Corpus Juridisch Nederlands application manual

Table of contents

Introduction

This manual describes the corpus exploitation environment for the *Corpus Juridisch Nederlands*. 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 (https://blacklab.ivdnt.org/). The web-based frontend is a further development of the corpus-frontend application developed by INT (https://github.com/instituutnederlandsetaal/blacklab-frontend). Its design is inspired by the first version of the OpenSoNaR user interface by Tilburg University 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*. Other metadata categories, like the fact that the Language Variant is NN (Netherlandic Dutch) for this collection, have not been made searchable. This metadata category is described below. In the corpus application it is possible to limit a search by filtering on this metadata category.

Year

The year the law texts were written.

Application user manual

The language of the corpus application is set to Dutch by default. Press the globe icon in the top right corner to select English.

Getting started

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

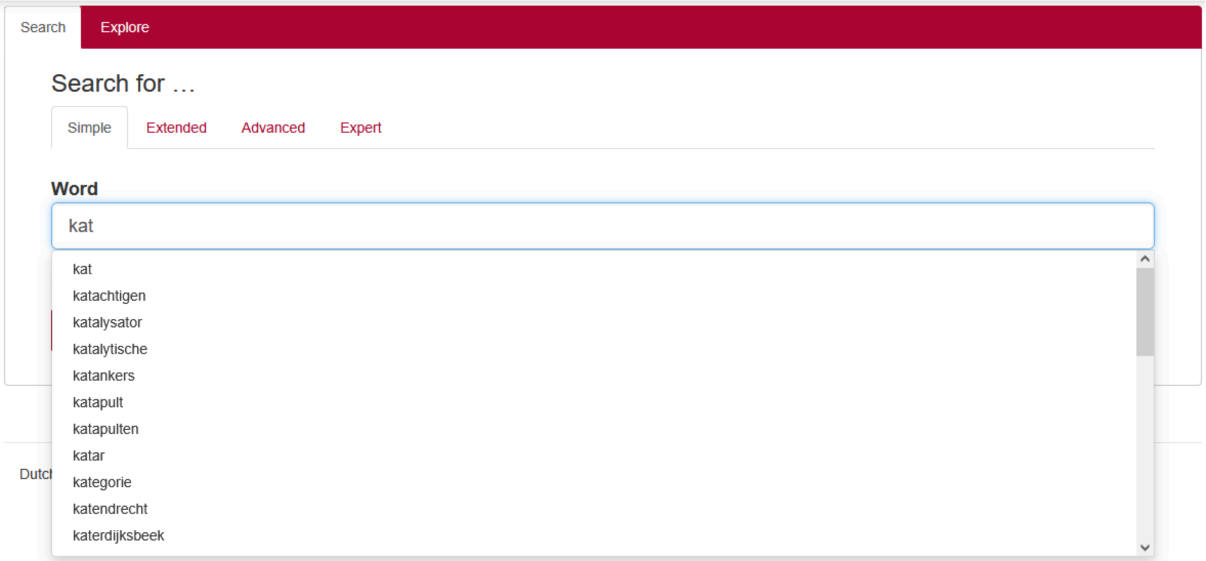
- To search for a word literally in the form you specify, use Simple Search or Extended Search.
    - Simple Search for Word *woningbouw*
    - Extended Search for Word *wetgever*
- To search for words or lemmata satisfying a certain pattern, use *wildcards* in Simple Search or Extended Search, or *regular expressions* in Advanced Search or Expert Search.
    - words starting with *ver* and ending with *len* in Simple Search
    - words starting with *ver* and ending with *len* in Extended Search
    - lemmata starting with *ver* and ending in *eren* with (mostly) one syllable in between in Advanced Search
    - lemmata starting with *ver* and ending in *eren* with (mostly) one syllable in between in Expert Search
- To see which unique forms occur as a result of your search, use Group Results.
    - example Group by Annotation: different 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 of the year

Searching the corpus

Simple search

Search

The Simple Search allows you to quickly search for specific word forms (e.g. *kat*). After entering a search term, a running list appears, which contains suggestions for possible search terms in alphabetical order, based on the characters typed in.



Word: kat

kat
katachtigen
katalysator
katalytische
katankers
katapult
katapulten
katar
kategorie
katendrecht
katerdijksbeek

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. Furthermore, you can 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|bedrij*|vergunning*.

Note that in Simple Search the patterns will be matched case-insensitively**:** *paragraaf* for instance will deliver the same results as *Paragraaf* or *PARAGRAAF*. See the paragraph Grouping results in Per Hit view to see how it is nevertheless possible to distinguish between uppercase and lowercase letters, or go to Extended Search.

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, searching for *a*n* matches all word forms 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* and *Assen*.

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.]

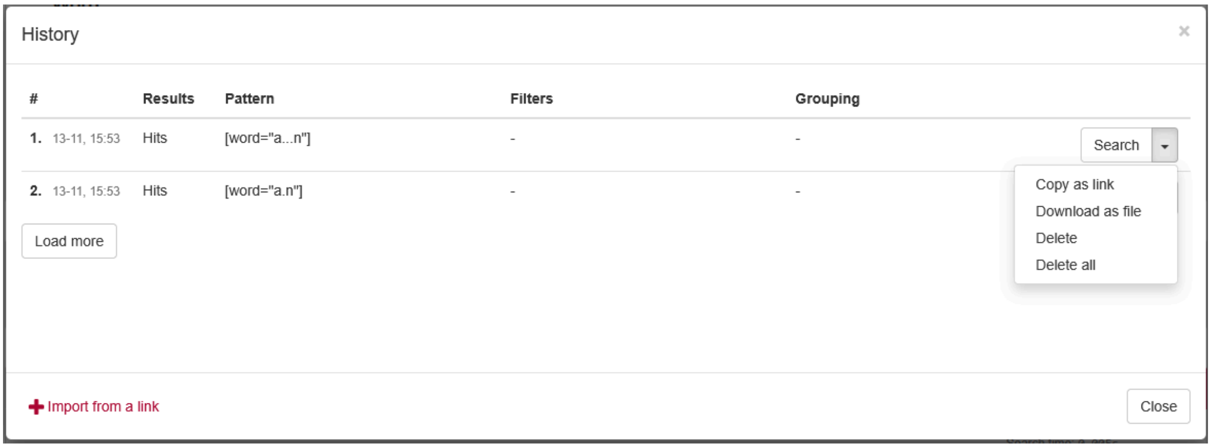
Reset

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.

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).



History

#	Results	Pattern	Filters	Grouping
1. 13-11, 15:53	Hits	[word="a...n"]	-	-
2. 13-11, 15:53	Hits	[word="a.n"]	-	-

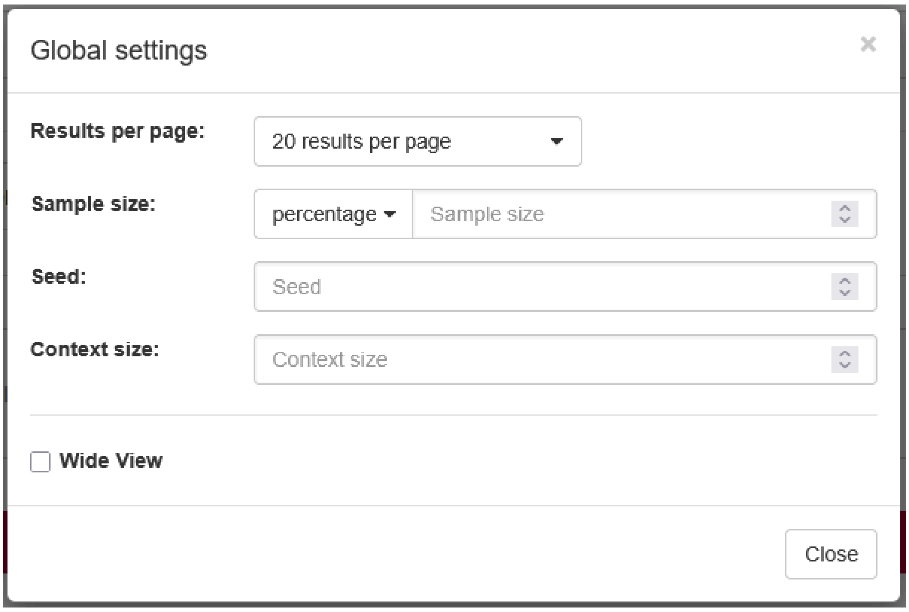
Every search query has its own url. If you copy this url via History (Copy as link) or directly from the address bar of your browser, you can send it to someone else who can import this link via Import from a link. It offers that person the possibility to run the search on his or her own computer.

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. 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.



Global settings ×

Results per page: 20 results per page

Sample size: percentage | Sample size

Seed: Seed

Context size: Context size

□ **Wide View**

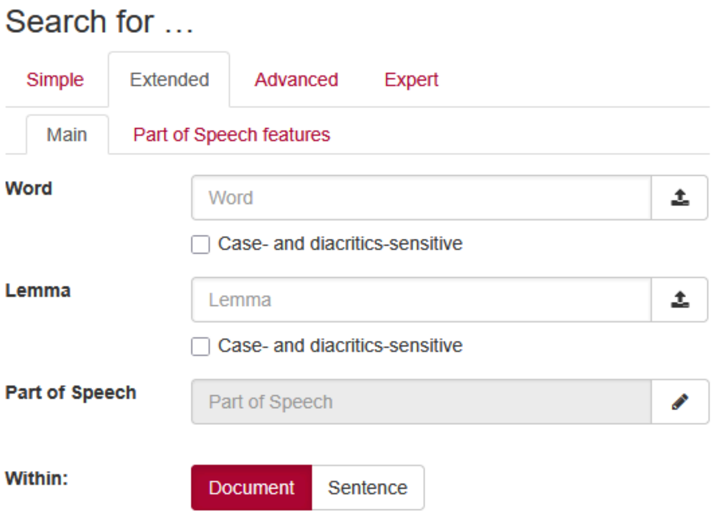
Close

Extended search

Like in Simple search, Extended Search allows you to quickly search for specific word forms. The search is performed in the same way as described for Simple Search. There are two tabs here, basics and Part of Speech features.

Main

In this corpus the three main attributes you can search for are Word (more precise: word form), Lemma and Part of Speech. In the search field Word and Lemma enter the word(s) or lemma(ta) (or Upload a list of values) you are looking for. Like in Simple Search, you can also enter a phrase for Word and/or Lemma. In the search field Part of Speech you can select the desired values. All supported attributes are shown in the search form:



Search for …

Simple | Extended | Advanced | Expert

Main | Part of Speech features

Word Word

□ Case- and diacritics-sensitive

Lemma Lemma

□ Case- and diacritics-sensitive

Part of Speech Part of Speech

Within: Document | Sentence

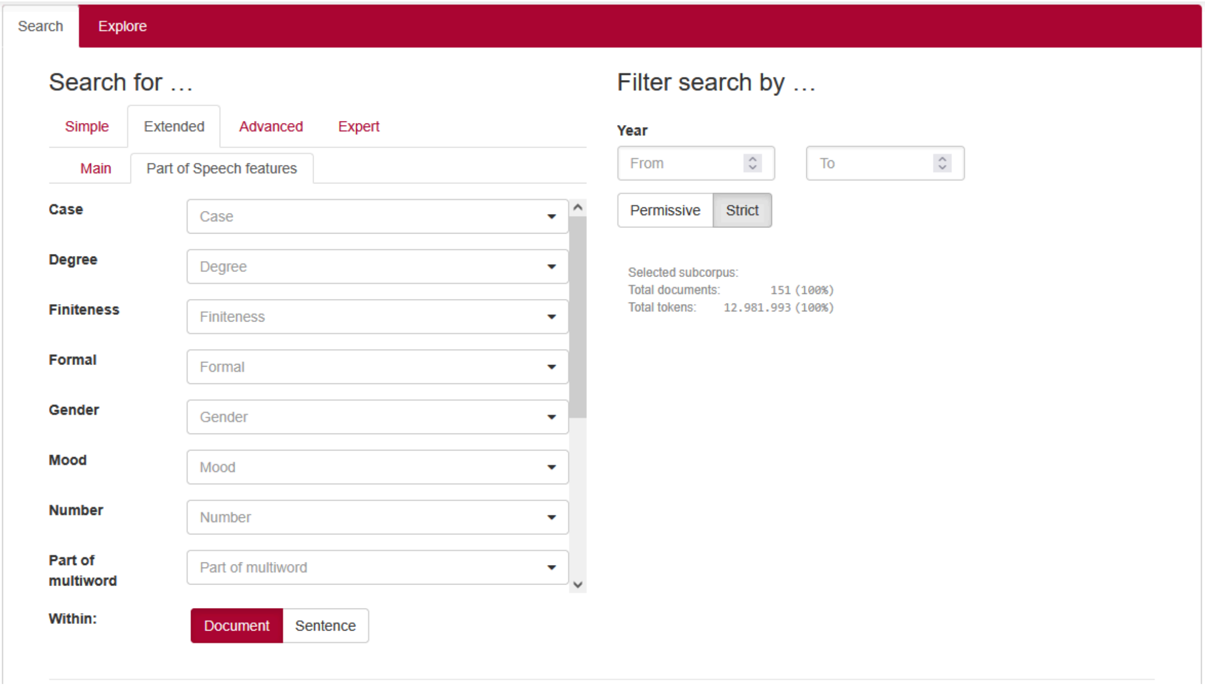
In Extended Search it is also possible to search case- and diacritics-sensitive. Note that the default setting for search is case- and diacritics-insensitive. For example, searching for the Word *minister* will result in 51.623 occurrences. By ticking the box Case- and diacritics-sensitive you will find 7678 occurrences of the Word *minister*, but none of *Minister, MInister* or *MINISTER*. In order to directly find only occurrences of the Word (form) *Minister* (41.928x), use the search term *Minister* and tick the box Case- and diacritics-sensitive under the search field Word.

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

You can combine multiple fields (such as Word and Part of Speech) to form a single query. For example, searching for the Word(form) *was* together with the Part of Speech VRB will result in a list of all verbs containing the word form *was*. The syntax of your query is shown in the results: [word="was"&pos="ww"].

Part of Speech features

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



Selected subcorpus:
Total documents: 151 (100%)
Total tokens: 12.981.993 (100%)

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* and *verklaard*.

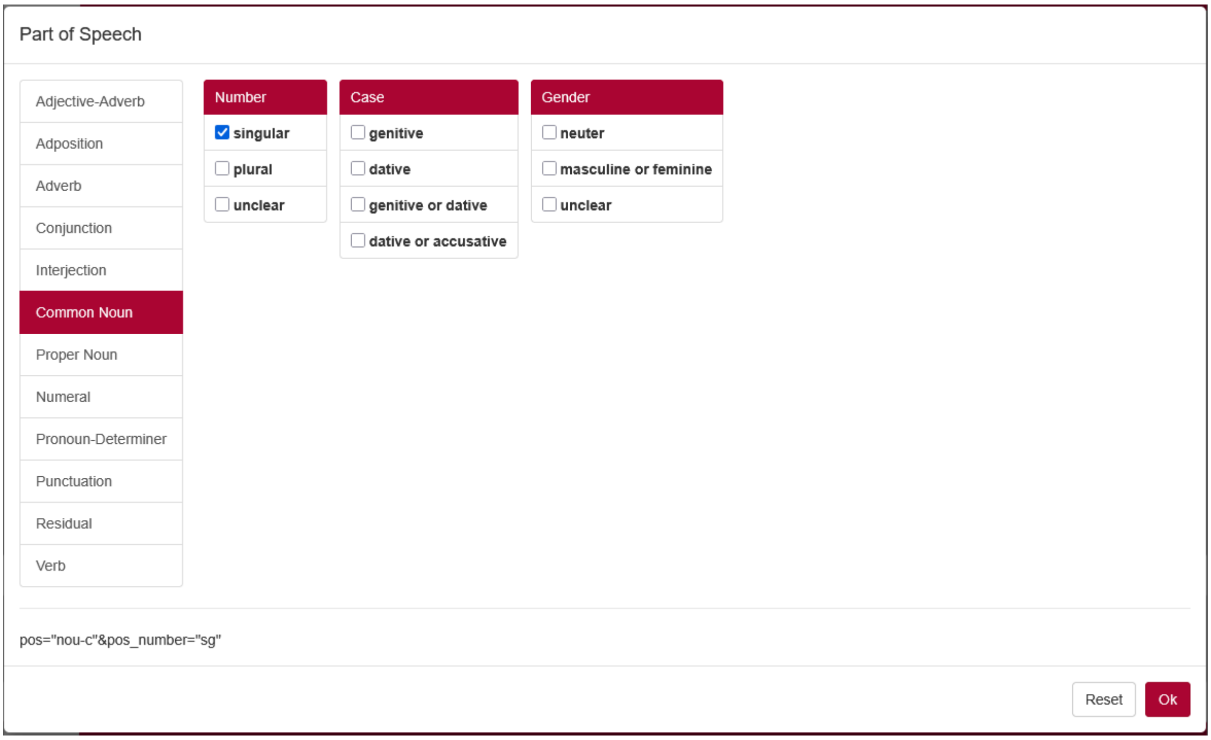
Upload a list of values

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 dialogue box

Clicking on the pencil next to the search field Part of Speech provides you with the Part of Speech dialogue box.



Part of Speech

- Adjective-Adverb
- Adposition
- Adverb
- Conjunction
- Interjection
- Common Noun
- Proper Noun
- Numeral
- Pronoun-Determiner
- Punctuation
- Residual
- Verb

Number	Case	Gender
☑ singular	□ genitive	□ neuter
□ plural	□ dative	□ masculine or feminine
□ unclear	□ genitive or dative	□ unclear
	□ dative or accusative	

```
pos="nou-c"&pos_number="sg"
```

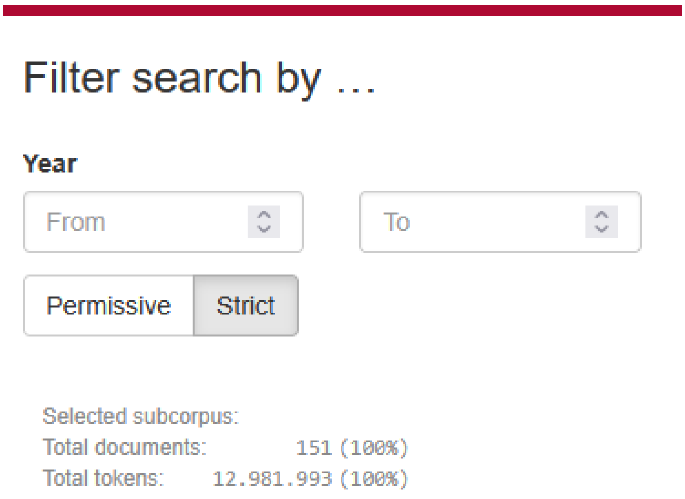
For some of the categories on the left you can tick certain features to further specify your search query. By doing so you can for instance delimit your search, as shown in the above screenshot for Noun Common.

Within

It is possible to apply your search query to all documents of the corpus, which is the default setting. However, it may be desirable to limit a search to a document or a sentence. The main reason to do this is to exclude multi-word matches that stretch over a sentence boundary. To do so you can click on these respective buttons.

Filter search by

At the right side you will find the option to limit your query to a subset of documents within a certain period, using the filter Year. You can find documents from a specific year by entering the same year in the 'from' row as in the 'to' row (see screenshot below). If you do not enter a specific year, the entire corpus is searched. If you want to filter by another year or another period, please press the reset button.



Filter search by …

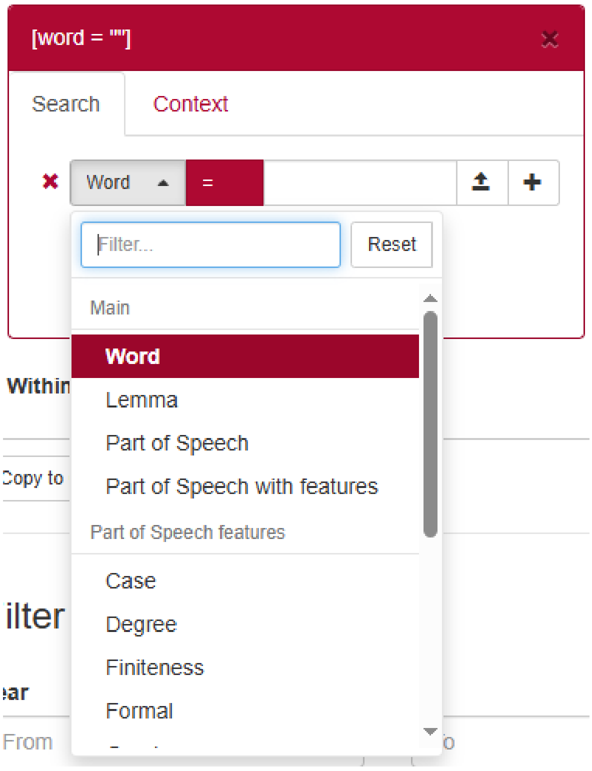
Selected subcorpus:
Total documents: 151 (100%)
Total tokens: 12.981.993 (100%)

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.



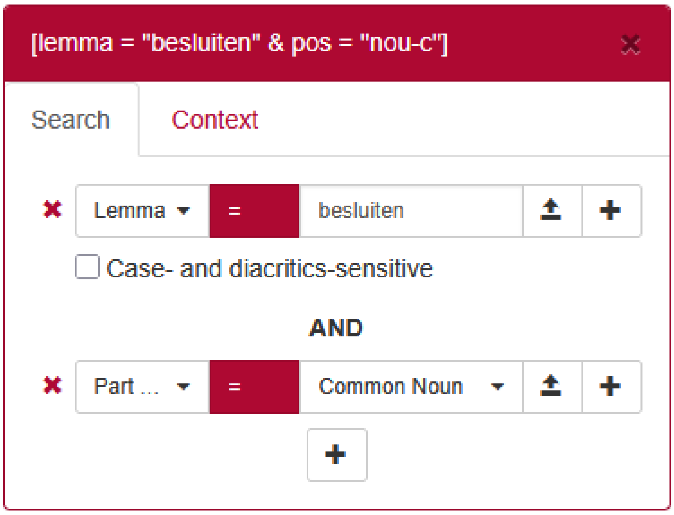
[word = ""]

A token box in the querybuilder has two tabs: Search and Context.

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* & Part of Speech=Common Noun 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:



```
[lemma = "besluiten" & pos = "nou-c"]
```

Lemma = besluiten

□ Case- and diacritics-sensitive

AND

Part ... = Common Noun

Token attributes

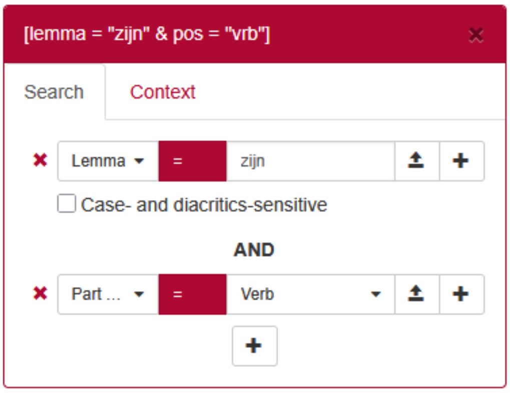
Specifying token attributes is similar to the Extended Search form. Select which attribute a token should have, and enter the value that the attribute must have for the token to be matched. Attributes in the query builder are interpreted as *regular expressions*. Note that this is different from the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



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

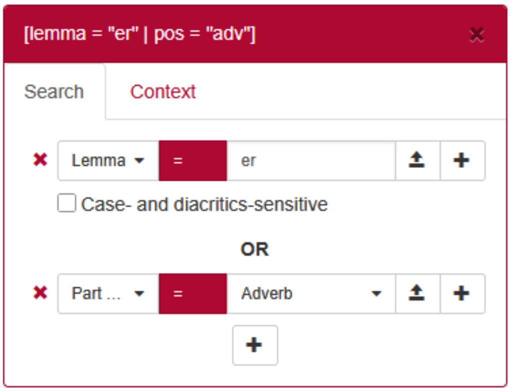
Lemma = zijn

□ Case- and diacritics-sensitive

AND

Part ... = Verb

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*.

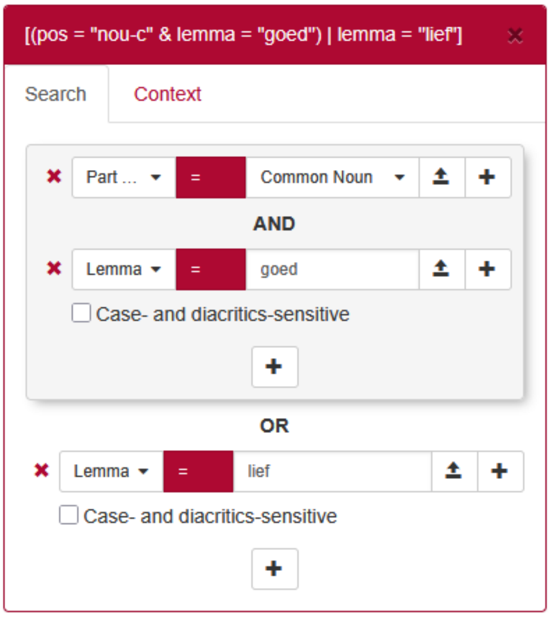

[lemma = "er" | pos = "adv"]

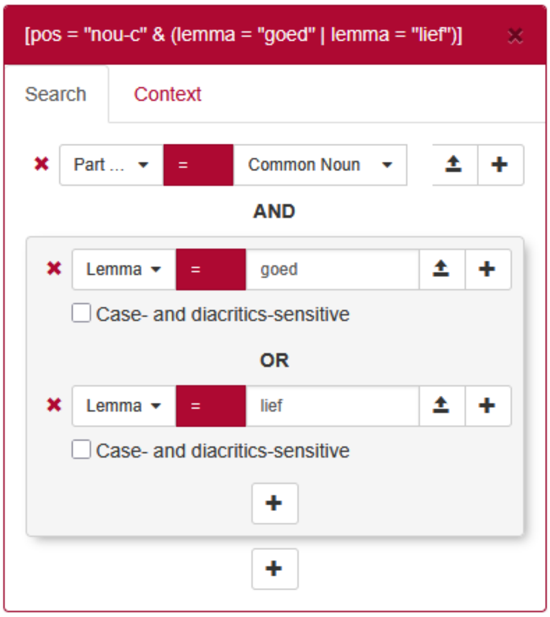
Function of the two +-buttons in a token box

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 *kadastraal bekend gemeente Lieve Vrouwe Parochie*, where *Lieve* is an adjective, so this is not what we were after.

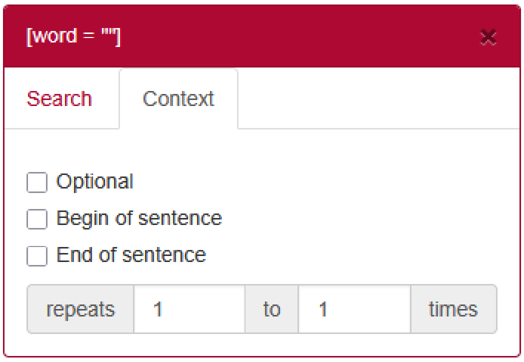
If, on the other hand, we add OR lemma=*lief* with the +-sign to the **right** of the attribute Lemma=*goed*, it will be inserted in a subclause (Lemma=*goed* OR Lemma=*lief*), thus resulting in the correct query [pos = "nou-c" & (lemma = "goed" | lemma = "lief")], as shown in the right screenshot below. Now it appears that the lemma *lief* does not occur as a noun in this corpus.


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


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

The tab Context

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

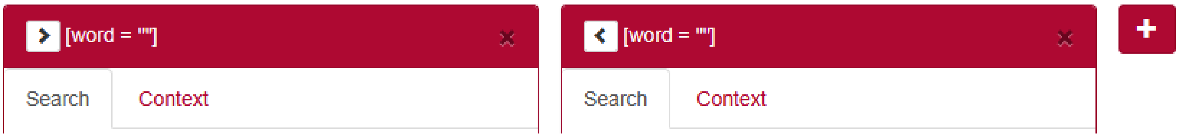


Managing sequences of token boxes

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

- *Rearrange* a token by clicking on the arrow in the top-left corner of a box (1). This arrow only appears if there are multiple token boxes.
- *Delete* a token by clicking the x in the top-right corner of a box (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) ↓ (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.

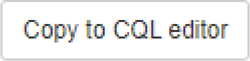
To remove an uploaded file and go back to typing a value, click on the cross (x) next to the yellow text box. Another possibility to clear the uploaded values is by clicking the yellow marked text field and then pressing the Clear button on the bottom left corner of the Edit box. Using the Reset button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

Within

Like in Expert Search, it is possible to apply your search query to all documents of the corpus, which is the default setting, or to sentences.

Copy to CQL editor

You can use the query builder to create complex queries without writing CQL. Any time a query is created in the querybuilder, it can be copied to the CQL editor, where you can further edit the query by hand. 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 copy your query into the query builder (in Advanced Search), 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.)
- Simple sequence: [pos="pd"][lemma="besluiten"] matches all occurrences of the lemma *besluiten* preceded by a pronoun.
- Combination of attributes (combining operators are &, |, !), e.g. [word="hoop|geloof|liefde"] matches either the word *geloof*, the word *hoop* or the word *liefde*.
- Repetition operators: [pos="aa"]{3} matches a sequence of 3 adjectives, [pos="aa"]{2,4} matches a sequence of 2 to 4 adjectives, [pos="aa"]{3,} matches a sequence of 3 or more adjectives.
- The empty [] matches any token, e.g. [word="rechter"][]{3}[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*. Note that, while most queries up to this point could also have been constructed with the query builder, we really need the power of CQL from here on.

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.

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

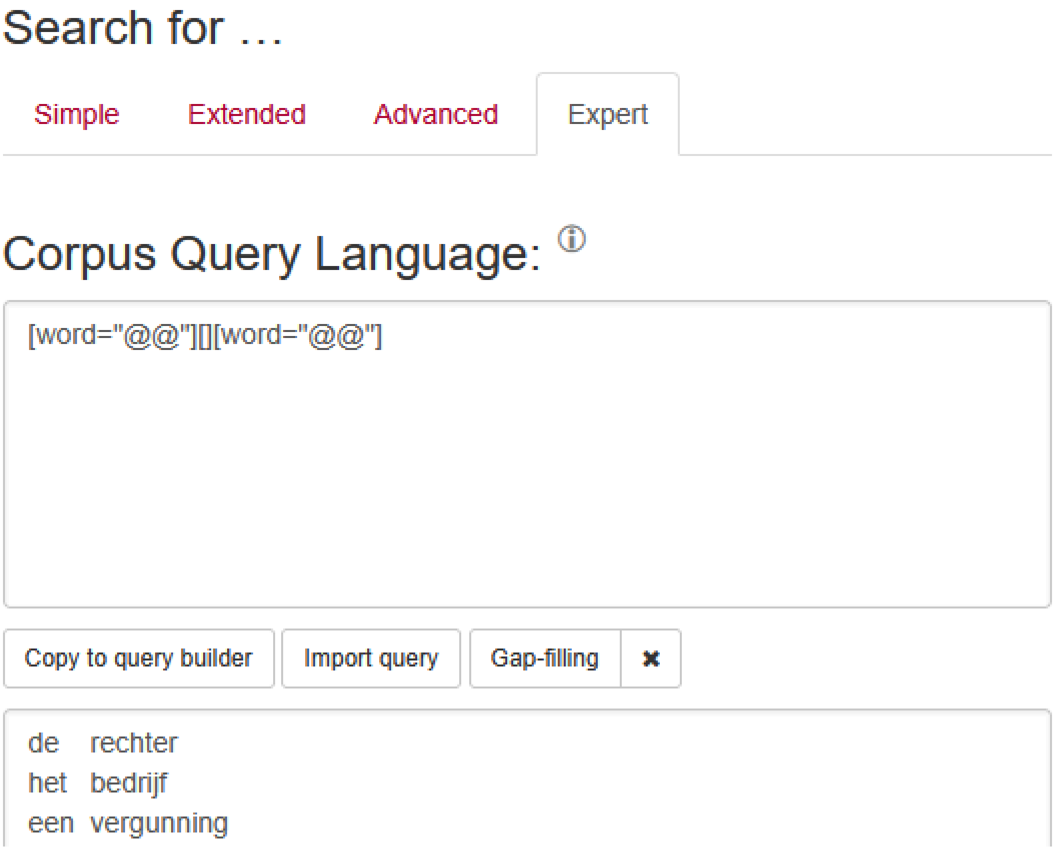
Use this button to upload a Tab Separated Values (TSV) file, which is a simple text format for storing data in a tabular structure. Each record in the table is one line of the text file. Each field value of a record is separated from the next by a tab character. It is also possible to upload a plain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

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:



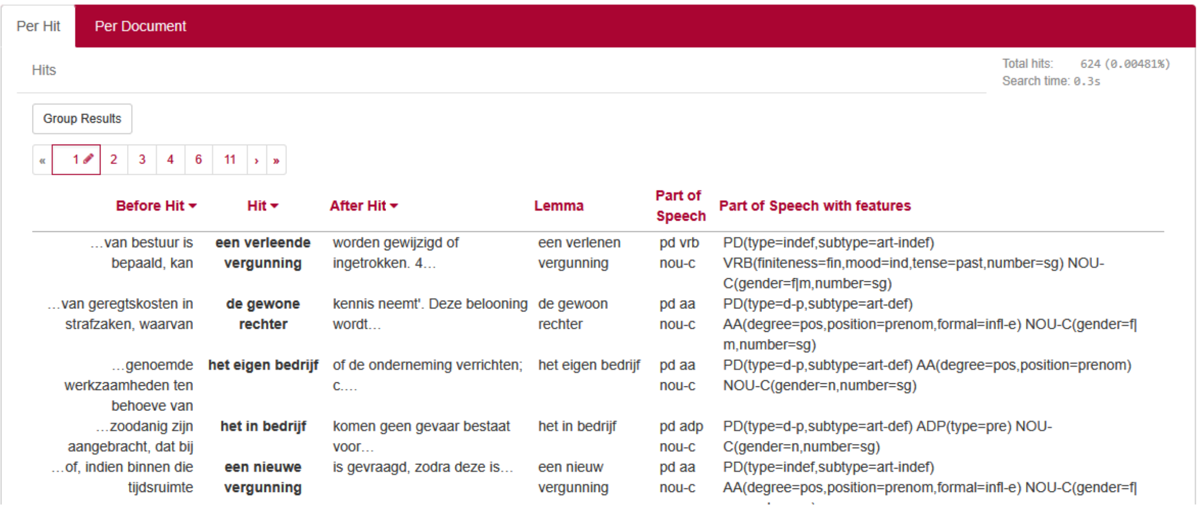
Search for …

Corpus Query Language:

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

de	rechter
het	bedrijf
een	vergunning

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):



Total hits: 624 (0.00481%)
Search time: 0.3s

Before Hit	Hit	After Hit	Lemma	Part of Speech	Part of Speech with features
…van bestuur is bepaald, kan	**een verleende vergunning**	worden gewijzigd of ingetrokken. 4…	een verlenen vergunning	pd vrb nou-c	PD(type=indef,subtype=art-indef) VRB(finiteness=fin,mood=ind,tense=past,number=sg) NOU-C(gender=f|m,number=sg)
…van geregtskosten in strafzaken, waarvan	**de gewone rechter**	kennis neemt'. Deze belooning wordt…	de gewoon rechter	pd aa nou-c	PD(type=d-p,subtype=art-def) AA(degree=pos,position=prenom,formal=infl-e) NOU-C(gender=f|m,number=sg)
…genoemde werkzaamheden ten behoeve van	**het eigen bedrijf**	of de onderneming verrichten; c.…	het eigen bedrijf	pd aa nou-c	PD(type=d-p,subtype=art-def) AA(degree=pos,position=prenom) NOU-C(gender=n,number=sg)
…zoodanig zijn aangebracht, dat bij	**het in bedrijf**	komen geen gevaar bestaat voor…	het in bedrijf	pd adp nou-c	PD(type=d-p,subtype=art-def) ADP(type=pre) NOU-C(gender=n,number=sg)
…of, indien binnen die tijdsruimte	**een nieuwe vergunning**	is gevraagd, zodra deze is…	een nieuw vergunning	pd aa nou-c	PD(type=indef,subtype=art-indef) AA(degree=pos,position=prenom,formal=infl-e) NOU-C(gender=f|

This mimics the functionality to upload a list of values in the Extended Search and Advanced Search interfaces.

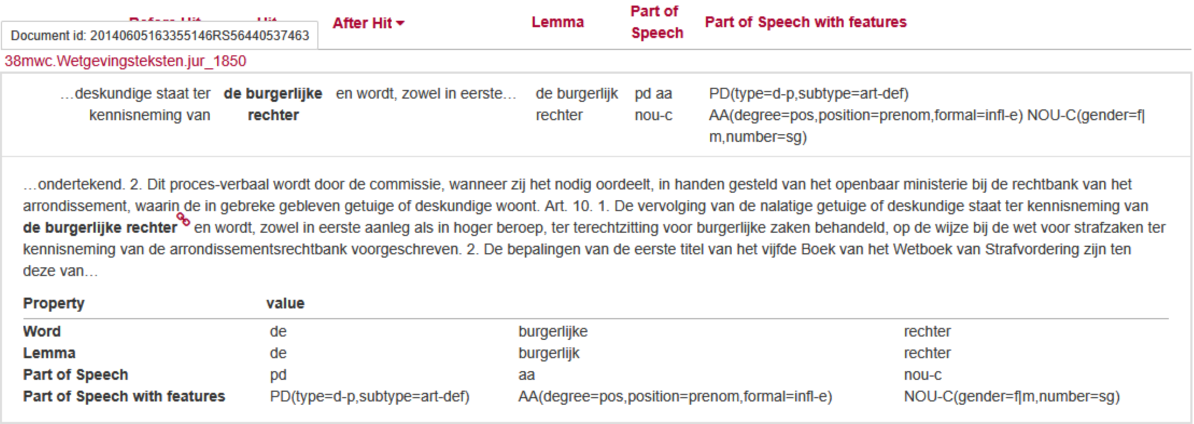
Please note that for this to work, you do need to enter @@ in the field where you want the substitution to take place. An empty field ([]) will match any term.

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.



Document id: 20140605163355146RS56440537463

		After Hit ▾	Lemma	Part of Speech	Part of Speech with features
38mwc.Wetgevingsteksten.jur_1850					
…deskundige staat ter kennisneming van	**de burgerlijke rechter**	en wordt, zowel in eerste…	de burgerlijk rechter	pd aa nou-c	PD(type=d-p,subtype=art-def) AA(degree=pos,position=prenom,formal=infl-e) NOU-C(gender=f|m,number=sg)

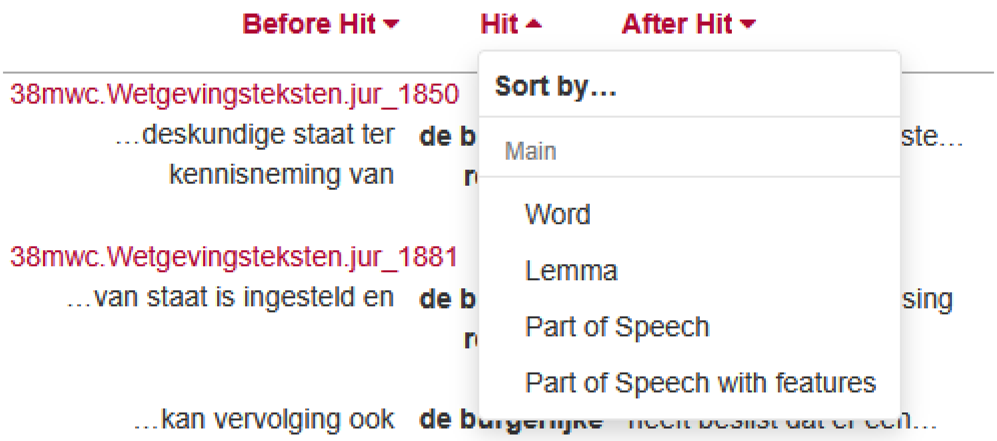
…ondertekend. 2. Dit proces-verbaal wordt door de commissie, wanneer zij het nodig oordeelt, in handen gesteld van het openbaar ministerie bij de rechtbank van het arrondissement, waarin de in gebreke gebleven getuige of deskundige woont. Art. 10. 1. De vervolging van de nalatige getuige of deskundige staat ter kennisneming van **de burgerlijke rechter** en wordt, zowel in eerste aanleg als in hoger beroep, ter terechtzitting voor burgerlijke zaken behandeld, op de wijze bij de wet voor strafzaken ter kennisneming van de arrondissementsrechtbank voorgeschreven. 2. De bepalingen van de eerste titel van het vijfde Boek van het Wetboek van Strafvordering zijn ten deze van…

Property	value		
Word	de	burgerlijke	rechter
Lemma	de	burgerlijk	rechter
Part of Speech	pd	aa	nou-c
Part of Speech with features	PD(type=d-p,subtype=art-def)	AA(degree=pos,position=prenom,formal=infl-e)	NOU-C(gender=f|m,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: 20140605163355146RS56440537463.)

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 Word, Lemma, Part of Speech and Part of Speech with features.

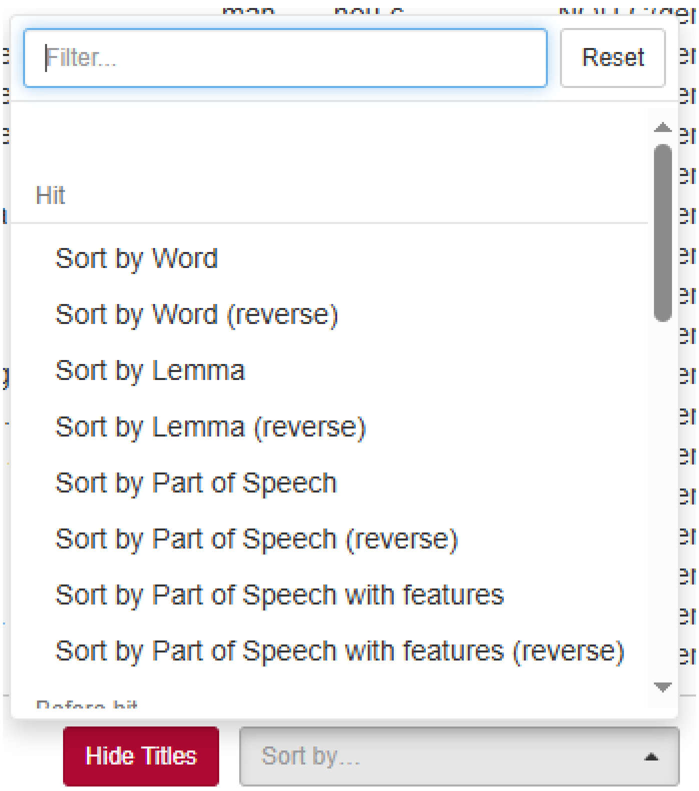


38mwc.Wetgevingsteksten.jur_1850
…deskundige staat ter kennisneming van **de b** … ste…

38mwc.Wetgevingsteksten.jur_1881
…van staat is ingesteld en **de b** … sing

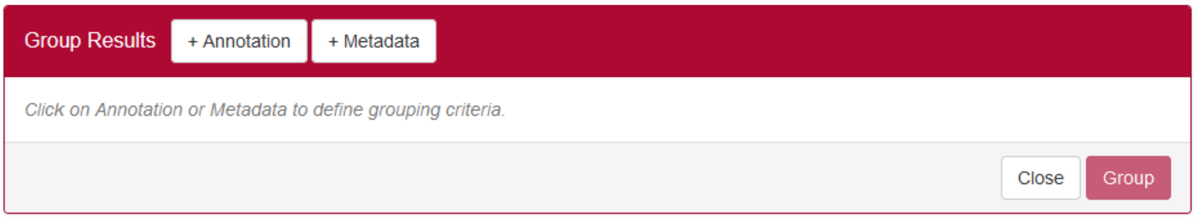
…kan vervolging ook **de burgerlijke** heeft beslist dat er een…

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 for Hit, Before hit, After hit and Year.



Grouping results

It is possible to group the results by clicking on the button Group Results, after which the following menu appears:



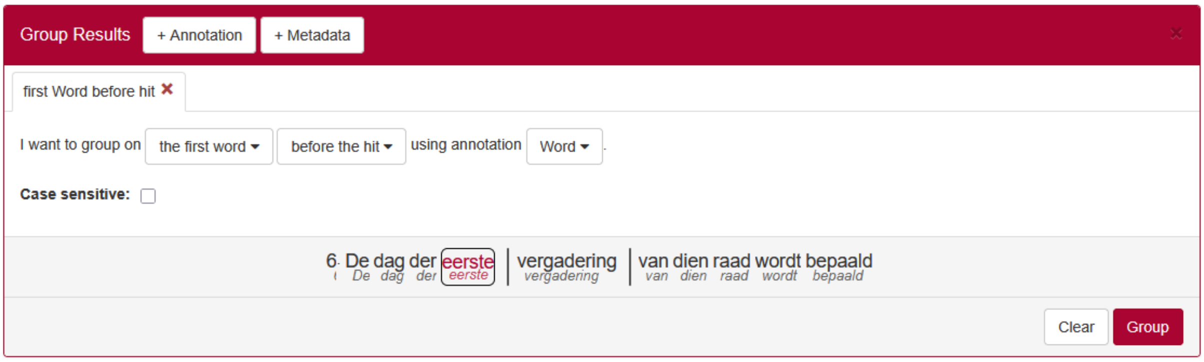
Click on Annotation or Metadata to define grouping criteria.

Results can be grouped by Annotation and by Metadata.

By clicking +Annotation you can group by the first word, by all words or by specific words, whether before the hit, within the hit or after the hit, and based on the annotation Word, Lemma, Part of Speech or Part of Speech with features. When grouping by the first word or specific words, you can also group from the end of the hit. The default grouping is grouping all words within the hit using annotation Word. Clicking +Metadata allows you to group by metadata assigned to the document (Year).

By clicking the Case sensitive box it is possible to distinguish between case sensitive and case insensitive.

The example below is grouped by the first word before the hit. The example dynamically updates when the grouping options are changed.

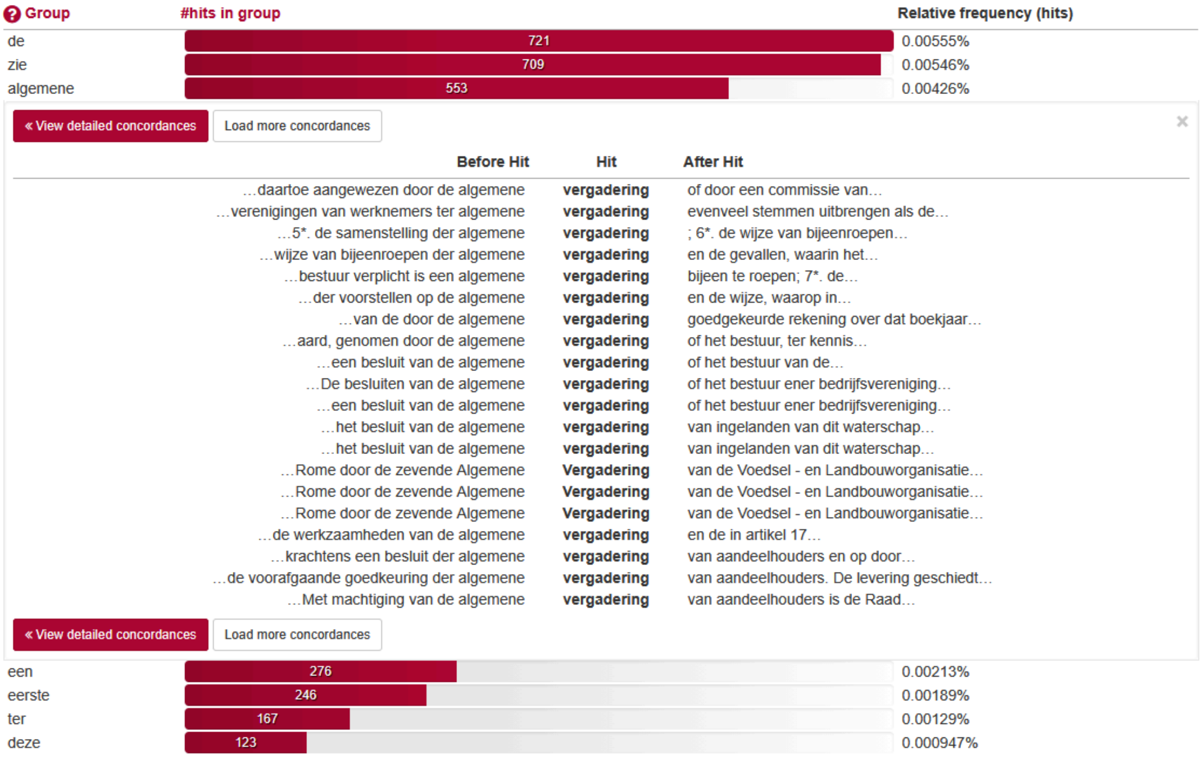


I want to group on the first word ▾ before the hit ▾ using annotation Word ▾

Case sensitive: □

6. De dag der eerste | vergadering | van dien raad wordt bepaald

Click a group to show or hide hits within that group, as shown below. Click once more on the group to close it again. If more than twenty hits are found in a document, you can make them appear by clicking on Load more concordances.



Group	#hits in group	Relative frequency (hits)
de	721	0.00555%
zie	709	0.00546%
algemene	553	0.00426%

Before Hit	Hit	After Hit
…daartoe aangewezen door de algemene	vergadering	of door een commissie van…
…verenigingen van werknemers ter algemene	vergadering	evenveel stemmen uitbrengen als de…
…5*. de samenstelling der algemene	vergadering	; 6*. de wijze van bijeenroepen…
…wijze van bijeenroepen der algemene	vergadering	en de gevallen, waarin het…
…bestuur verplicht is een algemene	vergadering	bijeen te roepen; 7*. de…
…der voorstellen op de algemene	vergadering	en de wijze, waarop in…
…van de door de algemene	vergadering	goedgekeurde rekening over dat boekjaar…
…aard, genomen door de algemene	vergadering	of het bestuur, ter kennis…
…een besluit van de algemene	vergadering	of het bestuur van de…
…De besluiten van de algemene	vergadering	of het bestuur ener bedrijfsvereniging…
…een besluit van de algemene	vergadering	of het bestuur ener bedrijfsvereniging…
…het besluit van de algemene	vergadering	van ingelanden van dit waterschap…
…het besluit van de algemene	vergadering	van ingelanden van dit waterschap…
…Rome door de zevende Algemene	Vergadering	van de Voedsel - en Landbouworganisatie…
…Rome door de zevende Algemene	Vergadering	van de Voedsel - en Landbouworganisatie…
…Rome door de zevende Algemene	Vergadering	van de Voedsel - en Landbouworganisatie…
…de werkzaamheden van de algemene	vergadering	en de in artikel 17…
…krachtens een besluit der algemene	vergadering	van aandeelhouders en op door…
…de voorafgaande goedkeuring der algemene	vergadering	van aandeelhouders. De levering geschiedt…
…Met machtiging van de algemene	vergadering	van aandeelhouders is de Raad…

Group	#hits in group	Relative frequency (hits)
een	276	0.00213%
eerste	246	0.00189%
ter	167	0.00129%
deze	123	0.000947%

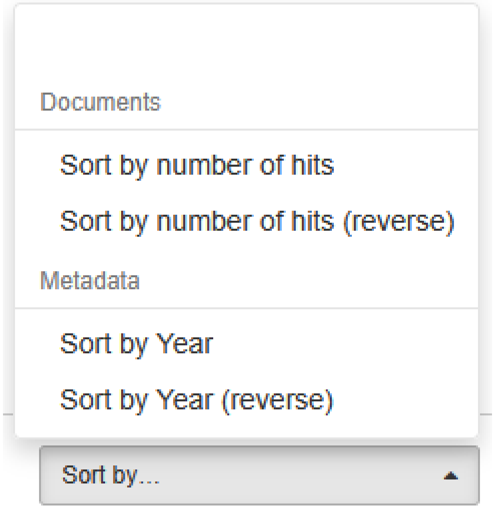
Click on View detailed concordances to go back to the normal hits view to see more detailed information for the hits in this group. The button Go back to grouped results brings you back to the list of groups.

Per Document view

Sorting results

Click on any of the column headings to sort the documents by Document (name), Year or 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 Hit (Documents) or Year (Metadata).



Grouping results

Results Per Document can be grouped by metadata assigned to the document (Year).

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.

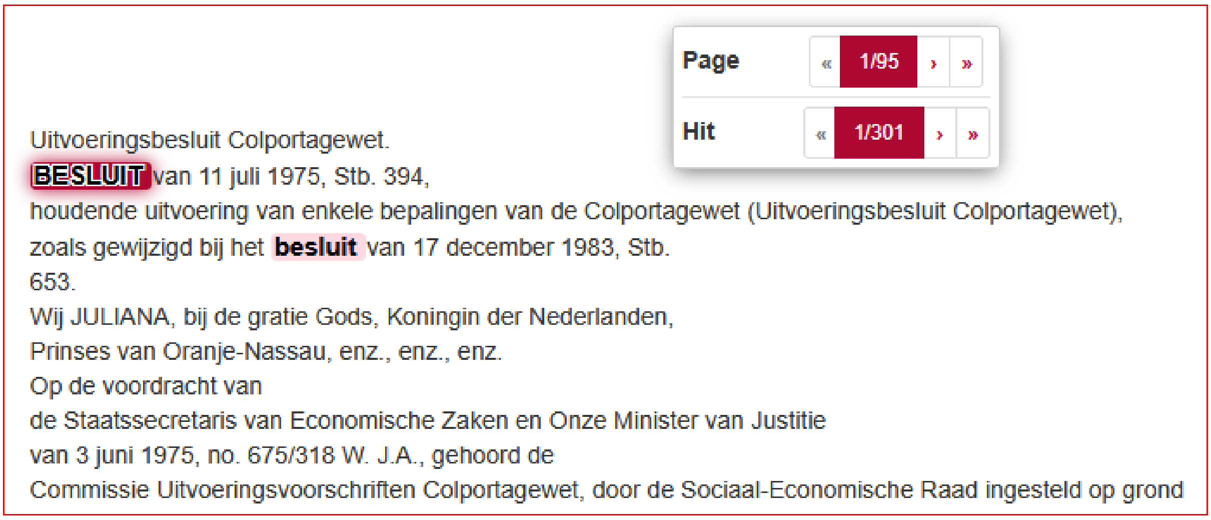
Grouped results can be exported in the same way. However, if you would like to have the metadata with each concordance of a group, you must first click on the red bar of a specific group and then on View detailed concordances. The results you then see can be exported by the use of the Export buttons.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

Information about a document

Click on a document title or the chain icon in the per hit view to open this document in a new window: the Content window.

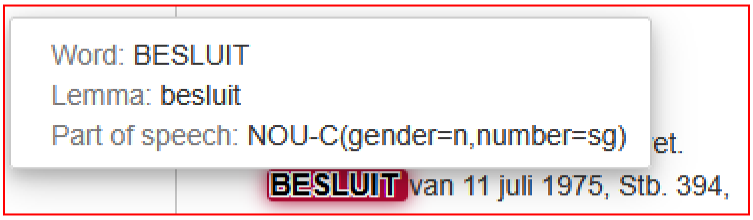
Content

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 (such as *besluit* in the example below). You can navigate from one hit to another by using the arrows at the Hits button (this button can be dragged around), and you can also browse through the pages if the document consists of more than one page.



Uitvoeringsbesluit Colportagewet.
BESLUIT van 11 juli 1975, Stb. 394,
houdende uitvoering van enkele bepalingen van de Colportagewet (Uitvoeringsbesluit Colportagewet),
zoals gewijzigd bij het **besluit** van 17 december 1983, Stb.
653.
Wij JULIANA, bij de gratie Gods, Koningin der Nederlanden,
Prinses van Oranje-Nassau, enz., enz., enz.
Op de voordracht van
de Staatssecretaris van Economische Zaken en Onze Minister van Justitie
van 3 juni 1975, no. 675/318 W. J.A., gehoord de
Commissie Uitvoeringsvoorschriften Colportagewet, door de Sociaal-Economische Raad ingesteld op grond

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:



Word: BESLUIT
Lemma: besluit
Part of speech: NOU-C(gender=n,number=sg)

et.
BESLUIT van 11 juli 1975, Stb. 394,

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, Types (unique word forms), Lemmas, Part of Speech 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

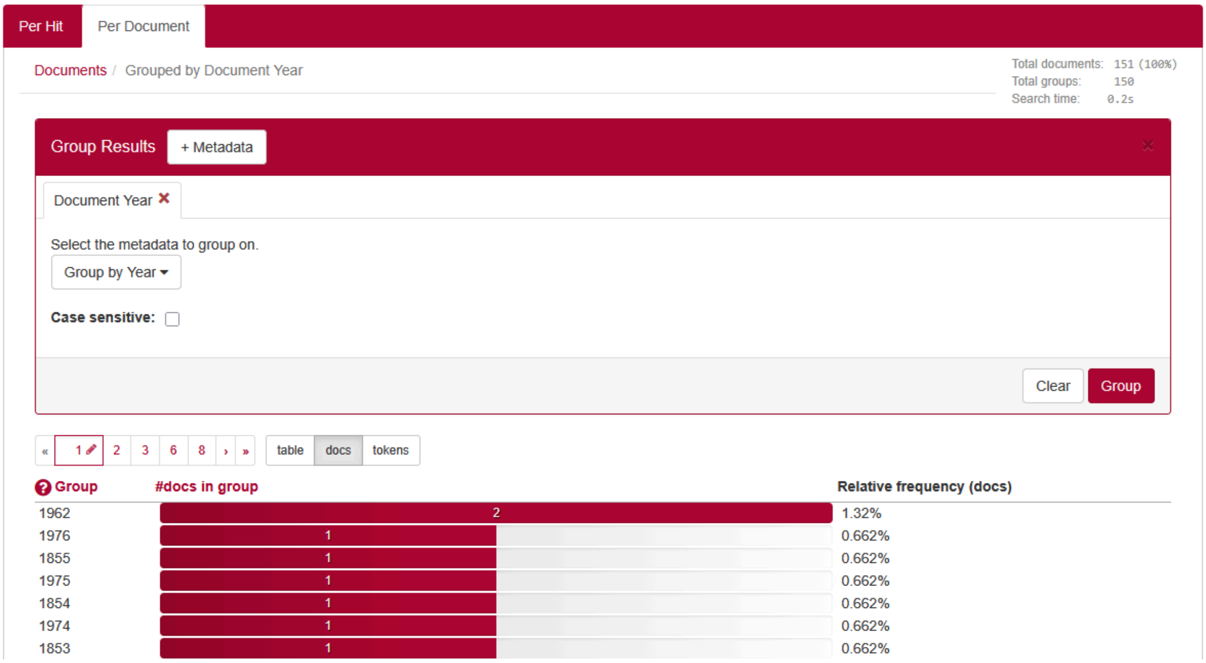
This subtab allows you to investigate the documents. It consists of two drop-down menus to specify the grouping of the metadata and to specify the way the groups are to be shown.

A simple example: suppose we want to obtain information about the average document length for a certain year in the *Corpus Juridisch Nederlands*.

- In the Group documents by metadata drop-down menu, choose Group by Year
- In Show groups as, select *Docs*

- Press Search

You will get this result:



Total documents: 151 (100%)
Total groups: 150
Search time: 0.2s

Select the metadata to group on.
Group by Year

Case sensitive: □

Group	#docs in group	Relative frequency (docs)
1962	2	1.32%
1976	1	0.662%
1855	1	0.662%
1975	1	0.662%
1854	1	0.662%
1974	1	0.662%
1853	1	0.662%

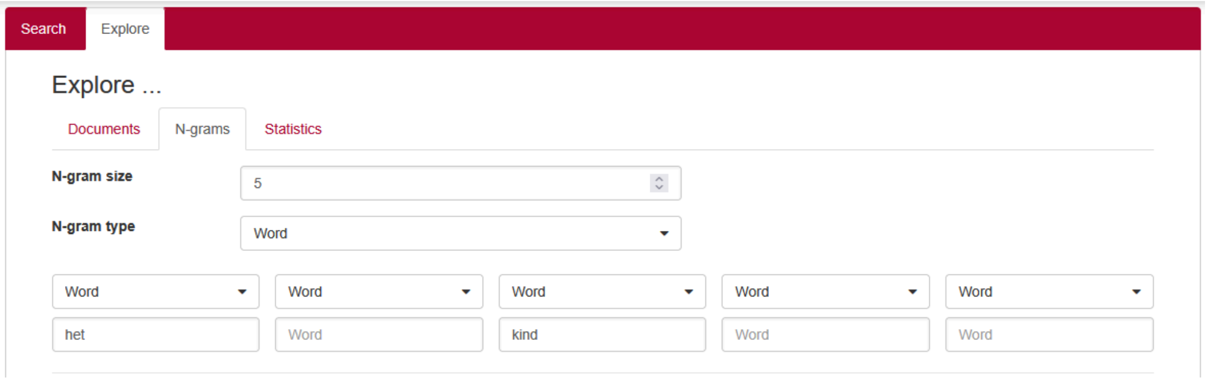
N-grams

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*: the attribute to search for. You can choose: Word (i.e. word form), Lemma, Part of Speech or Part of Speech with features. If you do not specify the search term further, a series of arbitrary tokens equal to the n-gram size will be searched for.
- It is also possible to restrict to, for instance, n-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 metadata.

Example



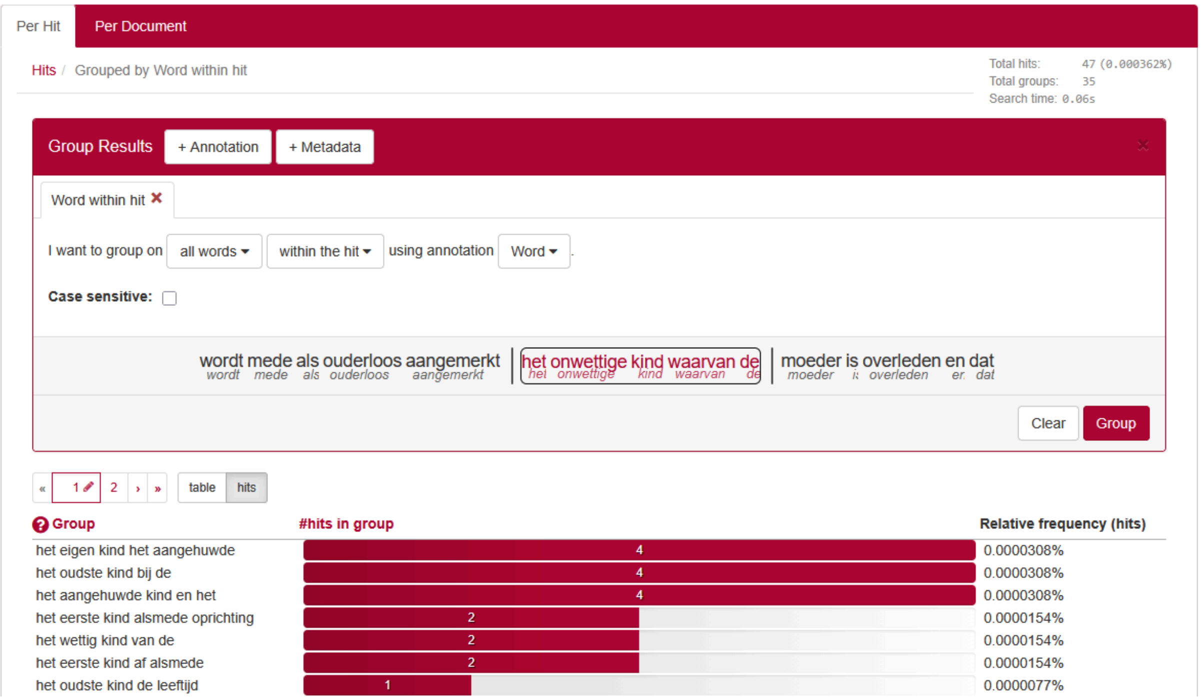
Explore ...

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.



Hits / Grouped by Word within hit

Total hits: 47 (0.000362%)
Total groups: 35
Search time: 0.06s

I want to group on all words within the hit using annotation Word.

Case sensitive: □

wordt mede als ouderloos aangemerkt | het onwettige kind waarvan de | moeder is overleden en dat

Group	#hits in group	Relative frequency (hits)
het eigen kind het aangehuwde	4	0.0000308%
het oudste kind bij de	4	0.0000308%
het aangehuwde kind en het	4	0.0000308%
het eerste kind alsmede oprichting	2	0.0000154%
het wettig kind van de	2	0.0000154%
het eerste kind af alsmede	2	0.0000154%
het oudste kind de leeftijd	1	0.0000077%

Statistics (frequency lists)

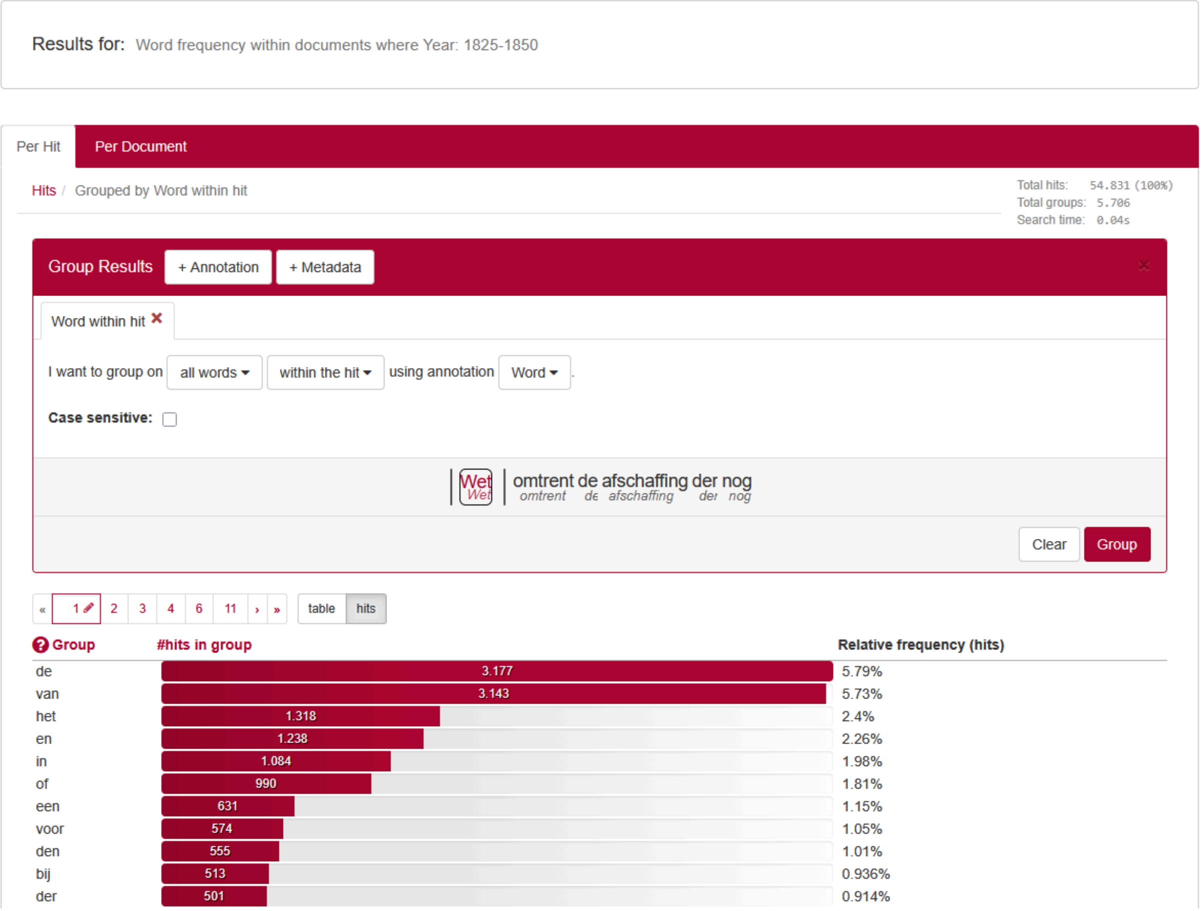
Here, you can produce frequency lists for the corpus. It is rather similar to the previous option, but restricted to 1-grams.

Options

- *Frequency list type:* choose for Word (i.e. Word form), Lemma, Part of Speech or Part of Speech with 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. This results in:



Results for: Word frequency within documents where Year: 1825-1850

Total hits: 54.831 (100%)
Total groups: 5.706
Search time: 0.04s

I want to group on all words within the hit using annotation Word .

Case sensitive: □

Wet omtrent de afschaffing der nog
Wet omtrent de afschaffing der nog
Clear
Group
« 1 2 3 4 6 11 › »
table
hits

Group	#hits in group	Relative frequency (hits)
de	3.177	5.79%
van	3.143	5.73%
het	1.318	2.4%
en	1.238	2.26%
in	1.084	1.98%
of	990	1.81%
een	631	1.15%
voor	574	1.05%
den	555	1.01%
bij	513	0.936%
der	501	0.914%

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!"

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.

It is possible to search for punctuation marks. E.g. to find occurrences of the word "want" preceded by a comma use the following query:

```
[punctBefore="," & word="want"]
```

To find occurrences of the lemma "krant" that are followed by an exclamation mark, use:

```
[lemma="krant" & punctAfter="!"]
```

Some punctuation marks have a special function in regular expressions and therefore must be preceded by a backslash (\) when used in queries. For instance, to search for a period (`.`) after the word **"geweest"**, use:

```
[word="sentence" & punctAfter="\."]
```

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.