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:

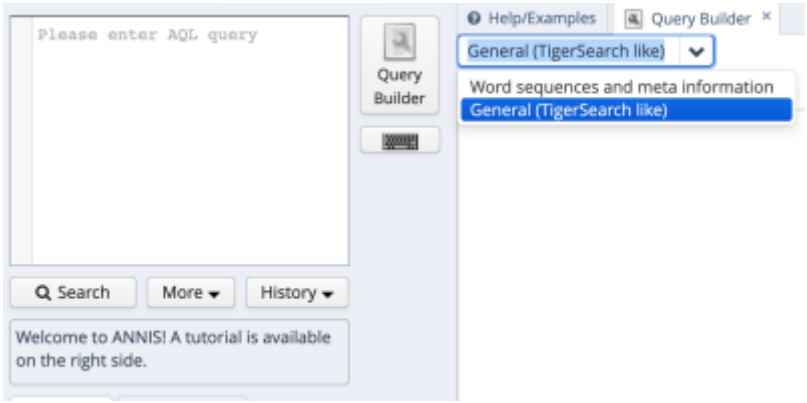


Figure 1: Starting the query builder

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 different sections, which we will discuss in more detail in the following. tok="ora" & tt_pos="NOM" & tok="siamo" & age_range = "18-24" & #1_=_#2 & #2 . #3 & #4_i_#1 & #4_i_#3

Linguistic sequence

In the first section you define the sequence of tokens and their attributes that you want to query for. Figure 2 represents the following query:

1. A first token with the value *ora*
2. The same token has to have the part of speech annotation `tt_pos="NOM"`
3. A second token with the value *siamo* has to followed follow the first one.
4. The second token has to follow the first one directly, i.e. with no other tokens in between.

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

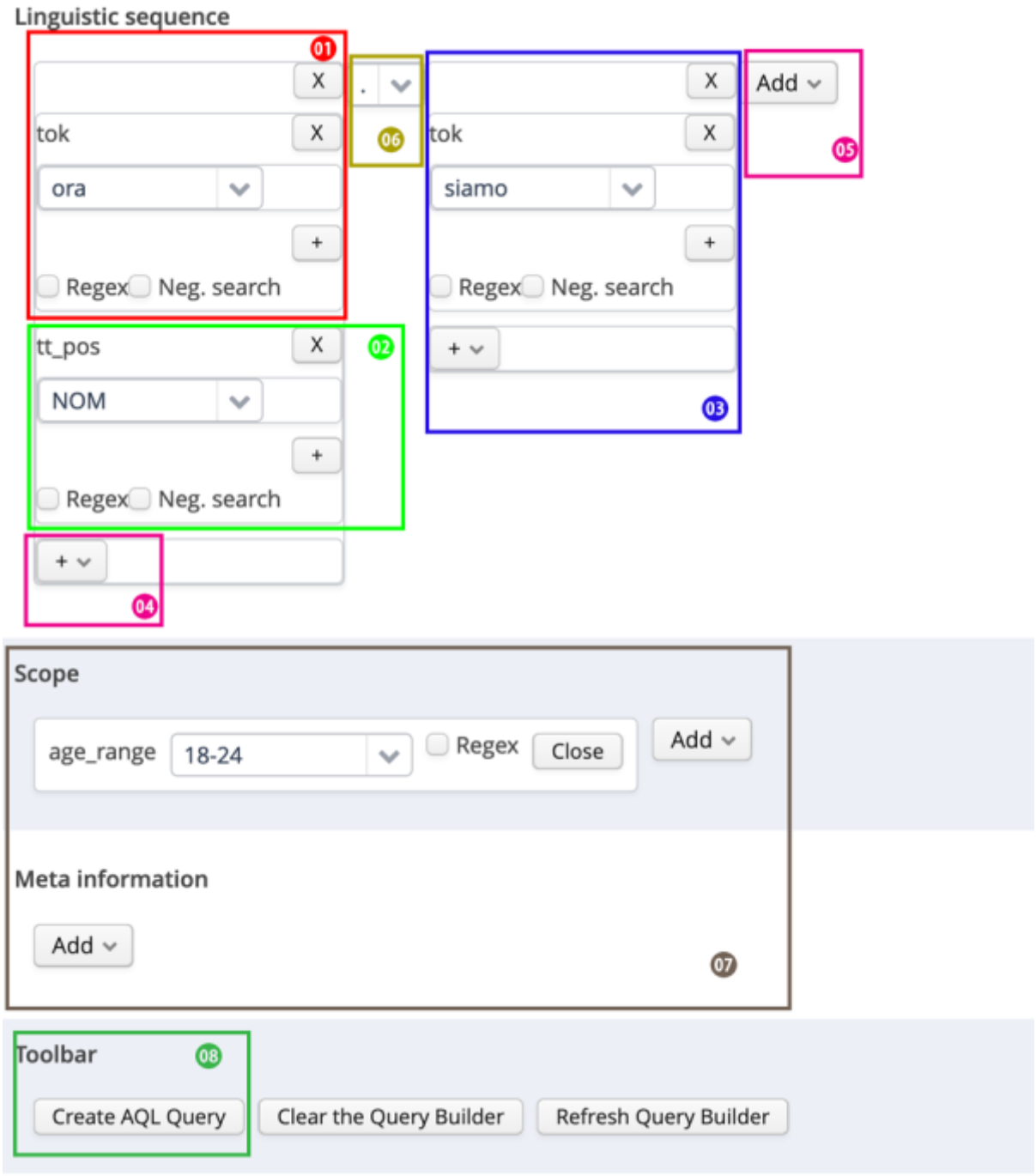


Figure 2: The query "was (part of speech: PRELS) me" built with word sequence builder.

However, there are different way in which tokens can follow each other in a sequence, since a token can follow another directly (with no other token in between) or indirectly with one or several other tokens in between. This option is represented by the rectangle 4 in figure 2 and explained in more detail in figure 3, where `.2` stands for a token that is preceding another one with one token in

between, `.1,2` represents a token that is directly preceding or with one token in between, `.*` represents a token that is indirectly preceding, this with any number of tokens in between and `.` represents a token that is directly preceding.

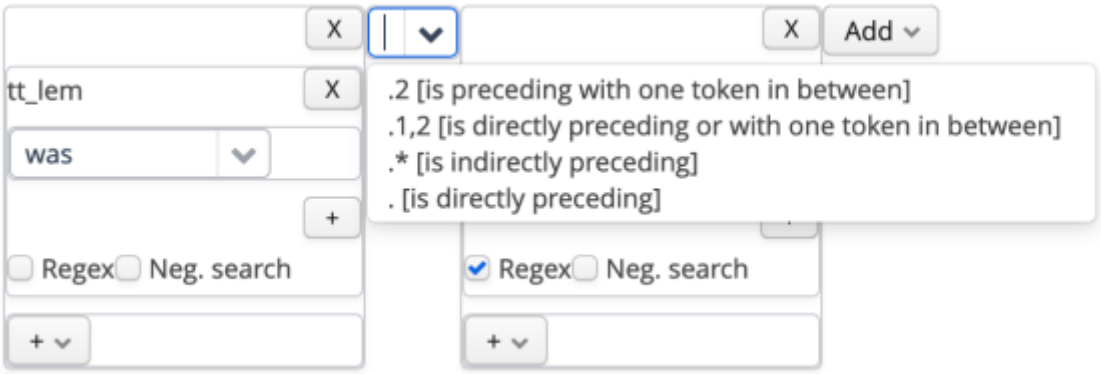


Figure 3: Describing different options for the sequence of tokens

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 you 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.

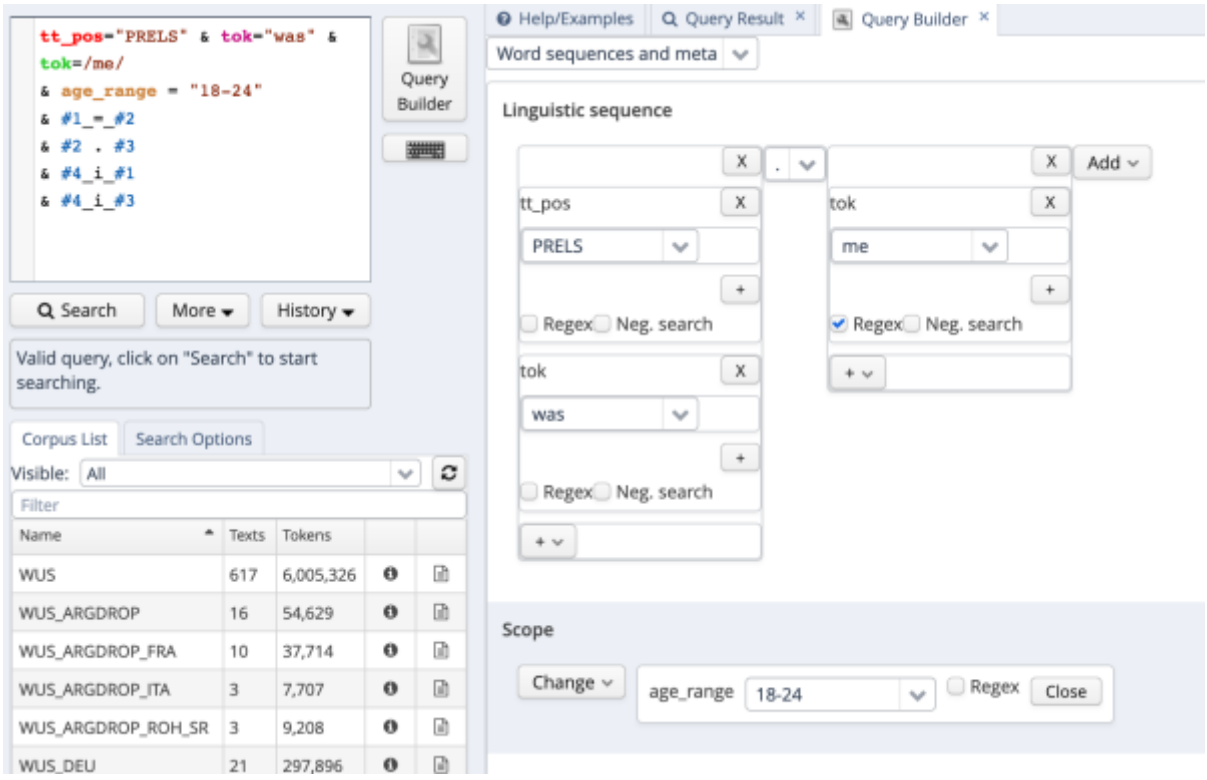


Figure 4: complex query from different layers.

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.

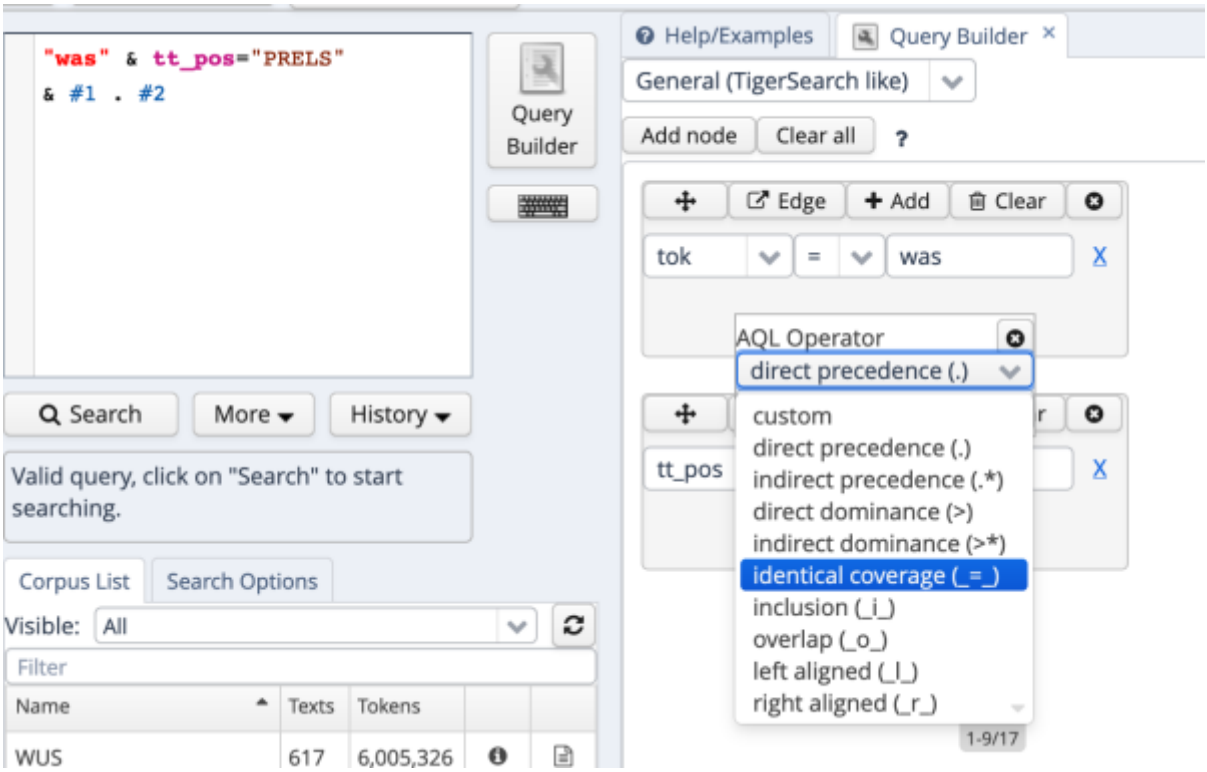


Figure 5: Building a query by selecting a relationship between entities

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=1587545373

Last update: **2022/06/27 07:21**