

# CJN application manual

## Table of contents

<b>Introduction</b>	<b>4</b>
Information about the corpus	4
Linguistic Annotation	4
Metadata categories	4
<b>Application user manual</b>	<b>5</b>
Getting started	5
Searching the corpus	5
Simple search	5
Search	5
Wildcards	6
Reset	6
History	6
Global settings	7
Extended search	7
Main	8
PoS features	8
Wildcards	9
Upload a list of values	10
Part of speech dialog box	10
Starting a new search	11
Filter search by	11
Advanced search	11
The query builder	11
The tab search	12
Adding attributes to a token box	13
Function of the two +-buttons in a token box	13
The tab options	14
Managing sequences of token boxes	15
Uploading value lists in the query builder	15

Copy to CQL editor	16
Expert search	16
Copy to query builder	17
Import query	17
Gap filling	17
Viewing results	19
Per Hit view	19
Sorting results	19
Grouping results	19
Per Document view	21
Sorting results	21
Grouping results	21
Exporting results	21
Information about a document	22
Content	22
Metadata of a document	23
Statistics	23
Exploring the corpus	23
Documents	23
N-grams	24
Options	24
Example	24
Statistics (frequency lists)	25
Options	25
Example	25
<b>Appendix: Corpus Query Language</b>	<b>26</b>
CQL support	26
Supported features	26
Differences from CWB	27
(Currently) unsupported features	28
Using Corpus Query Language	28
Matching tokens	28
Sequences	29
Regular expression operators on tokens	29

Case- and diacritics-sensitivity	30
Matching XML elements	30
Labeling tokens, capturing groups	31
Global constraints	31

# Introduction

This manual describes the corpus exploitation environment for the *Corpus Juridisch Nederlands*. The corpus application is developed by the 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 Corpus Juridisch Nederlands comprises a collection of 5.856 legal texts that could be consulted from the mid-1980s until 1992 as N-Lex, a database of current Dutch legislation. The material has been made available by the Centre for Informatics and Law of the Erasmus University in Rotterdam. The files have been compiled per year and run from 1814 to 1989. Only a few French-language texts and some undated texts have not been included in the corpus. [Note that the current website [N-Lex](#) contains the consolidated Dutch legislation which is or has been in force since 1 May 2002.]

The documents that now make up the Corpus Juridisch Nederlands were originally part of the [Corpus Hedendaags Nederlands](#). Because these texts date from 1814 to 1989, they are out of place in the latest version of the Corpus Contemporary Dutch. This is why these documents have been incorporated in a separate Corpus Juridisch Nederlands.

The corpus was first published as part of the *38 Million Words Corpus* of the Instituut voor Nederlandse Lexicologie, in 1996, and rereleased as part of the *Corpus Hedendaags Nederlands* in 2014. In its current form, it was released in September 2021.

## Linguistic Annotation

The corpus has been PoS-tagged by means of an SVM-based tagger trained on a mapped version of the SoNaR-1 corpus (<http://hdl.handle.net/10032/tm-a2-h5>) and lemmatized by a lemmatizer trained on the [GiGaNT-Molex lexicon](#). The tagset used is based on the working paper “[De morfosyntactische module van het GiGaNT-lexicon](#)”.

Since linguistic enrichment took place automatically and it was not feasible to correct all data manually, some imperfections in the data are inevitable.

## Metadata categories

The *Corpus Juridisch Nederlands* has been enriched with only a few metadata categories. The only metadata category that can be searched for is *Year*, i.e. the year the law texts were written. Other metadata categories, like the fact that the Language Variant is NN (Netherlandic Dutch) for this collection, have not been made searchable.

# 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 or the attribute Word in Extended Search:
  - Simple Search for Word [woningbouw](#)
  - Extended Search for Word [wetgever](#)
- To search for words 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](#) and [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 onwettige](#)
  - example Group by Word before: [different words preceding the word koningin](#)
- To explore the distribution of document properties in the corpus, use the Explore feature
  - example: [characteristics about the year](#)

## Searching the corpus

### Simple search



The screenshot shows the 'Search' tab of the application interface. At the top, there are two tabs: 'Search' (selected) and 'Explore'. Below the tabs is a search bar labeled 'Search for ...'. Underneath the search bar are four buttons: 'Simple' (selected), 'Extended', 'Advanced', and 'Expert'. Below these buttons is a text input field labeled 'Word' containing the text 'huis'. At the bottom of the search area are four buttons: 'Search' (highlighted in red), 'Reset', 'History', and a settings gear icon.

### Search

The Simple Search allows you to quickly search for specific word forms (e.g. *huis*). It is also possible to enter a phrase: *met ingang van* or *namens de minister*. You will then find all occurrences of that exact phrase.

Note that in Simple Search the patterns will be matched case-insensitively: *justitie* for instance will deliver the same results as *Justitie* or *JUSTITIE*. 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 or lemmata. 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. *aan*, *artikelen* and *aanzien*.
- ? The question mark matches a single character once. Therefore, searching for *a?n* matches *only* three-letter values starting with an *a* and ending with a *n*, e.g. *aan*, *a-n*, *arn* and *ann*. This wildcard can be used more than once. Thus *a???n* matches words like *allen*, *akten* en *Assen*.

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.]

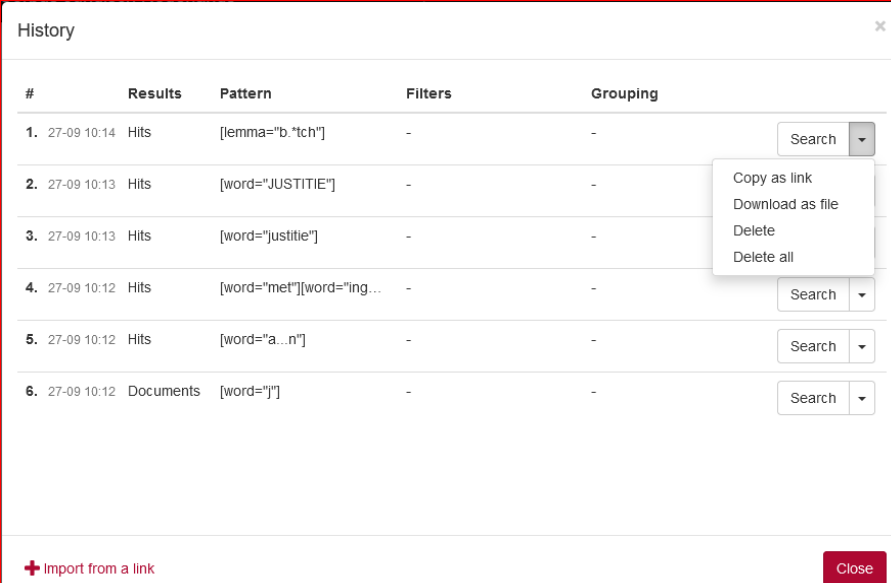
## Reset

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).



The screenshot shows a 'History' window with a table of search queries. The table has columns for '#', 'Results', 'Pattern', 'Filters', and 'Grouping'. Each row represents a search query with a 'Search' button and a dropdown menu. The dropdown menu for the second query is open, showing options: 'Copy as link', 'Download as file', 'Delete', and 'Delete all'. At the bottom of the window, there is a '+ Import from a link' button and a 'Close' button.

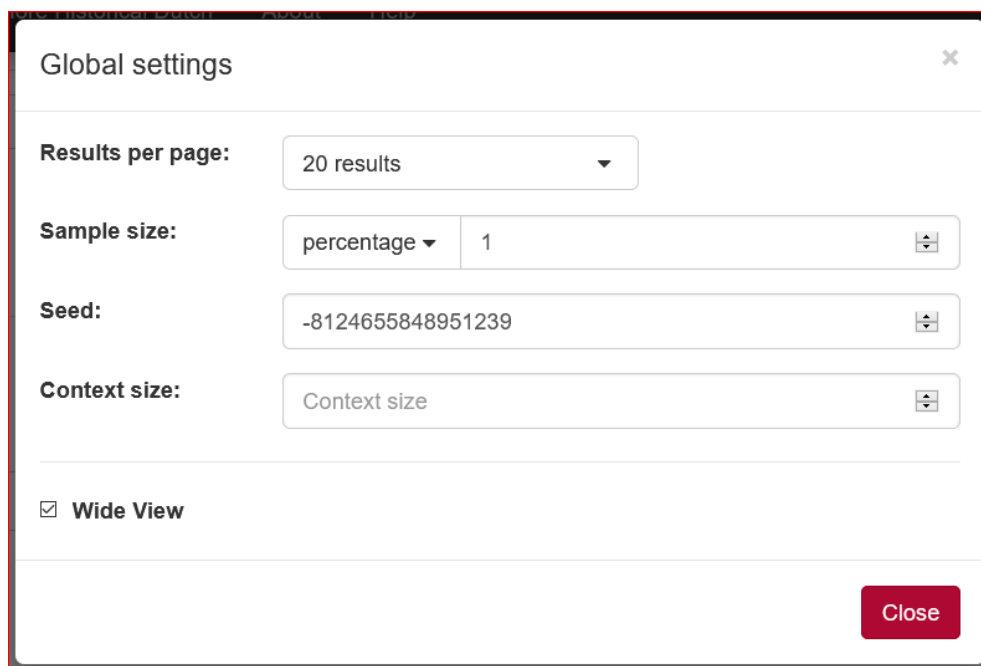
#	Results	Pattern	Filters	Grouping
1.	27-09 10:14 Hits	[lemma="b.*tch"]	-	-
2.	27-09 10:13 Hits	[word="JUSTITIE"]	-	-
3.	27-09 10:13 Hits	[word="justitie"]	-	-
4.	27-09 10:12 Hits	[word="met"][word="ing...]	-	-
5.	27-09 10:12 Hits	[word="a...n"]	-	-
6.	27-09 10:12 Documents	[word="j"]	-	-

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 dialog box titled "Global settings" with a close button (X) in the top right corner. The dialog contains the following settings:

- 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 "-8124655848951239".
- Context size:** A text input field containing "Context size".
- Wide View:** A checkbox that is checked.

A red "Close" button is located in the bottom right corner of the dialog.

## Extended search

The Extended Search allows you to find all occurrences of a *token* with its specific *attributes*. A *token* - usually just a single word - is the smallest unit within a corpus, whereas *attributes* are the different values that together make up a token.

## Main

In this corpus the three attributes you can search for are Word (more precise: word form), Lemma and Part of speech.

## PoS features

You can expand your search options by using the tab PoS features. The following screen will appear:

The screenshot displays a search interface with a red header bar containing 'Search' and 'Explore' tabs. Below the header, there are two main sections: 'Search for ...' and 'Filter search by ...'. The 'Search for ...' section has four tabs: 'Simple', 'Extended', 'Advanced', and 'Expert'. Underneath, there are two sub-tabs: 'Main' and 'PoS features'. The 'PoS features' tab is active, showing a list of attributes with drop-down menus: Type, Subtype, Gender, Number, Person, Tense, Mood, and Position. Below these is a 'Within:' section with buttons for 'Document', 'Sentence', and 'Paragraph'. The 'Filter search by ...' section includes a 'Year' filter with 'From' and 'To' input fields, and two radio buttons for 'Permissive' and 'Strict'. A summary of the selected subcorpus is displayed: 'Selected subcorpus: Total documents: 150 (100%), Total tokens: 12.981.471 (100%)'. At the bottom of the interface are four buttons: 'Search', 'Reset', 'History', and a settings gear icon.

All the values of these attributes can be selected by the use of a drop-down menu. The entered values in Main and those in PoS features will be combined in a search query. For instance, searching for the value *verklaren* (Main, Lemma) and the value *participle* (PoS features, Finiteness) will result in the word forms *verklarend* en *verklaard*.

By default, a search is performed across sentence boundaries, within documents. The search can be restricted to sentence by clicking on the word Sentence at Within. For example, the search terms *was verzekerd de* in the Word search bar yield six hits, but when Sentence is switched on there are no results at all.

In the search fields Word and Lemma enter the value of the attributes (or Upload a list of values; see below) you are looking for. In the search field Part of speech you can select the desired values. Then press enter or click the Search button below to execute the search and view the results. Note that the default setting for Word and Lemma in Extended search is case- and diacritics-insensitive. For example, searching for the Word *rechter* will result in all spelling variants as *rechter*, *Rechter* and *RECHTER*. In order to directly find only occurrences of the Word (form) *rechter*, tick the box Case- and diacritics-sensitive under the search field Word (as shown below).



Search
Explore

## Search for ...

Simple
Extended
Advanced
Expert

Main

PoS features

**Word**

↑

Case- and diacritics-sensitive

**Lemma**

↑

Case- and diacritics-sensitive

Please note that there is an important difference between the search fields Word and Lemma. As an example: entering the value *besluiten* in Word will only provide you with occurrences of that exact string of characters. When you enter *besluiten* in the search field Lemma you will - besides the lemma *besluiten* - also find all word forms that are linked to that lemma, such as the conjugated forms *besluit* and *besloten*.

### Wildcards

In Extended Search, the use of wildcards can prove good service to search for specific word forms or lemmata. 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* in Word matches all values that start with an *a* and end with a *n*, e.g. *aan*, *artikelen* and *aanzien*. Note that the same query in Lemma will give other results.
- ? The question mark matches a single character once. Therefore, searching for *a?n* in Lemma matches *only* three-letter values starting with an *a* and ending with a *n*, e.g. *aan*, *a-n*, *arn* and *ann*.

This wildcard can be used more than once. Searching for *e???n* in Word matches the word forms *eisen*, *eigen*, *ervan*, *eenen*, *erven* en *elken*, whereas searching for *e???n* in Lemma matches for instance the lemmata *eigen* (word forms *eigen*, *eigene*, *eigener*), *ervan* en *eisen* (word form *eist*, *eisen*, *eiste*, *geëist*).

Note that searching with wildcards is limited to Simple Search and Extended Search. (In Advanced Search and Expert Search you can use so-called regular expressions instead of wildcards.)

In the search fields Word and Lemma it is possible to search for different values simultaneously by separating them without spaces by a vertical line, e.g. *advocaat|bedrijf|vergunning* or - with the use of wildcards - *advocaat|bedrijf|ver\**.

For the search field Word it is also possible to search for a series of tokens by entering multiple values - including wildcards - separated by a space, e.g. *nodige wijzigingen, nodige \*, nodig geachte \**. Note that searching for *nodig wijziging, nodig \*, nodig geacht \** in the search field Lemma will give different results!

Values at the same position in different fields are grouped together as a single token, meaning that all values in the first position of each field are grouped to match a single token.

- A single-token example: searching for the Word(form) *loop* together with the part of speech Noun Common will result in a list of all verbs containing the word form *loop*. The syntax of your query is shown in the results: [\[word="loop"&pos="nou-c"\]](#).
- A multi-token example: searching for *mij gegeven* in the Word(form) field and *ik geven* in the Lemma field finds those occurrences of the bigram in which the first word is the declined form of the personal pronoun *ik* and the second belongs to the paradigm of the verb *geven*: [\[word="mij"&lemma="ik"\]\[word="gegeven"&lemma="geven"\]](#).

### Upload a list of values

At the right side of the search fields Word and Lemma there is an option to Upload a list of values; those values must all be separated by a white space. 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 first.)

Every word in the uploaded file will be added to the list of values to search for. To remove the word list simply delete all text in the search field or press the Reset button.

### Part of speech dialog box

Clicking on the pencil next to the search field Part of speech provides you with the Part of speech dialog box.

Part of Speech

Adjective-Adverb	<b>Number</b>	<b>Case</b>	<b>Gender</b>
Adposition	<input type="checkbox"/> singular	<input type="checkbox"/> genitive	<input type="checkbox"/> neuter
Adverb	<input type="checkbox"/> plural	<input type="checkbox"/> dative	<input type="checkbox"/> masculine or feminine
Conjunction	<input type="checkbox"/> unclear	<input type="checkbox"/> genitive or dative	<input type="checkbox"/> unclear
Interjection		<input type="checkbox"/> dative or accusative	
<b>Noun Common</b>			
Noun Proper			
Numeral			
Pronoun-Determiner			
Punctuation			
Residual			
Verb			

pos="nou-c"

OK Reset

For some of the categories on the left you can tick certain features to further specify your search query. By doing so you can for instance delimit your search, as shown in the above screenshot for Noun Common.

### Starting a new search

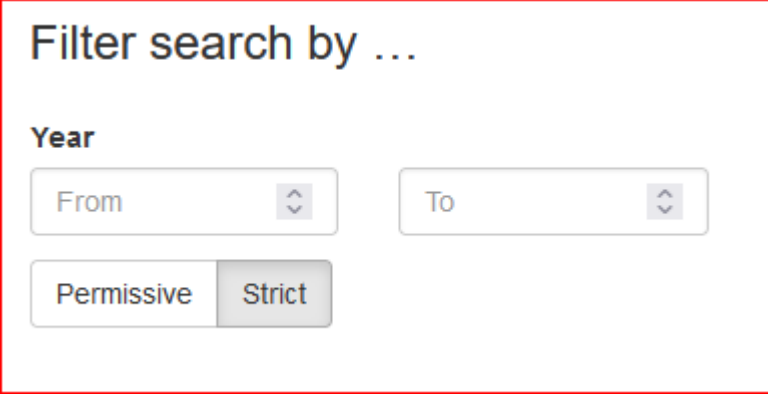
You can start a new search by pressing the Reset button. By doing so, both the search query and the hits found will disappear. Your search history, however, will remain unchanged.

The search fields Word and Lemma are provided with a list, which contains suggestions for possible search terms in alphabetical order, based on the characters typed in.

If you only use the fields Word or Lemma, there are two possibilities to start a search: fill in the desired value and press enter, or click the Search button. The only way to start a new search after a change in Part of Speech is to 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 to a certain period, using the filter Year. To view the results for all periods simply leave the attributes in the filtering form empty.



The screenshot shows a form titled "Filter search by ...". Under the heading "Year", there are two input fields: "From" and "To", each with a dropdown arrow. Below these fields are two buttons: "Permissive" and "Strict".

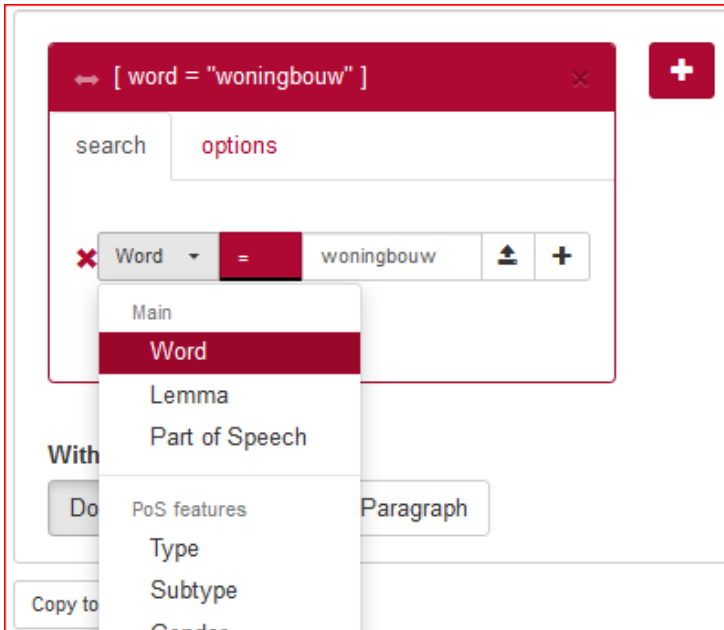
## Advanced search

### 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.

By default, a search is performed across sentence boundaries, within documents. The search can be restricted to sentence by clicking on the word Sentence at Within. For example, the search terms *was verzekerd de* in the Word search bar yield six hits, but when Sentence is switched on there are no results at all.

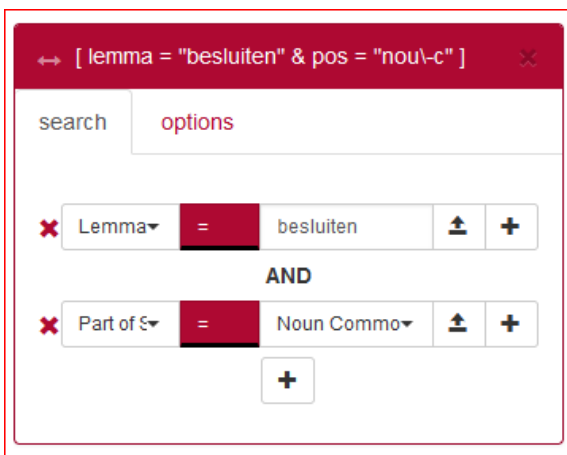


A token box in the querybuilder has two tabs: search and options.

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=*besluiten* and Part of Speech=Noun Common for example excludes all forms of the verb *besluiten*.

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:



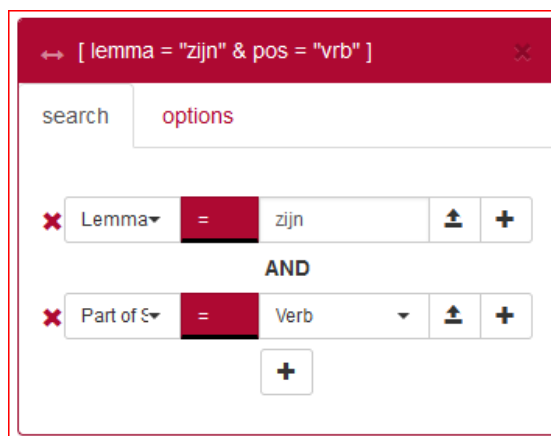
Specifying token attributes is similar to the Extended Search form. Select which attribute a token should have, and enter the value that the attribute must have for the token to be matched. Attributes in the query builder are interpreted as *regular expressions*. Note that this is different from the 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 choose Adverb for 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 using the +-signs **below** the attributes in the order Part of speech = Noun Common AND Lemma = *goed*, OR Lemma = *lief*, 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 "kadastraal bekend gemeente Lieve Vrouwe Parochie", where *Lieve* is an adjective, so this is not what we were after.

If, on the other hand, we add the attributes using the +-signs **right** of the attributes in the order Part of speech = Noun Common AND Lemma = *goed*, OR Lemma = *lief*, as is shown in the right screenshot below, we get the token query: [pos = "nou\c" & (lemma = "goed" | lemma = "lief")]. Now it appears that the lemma *lief* does not occur as a noun in this corpus.

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

search options

Part of  $\xi$  = Noun Common

AND

Lemma = goed

OR

Lemma = lief

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

search options

Part of  $\xi$  = Noun Common

AND

Lemma = goed

OR

Lemma = lief

Per Hit Per Document

Hits: Grouped by hit lemma

Total hits: 623 (0.02491%)  
Total groups: 3  
Search time: 0.18

Group by Lemma  Case-sensitive

hits

Group	hits in group	Relative frequency (hits)
goed	623	0.02491%
lief	1	0.000168%

View detailed concordances

Before	Hit	After
... het perceel, kadastraal bekend gemeente	Lieve	Vrouwe Paroche, sectie C, nr. ...
... het perceel, kadastraal bekend gemeente	Lieve	Vrouwe Paroche, sectie C, nr. ...
... de perceel, kadastraal bekend gemeente	Lieve	Vrouwe Paroche, sectie C, nr. ...
... het perceel, kadastraal bekend gemeente	Lieve	Vrouwe Paroche, sectie C, nr. ...
... het perceel, kadastraal bekend gemeente	Lieve	Vrouwe Paroche, sectie C, nr. ...
... de perceel, kadastraal bekend gemeente	Lieve	Vrouwe Paroche, sectie C, nr. ...
... het perceel, kadastraal bekend gemeente	Lieve	Vrouwe Paroche, sectie C, nr. ...
... het perceel, kadastraal bekend gemeente	Lieve	Vrouwe Paroche, sectie C, nr. ...
... het perceel, kadastraal bekend gemeente	Lieve	Vrouwe Paroche, sectie D, nr. ...
... het perceel, kadastraal bekend gemeente	Lieve	Vrouwe Paroche, sectie D, nr. ...
... het perceel, kadastraal bekend gemeente	Lieve	Vrouwe Paroche, sectie D, nr. ...
... het perceel, kadastraal bekend gemeente	Lieve	Vrouwe Paroche, sectie D, nr. ...

Per Hit Per Document

Hits: Grouped by hit word

Total hits: 623 (0.02491%)  
Total groups: 3  
Search time: 0.18

Group by Word  Case-sensitive

hits

Group	hits in group	Relative frequency (hits)
goed	623	0.02491%
goede	1	0.000562%
bede	1	0.0001682%

Sort by: Export Export to Excel

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 = "raad" ]

search options

Optional

Begin of sentence

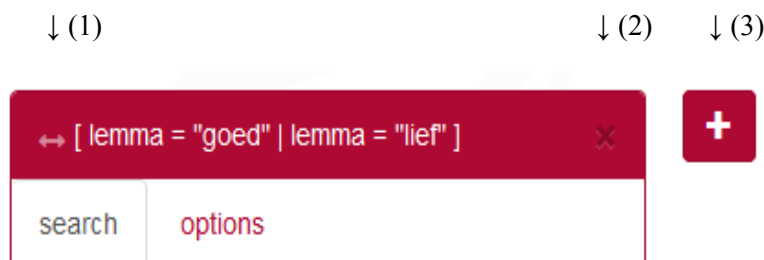
End of sentence

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).



## 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 to 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 press 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 `[ lemma = "zijn" & pos = "pd" ]` - 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.

Search for ...

Simple   Extended   Advanced   **Expert**

---

Corpus Query Language:

```
[ lemma = "zijn" & pos = "pd" ]
```

Copy to query builder   Import query   Gap-filling

## 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="adviesraad"]` or `[ word = "adviesraad" ]` (or just “adviesraad”) does not make any difference to the result. However, there is a difference between the queries `[word="adviesraad"]` and `[word="adviesraad"]`. The first search results in 67 hits, but the second one in zero!

Some examples:

- Simple: [\[word="adviesraad"\]](#), e.g. the attribute word matches the regular expression *adviesraad*; [\[word!="adviesraad"\]](#), e.g. the attribute word does **not** match the regular expression *adviesraad*; [\[word="\\*.man"\]](#) 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. [\[word="rechter"\]\[\]{}\[word="uitspraak"\]](#) matches a sequence of *rechter* followed by *uitspraak* with three arbitrary tokens in between.
- Operators `|`, `&` and parentheses `()` and the repetition operators `(+)`, `(*)`, `(?)` and `({})` can be used to build complex sequence queries. Example: [\["hare" | "zijne"\] "koninklijke" "hoogheid"](#), matching any sequence of *Zijne Koninklijke Hoogheid* or *Hare Koninklijke Hoogheid*.

This short list does not cover all CQL features. For more detailed information on how to write CQL, please consult the short [CQL manual in the appendix](#), which contains further pointers.



## Copy to query builder

When the query is relatively simple - like `[pos="AA"] [lemma="wet"]` - it can also be imported into the querybuilder using the *Copy to query builder* button. This will take you automatically to the Advanced Search screen, after which you can start the search or adjust the query if desired.

A message will be displayed next to the button if the query couldn't be parsed.

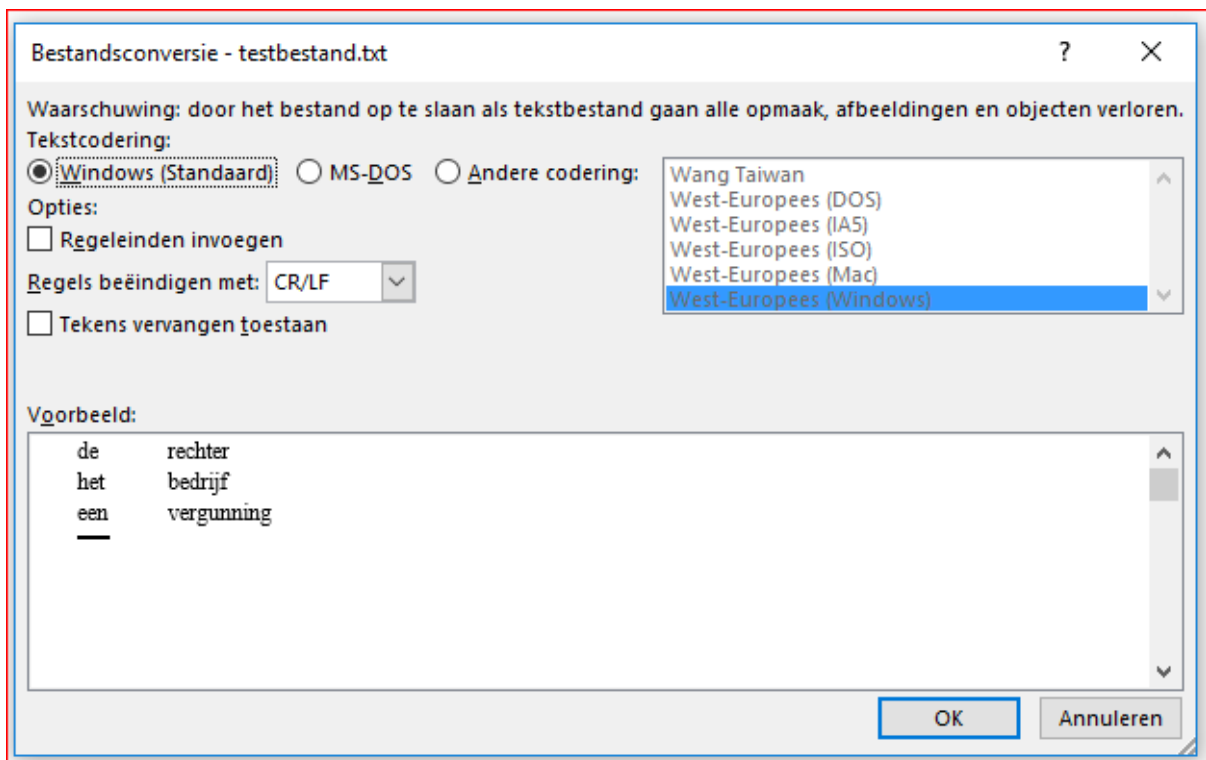
## Import query

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 screenshot shows a search interface with a red header. Below the header, there are tabs for 'Simple', 'Extended', 'Advanced', and 'Expert'. The 'Advanced' tab is selected. Underneath, there is a section for 'Corpus Query Language' with a text input field containing the query: `[word="@@"][word="@@"]`. Below the input field, there are buttons for 'Copy to query builder', 'Import query', and 'Gap-filling' (which is highlighted with a red 'x'). At the bottom, there is a list of values to be substituted: 'de', 'het', 'een' in the first column, and 'rechter', 'bedrijf', 'vergunning' in the second column.

The values in the first column - *de*, *het*, *een* - will be entered at the position of the first gap (@@) and the values in the second column - *rechter*, *bedrijf*, *vergunning* - at the position of the second gap. With these values, gap-filling yields the following results (titles are hidden):

The screenshot shows a search results page with a red header. Below the header, there are tabs for 'Per Hit' and 'Per Document'. The 'Per Hit' tab is selected. The page shows a table of hits with the following columns: 'Before hit', 'Hit', 'After hit', 'Lemma', 'Part of Speech', and 'Part of Speech + features'. The table contains several rows of results, with the first row highlighted. The 'Hit' column contains the words 'een zodanige vergunning' and 'het onderwerpelijke bedrijf' and 'het vrije bedrijf'. The 'Part of Speech' column contains 'PD AA' and 'NOU-C'. The 'Part of Speech + features' column contains various grammatical features like 'AA(degree=pos, position=prenom, formal=infl-e) NOU-C(gender=fjm, number=sg)'. There is also a 'Hits' section at the top right showing 'Total hits: 624 (0.00481%)' and 'Search time: 0.2s'.

Before hit	Hit	After hit	Lemma	Part of Speech	Part of Speech + features
...overeenkomstige toepassing. De beschikking, waarbij	<b>een zodanige vergunning</b>	wordt verleend, wordt afgekondigd in...	een zodanig vergunning	PD AA NOU-C	PD(type=indef, subtype=art-indef) AA(degree=pos, position=prenom, formal=infl-e) NOU-C(gender=fjm, number=sg)
...een vakvereniging van arbeiders in	<b>het onderwerpelijke bedrijf</b>	, die mededeling wenst te ontvangen...	het onderwerpeijk bedrijf	PD AA NOU-C	PD(type=d-p, subtype=art-def) AA(degree=pos, position=prenom, formal=infl-e) NOU-C(gender=n, number=sg)
...van het soortgelijke arbeid in	<b>het vrije bedrijf</b>	gebruikelijke loon. Art. 34. 1...	het vrij bedrijf	PD AA NOU-C	PD(type=d-p, subtype=art-def) AA(degree=pos, position=prenom, formal=infl-e) NOU-C(gender=n, number=sg)
...aan die welke geldt in	<b>het vrije bedrijf</b>	. Art. 38. De arbeid zal...	het vrij bedrijf	PD AA NOU-C	PD(type=d-p, subtype=art-def) AA(degree=pos, position=prenom, formal=infl-e) NOU-C(gender=n, number=sg)
...eerste lid, toepassing heeft gevonden,	<b>een tijdelijke vergunning</b>	voor de uitoefening van een...	een tijdelijk vergunning	PD AA NOU-C	PD(type=indef, subtype=art-indef) AA(degree=pos, position=prenom, formal=infl-e) NOU-C(gender=fjm, number=sg)
...de Commissie worden ingetrokken. 4.	<b>Een tijdelijke vergunning</b>	, verleend ter onverwijde voorziening in...	een tijdelijk vergunning	PD AA NOU-C	PD(type=indef, subtype=art-indef) AA(degree=pos, position=prenom, formal=infl-e) NOU-C(gender=fjm, number=sg)
...eerste lid, toepassing heeft gevonden,	<b>een tijdelijke vergunning</b>	voor de uitoefening van een...	een tijdelijk vergunning	PD AA NOU-C	PD(type=indef, subtype=art-indef) AA(degree=pos, position=prenom, formal=infl-e) NOU-C(gender=fjm, number=sg)

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 words in the column Hit - to display the properties and values of the hit (in the following example **de burgerlijke rechter**). Click the hit again to close.

Before hit -	Hit -	After hit -	Lemma	Part of Speech	Part of Speech + features
38mwc.Wetgevingsteksten.jur_1881					
...van staat is ingesteld en	<b>de burgerlijke rechter</b>	daarop een eindbeslissing heeft gegeven...	de burgerlijk rechter	PD AA NO U-C	PD(type=d-p,subtype=art-def) AA(degree=pos,position=prenom,formal=infl-e) NOU-C(gender=fjm,number=sg)
gevangenisstraf van ten hoogste vijf jaren of geldboete van de vierde categorie. 2. Ontzetting van de in artikel 28, eerste lid, onder 1*, 2 en 4*, vermelde rechten kan worden uitgesproken. 3. Vervolging heeft niet plaats dan nadat een rechtsvordering tot inroeping of tot betwisting van staat is ingesteld en <b>de burgerlijke rechter</b> daarop een eindbeslissing heeft gegeven. Indien de rechtsvordering echter door het stilzitten van de partijen onvoldoende voortgang vindt, kan vervolging ook plaatshebben nadat de burgerlijke rechter heeft beslist dat er een begin is van bewijs bij geschifte als bedoeld in artikel 209 van Boek 1 van het Burgerlijk Wetboek. Art.					
Property	Value				
Word	de		burgerlijke		rechter
Lemma	de		burgerlijk		rechter
Part of Speech	PD		AA		NOU-C
Part of Speech + features	PD(type=d-p,subtype=art-def)		AA(degree=pos,position=prenom,formal=infl-e)		NOU-C(gender=fjm,number=sg)

Hit rows are always preceded by a row containing the document title in which those hits occurred, in this case 38mwc.Wetgevingsteksten.jur\_1881. 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: 20140605163358569RS03043746155.)

### Sorting results

Click on any of the column headings (i.e. Before hit, Hit, After hit, Lemma, Part of Speech or Part of Speech+features) to sort the hits within that column, clicking again inverts the sorting.

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, Metadata.

### Grouping results

Results Per Hit can be grouped by properties of Hit, Before hit, After hit and by Metadata. 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.

Per Hit | Per Document

Hits / Grouped by hit:lemma

Total hits: 8 (0.0000616%)  
Total groups: 8  
Search time: 0.1s

Group by Lemma  Case-sensitive

« 1 »

table hits

Group	#hits in group	Relative frequency (hits)
rechter bevoegd bij de uitspraak	1	0.0000077%
rechter doen bij vonnis uitspraak	1	0.0000077%
rechter kunnen in zijn uitspraak	1	0.0000077%
rechter openbaarmaking van zijn uitspraak	1	0.0000077%
rechter de openbaarmaking zijner uitspraak	1	0.0000077%
rechter bepalen dat zijn uitspraak	1	0.0000077%
rechter te dier zaak uitspraak	1	0.0000077%
rechter mede wanneer de uitspraak	1	0.0000077%

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="d.?e"] [lemma="schoon"] []`. This produces the following hits (Titles are hidden):

Before hit	Hit	After hit
...beeindiging van het ballasten van	<b>de schone ballasttanks</b>	. Datum van aantekening. Verantwoordelijke officier...
...beeindiging van het uitpompen van	<b>de schone ballast</b>	. Datum van aantekening. Verantwoordelijke officier...
...een of meer ladingtanks bevat,	<b>die schoon en</b>	droog zijn en voor het...
...Art. 5. 1. Alvorens werknemers	<b>de schoon te</b>	maken ruimten betreden, dient te...
...van de afzonderlijke congenere	<b>de schone olie</b>	bedragen circa 1 ppm. BIJLAGEN...
...van onderhoud wordt gehouden en	<b>die schoon dient</b>	te zijn; g. de temperatuur...
...te bevinden. De toegang tot	<b>de schone ruimte</b>	mag alleen mogelijk zijn via...
...een sluis in of uit	<b>de schone ruimten</b>	worden gebracht. Slechts indien de...
...of papier waarvan de vezels	<b>de schone ruimte</b>	kunnen verontreinigen mag niet in...

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

Per Hit | Per Document

Hits / Grouped by context:word:i:H2-3

Total hits: 9 (0.0000693%)  
Total groups: 8  
Search time: 0.1s

Context (advanced) Apply

Word Before Hit After  Case-sensitive

From end of hit

New context group

« 1 »

table hits

Group	#hits in group	Relative frequency (hits)
schone ruimte	2	0.0000154%
schone ballasttanks	1	0.0000077%
schone olie	1	0.0000077%
schoon en	1	0.0000077%
schone ruimten	1	0.0000077%
schone ballast	1	0.0000077%
schoon dient	1	0.0000077%
schoon te	1	0.0000077%

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

The screenshot shows a search results interface. At the top, there is a table with columns: Group, #hits in group, and Relative frequency (hits). The first row shows 'schone ruimte' with 2 hits and a relative frequency of 0.0000154%. Below this, there is a button 'View detailed concordances'. The detailed view shows a concordance with columns 'Before', 'Hit', and 'After'. The 'Hit' column contains 'de schone ruimte'.

Group	#hits in group	Relative frequency (hits)
schone ruimte	2	0.0000154%
schone ballasttanks	1	0.0000077%
schone olie	1	0.0000077%
schoon en	1	0.0000077%
schone ruimten	1	0.0000077%

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 Metadata.

The screenshot shows a search results interface with a sorting dropdown menu open. The dropdown menu has options: Documents, Sort by hits, Sort by hits (ascending), Metadata, Sort by Year Metadata (highlighted), and Sort by Year (descending) Metadata. Below the dropdown, there is a table with columns Year and Hits. The table shows results for the years 1973, 1974, 1977, 1979, and 1986.

Year	Hits
1973	2
1974	2
1977	1
1979	1
1986	3

### Grouping results

Results Per Document can be grouped by Year, 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.

Results for: "[word="minister"][word="heeft]" within all documents

Per Hit **Per Document**

Hits / Grouped by wordleft:word Total hits: 106 (0.000817%)  
Total groups: 6  
Search time: 0.1s

Group by Word before  Case-sensitive

« 1 »

table hits

Group	#hits in group	Relative frequency (hits)
Onze	69	0.000532%
de	33	0.000254%

« View detailed concordances Load more concordances »

Before	Hit	After
... aanvraagt, ten genoee van de	<b>Minister heeft</b>	aangetoond dat hij a. voldoende ...
... vrijstelling wegens kostwinnerschap en de	<b>minister heeft</b>	bepaald, dat vergoeding zal worden ...
... die geen budgetgemeente zijn, de	<b>minister heeft</b>	om van streefsubsidies, streefaantallen of ...
... binnen twee maanden nadat de	<b>minister heeft</b>	beschikt. Art. 23 1. Aan ...
... binnen twee maanden nadat de	<b>minister heeft</b>	beschikt. Hoofdstuk 3. Voorzieningen aan ...
... 000 zullen bedragen, nadat de	<b>minister heeft</b>	beschikt op een verzoek als ...

## Information about a document

Click on a document title to open the document in a new 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 Pages and the Hits button:

Pages « 1 »

Hits « < 7 > »

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

op een  
als bed  
vastste  
indien l  
- ook n  
tot vaststelling **van het subsidiebedrag** - de minister blijkt dat  
de overheidsinstelling niet heeft voldaan aan het bepaalde in artikel

Word: subsidiebedrag  
Word id: w.8923  
Lemma: subsidiebedrag  
Part of speech: NOU-C(gender=n,number=sg)

ng zou hebben genomen,  
zodra

## Metadata of a document

In the Metadata tab all metadata properties of the document are displayed.

## Statistics

The Statistics tab shows several document statistics: the number of Tokens, the number of Types (unique word forms), the number of 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 or via the menu symbol right of 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 obtain information about the average document length for a certain year.

- In the Group documents by metadata drop-down menu, choose Group by Year
- In Show groups as, select *table*
- Press Search
- Click on Group to order the years chronologically

You will get this result:

Group	#docs in group	#tokens in group	Relative frequency (docs)	Relative frequency (tokens)	Average document length
1814	1	8.174	0.667%	0.063%	8.174
1815	1	2.022	0.667%	0.0156%	2.022
1818	1	815	0.667%	0.00628%	815
1820	1	300	0.667%	0.00231%	300
1822	1	707	0.667%	0.00545%	707
1823	1	482	0.667%	0.00371%	482
1824	1	351	0.667%	0.0027%	351
1827	1	11.727	0.667%	0.0903%	11.727

## 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*: Word (i.e. word form); in *Corpus Juridisch Nederlands*, N always stands for a Word;
- 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 *Corpus Juridisch Nederlands* for specific Witness Years.

### Example

Search Explore

Explore ...

Documents N-grams Statistics

N-gram size: 5

N-gram type: Word

Word Word Word Word Word

het Word kind Word Word

Within all the documents of the *Corpus Juridisch Nederlands*, you will find 47 occurrences of this so-called 5-gram.

Per Hit Per Document

Hits / Grouped by hit:word

Total hits: 47 (0.000362%)  
Total groups: 35  
Search time: 0.2s

Group by Word  Case-sensitive

« 1 2 »

table hits

Group	#hits in group	Relative frequency (hits)
het eigen kind het aangehuwde	4	0.0000308%
het aangehuwde kind en het	4	0.0000308%
het oudste kind bij de	4	0.0000308%
het wettig kind van de	2	0.0000154%
het eerste kind af alsmede	2	0.0000154%
het eerste kind alsmede oprichting	2	0.0000154%
het oudere kind wordt gericht	1	0.0000077%
het tweede kind die de	1	0.0000077%
het tweede kind Art. VI	1	0.0000077%



## Statistics (frequency lists)

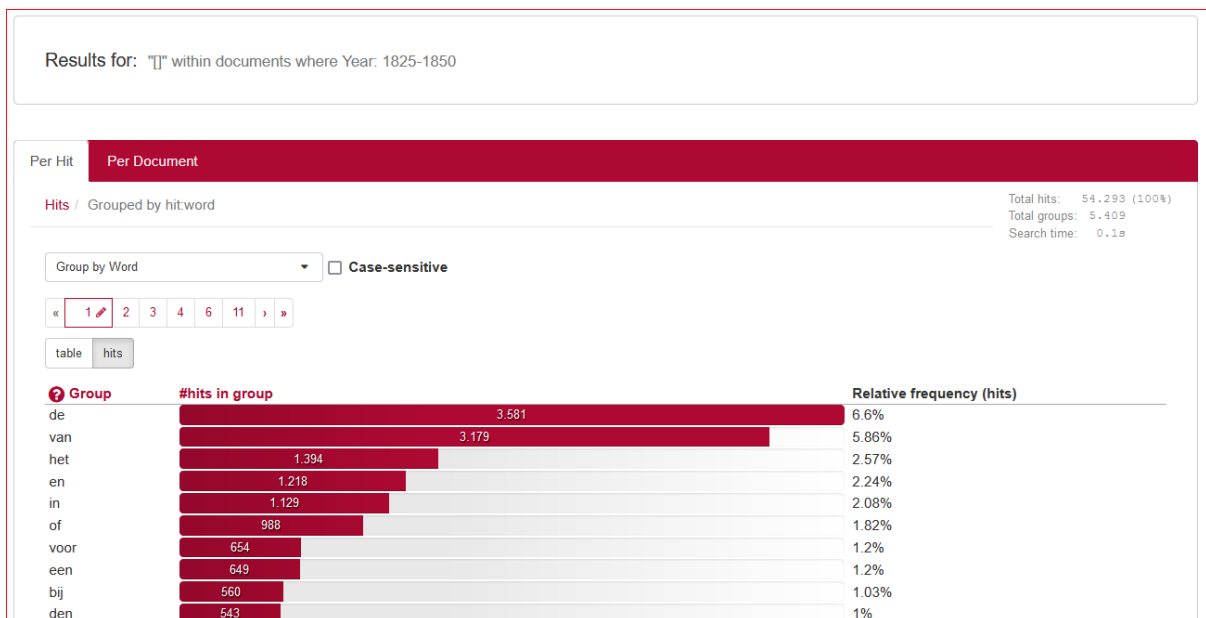
Here, you can produce frequency lists for the corpus. It is rather similar to the previous option, but restricted to 1-grams.

### Options

- *Frequency list type*: choose for lists of Word (i.e. Word form), Lemma, Part of speech and Part of Speech + features
- By using the Filter search by... you can create a subcorpus within the *Corpus Juridisch Nederlands* for specific metadata.

### Example

It is possible to determine the use of the most frequently used words from 1825 to 1850 by searching for Frequency list type Word and by filtering search by Year 1825-1850, which 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/accnt-sensitive matching. Note that, unlike in CWB, case-INsensitive matching is currently the default. To explicitly match case/accnt-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.