

## 2.4.1 Query building support

The query tool ANNIS supports you in different way. We list them up here.

### Annotations per sub-corpus

Every sub-corpus in the list has more information about its annotations behind the [<i>](#) next to its name. This helps you find available annotations per (sub-)corpus.

### Query Builder: Word sequences and meta information

This is probably the easiest way to build your (complex) query step by step. You find the Query Builder at the top, to the right of the query field:

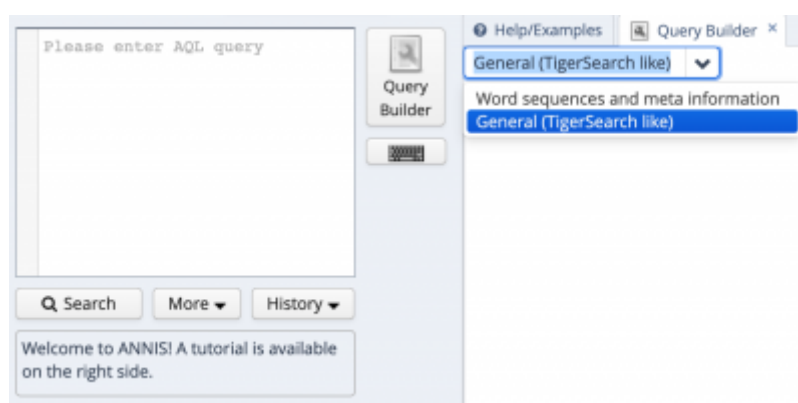


Figure 1: Starting the query builder

It comes in two flavors, as you can see. You can use the "General (TigerSearch like)" function, which is a bit more demanding or you can use the "Word sequences and meta information" function. To switch between these two functions, just press the arrow down.

Let us first look at the "Word sequences and meta information" function. This function helps you build queries per (sub-)corpus, so when you first open it, all sections have a button "initialize". When you press this button, the tool looks for available fields and options for the (sub-)corpus that you have selected at that moment. When you switch corpora, you have to re-initialize the tool by closing its tab and starting it anew.

The tool is set up into five different sections, which we will discuss in more detail in the following.

The first four sections describe the string that you want to query. They all can be found in the section *Linguistic sequence* in the query builder. Figure 2 shows the following query:

1. A token with the part of speech PRELS
2. This same token has to have the value *was* as an additional property
3. The token *was* as a PRELS has to be followed by a token with the value *me*
4. The token *me* has to follow immediately, i.e. with no other tokens in between the two tokens

As you can see in figure 1, properties that apply to the same token are listed below each other (1 and 2 in figure 1), while sequences of tokens are listed in a horizontal line (1 and 3 in figure 1).

01

X

.

▼

tok

X

ora

▼

+

☐ Regex

☐ Neg. search

06

03

X

Add ▼

tok

X

siamo

▼

+

☐ Regex

☐ Neg. search

+

▼

05

02

tt\_pos

X

NOM

▼

+

☐ Regex

☐ Neg. search

04

+

▼

Scope

age\_range

18-24

▼

☐ Regex

Close

Add ▼

Meta information

Add ▼

07

Toolbar

08

Create AQL Query

Clear the Query Builder

Refresh Query Builder

Figure 2: The query "was (part of speech: PRELS) me" built with word sequence builder.

X

▼

X

Add ▼

tt\_lem

X

was

▼

+

☐ Regex

☐ Neg. search

+

▼

▼

.2 [is preceding with one token in between]

.1,2 [is directly preceding or with one token in between]

.\* [is indirectly preceding]

. [is directly preceding]

☒ Regex

☐ Neg. search

+

▼

Figure 3: The relationship between the tokens

Once you have defined these attributes, you can click on *Create AQL Query* below and your query is created in the query field to the left of the Query Builder as you can see in figure 2.

In the fields below you can add additional annotations from the layers messages and chats. These will be added to the query you defined above. As you can see in figure 4, the resulting query is rather complex and thus justifies the use of the query builder.

The screenshot shows the Query Builder interface. On the left, a text field contains the following AQL query:

```
tt_pos="PRELS" & tok="was" &
tok=/me/
& age_range = "18-24"
& #1 = #2
& #2 = #3
& #4_i_#1
& #4_i_#3
```

Below the query field are buttons for "Search", "More", and "History". A message states: "Valid query, click on 'Search' to start searching." Below this is a "Corpus List" section with a "Search Options" button. A "Visible:" dropdown is set to "All". A table lists corpus data:

Name	Texts	Tokens		
WUS	617	6,005,326		
WUS_ARGDROP	16	54,629		
WUS_ARGDROP_FRA	10	37,714		
WUS_ARGDROP_ITA	3	7,707		
WUS_ARGDROP_ROH_SR	3	9,208		
WUS_DEU	21	297,896		

On the right, the "Query Builder" tab is active. It shows a "Linguistic sequence" builder with two columns. The left column contains "tt\_pos" with a dropdown set to "PRELS" and "tok" with a dropdown set to "was". The right column contains "tok" with a dropdown set to "me". Below each column are checkboxes for "Regex" and "Neg. search". At the bottom, a "Scope" section shows a "Change" dropdown, a field for "age\_range" set to "18-24", a "Regex" checkbox, and a "Close" button.

Figure 4: complex query from different layers.

## Query Builder. General (TigerSearch like)

This type of query builder focuses on relationships between entities. It basically works as follows: you select an entity (ie. the value of the token) and then select another entity (eg. the part of speech that this tokens is to have) and then link the two via a relationship. Figure 5 shows how this example is constructed.

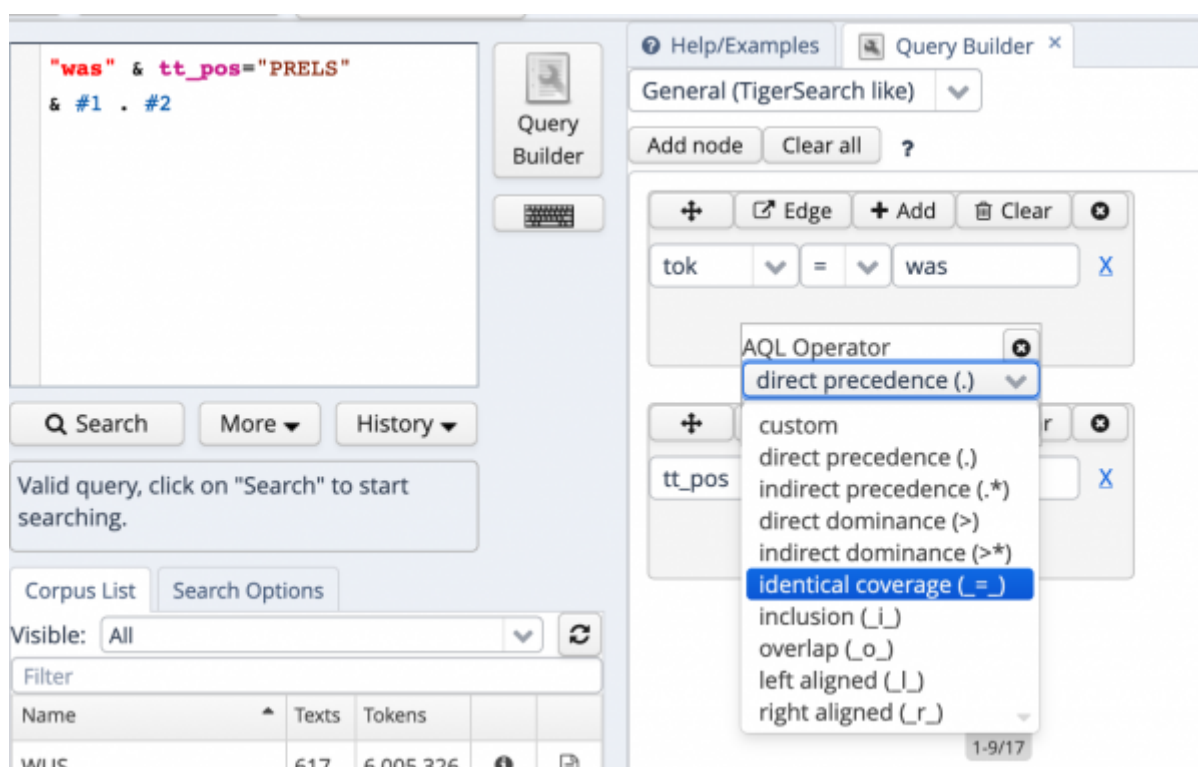


Figure 5: Building a query by selecting a relationship between entities

To work with this query builder, you first select all the nodes ("Add node") required for your query (i.e. the value of the token and the PoS in the example). You then add an edge, i.e. a relationship between these two entities (e.g. different types of precedence, inclusion, overlap etc.) and dock the selected entity to the other one.

**NB: you cannot have isolated identities, they all have to be linked via edges.**

From:

<https://whatsup.linguistik.uzh.ch/> -

Permanent link:

[https://whatsup.linguistik.uzh.ch/02\\_browsing/04\\_queries/01\\_support?rev=1587541406](https://whatsup.linguistik.uzh.ch/02_browsing/04_queries/01_support?rev=1587541406)

Last update: **2022/06/27 09:21**

