sem="hum"] [lex="være"] [func=".*>N.*"]+ [pos="N" & sem="A[a-z]+"]



● 1 ○ + ○ ? ○ *
Word □¬
□ ¬case, □ ¬diacritics
Base □¬
□ ¬case, □ ¬diacritics
Extra □¬
Sem hum □¬
Part of Speech - □¬
□ Noun
☑ Proper Noun
□ Adjective
□ Pronoun more
□ Verb
□ Adverb
□ Others more
Morphology + □¬
Function + □¬
Semantic Role + □¬
Dep Head Delete

● 1 ○ + ○ ? ○ *
Word □¬
□ ¬case, □ ¬diacritics
Base være □¬
□ ¬case, □ ¬diacritics
Extra □¬
Sem □¬
Part of Speech + □¬
Morphology + □¬
Function + □¬
Semantic Role + □¬
Dep Head Delete

○ 1 ● + ○ ? ○ *
Word □¬
□ ¬case, □ ¬diacritics
Base □¬
□ ¬case, □ ¬diacritics
Extra □¬
Sem □¬
Part of Speech + □¬
Morphology + □¬
Function - □¬
□ Subject more
□ Object more
□ Predicative more
□ Adverbial more
□ Verb-integrated particle
□ Arg. of prep. more
□ Adnominal more
☑ Prenominal
□ Postnominal
□ Apposition more
□ Adverbial Adject more
□ Argument of Comparison
□ Predicator more
□ Others more
Semantic Role + □¬
Dep Head Delete

● 1 ○ + ○ ? ○ *
Word □¬
□ ¬case, □ ¬diacritics
Base □¬
□ ¬case, □ ¬diacritics
Extra □¬
Sem A[a-z]+ □¬
Part of Speech - □¬
☑ Noun
□ Proper Noun
□ Adjective
□ Pronoun more
□ Verb
□ Adverb
□ Others more
Morphology + □¬
Function + □¬
Semantic Role + □¬
Dep Head Delete

10 Use cases: Research

10.1 Lexicography

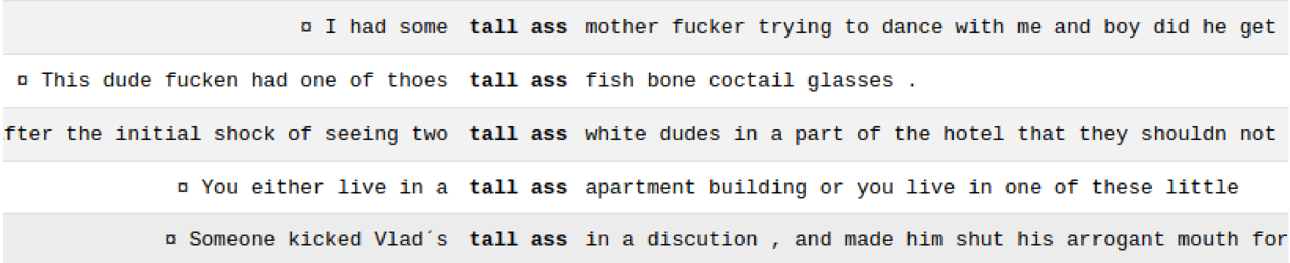
Lexicography is a traditional raison d'être for building and using corpora. Dictionary publishers use corpora to ensure good coverage, get usage examples, quantify the use of spelling variants and to decide on the grammatical information to be provided, e.g. inflection, valency and polysemy.

CorpusEye can support lexicographic work in various ways. For instance it can be used to discover fixed expressions, idioms and other multi-word expressions using collocational analysis. Let's say you are writing a dictionary entry for the English word *'tall'*, and look for typical noun collocates in the English blog corpus, with two adjacent word boxes: [word="tall"] [pos="N"]. By frequency-sorting the resulting concordance, you get a list of typical *tall* things: *guy, building, glass, grass, man, people, tales, order, tree, woman, boy, girl, ships … ass, mountain.*

Using relative rather than absolute frequencies should move the more typical collocates to the top (use Rel C for an in-corpus weighting rather than Rel G for "general language"). However, the English blog corpus is a large, unclean corpus with a lot of hapaxes (rare forms with only 1 or 2 occurrences), so you can get a much more readable frequency list by setting a minimum cutoff for the number of occurrences. '20' will do nicely, but even '3' or '4' will make a big difference.

The list shows that, obviously, people and things can be tall in the cm/m/km sense. Also, it needs to be vertically shaped things, there are no examples of tall cars, plates or balls. And if you force the mind to picture a 'tall plate', it would probably be one stacked with food. In an active dictionary, for a non-English speaker, this vertical dimensional trait should be mentioned to explain when to use *‘tall’* rather than *‘large’* or *‘big’*.

But the real linguistic treats in the frequency list are the non-literal meanings and uses. So as a lexicographer, you should pick up on *tall tales* and *tall order,* click the words and inspect a new, focused concordance to get inspiration for a definition or example. Even things may have non-literal meanings, like *tall ships* referring to sailing ships. And most current dictionaries would probably be unaware of the *‘tall ass’* meaning, a two-word adjectival (!) expression used in English slang, and making up 4 out of 5 examples found in the corpus:



```
                         ¤ I had some  tall ass  mother fucker trying to dance with me and boy did he get
 ¤ This dude fucken had one of thoes  tall ass  fish bone coctail glasses .
fter the initial shock of seeing two  tall ass  white dudes in a part of the hotel that they shouldn not
              ¤ You either live in a  tall ass  apartment building or you live in one of these little
             ¤ Someone kicked Vlad´s  tall ass  in a discution , and made him shut his arrogant mouth for
```

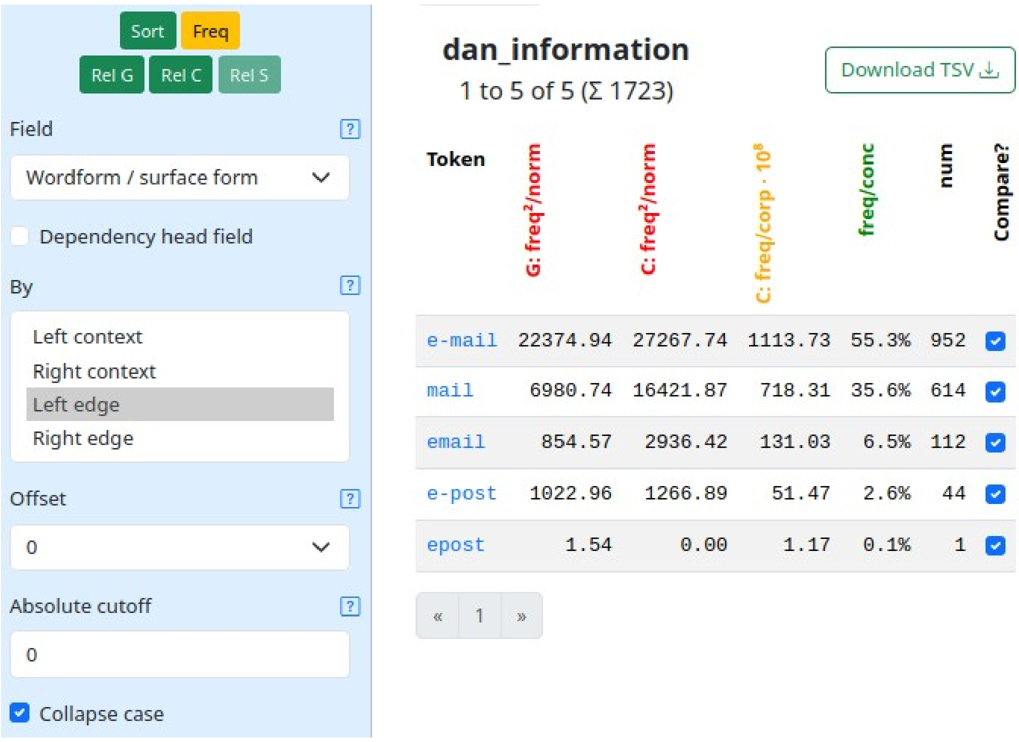
10.2 Language change

Corpora that have been compiled with data from a longer period of time[15], or a pair of comparable corpora from different periods, can be used to explore diachronic language change. Typical genres with a well-defined time-structure and time-related meta information are news corpora, social media corpora and literature corpora.

To study language change involving specific lexemes, search for the full set of historical and/or current variants and inspect the result in a time histogram or - for stacked bar charts – with the ‘group by’ option set to ‘year’.

As an example, let’s have a look at the battle between *e-mail, email, mail* and *e-post* in Danish over a 10-year period, using the newspaper corpus *Information.* First enter (e-?mail|mail|e-?post) in the word or lemma field, with case-insensitive on, and N (noun) set in the POS menu. Next tick ‘collapse case’ and press ‘Freq’ with wordform or lemma chosen in the field menu. This will yield overall statistics for the entire period/corpus:

[15] Including so-called monitor corpora, which are explicitly designed for ongoing compilation over an open-ended period of time.



Field: Wordform / surface form

☐ Dependency head field

By:
- Left context
- Right context
- Left edge
- Right edge

Offset: 0

Absolute cutoff: 0

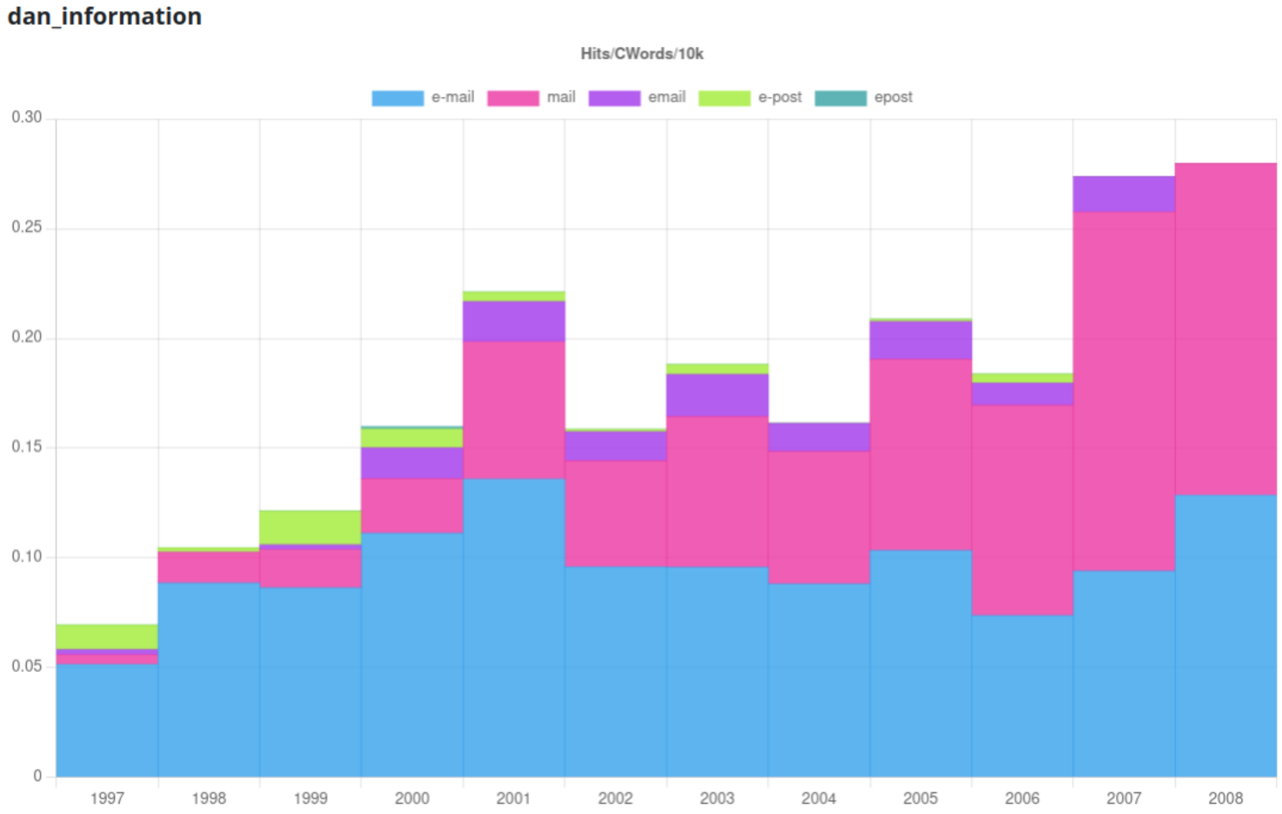
☑ Collapse case

dan_information

1 to 5 of 5 (Σ 1723)

Token	G: freq²/norm	C: freq²/norm	C: freq/corp · 10^8	freq/conc	num	Compare?
e-mail	22374.94	27267.74	1113.73	55.3%	952	☑
mail	6980.74	16421.87	718.31	35.6%	614	☑
email	854.57	2936.42	131.03	6.5%	112	☑
e-post	1022.96	1266.89	51.47	2.6%	44	☑
epost	1.54	0.00	1.17	0.1%	1	☑

You are now ready to create a stacked bar chart with yearly frequencies. Just set the ‘Group-by’ menu to ‘year’ and press ‘Group results’. The most neutral setting for ‘Compare hits per’ is probably ‘per 10K words’. The resulting bar chart should look like this:



Ignoring the colour coding, the chart can be read as a simple time histogram suggesting, that the concept of e-mail became more important during the period, with a four-fold frequency increase between 1997 and 2008.

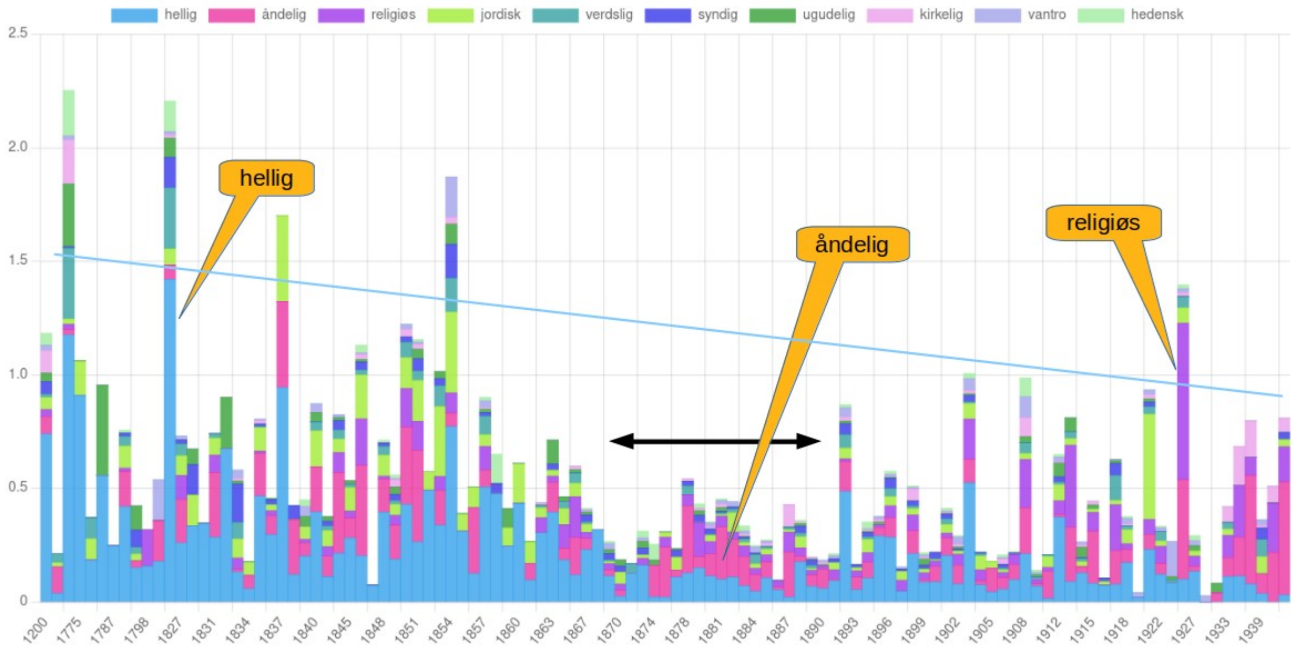
More interestingly, the colour-stacking illustrates the (lost) battle between the new, Danish form *e-post* (green), actively pushed by some conscious language users in the beginning of the period, and the English international terms. In addition, we can see the hyphenated form *e-mail* is preferred over *email*, and that the latter is loosing ground compared with the simple *mail*, probably reflecting the fact of life that e-mails started taking over instead of (once ordinary) paper mail, in the same sense that ‘mobile phones’ today are referred to as just ‘phones’.

10.3 Literary studies: Distant reading

Using linguistically inspired statistics, topic profiling, comparisons across time, authors and words, as well as network analysis to create visualization-based insights is a methodology used by the Distant Reading-approach to literary analysis. While *CorpusEye* supports many of these techniques, the arguably most interesting and useful aspect for literary studies is the option to inspect linguistic search results through the optics of literary metadata such as author, title and year of publication.

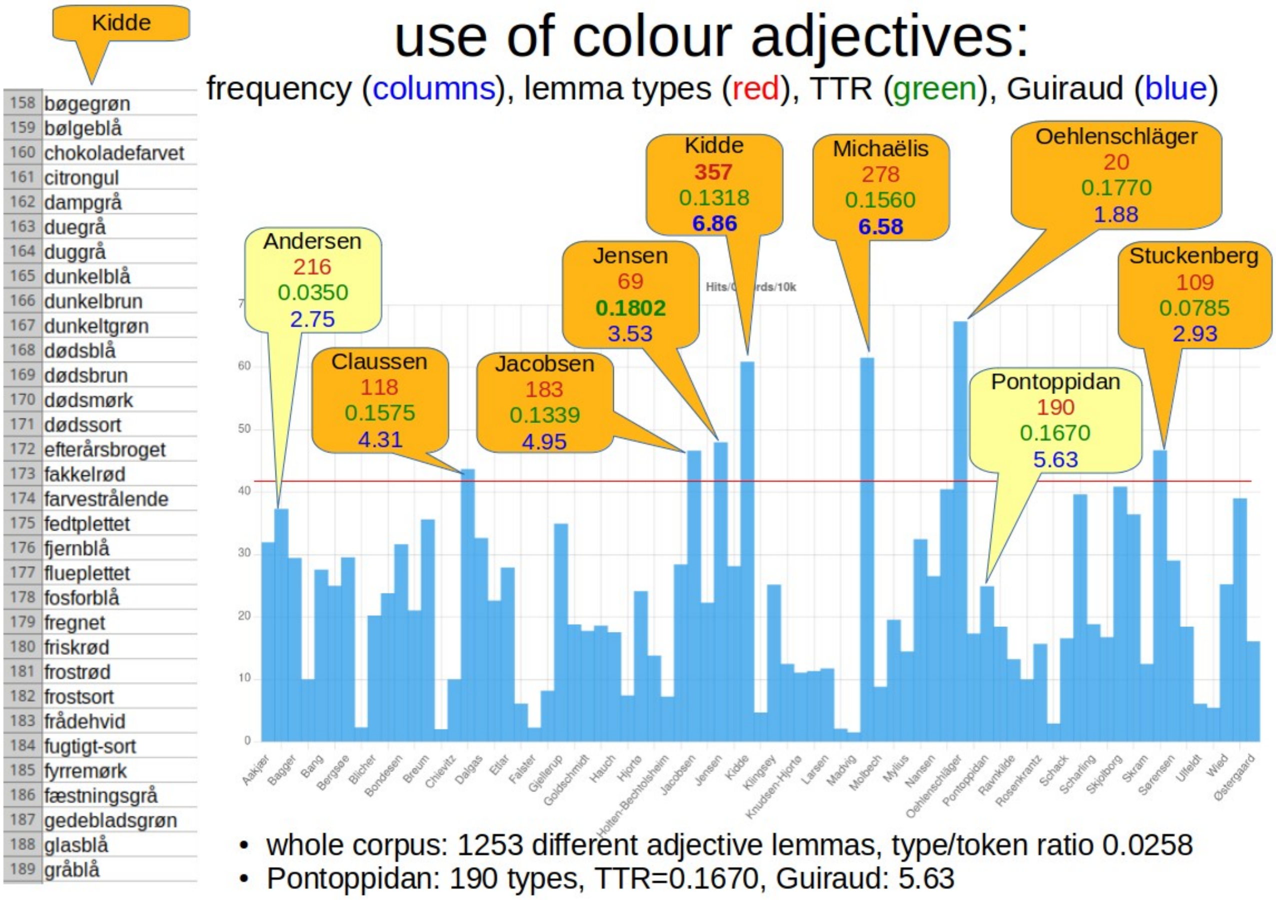
The role of religion in Danish literature can work as an example. First, we perform a search for religion adjectives (marked <jrel> or <Drel>) in the Danish literature corpus. In the top-ten of the resulting frequency list, we find the words *hellig* (holy), *åndelig* (spiritual) and *religiøs* (religious), as well as *verdslig* (worldly), *jordisk* (earthly), *syndig* (sinful) and others.

Using ‘Group by’ to create a year-by-year breakdown of hits, we get a histogram showing – on average – a fall in the use of “religious” adjectives. More interestingly, the word-stacked bar chart reveals that the religiously more involved *hellig* (holy) is typical of older text, while more modern text have a larger incidence of *religiøs* (religious) – a word more typical for a (non-religious) discourse *about* religion. The lowest numbers for religious adjectives are found in the period 1870-1890, the so-called “moderne gennembrud” (rise of modernism), with an elite focus on atheism, communism and other new, non-religious ideologies. For this period, the number one “religious” adjective is *åndelig* (spiritual).



Another example for a distant reading task could be the lexical spread (variation) of adjectives for different authors. In the example below, the search was for colour adjectives [sem=”jcol”] in Danish classical literature, with part of a frequency list for the Danish writer Kidde. The histogram columns show the per-sentence frequencies of color adjective. The absolute number of colour tokens and the number of different colour lemmas (in red) can be found using individual frequency lists for relevant authors, and given these two values, type-token ratios (TTR, green) can be calculated, providing a rough measure for lexical variation. However, for a meaningful comparison, these values have to be seen in relation to text sizes, because independently of actual lexical spread, TTR is lower for longer texts than for shorter ones. The Guiraud index (blue) addresses this problem with a measure, where the number of types is divided by the square root of the token count.

The histogram analysis shows that H.C. Andersen uses more colour adjectives per sentence than Pontoppidan, and also employs slightly more *different* colour lemmas (216 vs. 190). However, Andersen’s adjective types get “diluted” over much more text in the corpus, so TTR is way higher for Pontoppidan, and even with the text size-balanced Guiraud metric, Pontoppidan wins the competition for greater lexical variation by a factor of two (5.63 vs. 2.75).



use of colour adjectives:

frequency (columns), lemma types (red), TTR (green), Guiraud (blue)

158	bøgegrøn
159	bølgeblå
160	chokoladefarvet
161	citrongul
162	dampgrå
163	duegrå
164	duggrå
165	dunkelblå
166	dunkelbrun
167	dunkeltgrøn
168	dødsblå
169	dødsbrun
170	dødsmørk
171	dødssort
172	efterårsbroget
173	fakkelrød
174	farvestrålende
175	fedtplettet
176	fjernblå
177	flueplettet
178	fosforblå
179	fregnet
180	friskrød
181	frostrød
182	frostsort
183	frådehvid
184	fugtigt-sort
185	fyrremørk
186	fæstningsgrå
187	gedebladsgrøn
188	glasblå
189	gråblå

- whole corpus: 1253 different adjective lemmas, type/token ratio 0.0258
- Pontoppidan: 190 types, TTR=0.1670, Guiraud: 5.63

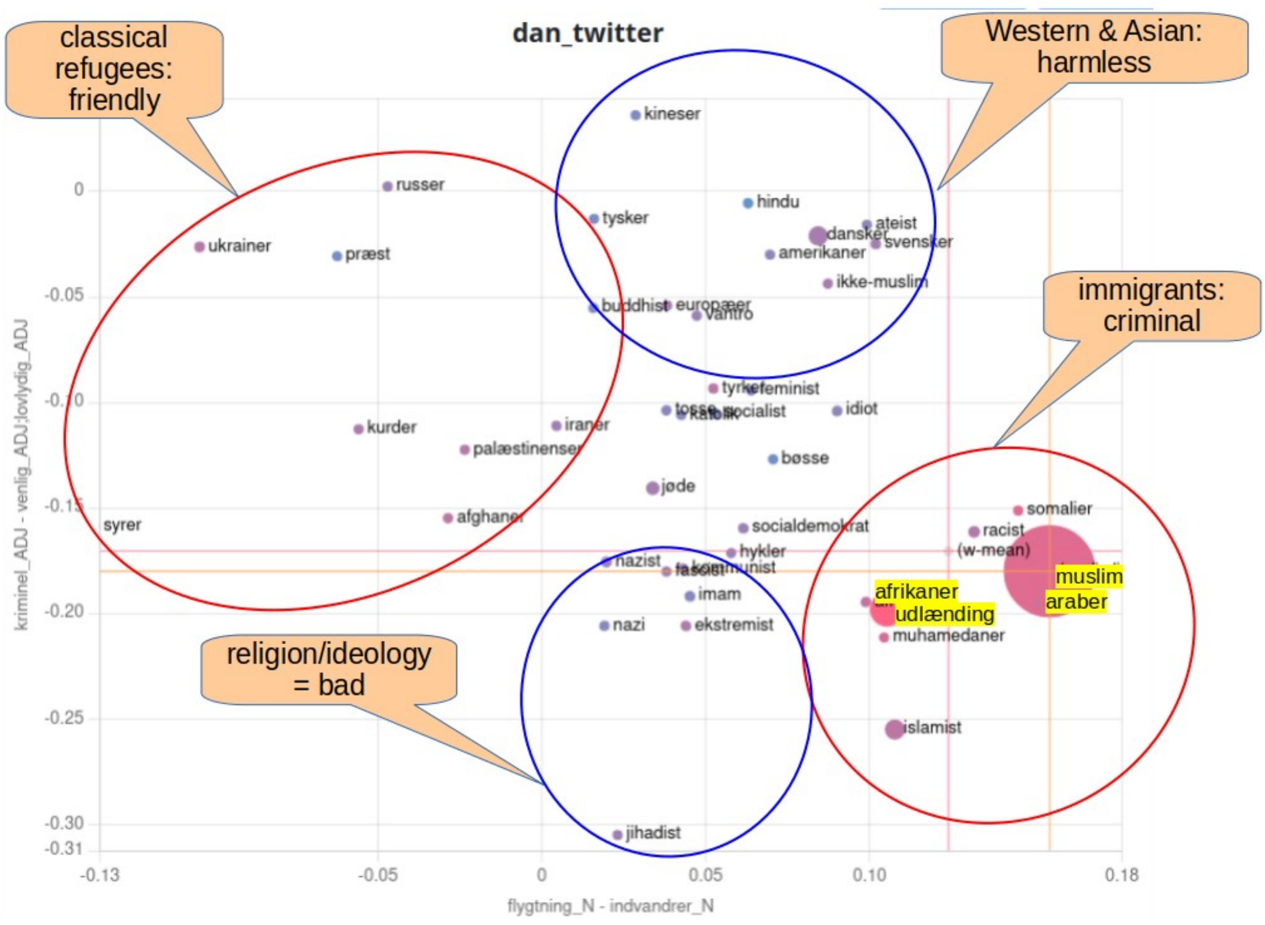
10.4 Hate speech

A lot of dedicated hate-speech research has been done using *CorpusEye* on German and Danish social media corpora compiled and linguistically annotated and evaluated during the *XPEROHS*[16] project at SDU's Institute of Language and Communication. For a comprehensive Danish perspective on hate speech, cf. (Bick, 2023), where findings are presented from various linguistic angles – lexicographical, morphosyntactic and semantic, addressing topics such as slurs, othering and overt sentiment markers (especially emojis), as well as metaphor and indirect hate speech.

As a high-end semantic tool, semantic scatter-plots can be used. Let's say we want to explore how different categories of foreigners are viewed in Danish social media, whether provenance matters, and whether the stereotype of the "criminal immigrant" can be linked to certain ethnic-national and religious groups more than others. To this end, we can define an "immigrant" subcorpus using a first-level search for a lemma list like (flygtning|invandr|udlænding|muslim|islam).*, followed by a sequential (SQ) search for the semantic noun classes of (Hnat|Hideo|Hattr) that will pick up on person nouns semantically marked by nationality, ideology and various attributes.

Frequency-sorting the resulting concordance for the searched-for nouns, we can then create a semantic scatter plot with a *refugee vs. immigrant* x-axis and a *criminal vs. law-abiding & friendly* y-axis.

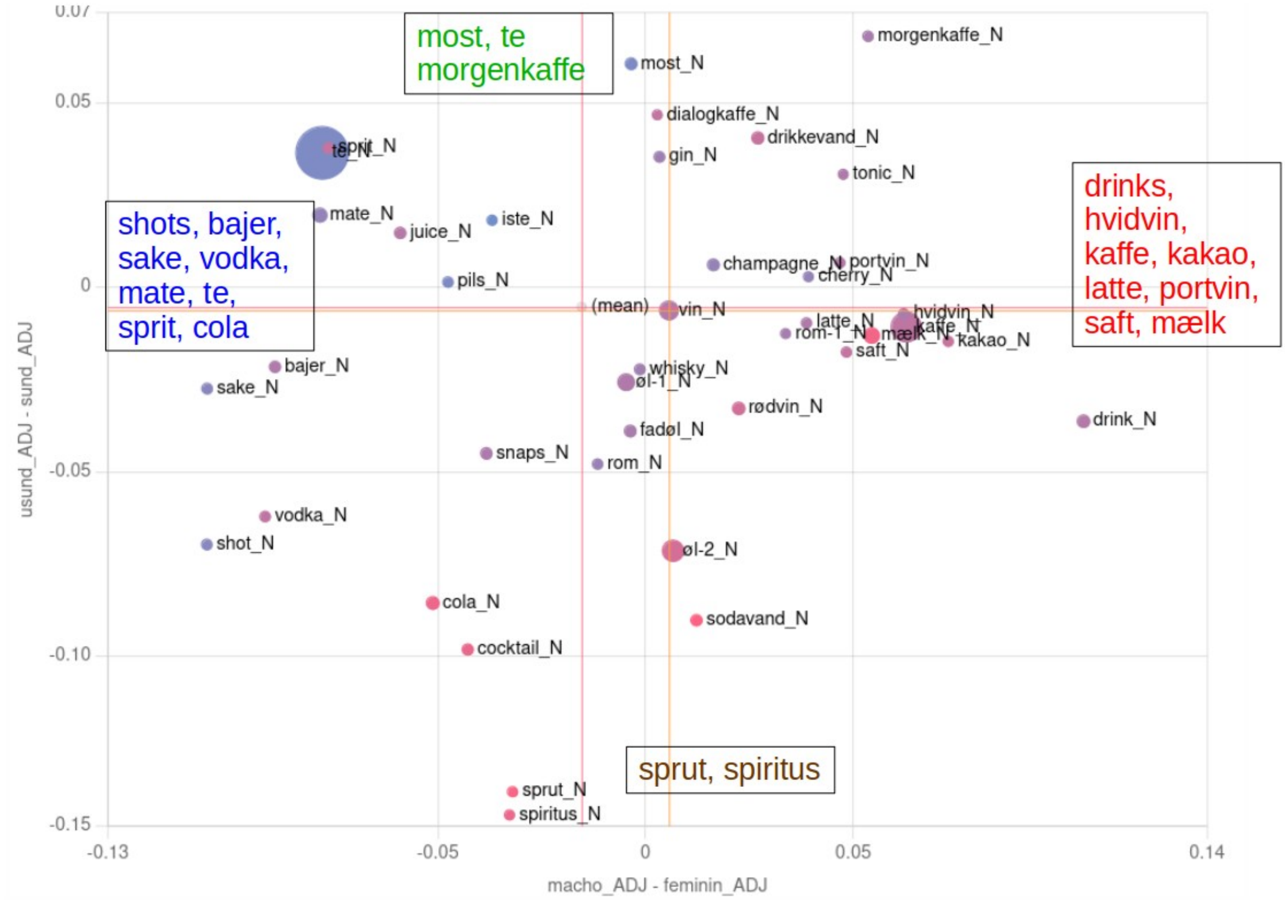
[16] https://xperohs.sdu.dk/



The result clearly illustrates a distinction between classical, friendly refugees and criminal immigrants. with Ukrainians, Kurds and Syrians being exponents of the former and Arab Muslims and Africans representing the latter group. That the criminality stereotype is linked to these groups specifically, and not the concept of *immigrant* as such, can be seen from the fact that Western and Asian immigrants form a separate bubble at the friendly, law-abiding end of the spectrum. Religion, finally, is not linked to a homogeneous stereotype either: Hindus, Buddhists, atheists and non-Muslims (!) have vectors at the positive end of the y-axis, while jihadists, Nazis and "extremists" can be found at the negative (criminal) end.

10.5 Gender studies

One way of exploring gender stereotypes is creating semantic scatter plots for certain topics, such as food, drink, cars, profession etc. For instance, a search for [sem="drink.*"] in the Danish social media corpus, with the stereotype axes *macho-feminin* and *usund-sund* will reveal which drinks are associated with men *(shots, bajer, vodka, mate, cola)* and women *(drinks, hvidvin, kaffe, kakao, latte, portvin, saft, mælk)*, respectively, and which are seen as healthy *(most, te, morgenkaffe)* or not *(sprut, spiritus)*.

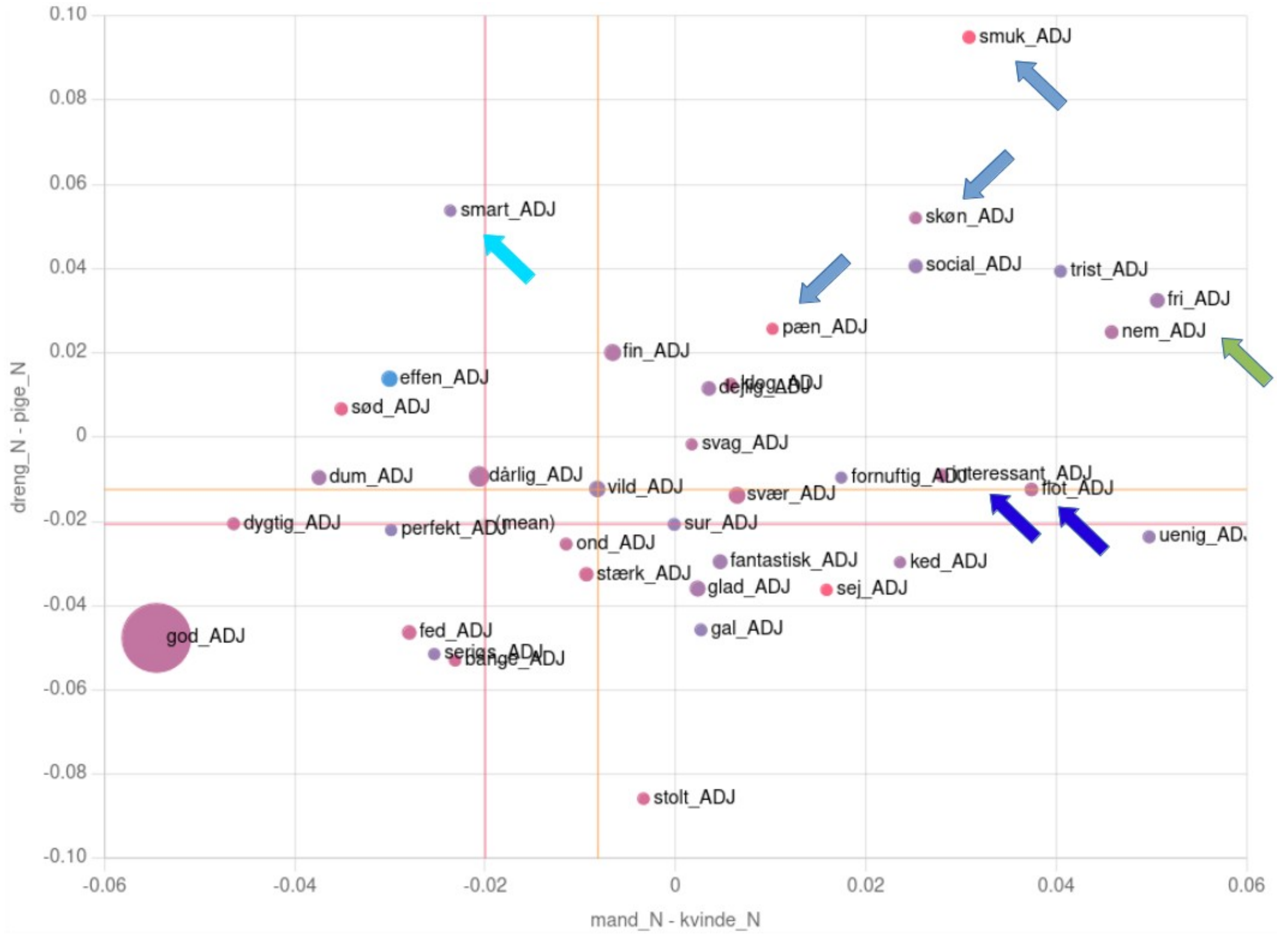


Stereotypical attributes of men and women can be addressed *through* collocation studies (ADJ + *man/woman/boy/girl).*

Alternatively, semantic similarity can be used for this task, too. First, extract relevant adjectives from the corpus and build a (frequency-ordered) lemma list:

[sem=".*(?:^|)(jqual|jpsych|jemo|jskill|jpower|jsoc|jcog|jcom|jsem)(?: |$).*" & pos="ADJ"]

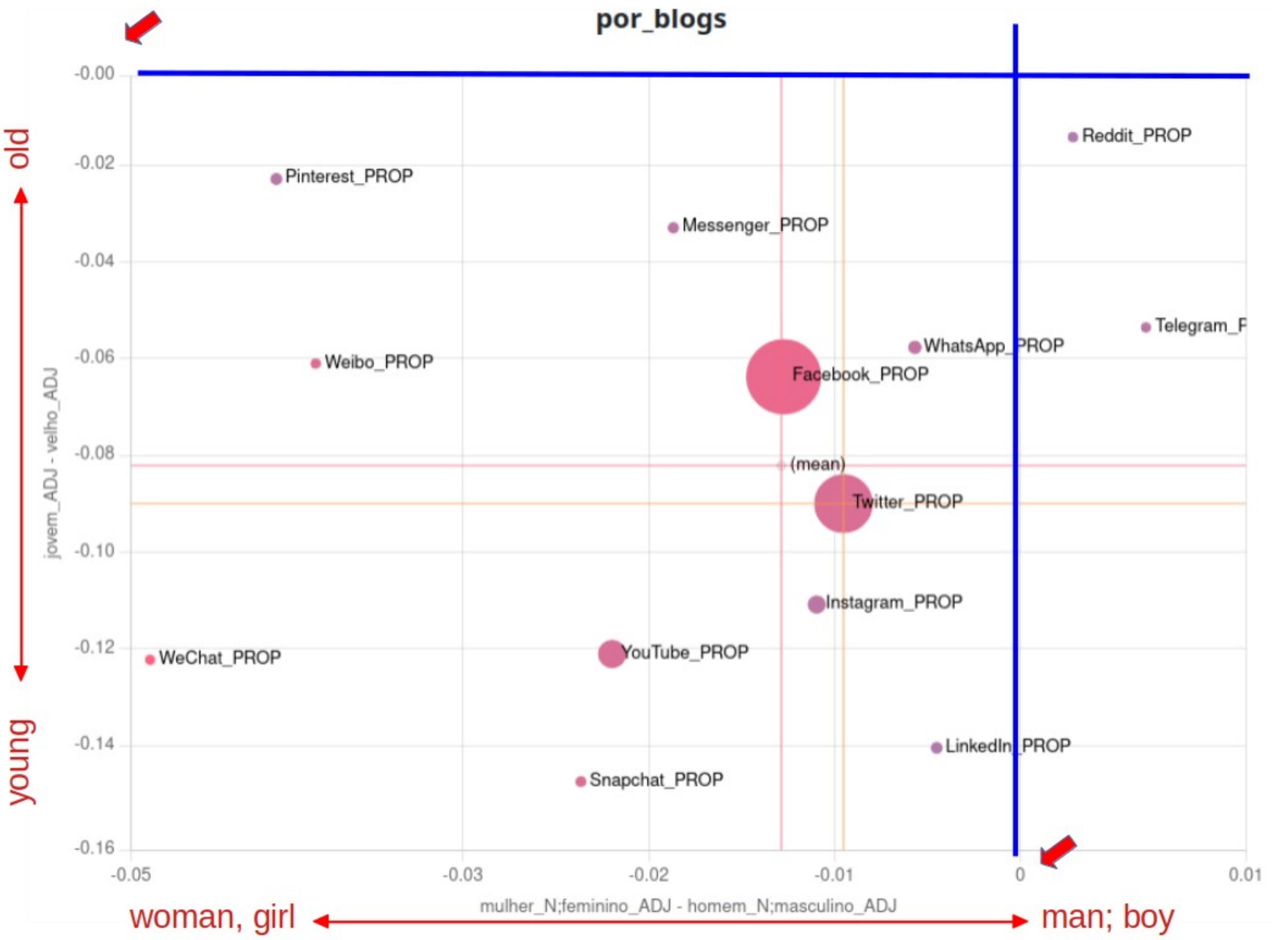
Then create a scatter plot with the axes *mand-kvinde* (man-woman) and *dreng-pige* (boy-girl). In the upper-right *kvinde/pige* quadrant we find a clustering of beauty adjectives: *smuk, skøn, pæn, flot, dejlig, fin*, but also the "availability" adjectives *nem, fri, interessant.* In the opposite, male corner we have positive *(god, dygtig, perfekt)* and negative *(dårlig, dum)* quality adjectives. More centered, but on the male side of the zero-line there are *vild* (wild), *ond* (mean) and *stærk* (strong).



10.6 Brand profiling

Blog and social media corpora can be used to evaluate how a given brand or company name is viewed by potential users or customers, i.e. which stereotypes people associate with the brand in question, which target groups resonate with it, and whether it is met with positive or negative sentiment. With diachronic data, it is also possible to spot and follow changes in sentiment and possibly link them to specific corporate, political or publicity events.

In the example below, the idea is to perform a profiling of social media channels using a Portuguese-language blog corpus. First, a concordance and frequency list is build with a search containing the names of the major social media services. Remember to click 'vector-plottable' in the statistics menu before creating the frequency list. Second, using the 'vector-plot' pop-up, we define an age axis and a gender axis, using the Portuguese words for *'young'* and *'old'*, and for *'woman;girl'* and *'man;boy'*, respectively. The resulting scatter plot has a strong bias towards *young* and *female* (cp. the blue zero-lines), but there are clear differences as to *how* young and *how* female. At the young end, there is Snapchat, and Instagram is "younger" than Facebook and Whatsapp. Reddit and Telegram are perceived as most "male", and WeChat as most "female".



Bibliography

Bick, Eckhard (2005), CorpusEye:Et brugervenligt web-interface for grammatisk opmærkede korpora, In: Peter Widell & Mette Kunøe (eds.), 10. Møde om Udforskningen af Dansk Sprog 7.-8.okt.2004, Proceedings. pp.46-57, Århus University

Bick, Eckhard (2023). VISL & CG-3: Constraint Grammar on the Move: An application-driven paradigm. In: Arvi Hurskainen, Kimmo Koskenniemi & Tommi Pirinen (eds.), Rule-Based Language Technology. NEALT Monograph Series vol. 2, pp. 112-140. University of Tartu. ISSN 1736-6291

Bick, Eckhard; Gorbahn, Katja; Kalwa, Nina (2023). Methodological Approaches to the Digital Analysis of Educational Media: Exploring Concepts of Europe and the Nation I: Katja Gorbahn, Erla Hallsteinsdóttir & Jan Engberg (eds.), Exploring Interconnectedness - Constructions of European and National Identities in Educational Media, pp. 143-186. Palgrave Macmillan & Springer.

Bick, Eckhard (2023). Derogatory Linguistic Mechanisms in Danish Online Hate Speech. In: Isabel Ermida (Ed.): Hate Speech in Social Media: A Linguistic Approach, pp. 165-202. London: Palgrave Macmillan

Kalwa, Nina (2019). Die Konstitution von Konzepten in Diskursen: Zoom als Methode der diskurslinguistischen Bedeutungsanalyse. I: Sprach(kritik)-kompetenz als Mittel demokratischer

Willensbildung. (ed. Sandro Moraldo et.al). Greifswalder Beiträge zur Linguistik, bd. 12, s. 11-26

Karlsson et al., 1994. Constraint Grammar: a Language-Independent System for Parsing Unrestricted Text. Mouton de Gruyter, Berlin, 1994.