Corpus Middelnederlands application manual

Table of contents

Introduction **3**

Information about the corpus published in the application 3

GiGaNT Lexicon service 3

Metadata categories 4

Date: Witness Year 4

Permissive / strict 4

Text type 4

Fictionality 4

Genre 5

Subgenre 5

Title and author 6

Words in title 6

Author 6

Application user manual **6**

Getting started 6

Searching the corpus 7

Simple search 7

Search 7

Wildcards 8

Reset 8

History 8

Global settings 9

Extended search 9

Starting a new search 11

Filter search by 11

Expert search 12

Import query 13

Gap filling 13

Viewing results 14

Per Hit view 14

Sorting results 15

Grouping results 15

Per Document view 17

Sorting results 17

Grouping results 18

Exporting results 18

Information about a document 18

Content 18

Metadata 19

Statistics 19

Exploring the corpus 19

Documents 19

N-grams 20

Options 20

Example 20

Statistics (frequency lists) 21

Options 21

Example 21

Appendix: Corpus Query Language **23**

CQL support 23

Supported features 23

Differences from CWB 24

(Currently) unsupported features 24

Using Corpus Query Language 25

Matching tokens 25

Sequences 25

Regular expression operators on tokens 26

Case- and diacritics sensitivity 26

Matching XML elements 26

Labeling tokens, capturing groups 27

Global constraints 27

Introduction

This manual describes the corpus exploitation environment for the *Corpus Middelnederlands*. 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 published in the application

The Corpus Middelnederlands in the current release is a collection of 372 documents from the period of 1300-1550. The main sources for this corpus are the rhyming texts and prose texts from the CD-ROM Middle Dutch, compiled bij the INL/INT and published in 1998 by the Flemish Standaard Uitgeverij and the Dutch Sdu. (The CD-ROM texts, taken from the corpus Gysseling have not been included here, for these texts please consult the online Corpus Gysseling.).

This corpus is in part an extension of the Corpus Middelnederlands that was integrated in the Nederlab portal in 2018. Two completely French texts were removed. The metadata has also been corrected.

The *Corpus Middelnederlands* presented in this application contains classical works of Middle Dutch literature like *Beatrijs*, *Van den vos Reynaerde*, the *abele spelen*, the stories about King Arthur or about Charlemagne, all texts from the famous Gruuthuse manuscript (including the Egidius song), but also many of the lesser known or less researched texts, such as prose adaptations of the rhyming knight's tales (the so-called 'chapbooks'), collections of songs such as the Antwerp Songbook, several Bible translations, hagiographies, books of prayer, chronicles, and all kinds of religious, didactic and scientific treatises, medical manuals and recipes.

A number of texts belonging to the so-called *artes literature* have been added to these source texts, such as the Hattem manuscript (C5) and *Van der proprieteyten der dinghen* ('On the Properties of Things') by Bartholomeus Anglicus – both published and made available by the Werkgroep Middelnederlandse Artesliteratuur (WEMAL) – and the *Circa Instans* and the *Trotula*.

GiGaNT Lexicon service

Contrary to other historical corpora of the *Dutch Language Institute* (*Instituut voor de Nederlandse Taal*), the *Corpus Middelnederlands* has not yet been annotated with part of speech and lemma. To make the corpus more accessible, suggestions for query expansion are given, using het INT lexicon service with the historical computational lexicon GiGaNT-HILEX.

The current version of GiGaNT-HILEX in the lexicon service contains the lexicon modules based on the Dictionary of the Dutch Language and the Dictionary of Middle Dutch.

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

Metadata categories

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

Date: Witness Year

With respect to dating the source, a distinction can made between the date of the manuscript in which a text was handed down (Witness Year) and the period in which a text was produced (Text Year). For each document in this corpus, we indicate the period in which the manuscript, providing us the text, was written. Witness Year does not necessarily refer to the period in which the text itself was written. It only concerns the carrier of the text.

Witness Year cannot be stated with the same accuracy for every document.. For example, the manuscript with the *Cyrurgie* of Jan Yperman can be dated exactly to 1351, while the source in which the text of *Wrake van Ragisel* is included, originated between 1340-1360.

Permissive / strict

It is possible to do a permissive and strict search for Witness Year. What exactly is the difference between the two options? An example can clarify this. Suppose you want to investigate sources that came into being between 1425 and 1450.

If you choose to do a Strict search by Witness Year, the search query will only result in manuscripts that were produced later than 1425 but before 1450. Among the 11 results is a manuscript from the period 1434-1436 in which the *Spiegel der sonden* is included.

If, on the other hand, you choose the option Permissive, no less than 42 documents are found, one of which is Geert Grote's *Getijdenboek*, which was handed down in a manuscript dating from the period 1450-1470.

Text type

All texts in this corpus are provided with metadata to help determine fictionality, genre and subgenre of the text. These metadata can be filtered during the search.

Fictionality

The documents have been divided into fictional texts (fiction) and non-fictional texts (non-fiction). Generally speaking, this contemporary classification will also apply to Middle Dutch texts. However, there are cases in which we will classify a text as fiction and medieval people as non-fiction. For this corpus, we applied the medieval point of view.

Genre

The texts are divided into two main genres: *prose* and *verse* and one mixed genre *prose+verse*, used for texts where prose and verse approximately balance each other.

Subgenre

The texts can be sorted out using one or more of the different subgenre labels (see the paragraph Filter search by). These labels may indicate a general text category as well as a more specific one; they may touch either the content of a text or its form (e.g. *stanzaic*).

- *Alchemy*: texts dealing with alchemical knowledge or information
- *Artes*: non-religious and non-fictional texts written for utilitarian and instructive purposes
- *Artifices*: descriptions of how to devise and use specific expedients or how to perform certain tricks
- *Astronomy / astrology*: texts describing the practical application of the medieval art or science of astronomy
- *Biblical texts*: Bible, lectionaria, diatessara
- *Biology*: texts with biological information
- *Chapbook*: printed books with popular literature, consisting chiefly of adaptations of chivalric romances, tales, ballads, tracts etc.
- *Chiromancy*: texts describing the art of telling the characters and fortunes of persons by inspections of their hands
- *Chivalric romance*: mostly versified stories about the idealized world of knights
- *Drama*: texts with one or more stage plays
- *Epic*: narrative literary texts
- *History*: texts with historical information
- *Legend*: texts offering a traditional story sometimes regarded as historical but not authenticated; a myth, a fable
- *Legendary hagiography*: descriptions of the life of saints
- *Lexicon*: text offering lexicographical information (a vocabulary or wordlist);
- *Linguistics*: texts offering linguistic information
- *Medicine*: texts with medical information or instructions
- *Metallurgy*: texts with information on (the use) of metals
- *Natural history*: texts providing in a popular form the scientific study of nature in general or animals, plants, even stones, in particular
- *Philosophy / ethics*: texts with life lessons, instructions of how to behave in life and social traffic, based on specific philosophical or secular grounds
- *Physiognomy*: texts describing the art of telling the characters and fortunes of persons by inspections of their physiognomy
- *Prayer*: texts containing a collection of prayers; also: book of hours
- *Religious*: texts dealing with religion or religious matters
- *Secular*: texts dealing with non-religious matters
- *Science*: texts offering scientific information
- *Song*: texts containing a collection of songs
- *Stanzaic*: texts composed in the form of stanzas

- *Technic*: texts with technical information or instructions
- *Theology / ethics*: texts with life lessons, instructions of how to behave in life and social traffic, based on theological or religious grounds
- *Travelogue*: travel books

Title and author

Words in title

Every document has a title. Usually this is the title as we know it from the editions used for this corpus.

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

Author

It is possible to search by author name. However, for most of the documents in this corpus the author is unknown or uncertain.

Er zijn ook nog gegevens te vinden over metadata bij Information about a document, o.a. localization, edition, corpus of the document in which the hit was found.

Application user manual

Getting started

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

- To search for a specific word form, use Simple Search or Extended search:
    - Simple search for Word *gesont*
    - Extended search for Word *gesont*
- 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 *ge-* and ending with *-en* in Simple Search or Extended Search
    - words starting with *be-*, ending in *-en* with one syllable in between in Expert Search
- To see which unique forms occur as a result of your search, use the Group hits by feature.
    - example Group by Context (advanced): all words following *lieve*
- To explore the distribution of document properties in the corpus, use the Explore feature
    - example: all authors whose books are dated between 1475-1500

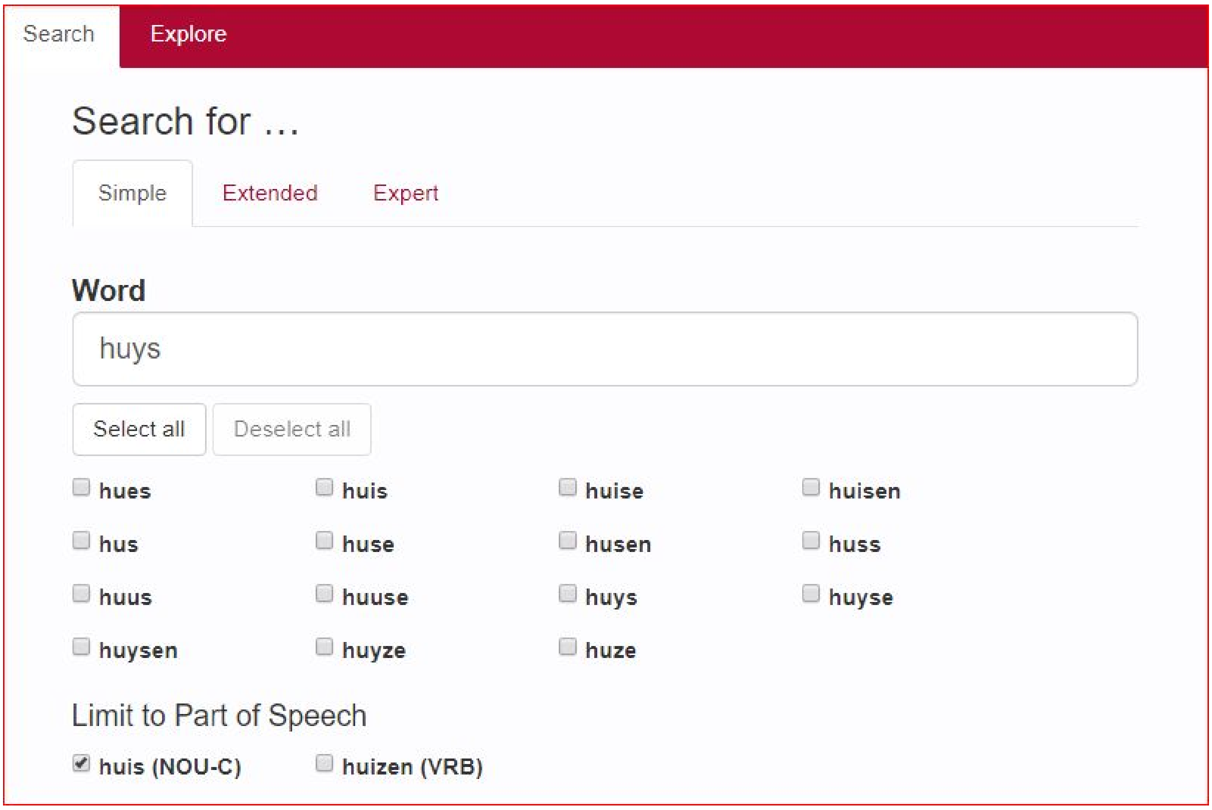
Searching the corpus

Simple search

Search

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

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



Search for …

Word

huys

□ hues □ huis □ huise □ huisen
□ hus □ huse □ husen □ huss
□ huus □ huuse □ huys □ huyse
□ huysen □ huyze □ huze

Limit to Part of Speech

☑ huis (NOU-C) □ huizen (VRB)

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

It is also possible to enter a phrase: *bij haers vaders huys* or *princen des huys*. You will then find all occurrences of that exact phrase.

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

Wildcards

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. *an*, *anzoemen* and *anderen*.

? The question mark matches a single character once. Therefore, *b?n* matches *only* three-letter values starting with an *b* and ending with a *n*, e.g. *ban*, *ben*, *bin*, *bon*, *bun* and *byn*.

This wildcard can be used more than once. Thus *b???n* matches *boven, buten*, *bouen*, *biden* and *began*.

Note that searching with wildcards are limited to Simple Search and Extended Search. [In Expert Search you can use so-called *regular expressions* instead of wildcards.]

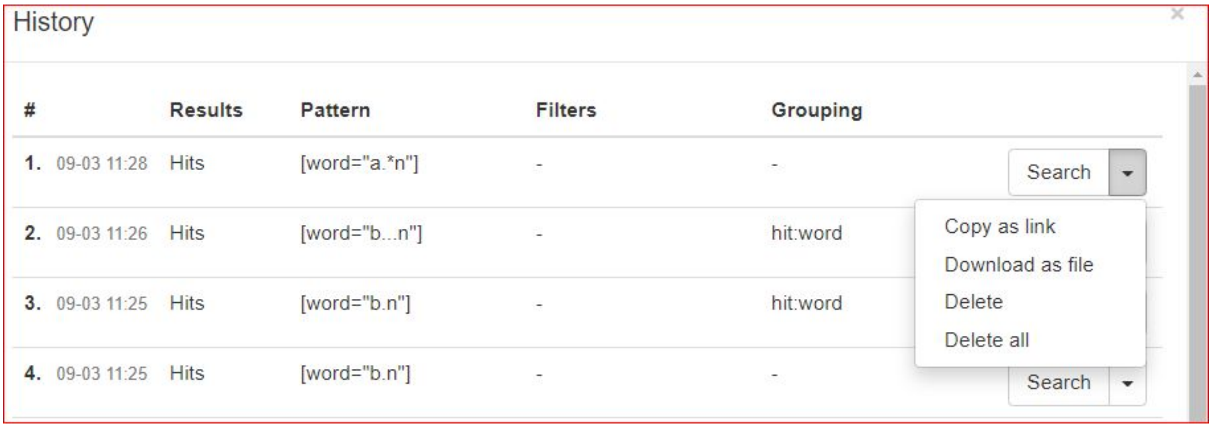
Reset

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).



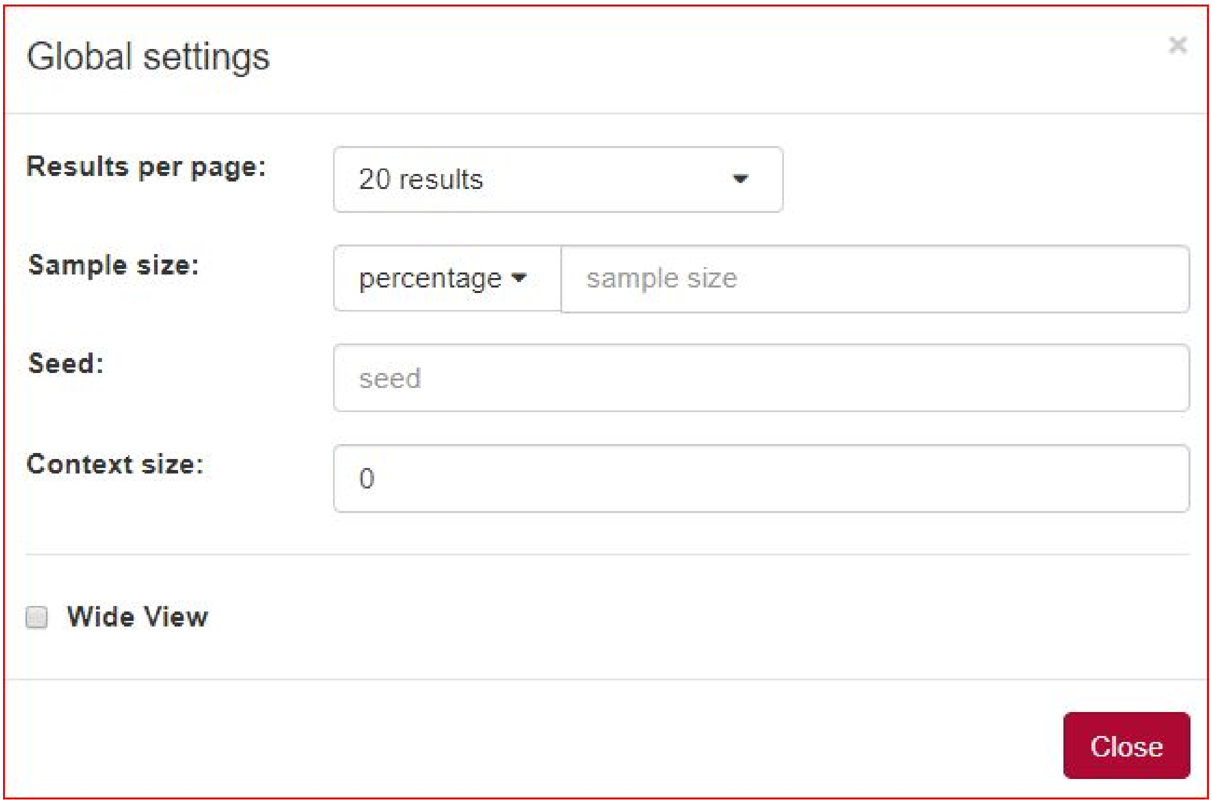
#	Results	Pattern	Filters	Grouping
1. 09-03 11:28	Hits	[word="a.*n"]	-	-
2. 09-03 11:26	Hits	[word="b...n"]	-	hit:word
3. 09-03 11:25	Hits	[word="b.n"]	-	hit:word
4. 09-03 11:25	Hits	[word="b.n"]	-	-

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 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

Sample size: percentage | sample size

Seed: seed

Context size: 0

☐ Wide View

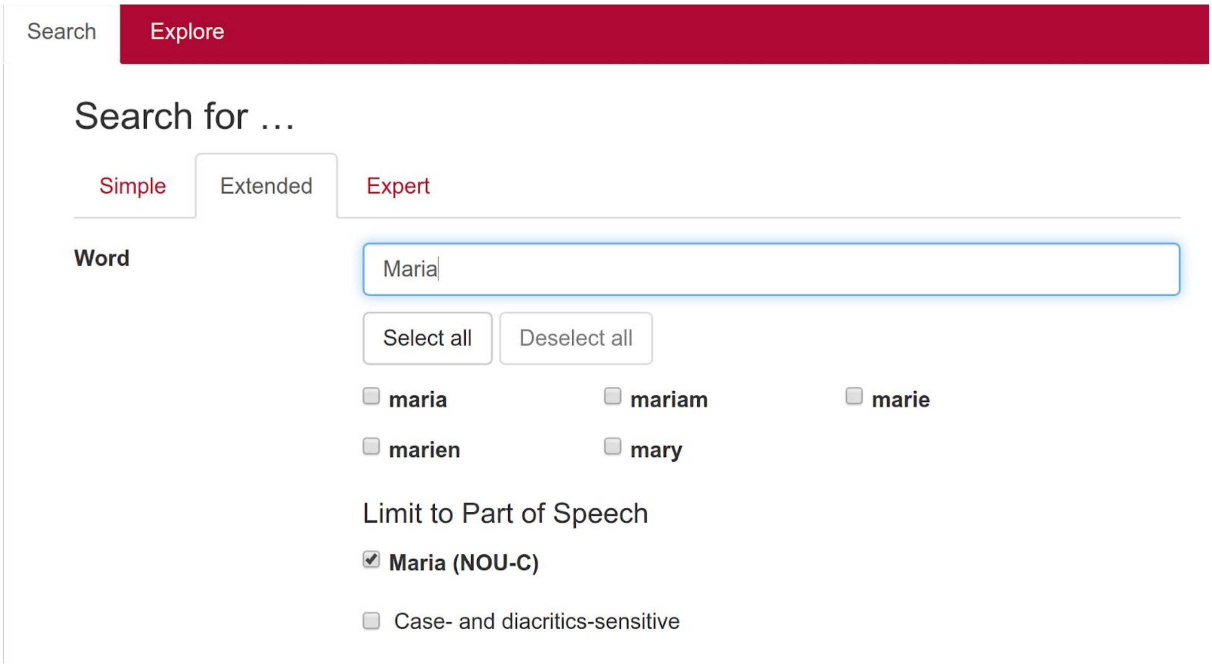
Extended search

Like in Simple search Extended Search allows you to quickly search for specific word forms (e.g. *huys*). In addition, Extended search offers some extra possibilities to refine your search.

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

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

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 *Maria* will result in 5120 occurrences of this name. By ticking the box Case- and diacritics-sensitive you will only find the Word *Maria* (1854x), but not the variant *maria.* In order to directly find only occurrences of the Word (form) *Maria*, tick the box Case- and diacritics-sensitive under the search field Word (as shown below).



Word: Maria

☐ maria ☐ mariam ☐ marie
☐ marien ☐ mary

Limit to Part of Speech

☑ Maria (NOU-C)

☐ Case- and diacritics-sensitive

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

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

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

For the search field Word it is possible to search for a series of tokens by entering multiple values - including wildcards - separated by a space, e.g. *die coreel*, *die* * or * *coreel*.

Starting a new search

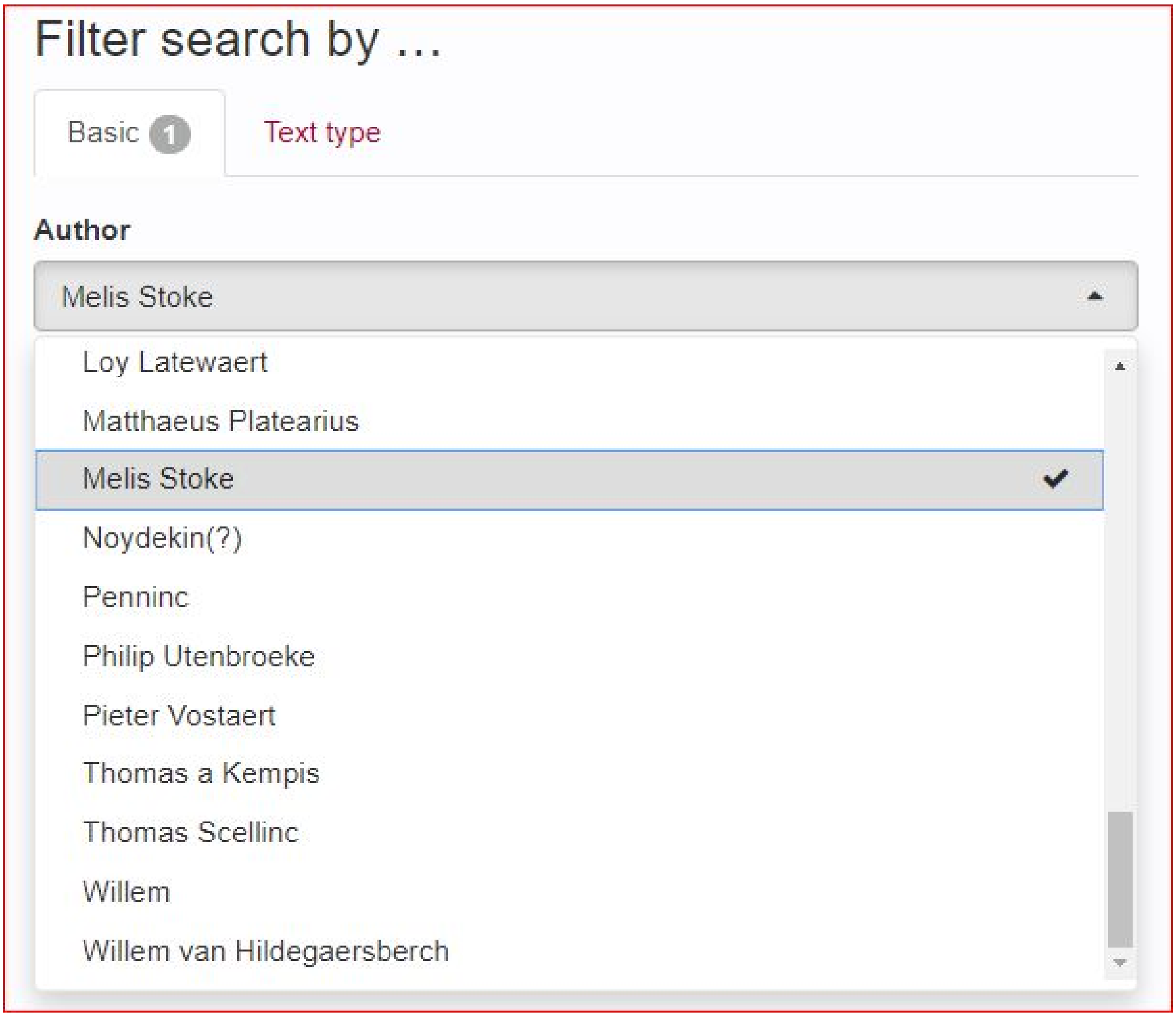
You can start a new search by pressing the Reset button. By doing so, both the search query and the hits found will disappear. Your search history, however, will remain unchanged.

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

Filter search by

At the right side you will find the option to limit your query to a subset of documents with specific metadata values. You can apply different filters for Author, Title, Witness Year (all in Basic) or for Fictionality, Genre and Subgenre (all in Text type). (To view the results for all documents simply leave the attributes in the filtering form empty.)

There are two different ways to specify a filter, depending on the field type. You can either fill in a value yourself - for instance 'Title' - or choose one or more values from a drop-down list. The drop-down list has been applied especially when the number of values to choose from is relatively small. You can pick one of these values by clicking on it; your choice will be marked with a tick. It is possible to choose several values. If you want to delete a selection, you can click on the corresponding line again.



Filter search by …

Author: Melis Stoke

When on the other hand the set of possible values is relatively large (e.g. Title), you have to type a specific value in the search field. After entering a character in the search field, a list of possible values is suggested. Clicking on an auto-completed value will paste that value in the field. Note that this only works with a single word, like *Brabantsche*. In order to search for an exact phrase, i.e. a multiple word value, it must be surrounded by double quotes. For instance, in the field Title "*Brabantsche* *" will result in three documents containing the word *Brabantsche*.

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

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="man"] or [word = "man"] does not make any difference to the result. However, there is a difference between the queries [word="man"] and [word=" man"]. The first search results in over 17.000 hits, but the second one in zero!

Some examples:

- Simple: `[word="man"]`, e.g. the attribute word matches the regular expression *man*; `[word!="man"]`, e.g. the attribute word does **not** match the regular expression *man*. `[word=".*man"]` matches all word forms ending with *man*, including *man* itself.
- Combination of attributes (combining operators are &, |, !), e.g. [word="huis"|"god"|"dier"] matches either the word *huis*, the word *god* or the word *dier*.
- The empty `[]` matches any token, e.g. `[word="man"][]{3}[word="god"]` matches a sequence of *man* followed by *god* with 3 arbitrary tokens in between. The query `[word="man"][]{1,3}[word="god"]` matches a sequence of 1 till 3 tokens/words between *man* and *god*.
- Querying with tag positions using e.g. `<s>` (start of sentence), `</s>` (end of sentence), `<s/>` (whole sentence) or `<s> ... </s>`. However, queries with tag positions are not very useful in the *Corpus Middelnederlands*, since punctuation is usually missing.
- Operators |, & and parentheses () and the repetition operators (+, *, ? and {}) can be used to build complex sequence queries. Example: `"goet" "wijf" | "quaet" "wijf"`, or even ("goet" "wijf" | "quaet" "wijf")+, matching any sequence of *goet wijf* or *quaet wijf*.

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.

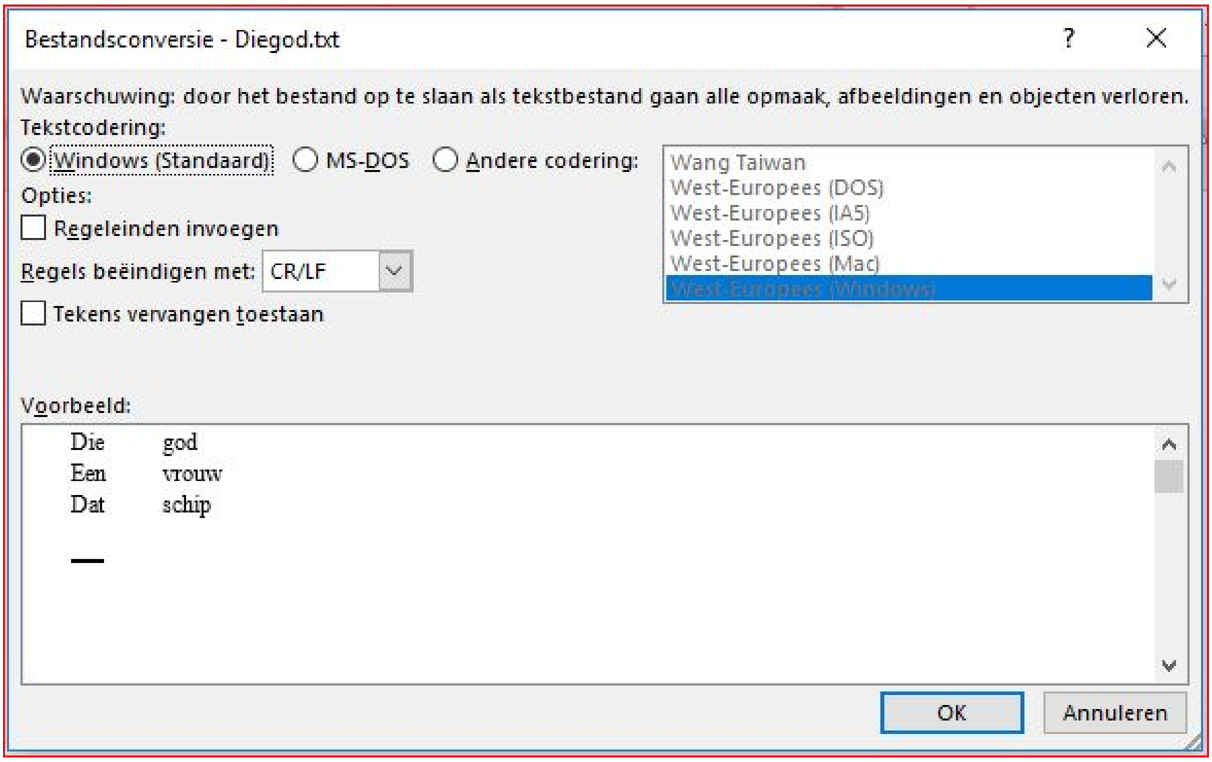
Import query

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 afterwards 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:



Bestandsconversie - Diegod.txt

Waarschuwing: door het bestand op te slaan als tekstbestand gaan alle opmaak, afbeeldingen en objecten verloren.

Tekstcodering:

● Windows (Standaard) ○ MS-DOS ○ Andere codering:

Wang Taiwan
West-Europees (DOS)
West-Europees (IA5)
West-Europees (ISO)
West-Europees (Mac)
West-Europees (Windows)

Opties:

□ Regeleinden invoegen

Regels beëindigen met: CR/LF

□ Tekens vervangen toestaan

Voorbeeld:

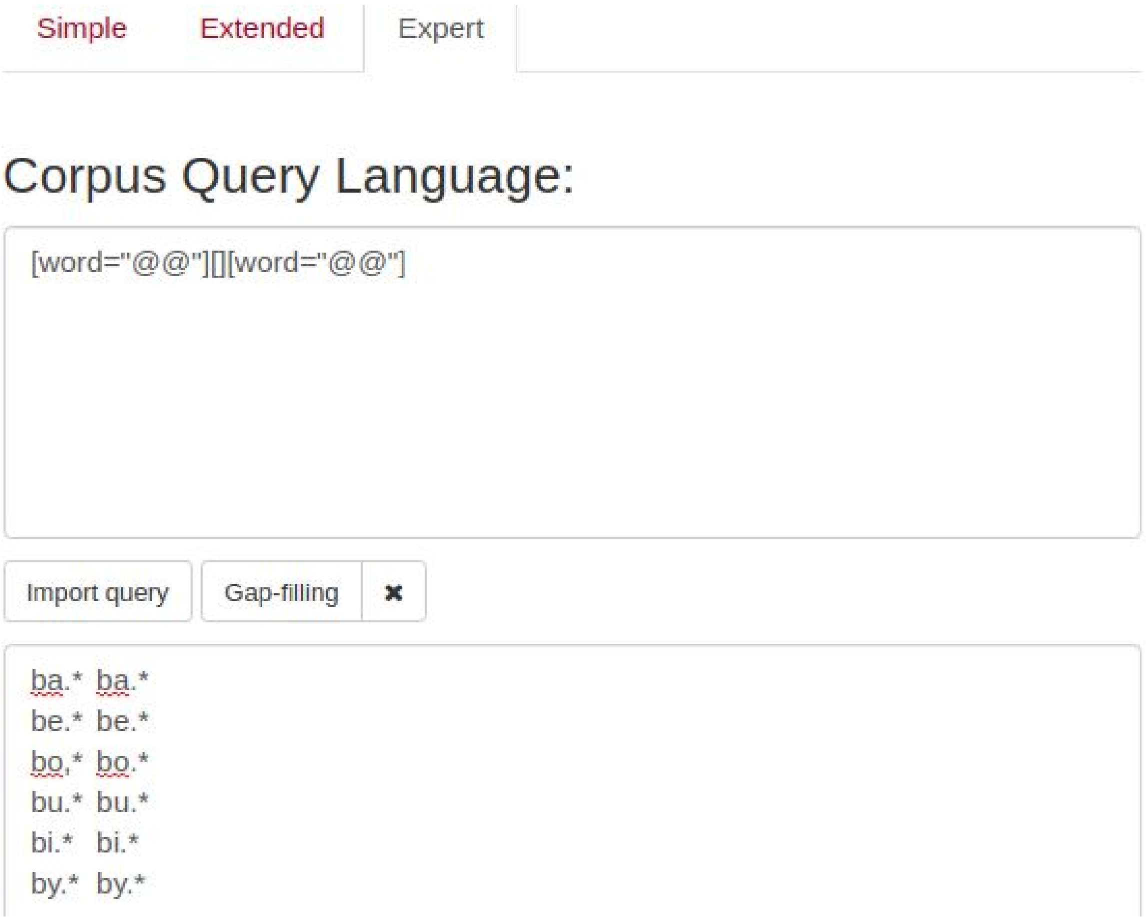
Die	god
Een	vrouw
Dat	schip

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 a pronoun/determiner and a noun 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:

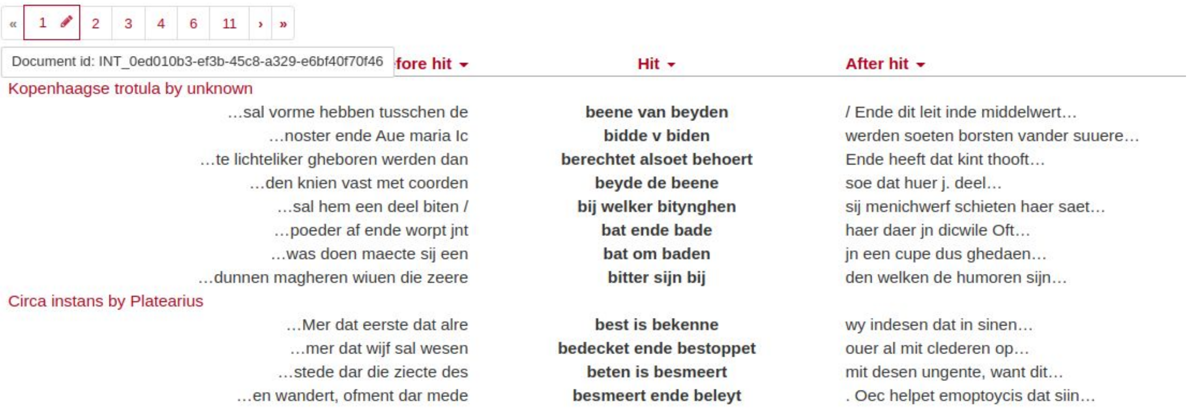


Corpus Query Language:

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

```
ba.*  ba.*
be.*  be.*
bo.*  bo.*
bu.*  bu.*
bi.*   bi.*
by.*  by.*
```

The patterns in the first column - *ba.*, be.*, bo.*, bu.*, bi.*, by.** - will be entered at the position of the first gap (@@) and the patterns in the second column - also *ba.*, be.*, bo.*, bu.*, bi.*, by.** - at the position of the second gap (@@). With these values, gap-filling yields the following results:



Document id: INT_0ed010b3-ef3b-45c8-a329-e6bf40f70f46

fore hit	Hit	After hit
Kopenhaagse trotula by unknown		
…sal vorme hebben tusschen de	**beene van beyden**	/ Ende dit leit inde middelwert…
…noster ende Aue maria Ic	**bidde v biden**	werden soeten borsten vander suuere…
…te lichteliker gheboren werden dan	**berechtet alsoet behoert**	Ende heeft dat kint thooft…
…den knien vast met coorden	**beyde de beene**	soe dat huer j. deel…
…sal hem een deel biten /	**bij welker bitynghen**	sij menichwerf schieten haer saet…
…poeder af ende worpt jnt	**bat ende bade**	haer daer jn dicwile Oft…
…was doen maecte sij een	**bat om baden**	jn een cupe dus ghedaen…
…dunnen magheren wiuen die zeere	**bitter sijn bij**	den welken de humoren sijn…
Circa instans by Platearius		
…Mer dat eerste dat alre	**best is bekenne**	wy indesen dat in sinen…
…mer dat wijf sal wesen	**bedecket ende bestoppet**	ouer al mit clederen op…
…stede dar die ziecte des	**beten is besmeert**	mit desen ungente, want dit…
…en wandert, ofment dar mede	**besmeert ende beleyt**	. Oec helpet emoptoycis dat siin…

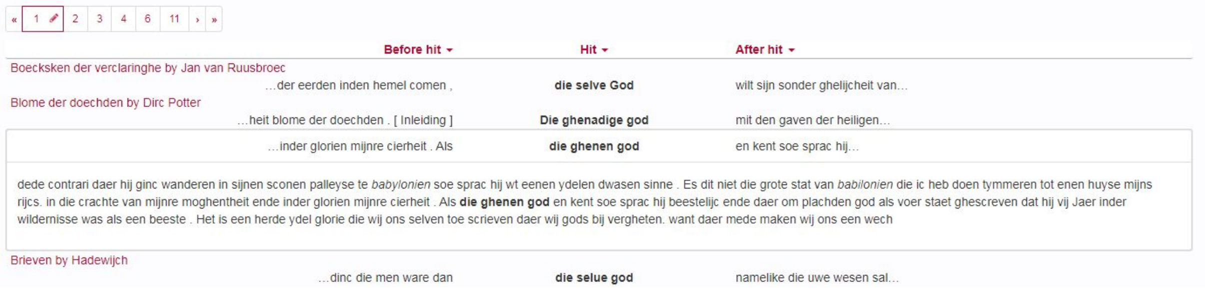
Please note that for this to work, you do need to enter @@ in the field where you want the substitution to take place. An empty field ([]) will match any term.

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 - the bold word(s) in the column Hit - to display the context of the hit. Click the hit again to close.



	Before hit	Hit	After hit
Boecksken der verclaringhe by Jan van Ruusbroec			
	...der eerden inden hemel comen ,	**die selve God**	wilt sijn sonder ghelijcheit van...
Blome der doechden by Dirc Potter			
	...heit blome der doechden . [Inleiding]	**Die ghenadige god**	mit den gaven der heiligen...
	...inder glorien mijnre cierheit . Als	**die ghenen god**	en kent soe sprac hij...

dede contrari daer hij ginc wanderen in sijnen sconen palleyse te *babylonien* soe sprac hij wt eenen ydelen dwasen sinne . Es dit niet die grote stat van *babilonien* die ic heb doen tymmeren tot enen huyse mijns rijcs. in die crachte van mijnre moghentheit ende inder glorien mijnre cierheit . Als **die ghenen god** en kent soe sprac hij beestelijc ende daer om plachden god als voer staet ghescreven dat hij vij Jaer inder wildernisse was als een beeste . Het is een herde ydel glorie die wij ons selven toe scrieven daer wij gods bij vergheten. want daer mede maken wij ons een wech

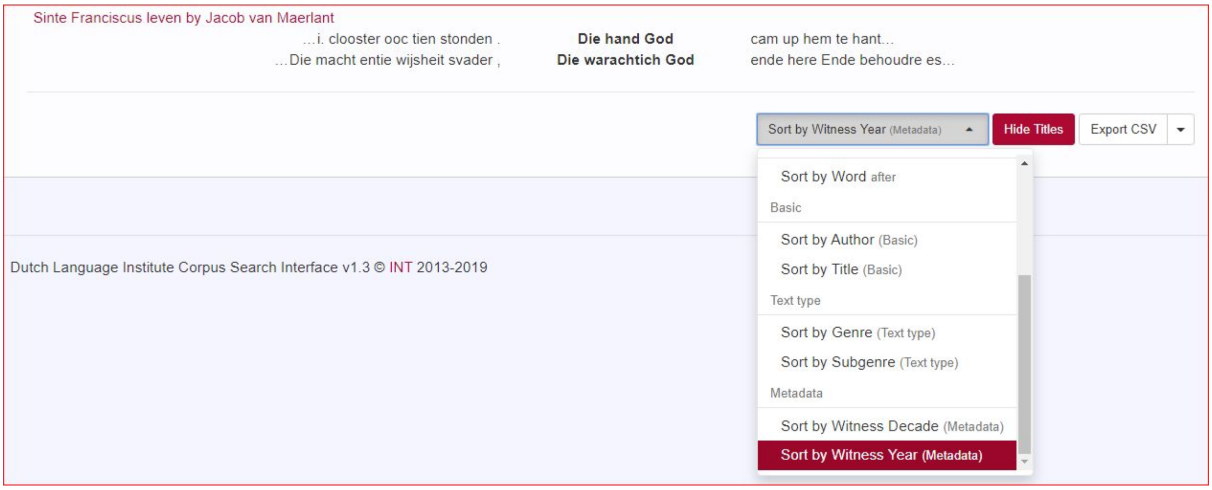
	Before hit	Hit	After hit
Brieven by Hadewijch			
	...dinc die men ware dan	**die selue god**	namelike die uwe wesen sal...

Hit rows are always preceded by a row containing the document title in which those hits occurred, in this case *Blome der doechden*. 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.

Sorting results

Click on any of the column headings (i.e. Before hit, Hit or After hit) to sort the hits on Words within the 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, Basic (Author, Title), Text type (Genre, Subgenre) and Metadata (Witness Decade, Witness Year):



Sinte Franciscus leven by Jacob van Maerlant		
...i. clooster ooc tien stonden .	**Die hand God**	cam up hem te hant...
...Die macht entie wijsheit svader ,	**Die warachtich God**	ende here Ende behoudre es...

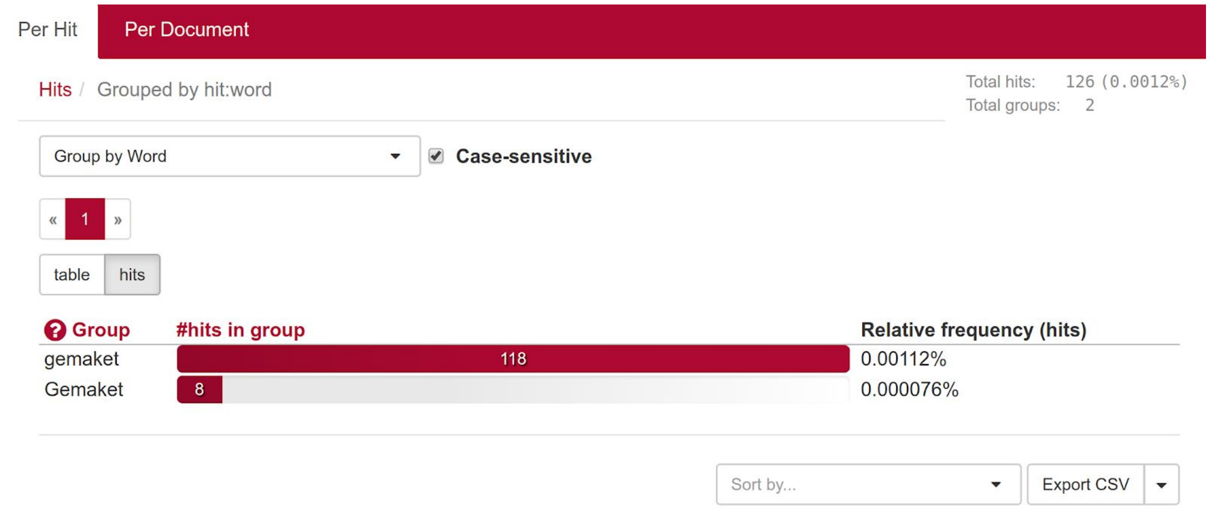
Sort by Witness Year (Metadata)
Hide Titles
Export CSV
Sort by Word after
Basic
Sort by Author (Basic)
Sort by Title (Basic)
Text type
Sort by Genre (Text type)
Sort by Subgenre (Text type)
Metadata
Sort by Witness Decade (Metadata)
Sort by Witness Year (Metadata)

Dutch Language Institute Corpus Search Interface v1.3 © INT 2013-2019

Grouping results

Results Per Hit can be grouped by properties of Hit, Before hit, After hit, Basic (Author, Title), Text type and of the metadata of the documents in which those hits occur (Author, Title, Genre, Subgenre, Witness Decade, Witness Year). Grouping is facilitated by the dropdown 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.

In the Per hit view, advanced grouping options are available by selecting the option Context (advanced). It is now possible to make a distinction between lower and upper case by ticking the box Case-sensitive.



Hits / Grouped by hit:word

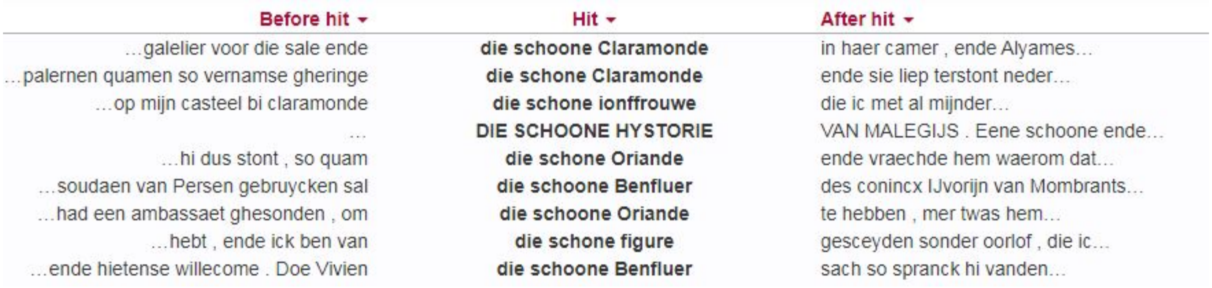
Total hits: 126 (0.0012%)
Total groups: 2

Group by Word ☑ Case-sensitive

Group	#hits in group	Relative frequency (hits)
gemaket	118	0.00112%
Gemaket	8	0.000076%

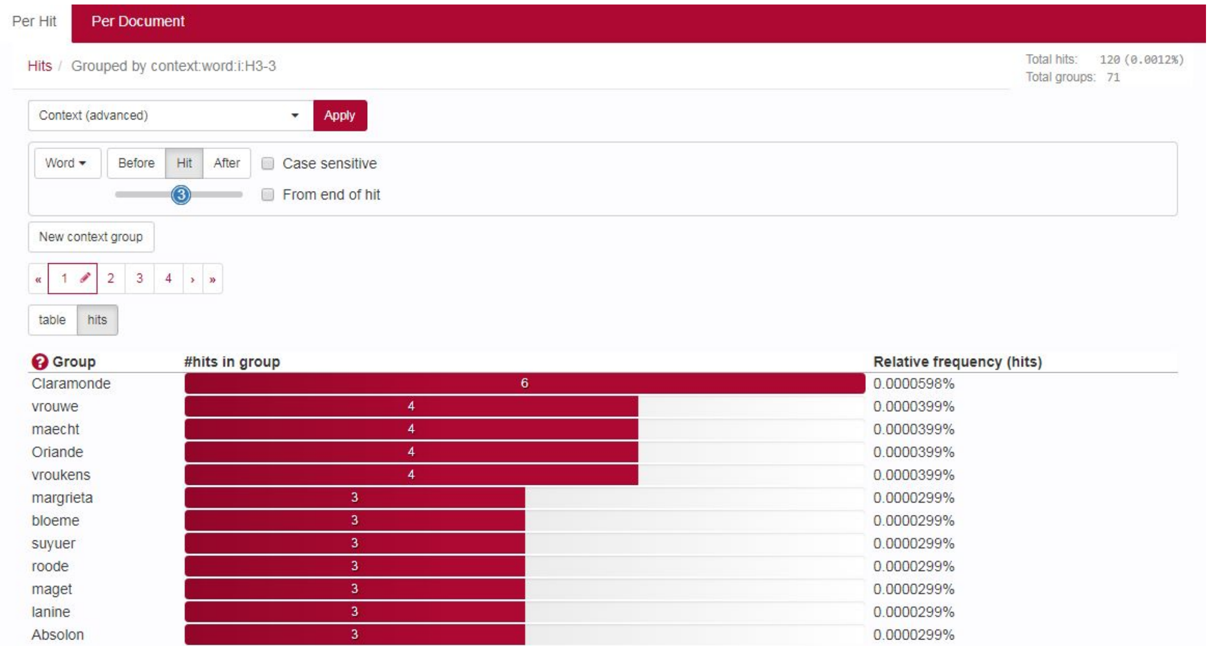
Context (advanced) allows you to group the results by up to 5 tokens before or after the hits. It also allows you to group the results based on (parts of) the hits. By pressing the New context group button you can group the results by another property or another range.

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 - in Expert Search: [word="die"] [word="scho.?ne"] [] - produces hits like the following (Titles are hidden):



Before hit	Hit	After hit
…galelier voor die sale ende	die schoone Claramonde	in haer camer , ende Alyames…
…palernen quamen so vernamse gheringe	die schone Claramonde	ende sie liep terstont neder…
…op mijn casteel bi claramonde	die schone ionffrouwe	die ic met al mijnder…
…	DIE SCHOONE HYSTORIE	VAN MALEGIJS . Eene schoone ende…
…hi dus stont , so quam	die schone Oriande	ende vraechde hem waerom dat…
…soudaen van Persen gebruycken sal	die schoone Benfluer	des conincx IJvorijn van Mombrants…
…had een ambassaet ghesonden , om	die schoone Oriande	te hebben , mer twas hem…
…hebt , ende ick ben van	die schone figure	gesceyden sonder oorlof , die ic…
…ende hietense willecome . Doe Vivien	die schoone Benfluer	sach so spranck hi vanden…

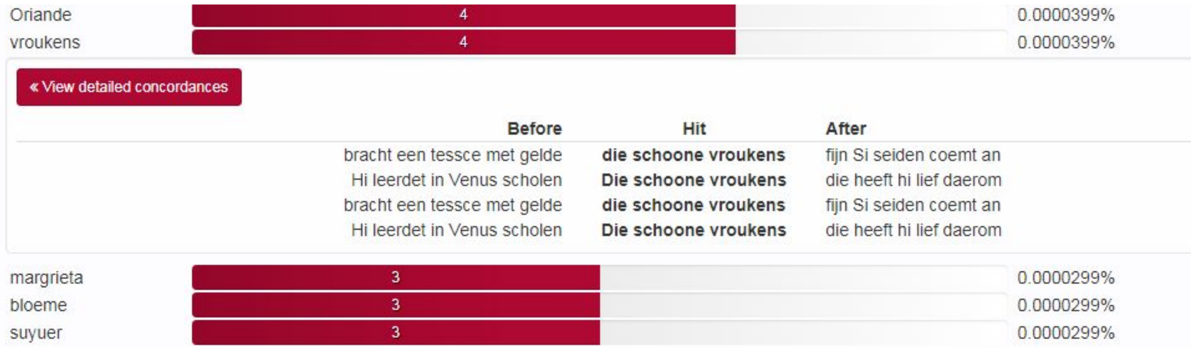
It is now possible to group the hits by the third tokens of those hits, e.g. the nouns. See below.



Total hits: 120 (0.0012%)
Total groups: 71

Group	#hits in group	Relative frequency (hits)
Claramonde	6	0.0000598%
vrouwe	4	0.0000399%
maecht	4	0.0000399%
Oriande	4	0.0000399%
vroukens	4	0.0000399%
margrieta	3	0.0000299%
bloeme	3	0.0000299%
suyuer	3	0.0000299%
roode	3	0.0000299%
maget	3	0.0000299%
lanine	3	0.0000299%
Absolon	3	0.0000299%

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



Oriande	4	0.0000399%
vroukens	4	0.0000399%

Before	Hit	After
bracht een tessce met gelde	**die schoone vroukens**	fijn Si seiden coemt an
Hi leerdet in Venus scholen	**Die schoone vroukens**	die heeft hi lief daerom
bracht een tessce met gelde	**die schoone vroukens**	fijn Si seiden coemt an
Hi leerdet in Venus scholen	**Die schoone vroukens**	die heeft hi lief daerom

margrieta	3	0.0000299%
bloeme	3	0.0000299%
suyuer	3	0.0000299%

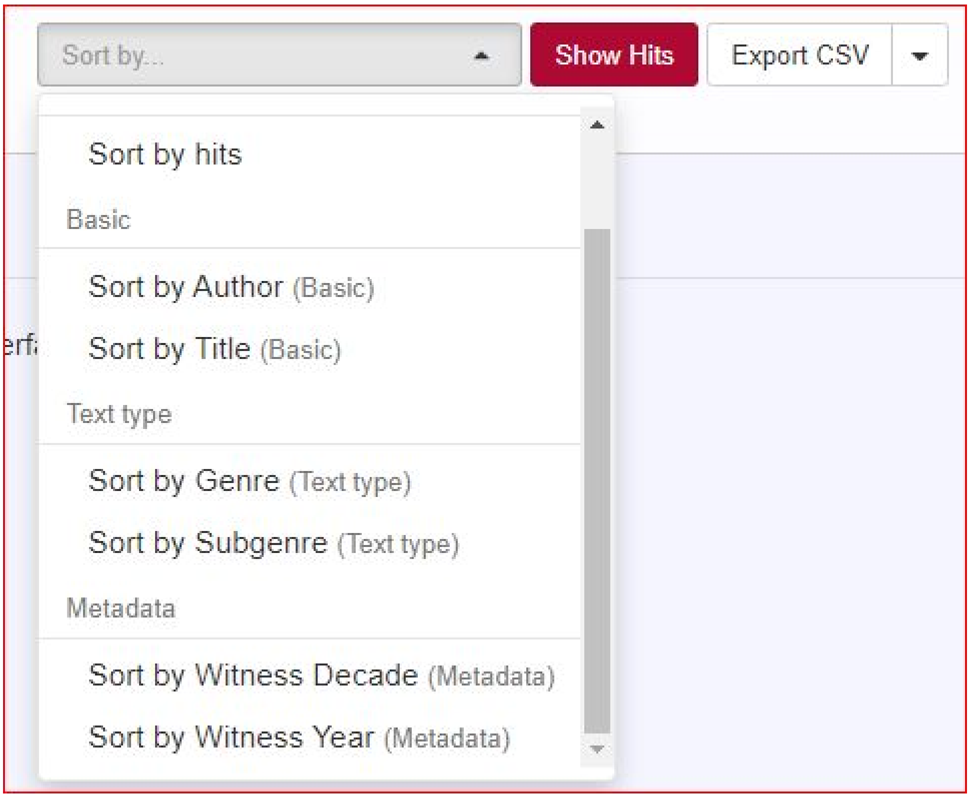
If more than twenty hits are found in a document, you can make them appear by clicking on Load more concordances; this button will appear right to the button View detailed concordances.

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 hits and by Author, Title, Genre, Subgenre, Witness Decade and Witness Year.


Sort by...
Show Hits
Export CSV
Sort by hits
Basic
Sort by Author (Basic)
Sort by Title (Basic)
Text type
Sort by Genre (Text type)
Sort by Subgenre (Text type)
Metadata
Sort by Witness Decade (Metadata)
Sort by Witness Year (Metadata)

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

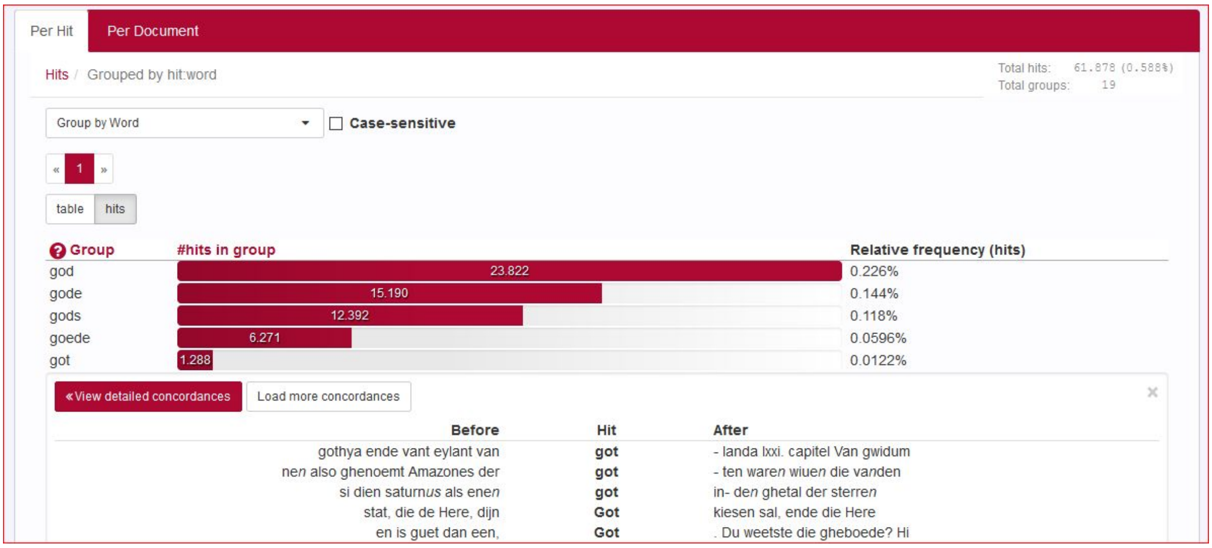
Grouping results

Results Per Document can be grouped by the metadata of the documents in which those hits occur (Author, Title, Genre, Subgenre, Witness Decade and Witness Year). Here, grouping is facilitated by the dropdown 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.



Total hits: 61.878 (0.588%)
Total groups: 19

Group	#hits in group	Relative frequency (hits)
god	23.822	0.226%
gode	15.190	0.144%
gods	12.392	0.118%
goede	6.271	0.0596%
got	1.288	0.0122%

Before	Hit	After
gothya ende vant eylant van	**got**	- landa lxxi. capitel Van gwidum
ne*n* also ghenoemt Amazones der	**got**	- ten ware*n* wiue*n* die va*n*den
si dien saturn*us* als ene*n*	**got**	in- de*n* ghetal der sterre*n*
stat, die de Here, dijn	**Got**	kiesen sal, ende die Here
en is guet dan een,	**Got**	. Du weetste die gheboede? Hi

Information about a document

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

Content

In the text hits from the current query will be highlighted bold. In the case of several hits only the current hit will also be underlined. You can go to the next (or previous) hit within the same document by pressing the Previous hit|Next hit button.

Metadata

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

Statistics

The Statistics tab shows some document statistics: the number of Tokens, the number of Types (unique word forms) and the Type/token ratio. It is possible to print or to download these statistics as an image or as a document 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

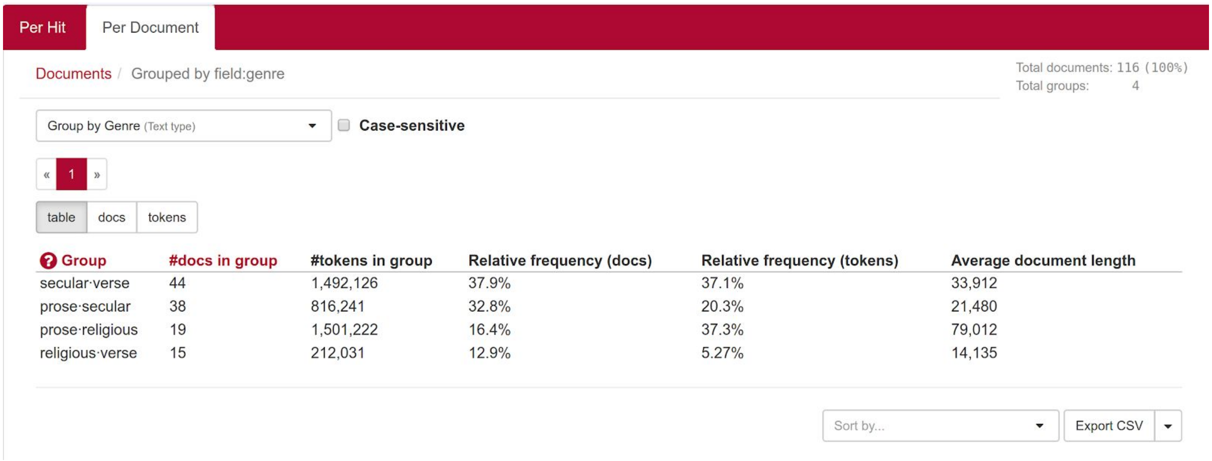
This subtab allows you to investigate the corpus. It consists of two drop-down menus to specify the grouping of the metadata and to specify the way the groups are to be shown.

A simple example: suppose we want to obtain information about the Text type of the documents from 1400 till 1500 within the *Corpus Middelnederlands*

- In the Group documents by metadata drop-down menu, choose Group by Genre (Text type)

- In Show groups as, select docs
- In the metadata search form (Filter search by), fill in at Witness Year *1400* and *1500*
- Press ‘Search’

This will be the result:



Group	#docs in group	#tokens in group	Relative frequency (docs)	Relative frequency (tokens)	Average document length
secular·verse	44	1,492,126	37.9%	37.1%	33,912
prose·secular	38	816,241	32.8%	20.3%	21,480
prose·religious	19	1,501,222	16.4%	37.3%	79,012
religious·verse	15	212,031	12.9%	5.27%	14,135

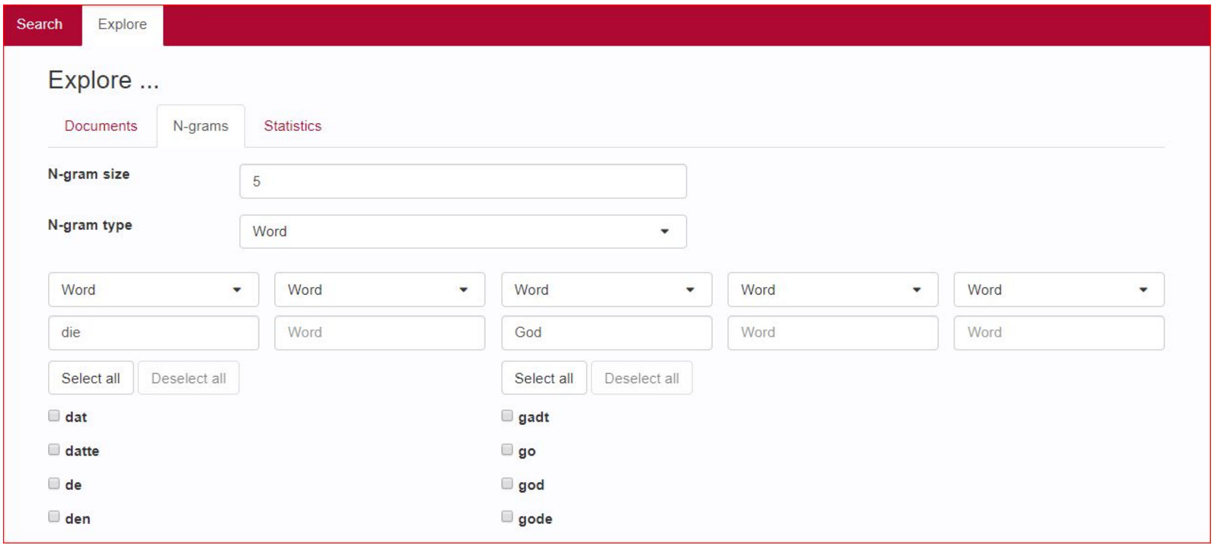
N-grams

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 the *Corpus Middelnederlands*, 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 the *Corpus Middelnederlands* for specific metadata.

Example



Search | Explore

Explore ...

Documents | N-grams | Statistics

N-gram size: 5

N-gram type: Word

Word	Word	Word	Word	Word
die	Word	God	Word	Word

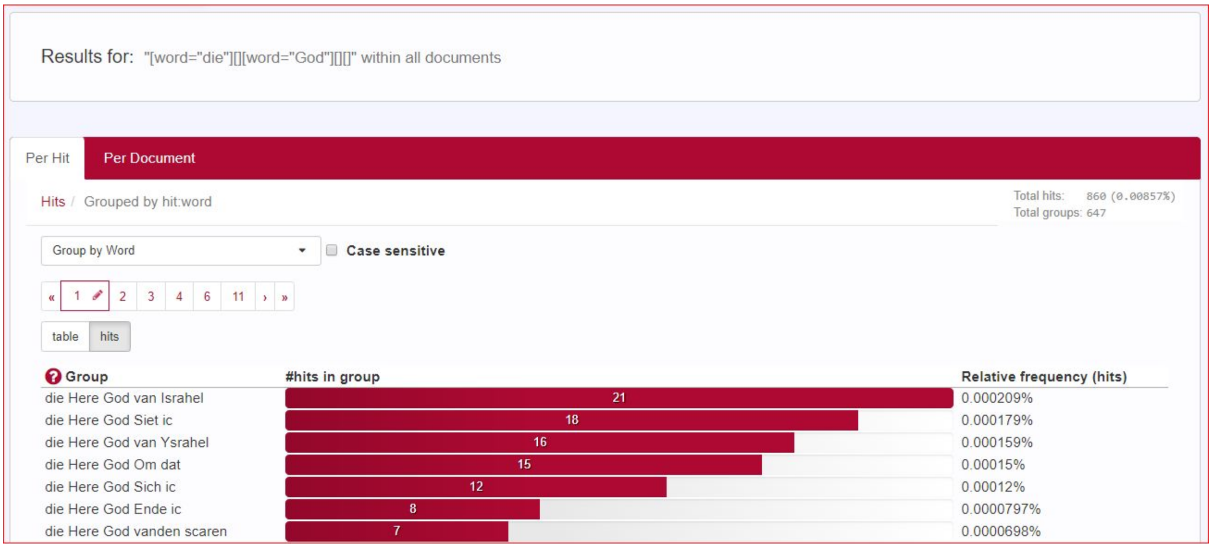
Select all | Deselect all

□ dat
□ datte
□ de
□ den

Select all | Deselect all

□ gadt
□ go
□ god
□ gode

This will result in a total of 860 hits:



Results for: "[word="die"][][word="God"][][]" within all documents

Per Hit | Per Document

Hits / Grouped by hit:word

Total hits: 860 (0.00857%)
Total groups: 647

Group by Word | □ Case sensitive

« 1 2 3 4 6 11 › »

table | hits

Group	#hits in group	Relative frequency (hits)
die Here God van Israhel	21	0.000209%
die Here God Siet ic	18	0.000179%
die Here God van Ysrahel	16	0.000159%
die Here God Om dat	15	0.00015%
die Here God Sich ic	12	0.00012%
die Here God Ende ic	8	0.0000797%
die Here God vanden scaren	7	0.0000698%

Statistics (frequency lists)

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

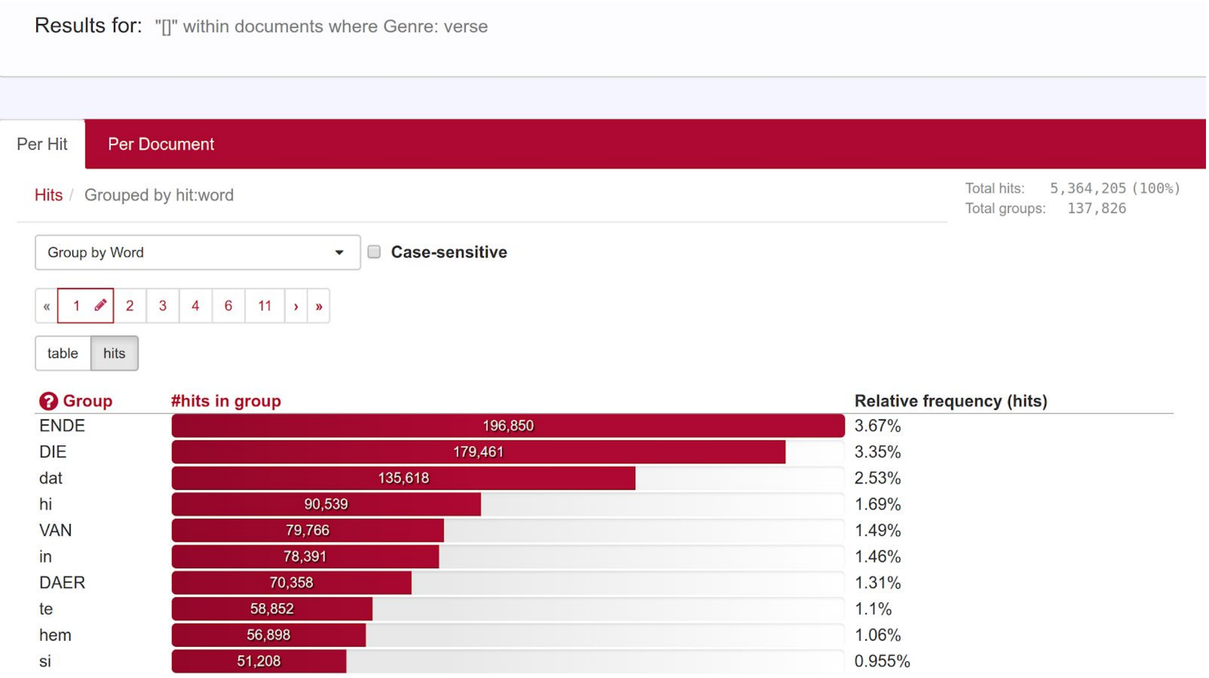
Options

- Frequency list type: the only possibility in the *Corpus Middelnederlands* is a list of Words (i.e. Word form)
- By using the Filter search by... you can create a subcorpus within the *Corpus Middelnederlands* for specific metadata

Example

It is possible to determine the use of the ten most frequently used Middle Dutch words in poetic texts and in prose texts in the *Corpus Middelnederlands* by searching for Frequency list type Word and by filtering search by Genre (*verse* and *prose*, respectively)

For verse this results in:

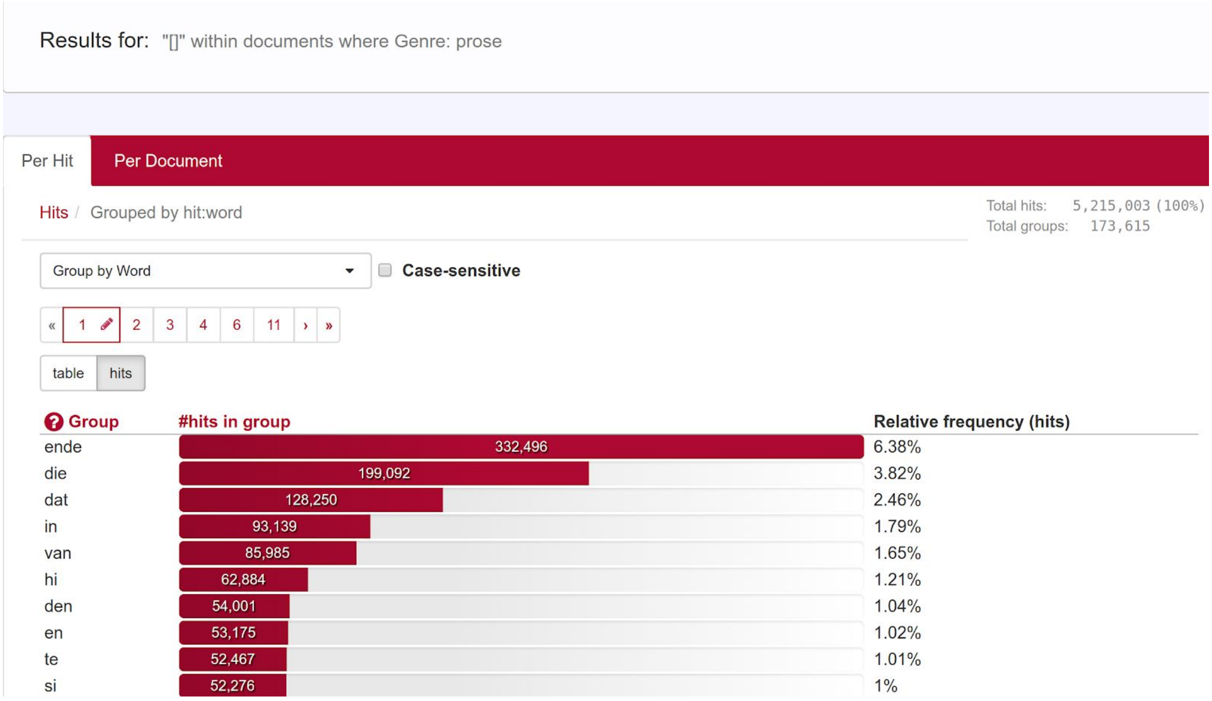


Results for: "[]" within documents where Genre: verse

Total hits: 5,364,205 (100%)
Total groups: 137,826

Group	#hits in group	Relative frequency (hits)
ENDE	196,850	3.67%
DIE	179,461	3.35%
dat	135,618	2.53%
hi	90,539	1.69%
VAN	79,766	1.49%
in	78,391	1.46%
DAER	70,358	1.31%
te	58,852	1.1%
hem	56,898	1.06%
si	51,208	0.955%

Whereas for prose this yields the following results:



Results for: "[]" within documents where Genre: prose

Total hits: 5,215,003 (100%)
Total groups: 173,615

Group	#hits in group	Relative frequency (hits)
ende	332,496	6.38%
die	199,092	3.82%
dat	128,250	2.46%
in	93,139	1.79%
van	85,985	1.65%
hi	62,884	1.21%
den	54,001	1.04%
en	53,175	1.02%
te	52,467	1.01%
si	52,276	1%

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 in 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 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-insensitively, 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 a 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 allow 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 that forward slash at the end of the tag. This way of referring to the element means "the whole element". Compare 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.
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.