

Couranten Corpus application manual

Table of contents

Introduction	3
Information about the corpus	3
GiGaNT Lexicon service	4
Linguistic Annotation	4
Metadata categories	4
Date	4
Title	5
Article title	5
Section	5
Newspaper	5
Article properties	5
Country	5
Place	5
Text type	5
Application user manual	6
Getting started	6
Searching the corpus	7
Simple search	7
Search	7
Wildcards	7
Reset	8
History	8
Global settings	8
Extended search	9
Basic	9
Linguistic annotation (alpha quality).	11
Starting a new search	11
Filter search by	12
Filter by date	12

Filter by title or article properties	12
Advanced search	13
The query builder	13
The tab search	14
Token attributes	14
Adding attributes to a token box	15
Function of the two +-buttons in a token box	15
The tab options	16
Managing sequences of token boxes	16
Uploading value lists in the query builder	17
Copy to CQL editor	17
Expert search	17
Import query	18
Gap filling	18
Viewing results	20
Per Hit view	20
Sorting results	21
Grouping results	21
Per Document view	23
Sorting results	23
Grouping results	24
Exporting results	24
Information about a document	24
Content	24
Metadata	25
Statistics	25
Exploring the corpus	25
Documents	25
N-grams	26
Options	26
Example	27
Statistics (frequency lists)	27
Options	27
Example	28

Appendix: Corpus Query Language	29
CQL support	29
Supported features	29
Differences from CWB	30
(Currently) unsupported features	31
Using Corpus Query Language	31
Matching tokens	31
Sequences	32
Regular expression operators on tokens	32
Case- and diacritics-sensitivity	33
Matching XML elements	33
Labeling tokens, capturing groups	34
Global constraints	34

Introduction

This manual describes the corpus exploitation environment for the *Couranten Corpus*. The corpus application is developed by the Dutch Language Institute (Instituut voor de Nederlandse Taal or INT). The backend of the application is the BlackLab Lucene based search engine developed for corpora with token-based annotation (<http://inl.github.io/BlackLab/>). The web-based frontend is a further development of the corpus-frontend application developed by INT (<https://github.com/INL/corpus-frontend>) in CLARIN and CLARIAH projects. Its design is inspired by the first version of the OpenSoNaR user interface by Tilburg and Radboud University (<https://github.com/Taalmonsters/WhiteLab2.0>).

Information about the corpus

The *Couranten Corpus* comprises the seventeenth-century Dutch newspapers available on [Delpher](#). The oldest surviving newspapers were published in 1618. For the Delpher-website the Koninklijke Bibliotheek in The Hague has scanned the newspapers. These scans have been read by optical character recognition (OCR). However, OCR could not deal with the old fonts and texts of these newspapers. That is why the Meertens Institute set up a citizen science project, led by Nicoline van der Sijs. By means of a collaborative web application, created by Rob Zeeman, all newspapers were transcribed and corrected by more than 300 volunteers of the Stichting Vrijwilligersnetwerk Nederlandse Taal. Subsequently, interns Thomas Angenent, Aafje Baarslag, Rianne de Koning, Guido Moerdijk, Jeroen Pelkman and Lennart van Winzum checked and corrected the metadata and added new metadata, for instance on genre (advertisements, national news, international news, etc.). The last correction of the metadata was done at the INT.

This sizeable corpus currently contains the contents of 13 newspapers, 109.532 articles and 18.926.425 words. The information from these newspapers is of interest to researchers of various disciplines, ranging from historians to historical linguists, literature scholars and art historians.

In the future, transcriptions of newly scanned newspapers from the seventeenth century and newspapers from the eighteenth century will be added to the *Couranten Corpus*.

This first online accessible version of the *Couranten Corpus* was released on 12th May 2022.

GiGaNT Lexicon service

To make the *Couranten Corpus* more accessible, suggestions for query expansion are given, using the INT lexicon service with the historical computational lexicon [GiGaNT-HILEX](#).

The current version of GiGaNT-HILEX in the lexicon service contains the lexicon modules based on the *Dictionary of the Dutch Language (Woordenboek der Nederlandsche Taal, WNT)* and the *Dictionary of Middle Dutch (Middelnederlandsch Woordenboek, MNW)*.

If you want to make use of this service, please contact Katrien Depuydt (katrien.depuydt@ivdnt.org).

Linguistic Annotation

It is also possible to use the linguistic annotation to search the corpus. The linguistic annotation has been created by means of a Support Vector Machine-based statistical part of speech tagger, trained on the Letters as Loot corpus, and a lemmatizer using the INT historical lexicon and a simple statistical spelling variation model. It is an alpha version, since no effort has yet been done to train the tagger with more appropriate training material.

The part of speech tagging has been done using the tagset and tagging principles for the annotation of diachronic corpora of historical Dutch, developed in the context of the CLARIAH+ project. This annotation layer has been added to the corpus, and can also be used to search the online corpus. A detailed description can be found [here](#).

More information about the used lemmatization principles can be found in Marijke Mooijaart, [Het lemma in the GiGaNT lexicon](#).

Metadata categories

The *Couranten Corpus* has been enriched with an elaborate set of metadata categories. These metadata will all be described below. In the corpus application it is possible to limit a search by filtering on metadata categories.

Date

The date on which a newspaper issue was printed, as evidenced by the date of the newspaper.

Title

Article title

Article title involves the place and date when a newspaper article was written, such as *Zurich den 13 September*. It is possible to search on any word or number from the article title. For the given example these are the search terms: *Zurich, den, 13, and September*.

Section

Section refers to the portion of a newspaper issue under which a particular message is printed. In most cases it expresses a country (CROATIEN, 'Croatia') or a geographical area (DUTSLANT en aangrenzende Ryken). It should be noted that the area boundaries at the time may differ significantly from those of today.

Newspaper

This search field is provided with a list, which contains suggestions for search terms in alphabetical order, based on the characters typed in.

All 13 newspaper titles are listed in the About of this corpus.

Article properties

Country

Country refers to the name of the country in which a particular place is situated today. For example, the former German city of *Breslau* became the Polish city of *Wroclaw* after the war. Therefore, *Breslau* can be found under *Polen* (Poland).

Place

Place refers to the current Dutch name for a particular place. This eliminates the need to look up all the different historical spelling variants of a place separately. With the search term *'s-Gravenhage* you will find newspaper reports from *'s Gravenhaghe, 's Gravenhage, 's Graven-Haghe, 's Graven-Hage, den Hage, den Haegh, etc.*

Text type

The type of text to which the letter belongs (advertisement, below the line, breaking news, domestic news, international news, notification).

Application user manual

Getting started

Here are a few examples of what you can do with the corpus application (the links will take you to the application):

- To search for a word literally in the form you specify, use Simple Search:
 - Simple Search for Word [schip](#)
- To search for different spellings and inflections/conversions of a given lemma, use Lemma in Extended Search:
 - Extended Search for Lemma [dochter](#)
- To search for words or lemmata satisfying a certain pattern, use *wildcards* in Simple Search or Extended Search, or *regular expressions* in Expert Search
 - words starting with *ver* and ending with *len* in [Simple Search](#)
 - lemmata starting with *ver* and ending with *len* in [Extended Search](#)
 - lemmata starting with *ver* and ending in *eren* with one syllable in between in [Expert Search](#)
- To see which unique forms occur as a result of your search, use the Group hits by feature.
 - example Group by Context (advanced): [all words following lieve](#)
 - example Group by Word before: [different words preceding the word huis](#)
- To explore the distribution of document properties in the corpus, use the Explore feature
 - example: [characteristics about the newspapers](#)
 - example: [place](#)

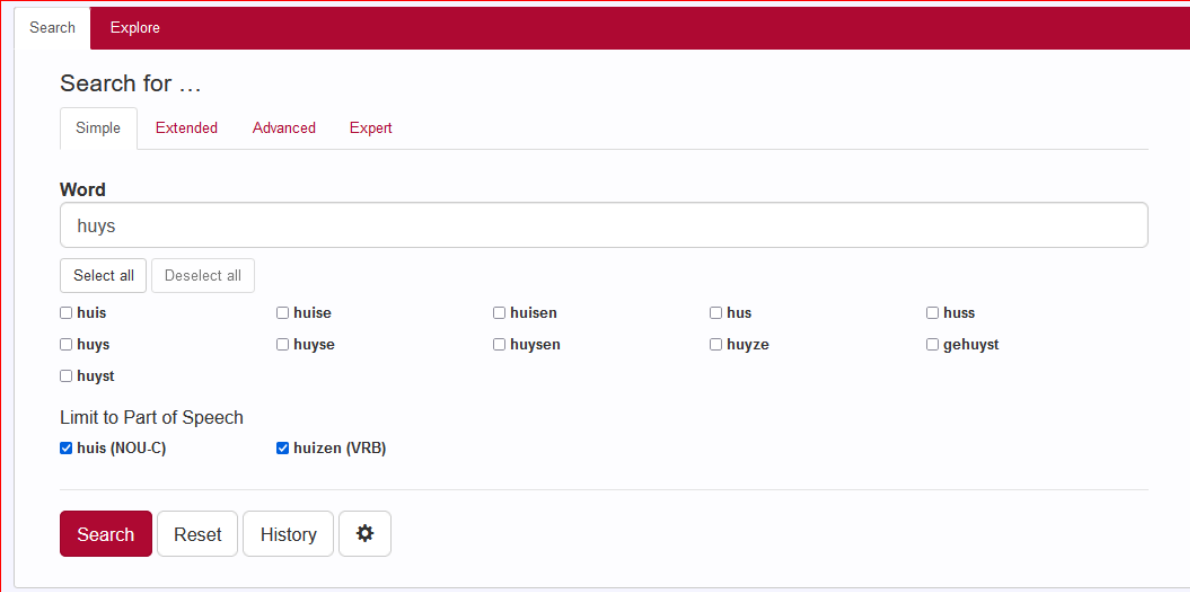
Searching the corpus

Simple search

Search

The Simple Search allows you to quickly search for specific word forms (e.g. *huys*). After entering a search term, a spinner briefly appears on the right side of the search bar. Based on the keyed in word, suggestions are given of possible variants of spelling and/or form from the GiGaNT-lexicon.

Based on the information in this lexicon all spelling variants of the search term found are suggested (see screenshot below). You can then choose from the presented suggestions or select all at the same time (Select all). To make your search even more targeted, it is also possible to limit the search to the parts of speech that were found in GiGaNT-HILEX in connection to the search term.



The screenshot shows the 'Simple Search' interface. At the top, there are tabs for 'Search' and 'Explore'. Below this is a search bar with the text 'Search for ...'. Underneath the search bar are four tabs: 'Simple' (selected), 'Extended', 'Advanced', and 'Expert'. A 'Word' input field contains the text 'huys'. Below the input field are two buttons: 'Select all' and 'Deselect all'. A list of suggestions is shown, each with an unchecked checkbox: 'huis', 'huise', 'huisen', 'hus', 'huss', 'huys', 'huyse', 'huysen', 'huyze', and 'gehuyst'. Below the list is a section titled 'Limit to Part of Speech' with two checked checkboxes: 'huis (NOU-C)' and 'huizen (VRB)'. At the bottom of the interface are four buttons: 'Search' (highlighted in red), 'Reset', 'History', and a settings gear icon.

If you know exactly which word you are looking for, you can also – while the wheel is spinning – press Enter directly. The search will then start immediately.

It is also possible to enter a phrase: *eene vrede maecken* or *wat aengaet*. You will then find all occurrences of that exact phrase.

Note that in Simple Search the patterns will be matched case-insensitively: *capitein* for instance will deliver the same results as *Capitein* or *CAPITEIN*. See the paragraph [Grouping results](#) in Per Hit view to see how it is nevertheless possible to distinguish between uppercase and lowercase letters.

Wildcards

In Simple Search, the use of wildcards can prove good service to search for specific word forms. A wildcard is a symbol used to replace or represent one or more characters. The following two wildcards are supported:

- * The asterisk matches any character zero or more times. Therefore, *a*n* matches all values that start with an *a* and end with a *n*, e.g. *aen*, *aengekomen* and *Antwerpen*.

? The question mark matches a single character once. Therefore, searching for *ann?* matches *only* four-letter values starting with *ann* and ending with a random character, e.g. *anna*, *ann.*, *anne* and *anno*.

This wildcard can be used more than once. Thus *ar????n* matches words like *armeen*, *ariaen* en *arduy*n. Note that searching with wildcards is limited to Simple Search and Extended Search. [In Expert Search you can use so-called regular expressions instead of wildcards.]

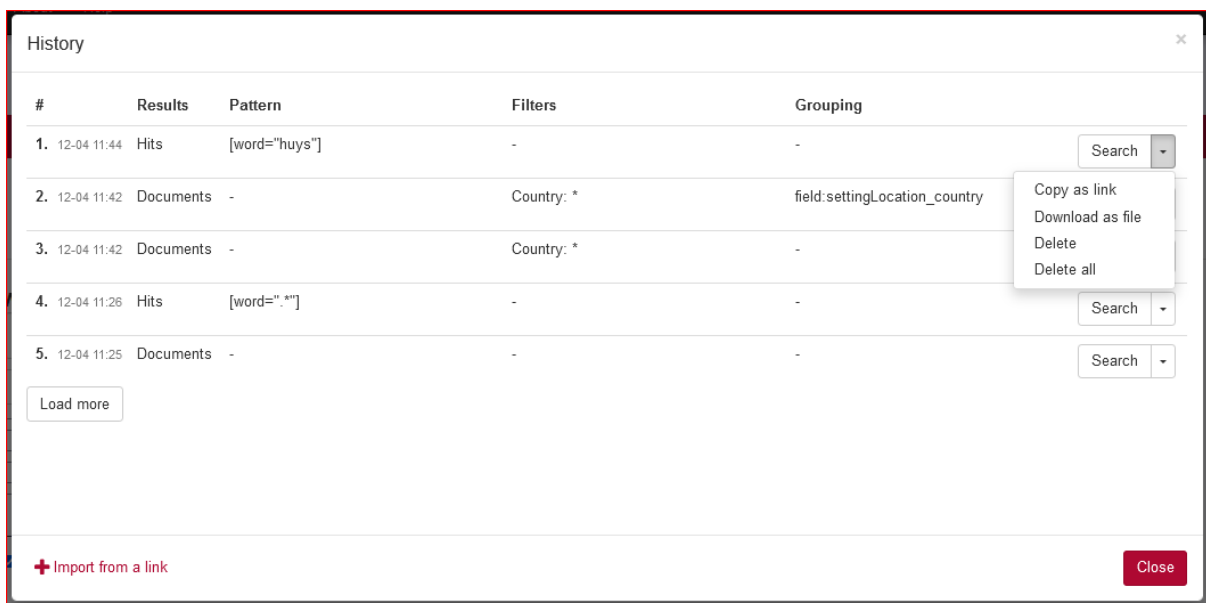
Reset

You can start a new search by pressing the Reset button. By doing so, both the search query and the hits found will be cleared. Your search history, however, will remain unchanged.

Note that it is also possible to start a new search by entering a new word or phrase in the search field Word.

History

The History button will display your query history. Per search query there are several possibilities (as shown in the screenshot below): you can perform the search query again (Search), you can copy the search query as a link (Copy as link), you can download the search query as a file (Download as file), you can delete a single search query (Delete) or delete all search queries (Delete all).



Every search query has its own url. If you copy this url via History (Copy as link) or directly from the address bar of your browser, you can send it to someone else who can import this link via Import from a link. It offers that person the possibility to run the search on his or her own computer.

Global settings

The Global settings dialogue, activated by pressing the wheel button, allows you to configure five settings: Results per page, Sample size, Seed, Context size and Wide View.

- *Results per page*: you can choose whether you want 20, 50, 100 or 200 results to be shown;

- *Sample size*: selecting a value here will instruct the search engine to return a random sample drawn from the complete result set. (Pressing the Reset button does not change the sample size. It must be changed manually.) The sample size can be limited by
 - a percentage of the total number of search results (percentage)
 - the number of results displayed (count);
- *Seed*: a ‘random seed’ is a number used to initialize a so-called pseudo-random number generator. Keeping the same seed will ensure that two samples drawn from the same result set are identical. A new seed will most likely result in a different sample;
- *Context size*: by entering a number you can determine the number of words Before hit and After hit;
- *Wide View*: the default setting is ‘small view’; you can change to Wide View by ticking the checkbox.

The image shows a 'Global settings' dialog box with the following elements:

- Results per page:** A dropdown menu showing '20 results'.
- Sample size:** A dropdown menu showing 'percentage' and a text input field containing '1'.
- Seed:** A text input field containing '-1468812526110331'.
- Context size:** A text input field containing 'Context size'.
- Wide View:** A checkbox that is checked.
- Close:** A red button at the bottom right.

Extended search

In extended search, it is possible to search in two different ways: *Basic* and *Linguistic annotation (alpha quality)*.

Basic

Like in Simple search, Extended Search allows you to quickly search for specific word forms in *Basic*. The search is performed in the same way as described for Simple Search.

After entering a search term, a spinner briefly appears on the right side of the search bar. Based on the keyed in word, suggestions are given of possible variants of spelling and/or form from the GiGaNT-lexicon.

Based on the information in this lexicon all spelling variants of the search term found are suggested (see screenshot below).

The screenshot shows the search interface of the GiGaNT-lexicon. At the top, there are tabs for 'Search' and 'Explore'. Below this, the search bar is labeled 'Search for ...'. There are four tabs for search levels: 'Simple', 'Extended', 'Advanced', and 'Expert'. Underneath, there are two sub-tabs: 'Basic' and 'Linguistic annotation (alpha quality)'. The search term 'wolf|wolven|wolff|wolfs|wolve|wulf|wulven' is entered in the search bar. Below the search bar, there are buttons for 'Select all' and 'Deselect all'. A list of search results is shown, each with a checked checkbox: 'wolf', 'wolfs', 'wulven', 'wolve', 'wolvén', 'wolff', and 'wulf'. Below the list, there is a section 'Limit to Part of Speech' with checkboxes for 'wolf (NOU-C)' and 'wolve (VRB)'. At the bottom, there is a checkbox for 'Case- and diacritics-sensitive' which is currently unchecked.

You can then choose from the presented suggestions or select all at the same time (Select all). To make your search even more targeted, it is also possible to limit the search to certain parts of speech that were found in GiGaNT-HILEX in connection to the search term. It is also possible to enter a phrase: *die daar toe soude meynen* or *zal hebben te beklaghen*.

In Extended Search it is also possible to search case- and diacritics-sensitive. Note that the default setting for search is case- and diacritics-insensitive. For example, searching for the Word *willem* (& Willem (NOU-P NOU-C)) will result in 1963 occurrences of this name (*Willem*, *WILLEM*, *willem*, *WILLEM*). By ticking the box Case- and diacritics-sensitive you will only find 7 occurrences of the Word *willem*, but none of *Willem*. In order to directly find only occurrences of the Word (form) *Willem* (1947x), use the search term *Willem* and tick the box Case- and diacritics-sensitive under the search field Word (as shown below).

The screenshot shows a search interface with a dark red header containing 'Search' and 'Explore'. Below the header, there's a 'Search for ...' section with tabs for 'Simple', 'Extended', 'Advanced', and 'Expert'. Under 'Extended', there's a 'Basic' tab and a 'Linguistic annotation (alpha quality)' option. The 'Word' field contains 'Willem'. Below the field are 'Select all' and 'Deselect all' buttons. There are two checkboxes: one for 'willem' (unchecked) and one for 'willems' (unchecked). Under 'Limit to Part of Speech', there are two checked options: 'Willem (NOU-P NOU-C)' and 'Case- and diacritics-sensitive'.

If you know exactly which word you're looking for, you can also – while the wheel is spinning – press Enter directly. The search will then start immediately.

Like in Simple Search, wildcards are supported in Extended Search. (See for a short explanation of wildcards [Simple Search](#).)

In the search field Word it is possible to search for different values simultaneously by separating them without spaces by a vertical line, e.g. *god|man|lief* or – with the use of wildcards – *god|aan*|hond*.

For the search field Word it is possible to search for a series of tokens by entering multiple values – including wildcards – separated by a space, e.g. *Portugeese Admiraal, Portugeese * or * Admiraal*. It will be obvious that these three searches give different results

Linguistic annotation (alpha quality).

Extended search offers some extra possibilities to refine your search in *Linguistic annotation (alpha quality)*. You can indicate the Part of speech of a word you are looking for. For some word types it is possible to extend a search with features. This applies to Proper noun, Pronoun and Residual. In addition, you can use the lemma search bar to simultaneously search for all spelling variants, inflections and conjugations.

Linguistic enrichment was carried out fully automatically and was not checked manually. Therefore, there will still be some imperfections. This feature is useful for adding structure to large amounts of data, but should be used with caution.

Starting a new search

You can start a new search by pressing the Reset button. By doing so, both the search query and the hits found will disappear. Your search history, however, will remain unchanged.

If you use the field Word (in Basic), there are two possibilities to start a search: fill in the desired value and press enter or fill in the desired value and then click the Search button.

Filter search by

At the right side you will find the option to limit your query to a subset of documents with specific metadata values. You can apply different filters for Date (*Date*), Title (*Article title*, *Section*, *Newspaper*) and Article properties (*Country*, *Place*, *Text type*). To view the results for all documents simply leave the attributes in the filtering form empty.

Filter by date

The newspapers in this corpus were printed in the period between 15 May 1619 and 2 January 1700. You can find a newspaper from a specific date by entering the same data in the "from" row as in the "to" row (see screenshot below). Newspapers from a certain period can be found by entering a start date and an end date. If you do not enter a date, the entire corpus is searched. If you want to filter by another date or another period, please press the "reset" button.

Filter search by ...

Date **1** Title Article properties

Date (1619-05-15 to 1700-01-02)

From: 1672 04 21

To: 1672 04 21

Permissive **Strict** reset

Date (Date): 1672-04-21

Selected subcorpus:
Total documents: 21 (0.0192%)
Total tokens: 2.706 (0.0143%)

Filter by title or article properties

There are two different ways to specify a filter, depending on the field type. You can either fill in a value yourself – for instance Country – or choose one or more values from a drop-down list – for instance Text type. The drop-down list has been applied especially when the number of values to choose from is relatively small. Text type for instance has only six possibilities (advertisement, below the line, breaking news, domestic news, international news and notification). You can pick one of these values by clicking on it; your choice will be marked with a tick. It is possible to choose several values. If you want to delete a selection, you can click on the corresponding line again. To close the drop-down list, you can either press the upward pointing arrow in the upper right corner or simply press escape.

Filter search by ...

Date Title Article properties 1

Country

Country

Place

Place

Text type

advertisement

advertisement ✓

below the line

breaking news

domestic news

international news

notification

When on the other hand the set of possible values is rather large (e.g. Article title), you have to type a specific value in the search field. After entering a single character, a list of possible values is suggested. Clicking on an auto-completed value will paste that value in the field. Note that this only works with a single word, like *zurich*. In order to search for an exact phrase, i.e. a multiple word value, it must be surrounded by double quotes. For instance, in the Article title “*zurich den 13 september*” will result in one specific newspaper issue.

By means of a number at the top of ‘Filter search by’, the number of values used to filter on, is displayed as can be seen in the above screenshot.

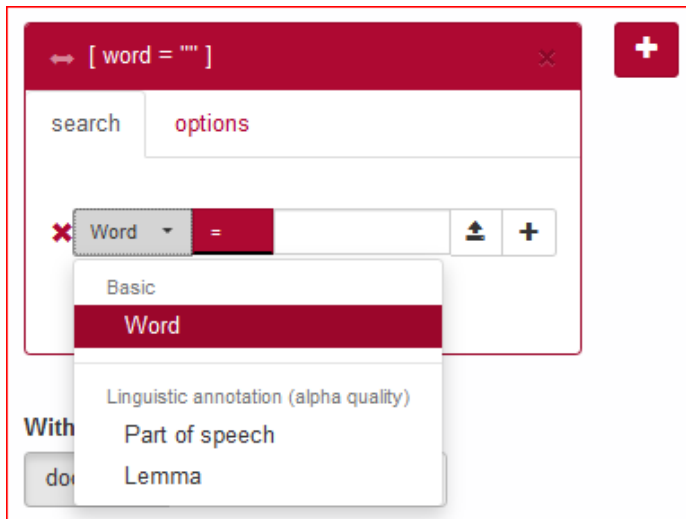
For a detailed description of the metadata, see the section [Metadata categories](#).

Advanced search

The query builder

The basic building block in the query builder is the *token box* (see below). Each box represents a token – usually just a single word – or a simple repetition of tokens; when multiple tokens are used, they are matched in order from left to right.

You can use the query builder to create complex queries without writing CQL (here: Corpus Query Language). Therefore, it is easy to use.

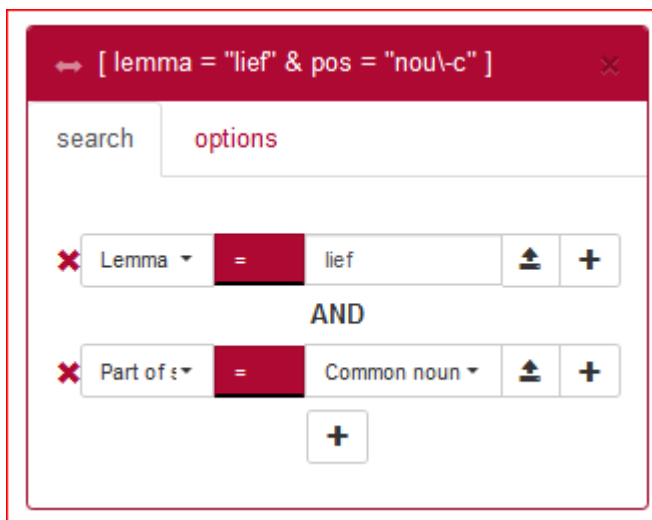


A token box in the querybuilder has two tabs: search and options.

The tab search

The tab search contains a set of attributes a token in the corpus must have to be matched by the query. By clicking the +-button on the right hand side of this token, you can add new attributes (see below). Then enter a value that the attribute must have for the token to be found. The search command `Lemma=lief` and `Part of Speech=Common noun` for example excludes all forms of *lief* as an adjective.

The CQL query generated to match this token (the *token query*) in the corpus is displayed in the top bar of the box, to help you understand what is happening internally. The following applies to our example:



Token attributes

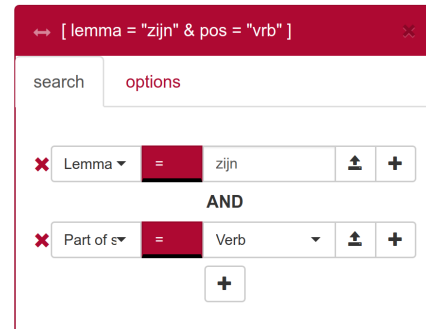
Specifying token attributes is similar to the Extended Search form. Select which attribute a token should have, and enter the value that the attribute must have for the token to be matched. Attributes in the query builder are interpreted as *regular expressions*. Note that this is different from the Extended Search, where token patterns use wildcards.

Going beyond single-attribute token queries, a token box also allows you to combine several attributes and to specify repetition options.

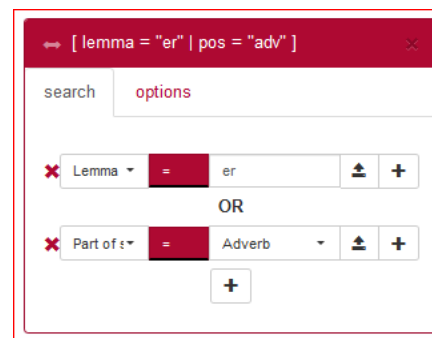
Adding attributes to a token box

Using the +-button, new attributes can be added. Two options exist: *AND* and *OR*.

The *AND* option creates a new attribute restriction that a token must match in addition to the ones which were already there. As an example: suppose we want to match *zijn* ('to be') as a verb, not as a pronoun. First, fill in the attribute Lemma with value *zijn*, then click +, choose *AND*, and choose the value Verb for Part of speech.



Similarly, creating a new attribute using *OR* will create a token query matching tokens that have the original attribute *or* the new attribute. For instance, enter *Word=er* first, add a new attribute with the *OR* option and enter Adverb as Part of Speech to match tokens with part of speech tag adverb or with word form equal to *er*.



Function of the two +-buttons in a token box

The difference between the +-sign on the right of an attribute and the one below it, is that the +-sign on the right keeps the newly added attribute “within a subclause”. This is most easily explained by means of an example.

Suppose we want to search for either *goed* or *lief*, used as a noun. If we add the attributes in the order Part of speech=Common noun AND Lemma=*goed*, OR Lemma=*lief* using the +-signs **below** the attributes, as in the left screenshot below, we get the token query [(pos = "nou\c" & lemma = "goed") | lemma = "lief"]. This will also match adjective forms of *lief*, as in “sy houden goede order, doch alsoo sy de Boeren op veel plaetsen niet thuyt vinden, en het haer aen Voeragie ontbreekt, moet het lieve Graen op het Velt aenhouden”, where *lieve* is an adjective, so this is not what we were after.

If, on the other hand, we add OR lemma=*lief* with the +-sign to the **right** of the attribute Lemma=*goed*, it will be inserted in a subclause (Lemma=*goed* OR Lemma=*lief*), thus resulting in the correct query [pos = "nou\c" & (lemma = "goed" | lemma = "lief")], as shown in the right screenshot below.

↔ [(pos = "nou\c" & lemma = "goed") | lemma = "lief"]

search options

Part of s = Common noun

AND

Lemma = goed

OR

Lemma = lief

↔ [pos = "nou\c" & (lemma = "goed" | lemma = "lief")]

search options

Part of s = Common noun

AND

Lemma = goed

OR

Lemma = lief

Before hit	Hit	After hit	Word	Lemma	Part of speech with features
Vvt Prague den 8. September. ...waren daer teghens gheresolveert haer	goet	ende bloet uyerlijck metten anderen...	goet	goed	NOU-C
Wt Dresden den 20. dito ...Ruyters, met ghewelt ofte met	lieve	door t Landt passeren willen...	lieve	lief	AA
Wt Weenen den 9. Februarj ...is) op des Heeren Budiani	goederen	seer grouwlijck hebben ghetranneiseert, soo...	goederen	goed	NOU-C
Wtlet legher van Sijn Princel Excell. ...met hun Vee ende andere	goederen	daer gevulcht waren, jammerlijc berooft...	goederen	goed	NOU-C
Vvt Straesburgh den 16. dito. ...inde Wantenou een Schip	goederen	gheladen, onversiens inde brant met gheraecht...	goederen	goed	NOU-C
...laten, andersints waren alle	goederen	verbrant...	goederen	goed	NOU-C
die					
Vvt Vveenen den 7. dito. ...Cosacken op des Heeren	goederen	seer ghetranneiseert. So hebben hun...	goederen	goed	NOU-C
Budians					
Vvt Breslauw den 2. Augusti. ...die Clooster ende ettelijcke	goederen	geplundert, die Geestlijcken gheransonneert, ende...	goederen	goed	NOU-C
Heeren					
Wt Venetien den 10. Februarj 1621. ...Turcksche, in den Golfo	goederen	geresen was, gantsch vergeleken ende...	goederen	goed	NOU-C
genomene					
Vvt Venetien den 13. dito. ...oepentlijck sal gestranguliert,	goederen	gheconfisgheert werden, overmits ende synne eenighe moorderijen...	goederen	goed	NOU-C
...gestelt, ende sou van synne	goederen	sesstich duysent Croonen nae Nederlant...	goederen	goed	NOU-C

Before hit	Hit	After hit	Word	Lemma	Part of speech with features
Vvt Prague den 8. September. ...waren daer teghens gheresolveert haer	goet	ende bloet uyerlijck metten anderen...	goet	goed	NOU-C
Wt Weenen den 9. Februarj ...is) op des Heeren Budiani	goederen	seer grouwlijck hebben ghetranneiseert, soo...	goederen	goed	NOU-C
Wtlet legher van Sijn Princel Excell. ...met hun Vee ende andere	goederen	daer gevulcht waren, jammerlijc berooft...	goederen	goed	NOU-C
Vvt Straesburgh den 16. dito. ...inde Wantenou een Schip	goederen	gheladen, onversiens inde brant met gheraecht...	goederen	goed	NOU-C
...laten, andersints waren alle	goederen	verbrant...	goederen	goed	NOU-C
die					
Vvt Vveenen den 7. dito. ...Cosacken op des Heeren	goederen	seer ghetranneiseert. So hebben hun...	goederen	goed	NOU-C
Budians					
Vvt Breslauw den 2. Augusti. ...die Clooster ende ettelijcke	goederen	geplundert, die Geestlijcken gheransonneert, ende...	goederen	goed	NOU-C
Heeren					
Wt Venetien den 10. Februarj 1621. ...Turcksche, in den Golfo	goederen	geresen was, gantsch vergeleken ende...	goederen	goed	NOU-C
genomene					
Vvt Venetien den 13. dito. ...oepentlijck sal gestranguliert,	goederen	gheconfisgheert werden, overmits ende synne eenighe moorderijen...	goederen	goed	NOU-C
...gestelt, ende sou van synne	goederen	sesstich duysent Croonen nae Nederlant...	goederen	goed	NOU-C

The tab options

The tab options specifies the contextual properties, such as whether the token occurs at the end of a sentence, and the repetition pattern:

↔ [word = ""]

search options

Optional

Begin of sentence

End of sentence

repeats 1 to 1 times

Managing sequences of token boxes

There are three ways to manage the sequence and the number of token boxes:

- *Rearrange* a token by clicking and dragging the little arrow handle in the top-left corner simultaneously (1).
- *Delete* a token by clicking the x in the top-right corner (2).
- *Create a new token box* by clicking the +-button next to the upper right corner of the utmost right token box (3).

↓ (1)

↓ (2) ↓ (3)



Uploading value lists in the query builder

It is also possible to upload a list of values, separated by a white space. To do so, click the upload button (with the arrow pointing upwards) and select a text file. Tokens will then be matched for any of the values from the file.

Note that this function only works for *.txt-files. If you are using a text editor like Word, you have to save your file as a *.txt file or you can copy and paste the values into a *.txt file first.

After uploading a file, the text can be edited by clicking the yellow marked file name in the text field. Editing the text is temporary and will not modify your original file.

To remove an uploaded file and go back to typing a value, click on the cross (x) next to the yellow text box. Another possibility to clear the uploaded values is by clicking the yellow marked text field and then pressing the Clear button on the bottom left corner of the Edit box. Using the Reset button will start a complete new search.

Copy to CQL editor

It is possible to copy a query – like [\[pos="aa & lemma="goed"\]](#) – to the CQL editor using the *Copy to CQL editor* button. This will take you automatically to the Expert Search screen, after which you can start the search or adjust the query if desired.

Expert search

The Corpus Query Language (CQL) editor allows you to type your own CQL query, to import a previously downloaded query and to upload a tab separated list of values to substitute for gap values (see below for further explanation).

CQL queries are expressions built up with the help of a few sequence operators and brackets from basic blocks enclosed by square brackets, in each of which one or more token attributes are specified.

In CQL, spaces only affect a search if they are included in quotes. Whether the search command is [word="schip"] or [word = "schip"] (or just “schip”) does not make any difference to the result. However, there is a difference between the queries [word="schip"] and [word=" schip"]. The first search results in exactly 28.784 hits, but the second one in zero!

Some examples:

- Simple: [\[word="schip"\]](#), e.g. the attribute word matches the regular expression *schip*; [\[word!="schip"\]](#), e.g. the attribute word does **not** match the regular expression *schip*; [\[word="*.man"\]](#) matches all words ending with *man*, including *man* itself. (Note that [\[word="*man"\]](#) will not give any results, because in Expert Search an asterisk is not a wildcard but a repetition operator.)
- Combination of attributes (combining operators are `&`, `|`, `!`), e.g. [\[word="hoop"|"geloof"|"liefde"\]](#) matches either the word *geloof*, the word *hoop* or the word *liefde*.
- The empty `[]` matches any token, e.g. [\[lemma="man"\]\[\]{}3\[lemma="god"\]](#) matches a sequence of *man* followed by *god* with three arbitrary tokens in between.
- Operators `|`, `&` and parentheses `()` and the repetition operators `(+)`, `(*)`, `(?)` and `({})` can be used to build complex sequence queries. Example: ["laetste" "man" | "almachtige" "god"](#), matching any sequence of *laetste man* or *almachtige god*.

This short list does not cover all CQL features. For more detailed information on how to write CQL, please consult the short [Appendix: Corpus Query Language](#), which contains further pointers.

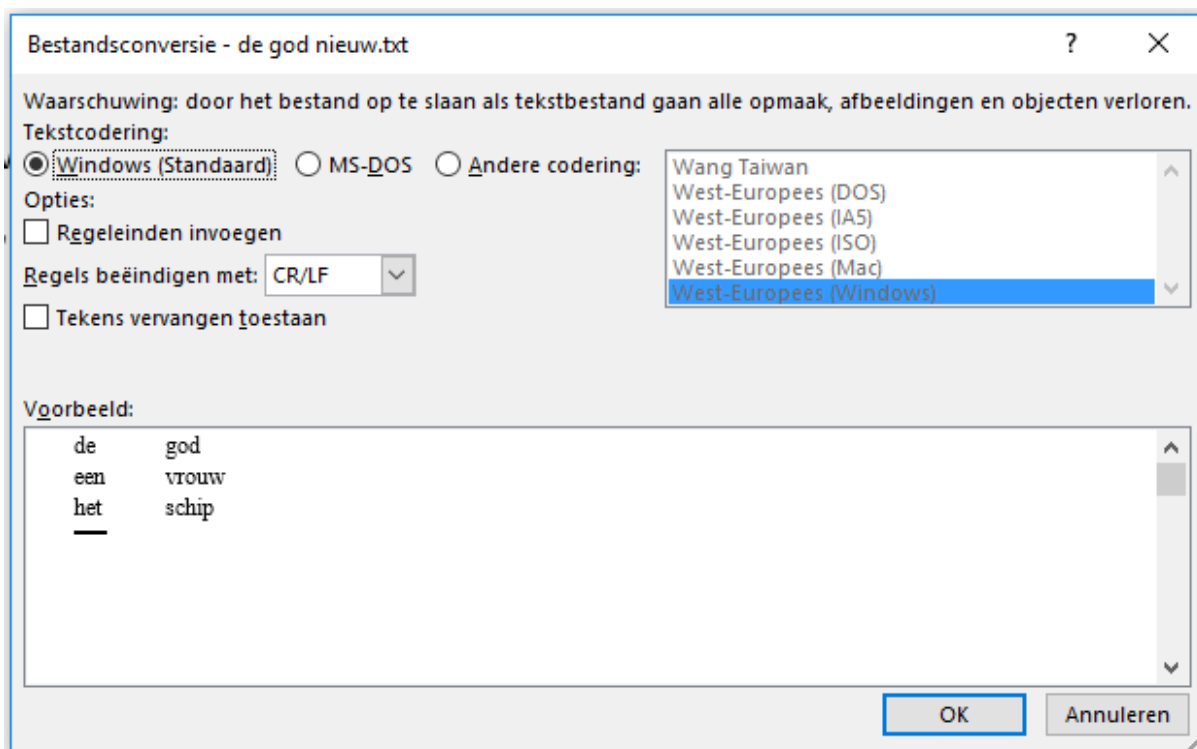
Import query

If you have entered a search query, you can find it back by clicking the History button. On the right hand side you can select Download as file in the drop-down menu (default value is Search) and save the file. (For a more elaborate description of the History button see [Simple Search](#).)

Previously saved queries can be used again by uploading them through the Import query button.

Gap filling

Use this button to upload a Tab Separated Values (TSV) file, which is a simple text format for storing data in a tabular structure. Each record in the table is one line of the text file. Each field value of a record is separated from the next by a tab character. It is also possible to upload a plain text file (.txt) that has the same properties, as is shown in the following screenshot:

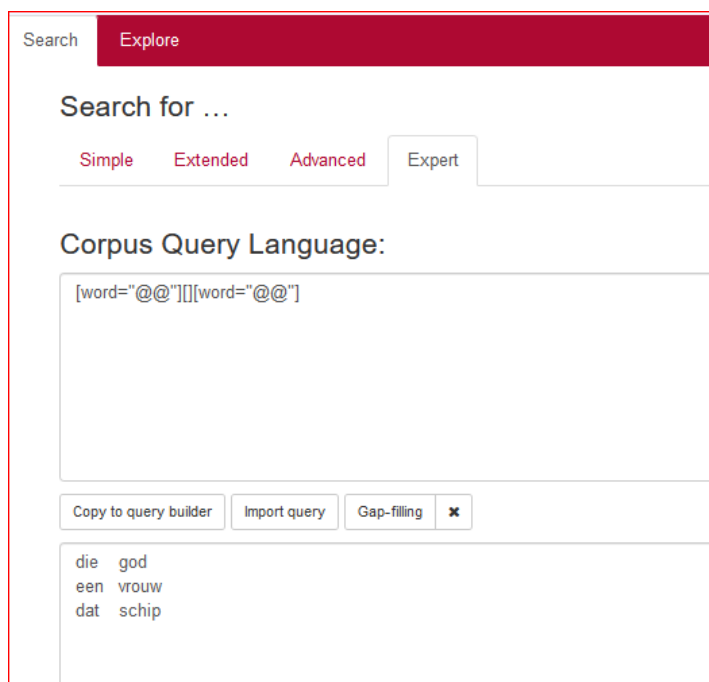


A .tsv file or a comparable .txt file enables you to complete a query with marked gaps.

If, for instance, you are interested in the distribution of words that can be placed between two specific words you can create this query in the Corpus Query Language field:

```
[word="@@" ] [ ] [word="@@" ]
```

By clicking Gap-filling you can upload a file with a tab-separated list of values from your computer to substitute them for the gap values, i.e. the at signs (@@) in your query. After the upload your values will appear in a separate box:



The values in the first column – *die, een, dat* – will be entered at the position of the first gap (@@) and the values in the second column – *god, vrouw, schip* – at the position of the second gap. With these values, gap-filling yields the following results (titles are hidden):

The screenshot shows a search interface with a 'Per Document' view. At the top, there's a 'Hits' section with a 'Group hits by...' dropdown and a pagination bar showing '1' selected. Below this is a table with three columns: 'Before hit -', 'Hit -', and 'After hit -'. The 'Before hit' column contains a long list of text fragments. The 'Hit' column shows various instances of the word 'Schip' with different determiners: 'Dat selvighe Schip', 'dat Enghelsche Schip', 'dat vyfde Schip', 'dat gheen Schip', 'dat eene Schip', 'dat seecker Schip', 'dat de Schip', 'dat een Schip', 'dat gheen Schip', 'dat het Schip', 'dat het Schip', 'dat het Schip', 'dat het Schip', 'dat het Schip', 'dat een Schip', 'dat het Schip', 'dat het Schip', 'dat seker Schip', and 'een Bovenlantse Vrouw'. The 'After hit' column contains the corresponding text fragments from the original document.

Please note that for this to work, you do need to enter @@ in the field where you want the substitution to take place. An empty field ([]) will match any term.

Viewing results

Results can be viewed in two ways: Per hit (hit is defined as one token or a group of tokens that matched the query), or Per document (each document listed contains at least one hit).

Per Hit view

Click a hit – i.e. a line with the bold word(s) in the column Hit – to display the properties and values of the hit (in the following example **dat selvighe schip**). Click the hit again to close.

The screenshot shows a detailed view of a search result. At the top, there's a 'Document id: kranten_17_80035' and a 'Hit -' column containing 'Dat selvighe Schip'. To the right are 'After hit -', 'Word' (Dat selvighe Schip), 'Lemma' (dat zelvig schip), 'Part of speech with features' (CONJ PD(subtype=oth) NOU-C), 'Newspaper' (Courante uyt Italien, Duytslandt, &c.), and 'Date from' (16230909). Below this is a text snippet: 'Prins een deel Soldaten volghen sal. Ladinghe van t Schip de paltsgraef, t welc voor eenighe daghen uyt Oost-Indien in Enghelandt aen ghecomen is, groot 800. Vaten. 221823 pont Naghelen. 64513 pont Noten. 4043 pont picol Catty peper. 153 Balen Masilipatansche Calicour. 80 larres. 2 Ocxhoofden Gember. 2 Stuck Gember. Dat selvighe Schip brenght tydinghe, hoe dat vyf Hollantsche Schepen in Compagnie waren, de vier waren waeren wel ghearriveert in S. Helena, het vyfde was in noot, lagh op 60 Vadem voor S. Helena, de Hollantsche zonden eene Sloepe met volck naer haer toe, maer alsoo het te hert waeyde cost daer niet'. Below the text is a table with 'Property' and 'Value' columns: Word (Dat selvighe), Lemma (dat), Part of speech with features (CONJ), and Value (selvighe, zelvig, PD(subtype=oth), Schip, schip, NOU-C).

Hit rows are always preceded by a row containing the document title in which those hits occurred, in this case *Courante uyt Italien, Duytslandt, &c.* The document titles can be toggled on or off by using

the Hide Titles (or Show Titles when titles are hidden) button at the bottom of the page. If you hover the mouse over the title, the identification number of the document appears, in this case: kranten_17_80035.

Sorting results

Click on any of the column headings to sort the hits on Words within that column, clicking again inverts the sorting. Extra sorting options are given when clicking on Before hit, Hit and After hit: you can sort by various attributes, as shown below.

The screenshot shows a search interface with a table of results. The table has columns: Before hit, Hit, After hit, Word, Lemma, Part of speech with features, Newspaper, and Date from. A context menu is open over the 'Hit' column, showing options: Word, Part of speech, Lemma, and Part of speech with features. The table contains three rows of data with corresponding text snippets.

Before hit	Hit	After hit	Word	Lemma	Part of speech with features	Newspaper	Date from
Venetien den 15 ...gelijk geschie	Word	onder resistentie	dat schoone	dat schoon	CONJ AA NOU-C	Oprechte Haerlemsche	16710829
Uyt Londen, den 7 ...heeft men va	Part of speech	enomen; en de...	Schip	schip		courant	
Amsterdam den 19 luy.	Lemma	ier thuy's hoorende, van	dat seker Schip	dat zeker	CONJ AA NOU-C	Ordinaris dingsdaeghse	16530218
	Part of speech with features	de...		schip		courante	

You can also sort the results by means of the drop-down menu at the bottom of the page (Sort by ...), which offers you the possibility to sort by various attributes as Hit, Before hit, After hit, Title, Date and Article properties.

Grouping results

Results Per Hit can be grouped by properties of Hit, Before hit, After hit, Title, Date and Article properties. Grouping is facilitated by the drop-down menu Group hits by. By selecting one of the properties a tick box appears that makes it possible to distinguish between case sensitive and case insensitive.

The screenshot shows a search interface with grouped results. The 'Group by Word' dropdown is set to 'hit:word' and 'Case-sensitive' is checked. The results are displayed as a table with columns: Group, #hits in group, and Relative frequency (hits). The table shows various forms of the word 'Schip' and their respective frequencies.

Group	#hits in group	Relative frequency (hits)
dat het Schip	771	0.00407%
dat t Schip	351	0.00185%
dat een Schip	186	0.000983%
dat het schip	97	0.000513%
dat t schip	32	0.000169%
dat haer Schip	23	0.000122%
dat dit Schip	20	0.000106%
dat sijn Schip	17	0.0000898%
dat een schip	11	0.0000581%
dat ons Schip	6	0.0000317%

Advanced grouping options are available by selecting the option Context (advanced). It allows you to group the results by up to 5 tokens before or after the hits. It also allows you to group the results based on (parts of) the hits. By pressing the New context group you can group the results by another property or another range.

We will work that out using an example. A noun phrase consisting of a pronoun/determiner *die*, the adjective *schone* or *schoone* and an arbitrary noun *may* be found in Expert Search with the following query: `[word="die"] [word="scho.?ne"] [pos="nou-c"]`. This produces the following hits (Titles are hidden):

Before hit	Hit	After hit	Lemma	Part of speech with features
... gheschiet, dan omtrent Presburgh, zijn	die schoone Marckten	Sint Ioris, Bossing, Modo ende...	die schoon markt	PD(subtype=oth) AA NOU-C
... ruyneren ghelijck tot Regensburgh met	die schoone steenen	brugge een aenvanck gemaect wordt...	die schoon steen	PD(subtype=oth) AA NOU-C
... om dat als de Vreemdelingen	die schoone plaets	gaen besien, sy de selve...	die schoon plaats	PD(subtype=oth) AA NOU-C
... werpen en te raseren, en	die schoone Stadt	tot een open Vleck te...	die schoon stad	PD(subtype=oth) AA NOU-C
... gekomen, weten nu weynig van	die schoone tydinge	van Babilonien: maer wel dat...	die schoon tijding	PD(subtype=oth) AA NOU-C
... verwacht, nevens de particulariteyten van	die schoone Actie	, die die Rovers by na...	die schoon actie	PD(subtype=oth) AA NOU-C
... is hy over sulcx opgebroken,	die schoone Marc	Hambach ende Slicht uyt gheplondert...	die schoon merk	PD(subtype=oth) AA NOU-C
... niet bytijts gemerckt had, had	die schoone Kerck	en een groot gedeelte der...	die schoon kerk	PD(subtype=oth) AA NOU-C
... van Riga komende rapporteert, dat	die schoone Stadt	een seer groot Ongeluck gehad...	die schoon stad	PD(subtype=oth) AA NOU-C
... veel schdae gelden werdt, werdende	die schoone Lantsdowen	, welke in hare Vruchten staen...	die schoon landouw	PD(subtype=oth) AA NOU-C
... te stellen, soo dat al	die schoone moeyte	van den Cardinael Azzolini, dewelcke...	die schoon moeyte	PD(subtype=oth) AA NOU-C
... niet meer, als 3 uren,	die schoone Kerck	t'eenemael verbrant, en niet, als...	die schoon kerk	PD(subtype=oth) AA NOU-C
... in vlam geraeckt, waer door	die schoone Abdy	in de assche gelegd is...	die schoon abjij	PD(subtype=oth) AA NOU-C

It is now possible to group the hits by the second and third tokens of those hits. See below.

Group	#hits in group	Relative frequency (hits)
schoone Kerck	2	0.0000106%
schoone Stadt	2	0.0000106%
schoone Marc	1	0.00000528%
schoone tydinge	1	0.00000528%
schoone Marckten	1	0.00000528%
schoone moeyte	1	0.00000528%
schoone Actie	1	0.00000528%
schoone Abdy	1	0.00000528%
schoone steenen	1	0.00000528%
schoone Lantsdowen	1	0.00000528%
schoone plaets	1	0.00000528%

Click a group to show or hide hits within that group, as shown below. Click once more on the group to close it again.

Group	#hits in group	Relative frequency (hits)
schoone Kerck	2	0.0000106%
schoone Stadt	2	0.0000106%

<< View detailed concordances

Before	Hit	After
werpen en te raseren, en van Riga komende rapporteert, dat	die schoone Stadt die schoone Stadt	tot een open Vleck te een seer groot Ongeluck gehad

schoone Marc	1	0.00000528%
schoone tydinge	1	0.00000528%
schoone Marckten	1	0.00000528%
schoone moeyte	1	0.00000528%

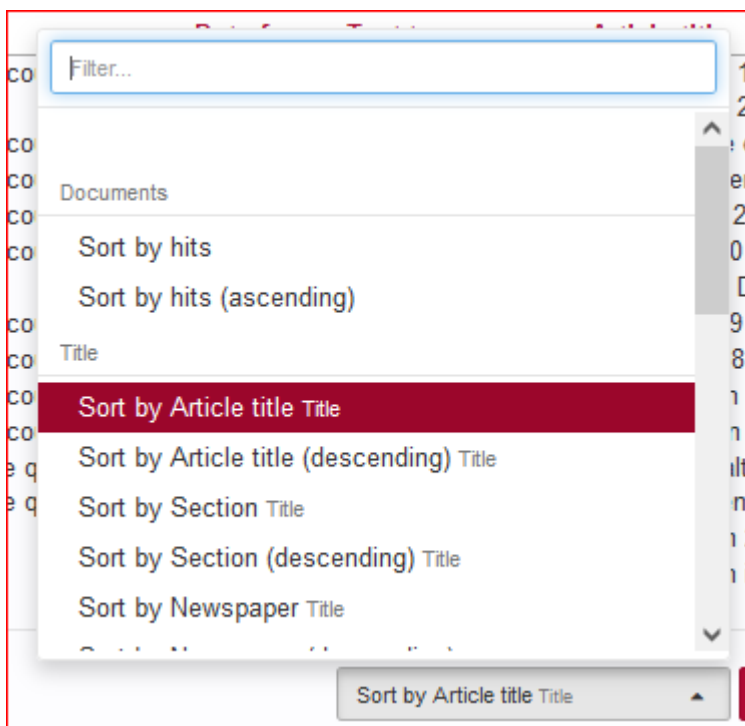
If more than twenty hits are found in a document, you can make them appear by clicking on Load more concordances; this button will appear right to the button View detailed concordances.

Click on View detailed concordances to go back to the normal hits view to see more detailed information for the hits in this group. The button Go back to grouped view brings you back to the list of groups.

Per Document view

Sorting results

Results can be sorted by means of the drop-down menu at the bottom of the page, which enables you to sort by Documents (e.g. the number of hits for your search query) and by Title, Date and Article properties.



Click on a Document title to show the Content of the document in a new window. Hits from the current query will be highlighted in bold in the opened document. In the case of several hits the first hit will also appear in shadow. You can go to the next (or previous) hit within the same document by pressing the Hits (and – in the case of many hits – Pages) button.

Grouping results

Results Per Document can be grouped by the metadata of the documents in which those hits occur (Title, Date and Article properties). Here, grouping is facilitated by the drop-down menu Group docs by.

Exporting results

The search results – both Per hit as Per document – can be exported by using the Export or the Export for Excel button at the bottom right of the page. The first button transfers the search results – including all metadata – to a Comma-Separated Values-file. These CSV-files consist only of text data, which makes it easy to implement (read and/or write) them into a spreadsheet or database program. The second button offers the possibility to export the results – including all metadata – to a CSV-file for use with Excel.

Grouped results can be exported in the same way. However, if you would like to have the metadata with each concordance of a group, you must first click on the red bar of a specific group and then on View detailed concordances (see screenshot below). The results you then see can be exported by the use of the Export buttons. This operation must be carried out for each individual group you wish to export.

Document	Day	Month	Year	Hits
Oprechte Haerlemsche courant, Regensburg den 3 Maert. (1689/3/12)	12	3	1689	2
Oprechte Haerlemsche courant, Venetien den 25 February. (1689/3/12)	12	3	1689	1
Oprechte Haerlemsche courant, Regensburg den 3 Maert. (1689/3/12)	12	3	1689	2
Oprechte Haerlemsche courant, Venetien den 25 February. (1689/3/12)	12	3	1689	1

Information about a document

Click on a document title to open this document in a new window: the Content window.

Content

Hits from the current query will be highlighted in bold in the opened document. In the case of several hits only the current hit will also appear in shadow. You can navigate from one hit to another by using the arrows at the Hits button:



When you hover with your mouse over a specific word in the document a pop-up will appear with the modern lemma and the option “Show details”. By clicking this link you will see extra information on word level:

Schryvens voor Alle-cante van Capiteyn Jan Pellecorn melt dat hy den voornoemden
 Roover, in onse[...] gaende nouvelen van verhaeldt, alder-eerst zijn[...] Stengh, Mars-
 mael aen boort gheweest, ende
 etst, ende met drie Bolck-bassie
 enen als voor desen verhaelt: Het
 en Coopman, groot 200. last, op
 hebbende 180. **Man**. Capiteyn Pellecorn is selver gequetst, hebbende twee schoten,
 een inde schouder, d'ander int been met vier dooden, den Schipper, Corporael,
 Opper-timmerman ende een Busschieter. Tot Argier is gevangen den Spaenschen

Word: Man
 Word id: INT_47bab8f5-cd5b-4b95-bd04-a4df88e221e3.w.678
 Lemma: man

You can also click on ‘View page in Delpher’ to look at the original document itself.

Metadata

In the Metadata tab all metadata properties of the document are displayed. They provide information about Title, Date, Article properties and Document length (tokens).

Statistics

The Statistics tab shows several document statistics: the number of Tokens, Types, Lemmas and the Type/token ratio. It is possible to print or to download these statistics via the menu symbol right of the title Token/Part of Speech Distribution respectively the title Vocabulary Growth.

Exploring the corpus

The Explore tab has three subdivisions: Documents, N-grams and Statistics.

Documents

This subtab allows you to investigate the documents. It consists of two drop-down menus to specify the grouping of the metadata and to specify the way the groups are to be shown.

A simple example: suppose we want to know which newspapers present news items about Poland (in Dutch: *Polen*) in the *Couranten Corpus*.

- In the Group documents by metadata drop-down menu, choose Group by Newspaper
- In Show groups as, select *docs*
- In the metadata search form (Filter search by), select in Article properties Country *Polen*
- Press Search

Search Explore

Explore ...

Documents N-grams Statistics

Group documents by metadata Group by Newspaper Title

Show groups as docs

Filter search by ...

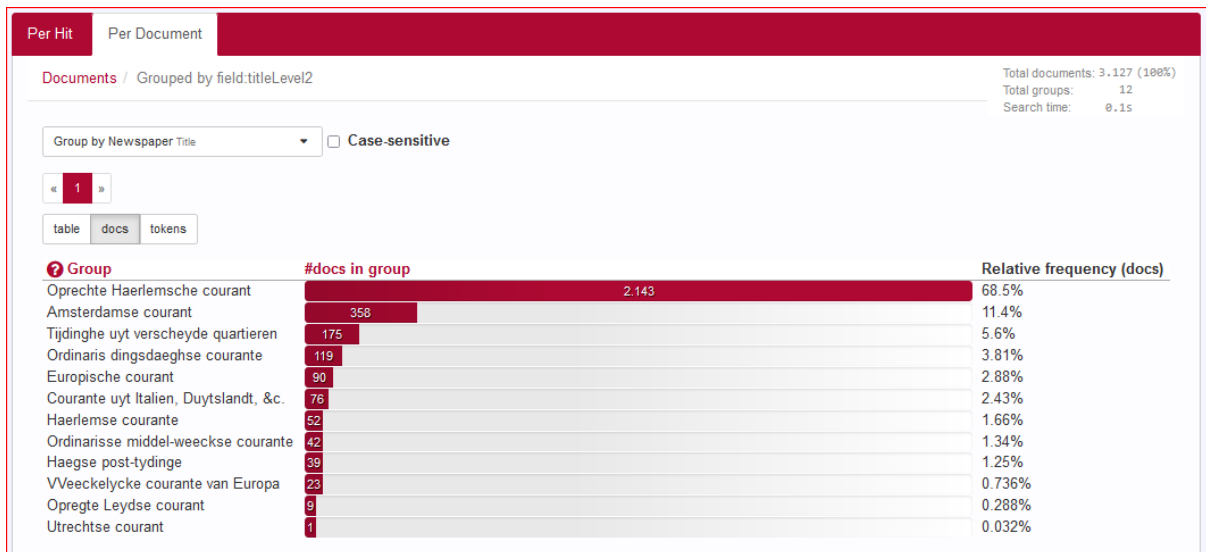
Date Title Article properties 1

Country polen

Place Place

Text type Text type

You will get this result:



N-grams

An *N-gram* is a sequence of *N* items. This option will list the frequency of different N-grams in a (sub-)corpus.

Options

- *N-gram size*: the length of the sequence (a number from 1 to 5; default setting is 5)
- *N-gram-type*: choose for sequences of Word (i.e. word form), Part of speech, Lemma or Part of speech with features. If you do not specify the search term further, a series of five consecutive Words, Parts of speech, Lemmas or Parts of speech with features will be searched for.

- It is also possible to restrict to, for instance, 5-grams with some slots already specified, as is shown in the following example.
- By using the Filter search by ... you can create a subcorpus within the *Couranten Corpus* for specific metadata.

Example

The screenshot shows the 'Explore ...' interface with the following settings:

- Documents** (selected tab)
- N-gram size**: 5
- N-gram type**: Word
- Five filter slots, each with a dropdown menu and a text input field:

Lemma	Part of speech	Word	Part of speech	Word
de	Adjective-Adverb	Word	Verb	Word

Within all the documents of the *Couranten Corpus*, you will find 1221 occurrences of this so-called 5-gram in news items from Poland (Filter search by ..., Country: Polen).

The screenshot shows the 'Per Document' view with the following statistics:

- Total hits: 1,221 (0.281%)
- Total groups: 1,174
- Search time: 0.25

The table below shows the results for the 5-gram search:

Group	#hits in group	Relative frequency (hits)
De tegenwoordige Convocationsdag soude aldaer	2	0.00046%
de Turckse manier gemaect 1200	2	0.00046%
de Coninckijcke Militie stonden met	2	0.00046%
den Grooten Heer heeft bemiddelt	2	0.00046%
De Sweetse Cavallerie blyft noch	2	0.00046%
Den Grooten Heer sal in	2	0.00046%
den Grooten Sultan continueert dat	2	0.00046%
De Duytsche Koopluyden saten 150	2	0.00046%
de Moorssche Wijs gekleet De	2	0.00046%
de geconfedereerde Armee af-reysen maer	2	0.00046%
de Sweedtse kant sullen alle	2	0.00046%
de gestelde tyd zyn voortgang	2	0.00046%
de volkomen Rouw sal zyn	2	0.00046%
de gantse Armee bevolen den	2	0.00046%
den andere niet sal verlaten	2	0.00046%
de Moscovische syde hebben begeert	2	0.00046%

Statistics (frequency lists)

Here, you can produce frequency lists for the corpus. It is rather similar to the previous option, but restricted to 1-grams.

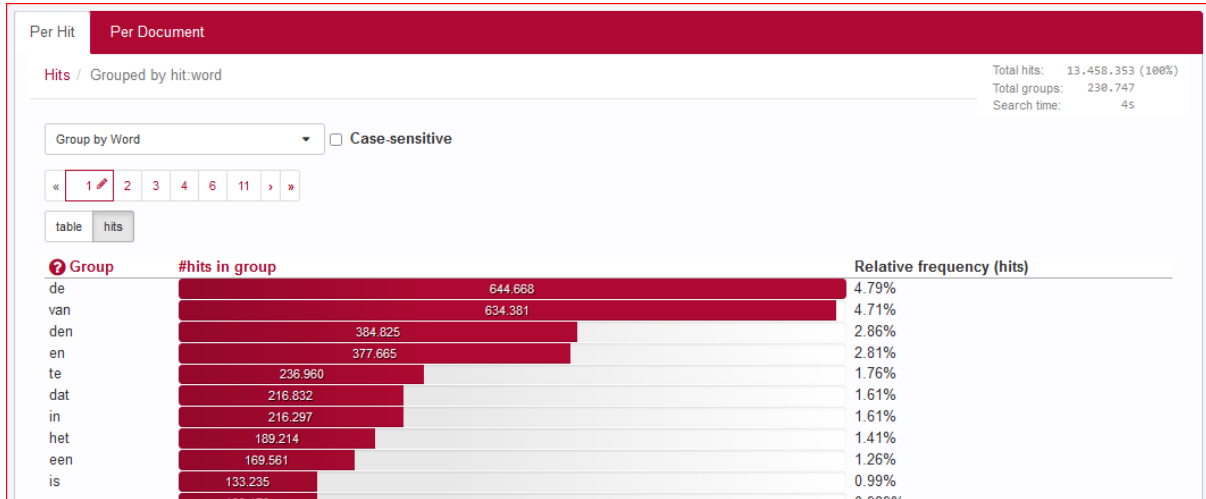
Options

- *Frequency list type*: choose for Word (i.e. word form), Part of speech, Lemma or Part of speech with features.

- By using the Filter search by... you can create a subcorpus within the *Couranten Corpus* for specific metadata.

Example

It is possible to determine the use of the ten most frequently used words in the *Oprechte Haerlemsche courant* in the *Couranten Corpus* by searching for Frequency list type Word and by filtering search by Title and Newspaper. This results in:



Appendix: Corpus Query Language

BlackLab supports Corpus Query Language, a full-featured query language introduced by the IMS Corpus WorkBench (CWB) and also supported by the Lexicom Sketch Engine. It is a standard and powerful way of searching corpus.

The basics of Corpus Query Language is the same in all three projects, but there are a few minor differences in some of the more advanced features, as well as some features that are exclusive to some projects. For most queries however, this will not be an issue.

This page will introduce the query language and show all features that BlackLab supports. If you want to learn even more about CQL, see [CWB CQP Query Language Tutorial](#) and [Sketch Engine Corpus Query Language](#).

CQL support

For those who already know CQL, here's a quick overview of the extent of BlackLab's support for this query language. If there is a feature we don't support, yet is important to you, please let us know. If it's quick to add, we may be able to help you out.

Supported features

BlackLab currently supports (arguably) most of the important features of Corpus Query Language:

- Matching on token annotations (also called properties or attributes), using regular expressions and =, !=, !. Example: [word="bank"] (or just "bank")
- Case/accent-sensitive matching. Note that, unlike in CWB, case-INsensitive matching is currently the default. To explicitly match case/accent-insensitivity, use "(?i)...". Example: "(?i)Mr\." "(?i)Banks"
- Combining criteria using &, | and !. Parentheses can also be used for grouping. Example: [lemma="bank" & pos="V"]
- Match-all pattern [] matches any token. Example: "a" [] "day"
- Regular expression operators +, *, ?, {n}, {n,m} at the token level. Example: [pos="AA"]+
- Sequences of token constraints. Example: [pos="AA"] "cow"
- Operators |, & and parentheses can be used to build complex sequence queries. Example: "happy" "dog" | "sad" cat"
- Querying with tag positions using e.g. <s> (start of sentence), </s> (end of sentence), <s/> (whole sentence) or <s> ... </s> (equivalent to <s/> containing ...). Example: <s> "The" . XML attribute values may be used as well, e.g. <ne type="PERS"/> ("named entities that are persons").
- Using within and containing operators to find hits inside another set of hits. Example: "you" "are" within <s/>
- Using an anchor to capture a token position. Example: "big" A:[]. Captured matches can be used in global constraints (see next item) or processed separately later (using the Java interface; capture information is not yet returned by BlackLab Server). Note that BlackLab can actually capture entire groups of tokens as well, similarly to regular expression engines.

- Global constraints on captured tokens, such as requiring them to contain the same word.

Example: "big" A:[] "or" "small" B:[] :: A.word = B.word

See below for features not in this list that may be added soon, and let us know if you want a particular feature to be added.

Differences from CWB

BlackLab's CQL syntax and behaviour differs in a few small ways from CWBs. In future, we'll aim towards greater compliance with CWB's de-facto standard (with some extra features and conveniences).

For now, here's what you should know:

- Case-insensitive search is currently the default in BlackLab, although you can change this if you wish. CWB and Sketch Engine use case-sensitive search as the default. We may change our default in a future major version.
If you want to switch case-/diacritics-sensitivity, use "(?i).." (case-sensitive) or "(?i).." (case-insensitive, usually the default). CWBs %cd flags for setting case/diacritics-sensitivity are not (yet) supported, but will be added.
- If you want to match a string literally, not as a regular expression, use backslash escaping: "e.g.". %l for literal matching is not yet supported, but will be added.
- BlackLab supports result set manipulation such as: sorting (including on specific context words), grouping/frequency distribution, subsets, sampling, setting context size, etc. However, these are supported through the REST and Java APIs, not through a command interface like in CWB. See [BlackLab Server overview](#).
- Querying XML elements and attributes looks natural in BlackLab: <s/> means "sentences", <s> means "starts of sentences", <s type='A'> means "sentence tags with a type attribute with value A". This natural syntax differs from CWBs in some places, however, particularly when matching XML attributes. While we believe our syntax is the superior one, we may add support for the CWB syntax as an alternative.
We only support literal matching of XML attributes at the moment, but this will be expanded to full regex matching.
- In global constraints (expressions occurring after ::), only literal matching (no regex matching) is currently supported. Regex matching will be added soon. For now, instead of A:[] "dog" :: A.word = "happy|sad", use "happy|sad" "dog".
- To expand your query to return whole sentences, use <s/> containing (...). We don't yet support CWBs expand to, expand left to, etc., but may add this in the future.
- The implication operator -> is currently only supported in global constraints (expressions after the :: operator), not in regular token constraints. We may add this if there's demand for it.
- We don't support the @ anchor and corresponding target label; use a named anchor instead. If someone makes a good case for it, we will consider adding this feature.
- backreferences to anchors only work in global constraints, so this doesn't work: A:[] [] [word = A.word]. Instead, use something like: A:[] [] B:[] :: A.word = B.word. We hope to add support for these in the near future, but our matching approach may not allow full support for this in all cases.

(Currently) unsupported features

The following features are not (yet) supported:

- intersection, union and difference operators. These three operators will be added in the future. For now, the first two can be achieved using & and | at the sequence level, e.g. "double" [] & [] "trouble" to match the intersection of these queries, i.e. "double trouble" and "happy" "dog" | "sad "cat" to match the union of "happy dog" and "sad cat".
- _ meaning "the current token" in token constraints. We will add this soon.
- lbound, rbound functions to get the edge of a region. We will probably add these.
- distance, distabs functions and match, matchend anchor points (sometimes used in global constraints). We will see about adding these.
- using an XML element name to mean 'token is contained within', like [(pos = "N") & !np] meaning "noun NOT inside in an tag". We will see about adding these.
- a number of less well-known features. If people ask, we will consider adding them.

Using Corpus Query Language

Matching tokens

Corpus Query Language is a way to specify a "pattern" of tokens (i.e. words) you're looking for. A simple pattern is this one:

```
[word="man"]
```

This simply searches for all occurrences of the word "man". If your corpus includes the per-word properties lemma (i.e. headword) and pos (part-of-speech, i.e. noun, verb, etc.), you can query those as well. For example, to find a form of word "search" used as a noun, use this query:

```
[lemma="search" & pos="NOU-C"]
```

This query would match "search" and "searches" where used as a noun. (Of course, your data may contain slightly different part-of-speech tags.)

The first query could be written even simpler without brackets, because "word" is the default property:

```
"man"
```

You can use the "does not equal" operator (!=) to search for all words except nouns:

```
[pos != "NOU-C"]
```

The strings between quotes can also contain wildcards, of sorts. To be precise, they are [regular expressions](#), which provide a flexible way of matching strings of text. For example, to find "man" or "woman", use:

```
"(wo)?man"
```

And to find lemmata starting with "under", use:

```
[lemma="under.*"]
```

Explaining regular expression syntax is beyond the scope of this document, but for a complete overview, see [here](#).

Sequences

Corpus Query Language allows you to search for sequences of words as well (i.e. phrase searches, but with many more possibilities). To search for the phrase "the tall man", use this query:

```
"the" "tall" "man"
```

It might seem a bit clunky to separately quote each word, but this allows us the flexibility to specify exactly what kinds of words we're looking for. For example, if you want to know all single adjectives used with man (not just "tall"), use this:

```
"an? | the" [pos="AA"] "man"
```

This would also match "a wise man", "an important man", "the foolish man", etc.

Regular expression operators on tokens

Corpus Query Language really starts to shine when you use the regular expression operators on whole tokens as well. If we want to see not just single adjectives applied to "man", but multiple as well:

```
"an? | the" [pos="AA"]+ "man"
```

This query matches "a little green man", for example. The plus sign after [pos="AA"] says that the preceding part should occur one or more times (similarly, * means "zero or more times", and ? means "zero or one time").

If you only want matches with two or three adjectives, you can specify that too:

```
"an? | the" [pos="AA"] {2,3} "man"
```

Or, for two or more adjectives:

```
"an? | the" [pos="AA"] {2,} "man"
```

You can group sequences of tokens with parentheses and apply operators to the whole group as well. To search for a sequence of nouns, each optionally preceded by an article:

```
("an? | the"? [pos="NOU-C"])+
```

This would, for example, match the well-known palindrome "a man, a plan, a canal: Panama!" (A note about punctuation: in BlackLab, punctuation tends to not be indexed as a separate token, but as a property of a word token - CWB and Sketch Engine on the other hand tend to index punctuation as a separate token instead. You certainly could choose to index punctuation as a separate token in BlackLab, by the way -- it's just not commonly done. Both approaches have their advantages and disadvantages, and of course the choice affects how you write your queries.)

Case- and diacritics-sensitivity

CWB and Sketch Engine both default to (case- and diacritics-)sensitive search. That is, they exactly match upper- and lowercase letters in your query, plus any accented letters in the query as well.

BlackLab, on the contrary, defaults to *IN*sensitive search (although this default can be changed if you like). To match a pattern sensitively, prefix it with "(?-i)":

```
" (?-i) Panama "
```

If you've changed the default search to sensitive, but you wish to match a pattern in your query insensitively, prefix it with "(?i)":

```
[pos=" (?i) NOU-C "]
```

Although BlackLab is capable of setting case- and diacritics-sensitivity separately, it is not yet possible from Corpus Query Language. We may add this capability if requested.

Matching XML elements

Corpus Query Language allows you to find text in relation to XML elements that occur in it. For example, if your data contains sentence tags, you could look for sentences starting with "the":

```
<s>"the"
```

Similarly, to find sentences ending in "that", you would use:

```
"that" </s>
```

You can also search for words occurring inside a specific element. Say you've run named entity recognition on your data and all person names are surrounded with <person>...</person> tags. To find the word "baker" as part of a person's name, use:

```
"baker" within <person/>
```

Note the forward slash at the end of the tag. This way of referring to the element means "the whole element". Compare this to <person>, which means "the element's open tag", and </person>, which means "the element's close tag".

The above query will just match the word "baker" as part of a person's name. But you're likely more interested in the entire name that contains the word "baker". So, to find those full names, use:

```
<person/> containing "baker"
```

Or, if you simply want to find all persons, use:

```
<person/>
```

As you can see, the XML element reference is just another query that yields a number of matches. So as you might have guessed, you can use "within" and "containing" with any other query as well. For example:

```
( [pos="AA"]+ containing "tall" ) "man"
```

will find adjectives applied to man, where one of those adjectives is "tall".

Labeling tokens, capturing groups

Just like in regular expressions, it is possible to "capture" part of the match for your query in a "group".

CWB and Sketch Engine offer similar functionality, but instead of capturing part of the query, they label a single token. BlackLab's functionality is very similar but can capture a number of tokens as well. For example:

```
"an?|the" Adjectives: [pos="AA"]+ "man"
```

This will capture the adjectives found for each match in a captured group named "Adjectives".

BlackLab also supports numbered groups:

```
"an?|the" 1: [pos="AA"]+ "man"
```

Global constraints

If you tag certain tokens with labels, you can also apply "global constraints" on these tokens. This is a way of relating different tokens to one another, for example requiring that they correspond to the same word:

```
A: [] "by" B: [] :: A.word = B.word
```

This would match "day by day", "step by step", etc.