

## 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:

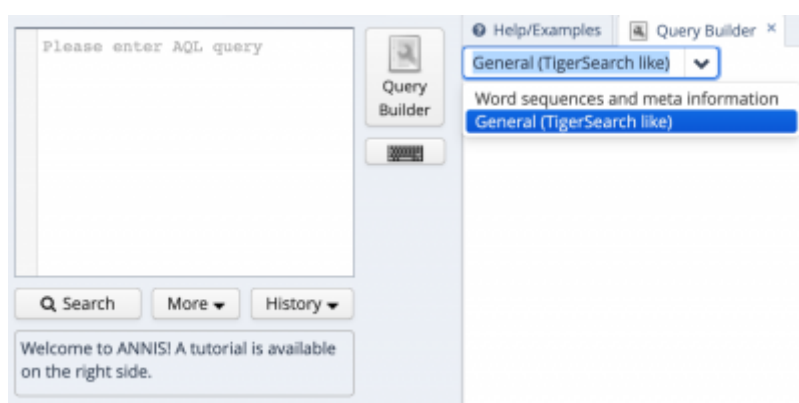


Figure 1: Starting the query builder

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*. 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*
3. The token *was* as a PRELS has to be followed by a token *me*
4. The token *me* as to follow directly, i.e. with no other tokens in between the two tokens

Sequences of tokens are listed horizontally, different annotations of a specific token are listed vertically. In figure 2, the top left attribute is `tt_pos="PRELS"`. Below that, thus, describing the same token is the attribute `tok="was"`. To the right of these two attributes is the attribute for the following token: `tok="me"`

**Linguistic sequence**

01

X

tok

X

ora

+

Regex

Neg. search

06

X

tok

X

siamo

+

Regex

Neg. search

03

05

Add

02

tt\_pos

X

NOM

+

Regex

Neg. search

04

+

age\_range

18-24

+

Regex

Close

Add

Add

07

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.

In the horizontal, additional tokens and their conditions can be added. In the example above, the token *me* has to follow the first token. In between those two tokens, you can select their relationship, i.e. whether the second token is to precede with one token in between or whether it is to precede directly or with one token in between or whether it is to precede directly or indirectly as you can see in the following example:

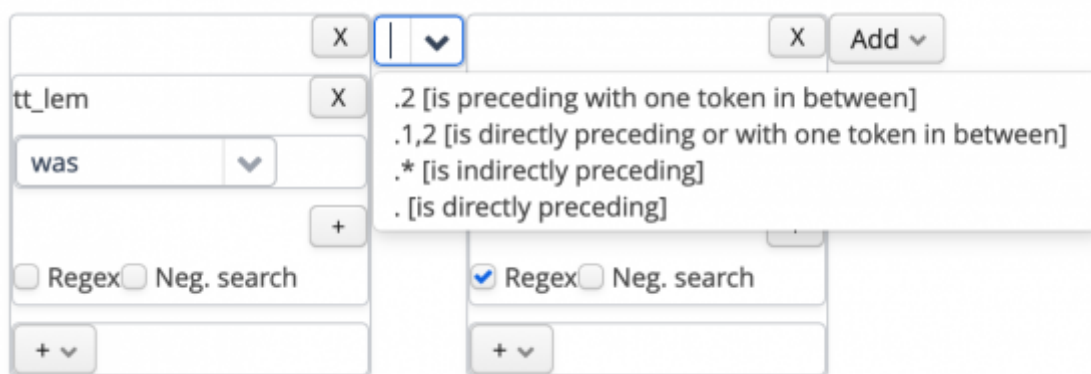


Figure 3: The relationship between the tokens

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.

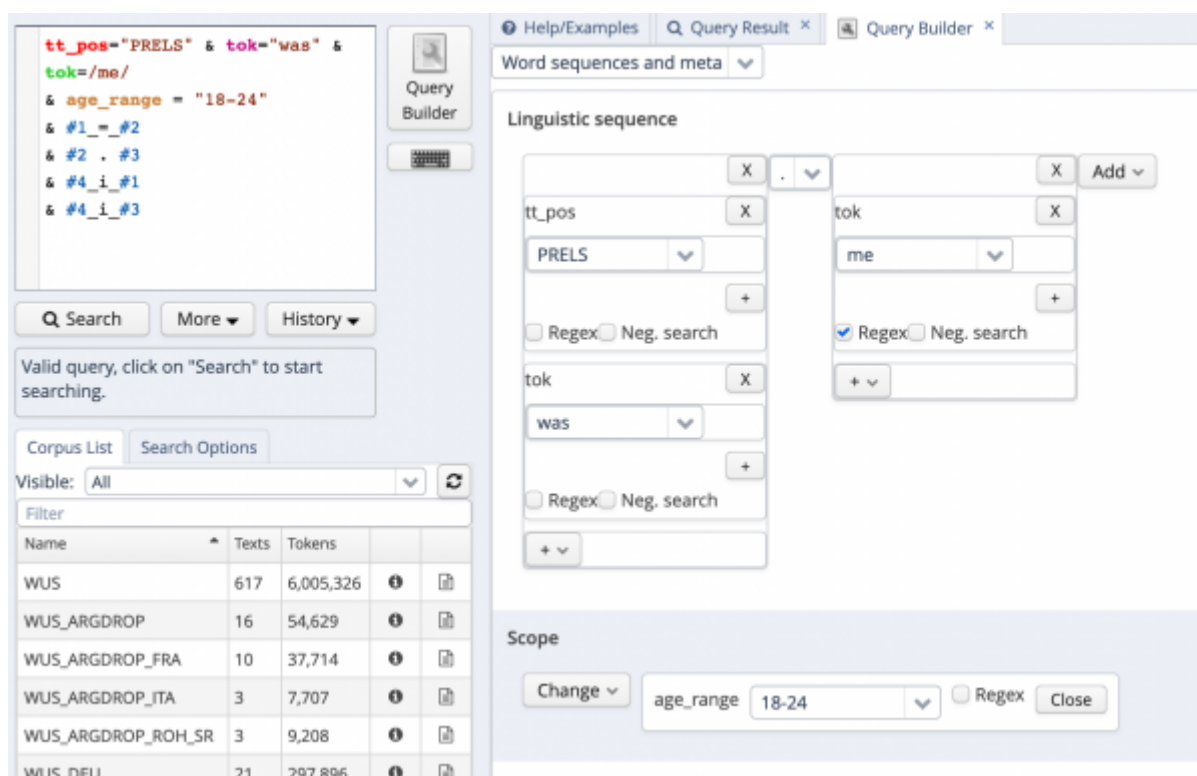


Figure 4: complex query from different layers.

## 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.

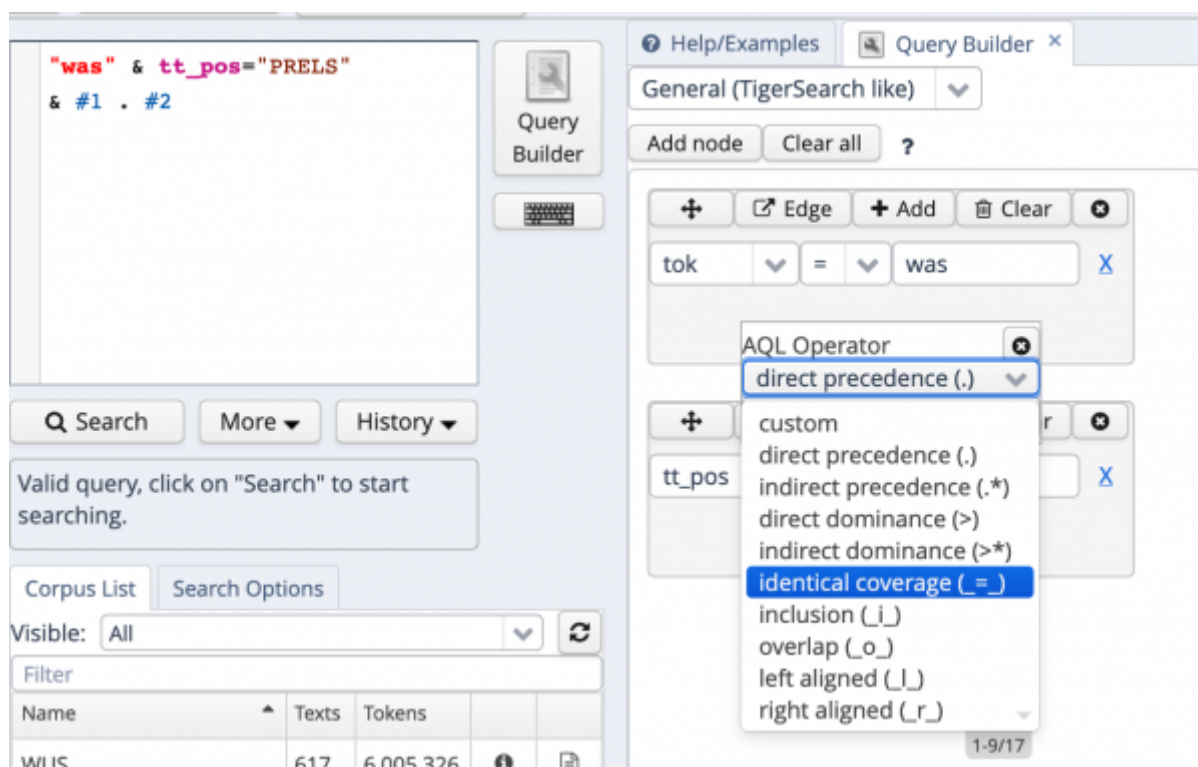


Figure 5: Building a query by selecting a relationship between entities

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=1587541055](https://whatsup.linguistik.uzh.ch/02_browsing/04_queries/01_support?rev=1587541055)

Last update: 2022/06/27 09:21

