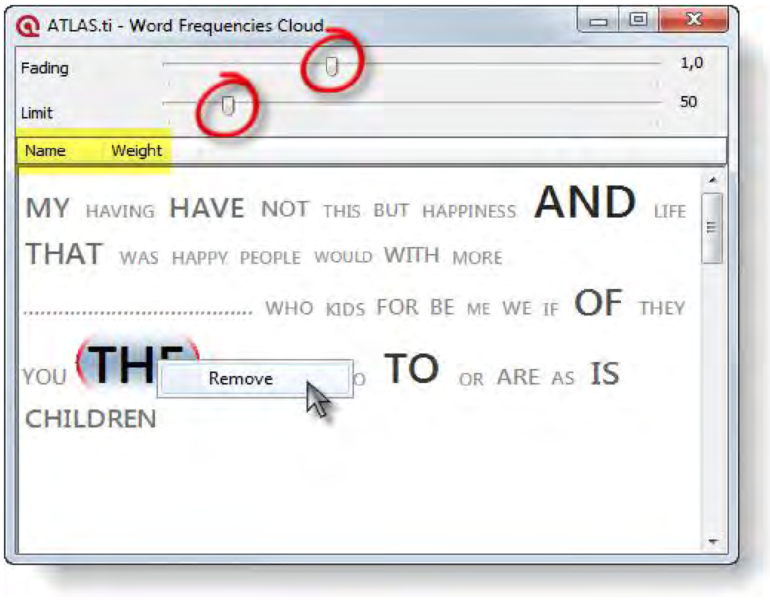

ATLAS.ti - Word Frequencies Cloud
Fading
1,0
Limit
50
Name
Weight
MY HAVING HAVE NOT THIS BUT HAPPINESS AND LIFE
THAT WAS HAPPY PEOPLE WOULD WITH MORE
WHO KIDS FOR BE ME WE IF OF THEY
YOU THE TO OR ARE AS IS
CHILDREN
Remove

Figure 209: Word Frequency Cloud

Fading fades out the less frequent words.

With the slider **Limit**, you can select how often a word should occur to be displayed in the list.

Right-click on a word to remove it from the view and to add it to the **stop list**.

Switch the sort order by name or weight by clicking on either the name or the weight tab.

Query Tool

You need the Query Tool for queries using more than a single code.

The **Query Tool** is used for retrieving quotations using the codes they were associated with during the process of coding. This is different from a *text* search: To search for occurrences of text that match a specified pattern or string, you have to use the search function or the Object Crawler (see "Text Search" on page 205 and "The Object Crawler" on page 361).

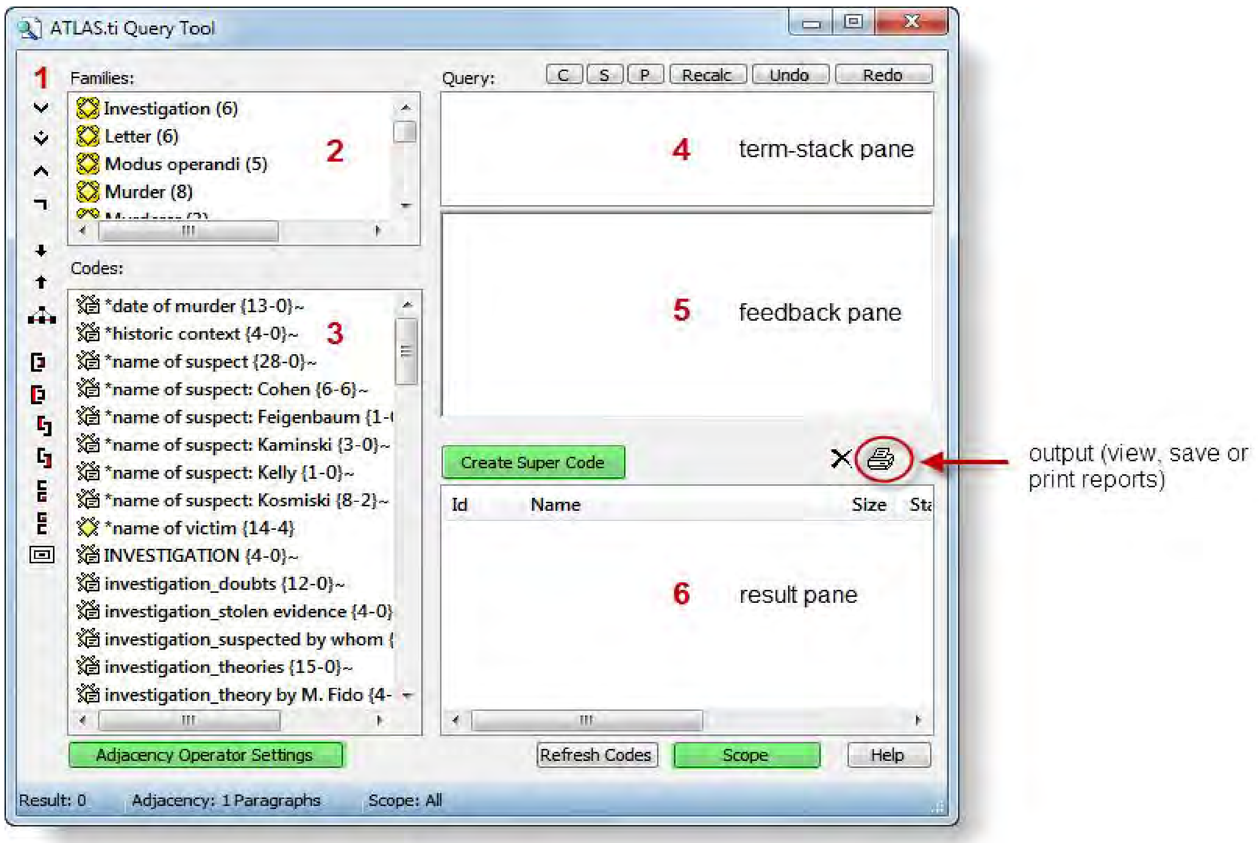
The simplest retrieval of this kind ("search for quotations with codes") is what you frequently do with the Code Manager: double-clicking on a code retrieves all its quotations. This may already be regarded as a query, although it is a simple one. The Query Tool is more complex in that it can be used to create and process queries that include combinations of codes.

A **query** is a search expression built from operands (codes and code families) and operators (e. g. NOT, AND, OR, etc.) that define the conditions that a quotation must meet to be retrieved (e. g., all quotations coded with both codes A and B).

By selecting codes or code families and operators, a query can be built incrementally which is instantaneously evaluated and displayed as a list of quotations. This incremental building of complex search queries gives you an exploratory approach toward even the most complex queries.

The Query Tool Window

The Query Tool is launched by clicking the Query Tool button (see left), or by choosing **ANALYSIS / QUERY TOOL** from the HU Editor's main menu.



Investigation (6)
Letter (6)
Modus operandi (5)
Murder (8)

*date of murder {13-0}~
*historic context {4-0}~
*name of suspect {28-0}~
*name of suspect: Cohen {6-6}~
*name of suspect: Feigenbaum {1-
*name of suspect: Kaminski {3-0}~
*name of suspect: Kelly {1-0}~
*name of suspect: Kosmiski {8-2}~
*name of victim {14-4}
INVESTIGATION {4-0}~
investigation_doubts {12-0}~
investigation_stolen evidence {4-0}
investigation_suspected by whom {
investigation_theories {15-0}~
investigation_theory by M. Fido {4-

Result: 0 Adjacency: 1 Paragraphs Scope: All

Figure 210: The Query Tool window

The Query Tool has the following main components:

[1] The **operator toolbar**, located near the left margin of the window.

[2] The **code-family pane** in the upper left lists code-families to be used in queries.

[3] The **codes pane** below the code-family pane contains all current codes (set filters do apply).

[4] The **term-stack pane** in the upper right displays the stack of all expressions entered in the current query. If more than one entry is visible, there are arguments still waiting to be used in the query. The topmost entry is the current query.

[5] The current query is also displayed in the **feedback pane** directly below the term-stack pane. Here a different notation is used, one that uses parentheses and resembles the calculator style of entering queries.

[6] The result of the query is displayed in the **results list** located in the lower right of the window.

Above the term-stack pane are several buttons for manipulating the stack: swapping (**S**) or duplicating terms (**P**), clearing the stack (**C**), etc.

Close to the results list are two buttons for removing unwanted hits and creating a report.

In figure 204 you see three other buttons highlighted in green. A super code is a saved query (see Super Codes on page 267 for further detail). You need the adjacency operator settings if you want to search for codes near to each other (see Adjacency Operators on page 259). Behind the Scope button you find another important feature. When you click on the Scope button, a second window opens showing the PD families (see page 218) you have created. These are often variables like age, gender, education, profession, location, time intervals etc. The scope function allows you to combine a code query with variables. For instance you can ask for all quotations where you have applied code A and code B, but only for females between the ages of 21 to 30.

Operands

Basic Operands

Two sorts of *basic or atomic operands* may be used in a query: **Codes** and **code families**.

A code represents a set of quotations, while a code family yields the quotations of all the codes that its members have. In other words, a family is interpreted as its member codes connected by the Boolean operator OR. Selecting a code family F1 which contains five codes C1-C5 is equivalent to the query: "C1 OR C2 OR C3 OR C4 OR C5".

Complex Operands

"Operand" does not only apply to basic descriptors. An operand can be **any expression** that itself is used as an argument. An expression "A AND B" may be used in a more complex query as an operand: "NOT(A AND B)", "(A AND B) OR (C AND NOT D)", etc.

All types of operands can be freely mixed in a query using any of the operators described below.

Operators

Three sets of operators are available. They are located within the toolbar at the left edge of the Query Tool.

Boolean operators allow combinations of keywords according to set operations. They are the most common operators used in information retrieval systems.

Semantic operators exploit the network structures that were built from the codes.

Proximity operators are used to analyze the spatial relations (e. g., distance, embeddedness, overlapping, co-occurrence) between coded data segments.

You can display a short help message for each operator by right clicking on its corresponding button in the toolbar.

Boolean Operators

Four Boolean operators are available with the Query Tool: OR, XOR, AND, and NOT.

OR, XOR, and AND are *binary operators* which need exactly two operands as input. NOT needs only one operand. However, as stated above, the operands themselves may be of arbitrary complexity. Codes, code families, or arbitrary expressions can be used as operands: "(A OR B) AND (NOT C AND D)".

⌄	**OR** The OR operator retrieves all data segments (i. e., quotations) that are coded with any of the codes used in the expression. Example: "All quotations coded with 'Earth' OR 'Fire'". An example of a more complex formulation based on a combination of queries is: "All quotations coded with 'Earth' OR coded by both 'Fire AND Water'."
⌄	XOR The OR operator does not really match the everyday usage of "OR." Its meaning is "At least one of…," including the case where ALL conditions match. The XOR operator, in contrast, asks that "EXACTLY one of…" the conditions must meet. It translates into everyday "either-or." Example: "All quotations coded with EITHER 'Earth' OR 'Fire' (but not with both)."
⌃	**AND** The AND operator finds quotations that match ALL the conditions specified in the query. This means you have applied two or more codes to the same quotation. Example: "All quotations coded with 'Earth' AND 'Fire'." The AND operator is very selective and often produces an empty result set. "Precision" of this operator is high, but the "recall" is rather low. It produces best results when

	combined with less restrictive operators or when the overall number of the available text segments is large.
¬	**NOT** The NOT operator tests for the absence of a condition. Technically, it subtracts the findings of the non-negated term from all data segments available. Given 120 quotations in the HU and 12 quotations assigned to code "Fire," the query "NOT Fire" retrieves 108 quotations - those which are *not* coded with "Fire." Of course, the operator can be used with an arbitrary expression as in the argument "NOT (Earth OR Fire)" which is the equivalent of "neither Earth nor Fire."

The OR operator has the potential to generate a HUGE number of hits. It has high "recall" (a lot is retrieved), but low "precision" (many of the retrieved quotations may not make sense).

Venn diagrams are descriptive schemes for illustrating the different set operations associated with Boolean operators.

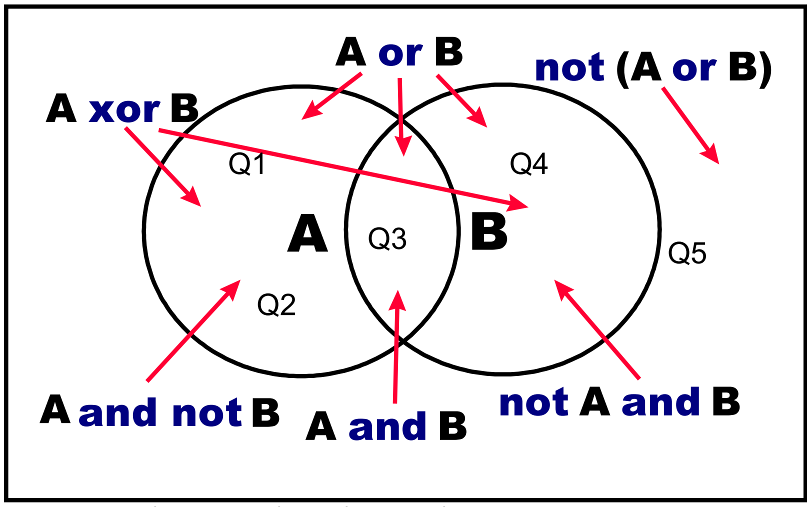


Figure 211: Boolean queries depicted as Venn diagrams

The rectangle encloses the set of all retrievable quotations, e. g. the "document universe." The two circles represent two codes A and B. Q1 to Q5 are quotations coded with A, B, or none (Q5).

Semantic Operators

The Semantic Operator buttons

The operators in this section exploit connected codes resulting from previous theory-building work. While Boolean-based queries are *extensional* and simply enumerate the elements of combined sets (e. g., LOVE or KINDNESS), semantic operators are *intentional*, as they already capture some meaning expressed in appropriately linked concepts (e. g., SUB(POSITIVE ATTITUDES)).

↓	SUB The SUB (or DOWN) operator traverses the network from higher to lower concepts, collecting all quotations from any of the sub codes. Only "transitive" relations between the codes are processed (see “Relations“on page 305; all others are types ignored. When building a terminology from your codes, use the ISA relation for sub-term links. Example: "All quotations coded with Magic or any (immediate or indirect) sub-term of Magic". Like the OR operator in the set of Boolean operators, the SUB may produce large result sets. However, unlike the OR operator, because you make use of a theory using SUB, the "precision" is much better (i. e., you get only what you expect). Of course, if your network contains dubious connections ("computer ISA intelligent entity"), the quality of your retrieval will decline.
↑	**UP** The UP operator looks at all directly linked codes and their quotations on the next higher level.. Unlike the SUB operator, it does not recursively traverse the structure. Only the next level is considered.
⋔	**SIBlings** The SIBlings operator finds all quotations that are connected to the selected code or any other descendants of its parents. Example: "All quotations coded with Love or any other Positive Attitude (here: kindness).”

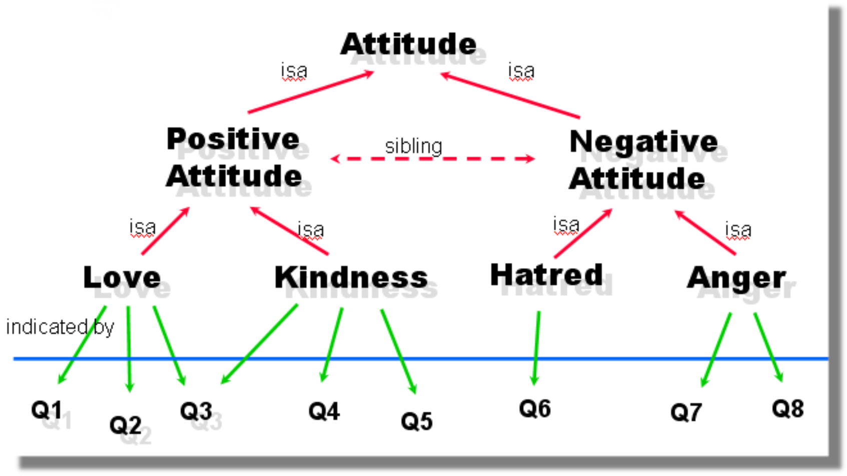


Figure 212: A hierarchy of concepts suitable for semantic retrieval

With such a network of codes the following queries would make sense (Q1 to Q8 = quotations):

SUB (Positive Attitude) => {Q1, Q2, Q3, Q4, Q5}

SUB (Negative Attitude} => {Q6, Q7, Q8}

SUB (Attitude) => {Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8}.

Proximity Operators

The Proximity Operator buttons

Proximity describes the spatial relation between quotations. Quotations can be embedded in one another, one may follow another, etc. The operators in this section exploit these relationships. They require two operands as their arguments. They differ from the other operators in one important aspect: proximity operators are *non-commutative*. This property makes their usage a little more difficult to learn.

Non-commutativity requires a certain input *sequence* for the operands. While "A OR B" is equal to "B OR A", this does not hold for any of the proximity operators: "A FOLLOWS B" is not equal to "B FOLLOWS A". When building a query, always enter the expressions in the order in which they appear in their natural language manifestation.

Because of non-commutativity, every proximity operator comes in two versions.

Another important characteristic for these operators is the specification of the operand for which you want the quotations retrieved. "A WITHIN B" specifies the constraint, but you must also specify if you want the quotations for the As or the Bs. This is done implicitly by the sequence. The code (or term) that is entered first is the one in which you are interested. If B's quotations are requested, you have to enter "B ENCLOSES A" using the query language described below.

EMBEDDING OPERATORS

The *embedding* operators describe quotations that are contained in one another and that are coded with certain codes.

	WITHIN A WITHIN B retrieves all quotations coded with A that are contained within data segments coded with B.
	ENCLOSES A ENCLOSES B retrieves all quotations coded with A that contain quotations coded with B.

OVERLAP OPERATORS

The *overlap* operators describe quotations that overlap one another.

	OVERLAPPED_BY A OVERLAPPED_BY B retrieves all quotations coded with A that are overlapped by quotations coded with B.
	OVERLAPS A OVERLAPS B retrieves all quotations coded with A that overlap quotations coded with B.

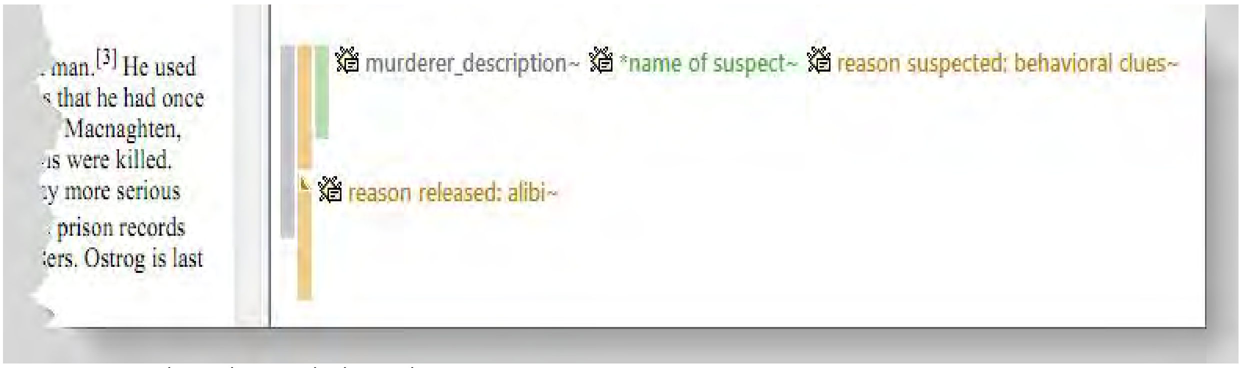


Figure 213: Visualizing the spatial relations between segments

If you want to retrieve all segments for “reason suspected: behavioral clues” related to the code “murderer_description”, you would need to click:

“reason suspected: behavioral clues”, “murderer_description”, WITHIN

If you enter: “murderer_description”,“reason suspected: behavioral clues”, WITHIN, the query tool would not deliver any results for the data segments shown in figure 213.

If you enter , “murderer_description”, “reason suspected: behavioral clues” ENCLOSES, then the query tool retrieves the larger segment “murderer_description” that does contain the behavioral clue. But then you need to read more than you need if you are only interested in the behavioral clues.

If you are interested in the code “alibi” as reasons for having released a suspect in relation to the description of the murderer, then you click: “reason released: alibi”, “murderer_description” overlaps.

If you want to find out about the name of the suspect related to behavioral clues, you enter “name of suspect”, “reason suspected: behavioral clues” WITHIN.

From the above example we have learned that a) you begin with the codes whose content you are most interested in, and b) you first enter the codes and then you select an operator. See is explained in more detail below in the section “The Query Language“.

Often when interested in the relation between two or more codes, you don't really care whether something overlaps or is overlapped by, or is within or encloses. It this is the case, you simply use the Co-occuRE operator, which is a combination of WITHIN, ENCLOSES, OVERLAPS, OVERLAPPED BY and AND.

Nonetheless these very specific operators are also very useful for specific type of data. Think of video data where it might be important wether action A was already going on before action B started or vice versa. Or if you have coded longer section in your data like biographical time periods in a persons life and then did some more fine-grained coding within these time periods. Then the WITHIN operator comes in handy. The same applies when working with pre-coded survey data. ATLAS.ti pre-codes your questions, then you do some further coding. This enables you to ask for instance for all quotations coded with “topic x” WITHIN question 5.

ADJACENCY OPERATORS

The *distance* operators describe a sequence of disjoint quotations. The maximum distance may be specified. Possible base units are characters and paragraphs for text, milliseconds for audio files, frames for video data and pixels for images.

	FOLLOWS A FOLLOWS B retrieves all quotations coded with A that follow quotations coded with B.
	PRECEDES A PRECEDES B retrieves all quotations coded with A followed by quotations coded with B.

ADJACENCY SETTINGS

To set the distance, click on the “Adjacency Operator Settings” button. Then select a base unit and specify the maximum distance.

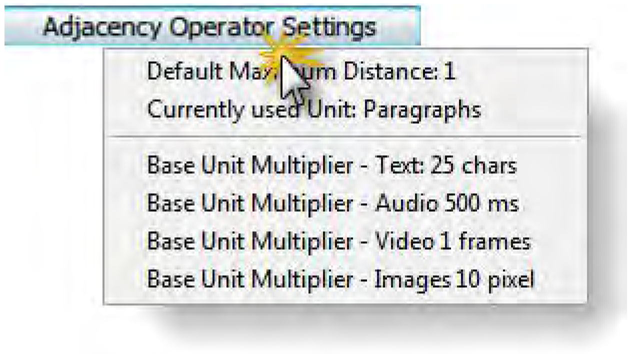

Adjacency Operator Settings
Default Maximum Distance: 1
Currently used Unit: Paragraphs
Base Unit Multiplier - Text: 25 chars
Base Unit Multiplier - Audio 500 ms
Base Unit Multiplier - Video 1 frames
Base Unit Multiplier - Images 10 pixel

Figure 214: Adjacency operator settings

THE CO-OCCURRENCE OPERATOR

Co-occurrence is essentially a short-cut for a combination of all the basic proximity operators except **FOLLOWS** and **PRECEDES**.

A CO-OCCURRING WITH B: Find all quotations that co-occur with B (in whatever way).

The procedures used for calculating co-occurrence for two codes is also used in the Network Editor when importing co-occurring codes into a network view. See "Import Co-occurring Codes" on page 328.

The Query Language

Queries are built step-by-step from operands and operators using the principle of **Reversed Polish Notation (RPN)**. This sounds complicated, but it is actually quite easy. See for example: http://en.wikipedia.org/wiki/Reverse_Polish_notation

RPN, invented by Polish mathematician Lukasiewicz, does not require parentheses to control the priority of operators, nor does it require any other characters like commas, periods, etc. Every click produces a meaningful result and it is impossible to create syntactically wrong queries.

Operands First, Operators Next

The most important point to understand about RPN is the *order* in which operands and operators of a search expression are entered. Using RPN, operands (codes, code families) are entered first, followed by one or more operators. This is an unusual method for most of us who are familiar with notations where operators are placed *between* the operands, as in "3 + 5". Most calculators use this type of notation, also called "infix" notation.

Infix notation: good for reading. Postfix notation: good for clicking.

Two aspects must be distinguished: how we read expressions and how we formulate them with a "point and click" language. The infix notation is usually easier to read, but the "postfix" notation is far easier to use when creating queries using mouse-controlled direct manipulation user interfaces like Windows.

An Arithmetic Example

Here are some simple arithmetic examples using an RPN calculator:

Arithmetic expression	**RPN expression**
Example 1: 3 + 4	3 4 +
Example 2: 3 + (4 * 5)	4 5 *3 +
Example 3: (3 + 4) * 5	3 4 + 5 *

No parentheses are needed in expressions using RPN notation. The precedence of the operators is controlled solely by the order in which operands and operators are entered.

Creating A Query With The Query Tool

The result of **any** query is a set of quotations.

The retrieval of quotations with the Query Tool differs from the arithmetic example above by the result in which we are interested.

We are really not interested in the operands (codes, code families) themselves, but in the set of quotations that is the result of *evaluating* an operand. By formulating a query "A OR B," this is what we really mean: "*Quotations coded with code A OR quotations coded with B.*" Therefore, entering the operand code "X" displays the quotation names which were coded with "X" in the results list. Next, you can either view the resulting quotations in context within the primary document, or generate a report that contains the full lenght quotations with or without their comments.

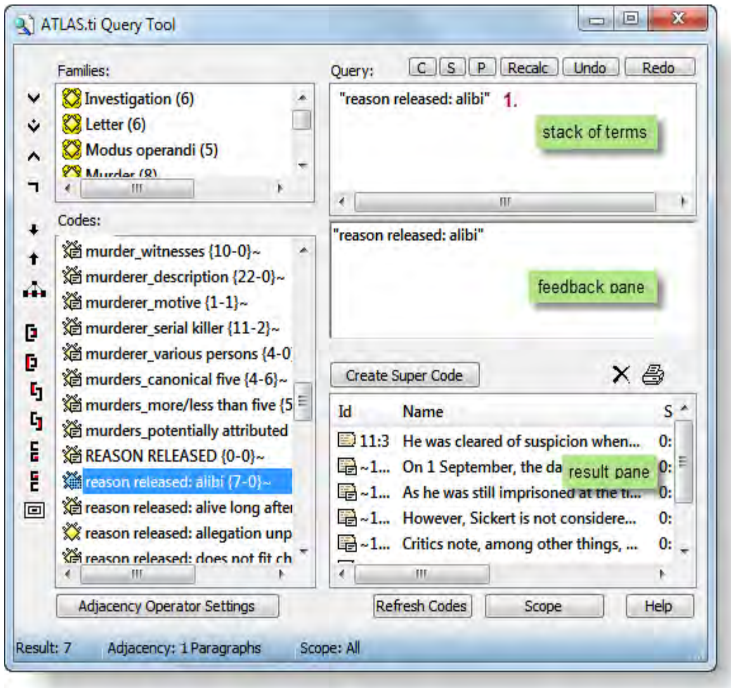
Build complex queries incrementally with immediate feedback after each step.

A Boolean Query

The example below uses the HU "Jack the Ripper_stage II." Please load and display this HU while reading the following. You can access the samples file , via TOOLS / EXPLORER / SAMPLES FOLDER. You find the HU in the sub folder JTR**.**

Our sample query, using Boolean operators, is this: "Find all quotations coded with either code "*reason released: alibi*" or code "*reason released: lack of evidence*".

- Open the Query Tool by clicking on the binoculars button in the main toolbar.
- Double-click on the code "reason released: alibi". The Query Tool displays the following entries:

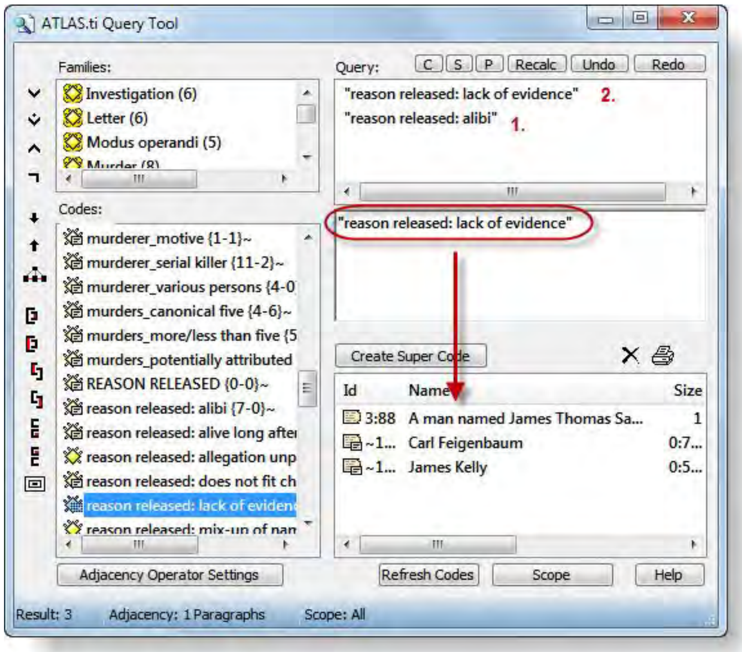

ATLAS.ti Query Tool
Families:
Investigation (6)
Letter (6)
Modus operandi (5)
Murder (8)
Codes:
murder_witnesses {10-0}~
murderer_description {22-0}~
murderer_motive {1-1}~
murderer_serial killer {11-2}~
murderer_various persons {4-0
murders_canonical five {4-6}~
murders_more/less than five {5
murders_potentially attributed
REASON RELEASED {0-0}~
reason released: alibi {7-0}~
reason released: alive long after
reason released: allegation unp
reason released: does not fit ch
Query:
C
S
P
Recalc
Undo
Redo
"reason released: alibi" 1.
stack of terms
"reason released: alibi"
feedback pane
Create Super Code
Id
Name
S
result pane
Adjacency Operator Settings
Refresh Codes
Scope
Help
Result: 7
Adjacency: 1 Paragraphs
Scope: All

Id	Name	S
11:3	He was cleared of suspicion when...	0:
~1...	On 1 September, the da	0:
~1...	As he was still imprisoned at the ti...	0:
~1...	However, Sickert is not considere...	0:
~1...	Critics note, among other things, ...	0:

Figure 215: Clicking a Boolean Query: Step 1

The *term stack* and feedback pane now display the code “reason released: alibi”. The results pane lists all quotations for this code.

Double-click on the code “reason released: lack of evidence”.



Families:
Investigation (6)
Letter (6)
Modus operandi (5)
Murder (8)

Codes:
murderer_motive {1-1}~
murderer_serial killer {11-2}~
murderer_various persons {4-0
murders_canonical five {4-6}~
murders_more/less than five {5
murders_potentially attributed
REASON RELEASED {0-0}~
reason released: alibi {7-0}~
reason released: alive long after
reason released: allegation unp
reason released: does not fit ch
reason released: lack of eviden
reason released: mix-up of nam

Query:
"reason released: lack of evidence"
"reason released: alibi"

"reason released: lack of evidence"

Id	Name	Size
3:88	A man named James Thomas Sa...	1
~1...	Carl Feigenbaum	0:7...
~1...	James Kelly	0:5...

Result: 3 Adjacency: 1 Paragraphs Scope: All

Figure 216: Clicking a Boolean Query: Step 2

Now there are two entries in the term stack, the codes “reason released: alibi” and “reasons released: lack of evidence.” The feedback pane displays the active query: code “reason released: lack of evidence”. And in the result pane you can immediately see the three quotations coded with this code.

With two operands on the term stack, we can combine them with an appropriate operator. The intention was to retrieve all quotations that contain information about an alibi or lack of evidence as reasons to release a suspect.

Click on the OR operator (see left) to combine the two expressions from the stack.

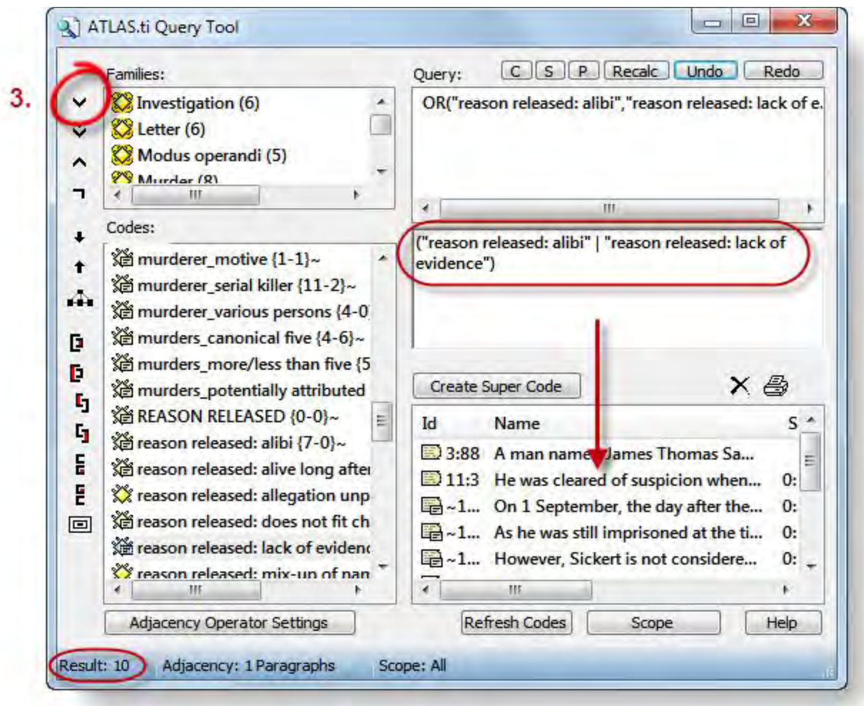

ATLAS.ti Query Tool
3.
Families:
Investigation (6)
Letter (6)
Modus operandi (5)
Murder (8)
Codes:
murderer_motive {1-1}~
murderer_serial killer {11-2}~
murderer_various persons {4-0
murders_canonical five {4-6}~
murders_more/less than five {5
murders_potentially attributed
REASON RELEASED {0-0}~
reason released: alibi {7-0}~
reason released: alive long after
reason released: allegation unp
reason released: does not fit ch
reason released: lack of evidenc
reason released: mix-up of nan
Query:
C S P Recalc Undo Redo
OR("reason released: alibi","reason released: lack of e.
("reason released: alibi" | "reason released: lack of evidence")
Create Super Code
Id Name S
3:88 A man named James Thomas Sa...
11:3 He was cleared of suspicion when... 0:
~1... On 1 September, the day after the... 0:
~1... As he was still imprisoned at the ti... 0:
~1... However, Sickert is not considere... 0:
Adjacency Operator Settings
Refresh Codes
Scope
Help
Result: 10
Adjacency: 1 Paragraphs
Scope: All

Figure 217: Clicking a Boolean Query: Step 3

The term stack now contains only one term, OR("reason released:alibi","reason released: lack of evidence"), i. e. the combination of the two codes. This term can be used as an operand to further extend the query, e. g. to negate the expression or add some more codes to it. But we will stop here for now.

The feedback pane displays the query in infix notation, as we would have entered it into a regular calculator ("reason released: alibi" OR "reason released: lack of evidence"). The results pane lists 10 quotations.

You can look at the quotations in the context of the document by clicking on a quotation in the list, or you can create a report (see "Output of Query Results" below).

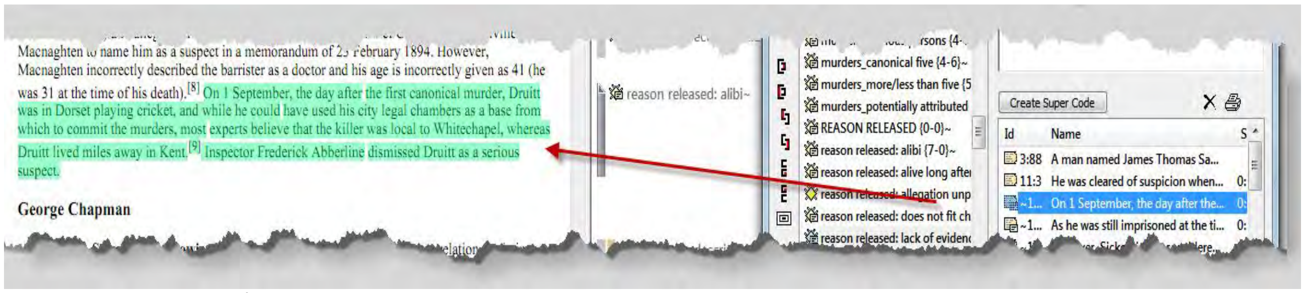
Output Of Query Results

Viewing Results In Context

Make sure that the Query Tool does not completely obstruct the area where PDs are displayed.

Click on a quotation in the results pane.

The quotation is highlighted in the primary document pane:



Macnaghten to name him as a suspect in a memorandum of 23 February 1894. However, Macnaghten incorrectly described the barrister as a doctor and his age is incorrectly given as 41 (he was 31 at the time of his death).[8] On 1 September, the day after the first canonical murder, Druitt was in Dorset playing cricket, and while he could have used his city legal chambers as a base from which to commit the murders, most experts believe that the killer was local to Whitechapel, whereas Druitt lived miles away in Kent.[9] Inspector Frederick Abberline dismissed Druitt as a serious suspect.

George Chapman

reason released: alibi~

murders_canonical five {4-6}~
murders_more/less than five {5
murders_potentially attributed
REASON RELEASED {0-0}~
reason released: alibi {7-0}~
reason released: alive long after
reason released: allegation unp
reason released: does not fit ch
reason released: lack of evidenc

Create Super Code

Id	Name	S
3:88	A man named James Thomas Sa...	
11:3	He was cleared of suspicion when...	0:
~1...	On 1 September, the day after the...	0:
~1...	As he was still imprisoned at the ti...	0:

Figure 218: Viewing results in context

Creating A Report

To print all hits found by a query, click the **PRINTER** button to the right of the results list.

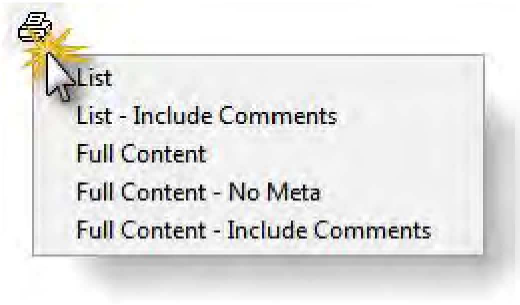


Figure 219: Output options

Select one of the following report options:

- **List:** Print a list of all quotations in a compact format showing only the quotation names.
- **List – Include Comments:** Same as **List** but includes the quotations' comments if any.
- **Full Content:** Output the *complete text* of the quotations (works for text-type quotations only, obviously).
- **Full Content – No Meta:** Output the *complete text* of the quotations and specify the kind of information that is included in the output. You may for example exclude the meta information for each quotation.
- **Full Content – Include Comments:** Same as **Full Content** but includes the quotations' comments if any.

Then, choose whether the output should be displayed in a text editor, saved to disk, or printed (see also "Output Destinations" on page 380).

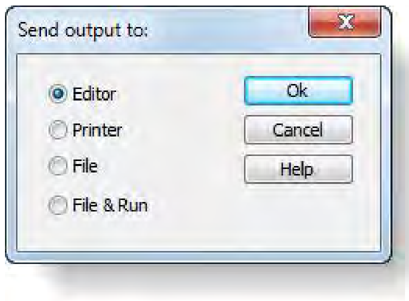


Figure 220: Select output destination

Cleaning Up The Hit List

Before creating a report of all the quotations found by the query, you have the option to remove entries from the hit list. You can remove unwanted hits from the list using the erase button right next to the printer button. You could, of course, reformulate your query to improve the precision.

This is how you clean up the hit list:

- Select an unwanted quotation in the hit list.
- Click the **Erase** button (see left).

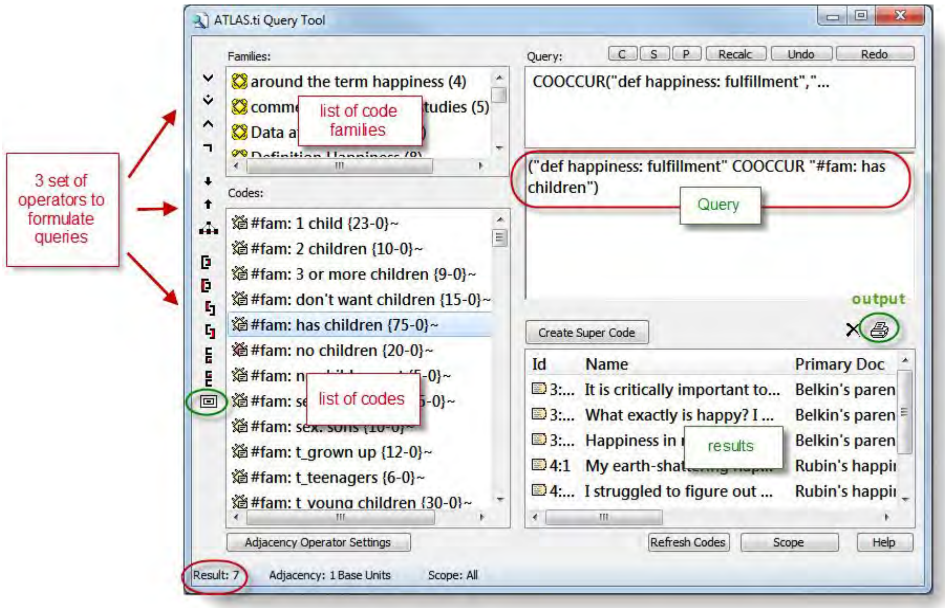
Another Example Based On The A7 Happiness Sample Project

- Select **Help / Quick Tour / Load "Children & Happiness Stage II"** to open the sample file.

For example, we can take a look at the code "*def happiness: fulfillment*" (respondents who define happiness as fulfillment) and inspect whether the responses are different dependent on whether someone has children or not.

- Double-click on the family "*def happiness: fulfillment*".
- Double-click on the code "*#fam: has children*". T
- Select the COOCCUR operator.

The results for this query are shown in the result pane. The number of results is shown in the bottom left corner of the window. To display the results in context, click on each quotation.



Families:
- around the term happiness (4)
- comme… …tudies (5)
- Data a…
- Definition Happiness (8)

Codes:
- #fam: 1 child {23-0}~
- #fam: 2 children {10-0}~
- #fam: 3 or more children {9-0}~
- #fam: don't want children {15-0}~
- #fam: has children {75-0}~
- #fam: no children {20-0}~
- #fam: n… …{5-0}~
- #fam: se… …5-0}~
- #fam: sex: sons {10-0}~
- #fam: t_grown up {12-0}~
- #fam: t_teenagers {6-0}~
- #fam: t_young children {30-0}~

Query: COOCCUR("def happiness: fulfillment","...

("def happiness: fulfillment" COOCCUR "#fam: has children")

Id	Name	Primary Doc
3:...	It is critically important to...	Belkin's paren
3:...	What exactly is happy? I ...	Belkin's paren
3:...	Happiness in	Belkin's paren
4:1	My earth-shat	Rubin's happi
4:...	I struggled to figure out ...	Rubin's happi

Result: 7 Adjacency: 1 Base Units Scope: All

Figure 221: Query tool window

- To create an output of these results, click on the printer button.

Let's now compare whether we get different results if we select respondents who do not have children:

- Click on the button **C** (CLEAR) at the top of the query tool.
- Double-click on the family "*def happiness: fulfillment*".
- Double-click on the code "*#fam: no children*".
- Select the *COOCCUR* operator.

The result pane shows 1 quotation.

Super Codes

"Intelligent" Super Codes compute their quotations "on demand".

Super Codes are a convenient way to store queries. Super Codes are very similar in look and feel to normal codes, with one important difference: instead of "hardwired" connections to quotations, Super Codes store a query to compute their virtual references whenever needed.

They "automatically" change their behavior during the course of theory building. If you have a Super Code "All about Magic" with a query "SUB Magic" and later add another sub code "White Magic,", all quotations to which the latter code refers are also retrieved by the (unchanged) query of "All about Magic." Super Codes can be clicked on in the code list like any other code and they will display their quotations in an identical way.

Normal codes are "hard-wired" to their quotations.

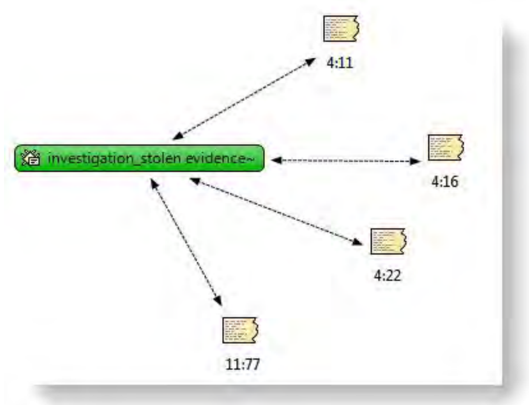


Figure 222: Codes are hard-wired to their quotations

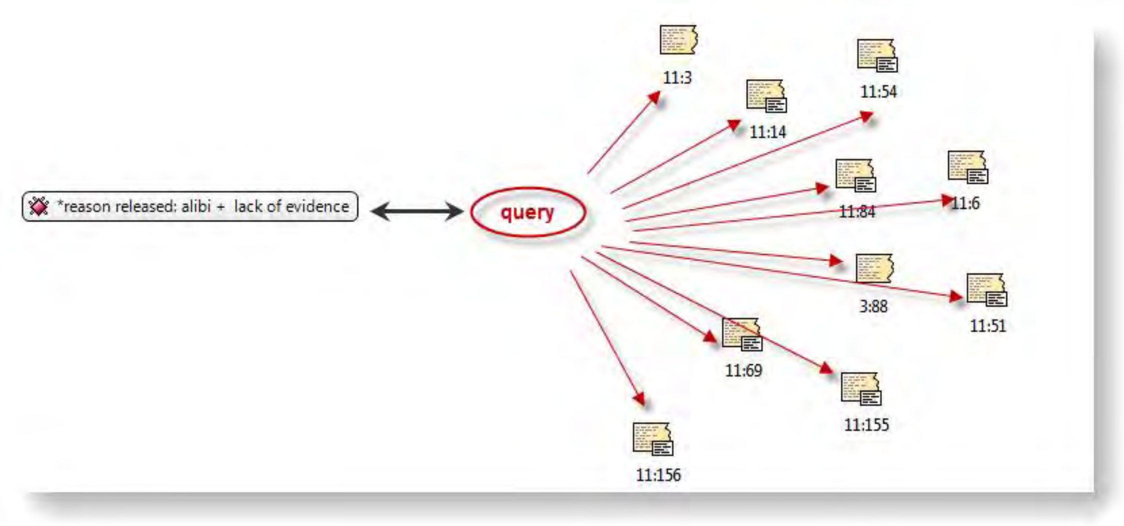


Figure 223: Super codes are linked to quoations via a query

Super codes are displayed in the Code Manager just like regular codes and can be recognized either by a red text color or by their red symbol, if images are switched on in the Views menu. The list of quotations associated with the Super Code can be displayed with a double-click, just as for any other code. Frequencies (density) are only indicated if you activated it, e. g. with a double-click in the Code Manager. If you start a new session, an asterisk (*) replaces the frequency count. The reason for this is that a Super Code is dynamic and its density/frequency count changes as soon as you modify any of the codes contained in the query of the Super Code. For the same reason, Super Codes are *not* displayed in the margin area. There is the possibility to create a regular code from a Super Code (see "Snapshot Codes" on page 271).

Super Codes can be used in code families, Network Views, and, last but not least, as powerful operands in queries, allowing you to incrementally build complex queries.

Creating Super Codes

To create a Super Code, you must have already constructed a query using the Query Tool which is displayed in the term stack. Note that because Super Codes are "intentional," you can also create a valid and useful Super Code with an empty results list (which might well change in a later stage of your analysis).

Click the "Super-Code" button in the query tool between the feedback pane and the result pane.

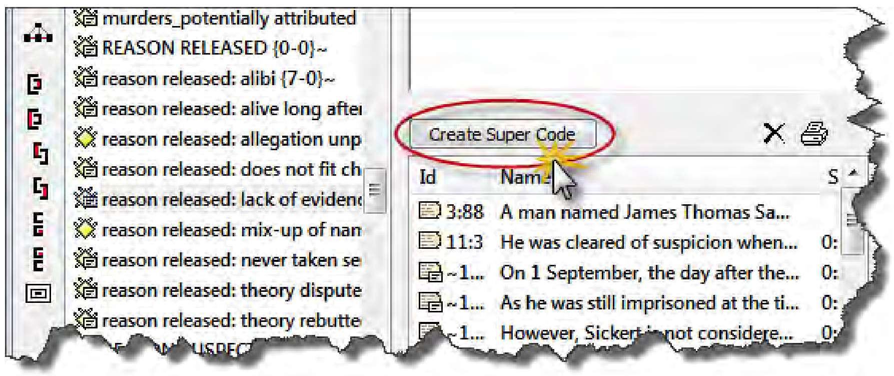

murders_potentially attributed
REASON RELEASED {0-0}~
reason released: alibi {7-0}~
reason released: alive long after
reason released: allegation unp
reason released: does not fit ch
reason released: lack of evidenc
reason released: mix-up of nan
reason released: never taken se
reason released: theory dispute
reason released: theory rebutte
Create Super Code
Id
Name
S
3:88 A man named James Thomas Sa...
11:3 He was cleared of suspicion when... 0:
~1... On 1 September, the day after the... 0:
~1... As he was still imprisoned at the ti... 0:
~1... However, Sickert is not considere... 0:

Figure 224: Location of super code button in the query tool

Enter a name for the new Super Code or accept the default name created from the query expression. Click **OK**.

The newly created Super Code immediately appears in the list of codes and can be used for new queries (and Super Codes) right away. Its icon and code name are red.

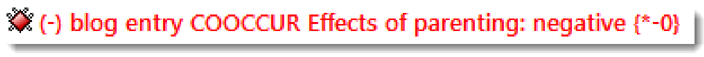

(-) blog entry COOCCUR Effects of parenting: negative {*-0}

Figure 225: Display of super codes in the Code Manager

You can access and edit the query later (see "Editing a super code Query" on page 270, but the notation that is used in the edit query window is not so easy to understand. Therefore it is advisable that you enter the query into the comment field of the super code. You can copy and paste it from the feedback pane into the comment field.

Creating Super Codes In The Code Manager

You can also create super codes using an OR combination in the Code Manager:

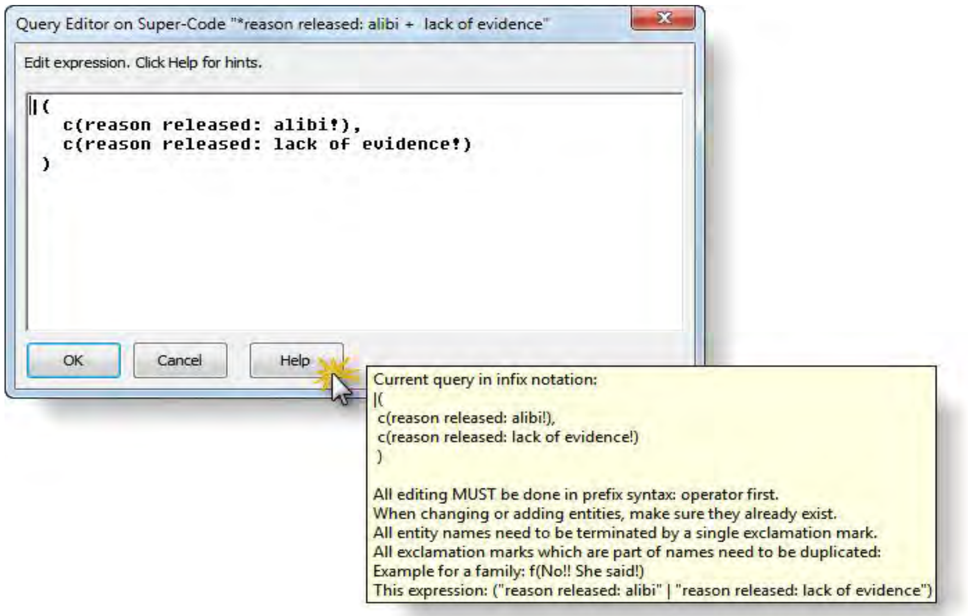
Select multiple codes in the Code Manager.

Select the menu option **CODES / MISCELLANEOUS / CREATE SUPER CODE.**

Enter a name for the super code and click **OK**.

Editing A Super Code Query

If you want to edit the query a super code is based upon, highlight the super code in the Code Manager and select **MISCELLANEOUS / EDIT QUERY** from the menu.


Query Editor on Super-Code "*reason released: alibi + lack of evidence"
Edit expression. Click Help for hints.
OK
Cancel
Help

```
|(
    c(reason released: alibi!),
    c(reason released: lack of evidence!)
 )
```

Current query in infix notation:

```
|(
 c(reason released: alibi!),
 c(reason released: lack of evidence!)
 )
```

All editing MUST be done in prefix syntax: operator first.
When changing or adding entities, make sure they already exist.
All entity names need to be terminated by a single exclamation mark.
All exclamation marks which are part of names need to be duplicated:
Example for a family: f(No!! She said!)
This expression: ("reason released: alibi" | "reason released: lack of evidence")

Figure 226: Editing a super code

Click on the **Help** button for instructions.

Auto-Optimization Of Super Code Queries

In order to take into account quotations that were manually removed from the hit list in the result pane of the Query Tool, the system must modified the query before a Super Code is created. Otherwise it would display the full set retrieved by the original query. To accomplish this task, a "suppressor" code is created, which refers to the quotations removed from the hit list.

Example: The original query "All quotations coded with Sanity or Health" (Sanity OR Health) yields 4 quotations 1:1, 1:2, 1:3 and 3:1. From the hit list, quotation 1:1 and 1:3 are removed. The query is now modified by creating a new "suppressor" code **Q1 referencing 1:1 and 1:3. The original query is modified as follows:

(Sanity OR Health) AND NOT (**Q1)

And returns exactly what you want: quotations 1:2 and 3:1.

Both the Super Code's and the suppressor code's automatically created comment reflect their mutual dependency.

A suppressor code cannot be deleted before the referring Super Code is deleted.

What You Cannot Do With Super Codes

As Super Codes are not directly associated with quotations, certain restrictions apply.

Super codes cannot used for manual coding.

Coding: The most important constraint is that you cannot associate them with quotations directly. Therefore, Super Codes are not presented when doing "code by list," and drag & drop onto data selections is prohibited.

Merging: Code Merge operations including Super Codes are also not possible.

Prevent Cycles: If you created a Super Code whose query contains a reference to a code family, you cannot assign this Super Code to the code family later. This would create a cyclic structure and is therefore disallowed.

Snapshot Codes

A Snapshot Code is a normal code that records the current state of a Super Code by way of "hard-wired" links to the derived quotations. By creating a snapshot from time to time, you can analyze the development of a Super Code.

Unlike the Super Code, a code created by the snapshot is displayed in the margin area and can be used for further coding. The default snapshot code names are suffixed with [SN<number>].

How To Create A Snapshot Code

- Select a Super Code in the Code Manager.
- From the **CODES / MISCELLANEOUS** menu, select option **CREATE SNAPSHOT**.
- The newly create code appears in the Code Manager. The code icon turns yellow, the characters of the code name appear in black and the post-fix [SN1 + a consecutive number] is added to the name. The frequency count is permanently displayed as the snapshot code is no longer dynamic.

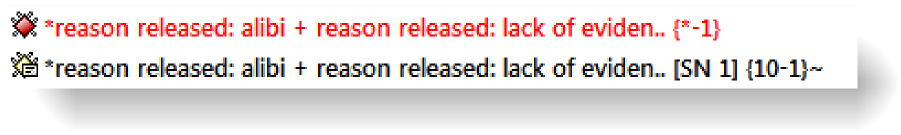
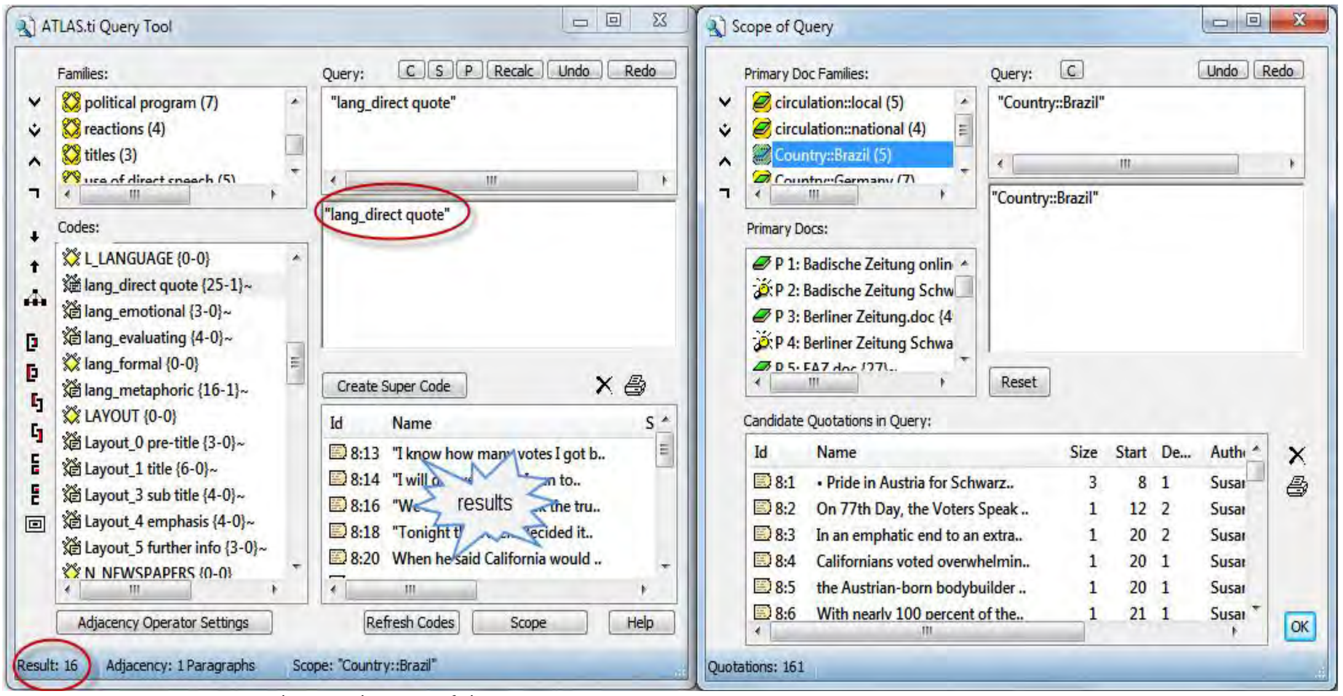

reason released: alibi + reason released: lack of eviden.. {-1}
*reason released: alibi + reason released: lack of eviden.. [SN 1] {10-1}~

Figure 227: A super code and its snapshot

Restricting Code Queries To Sub Groups

You can specify the documents that are to be considered in a query. By default, the query's "document universe" is *all* **PDs** currently filtered in the HU Editor. Clicking Scope opens another window that shows the PDs in the lower left pane and the **PD families** in the upper left. As PD families can be looked at as nominal variables, it is easy to preselect "all interviews with male interviewees aged between twenty and thirty from small towns.

A restricted set of operators is offered (note that only Boolean operators make sense here) and can be used to construct scope selection queries in much the same way as the query itself.



Id	Name	S
8:13	"I know how many votes I got b..	
8:14	"I will o... n to..	
8:16	"We... the tru..	
8:18	"Tonight t... ecided it..	
8:20	When he said California would ..	

Id	Name	Size	Start	De...	Auth
8:1	• Pride in Austria for Schwarz..	3	8	1	Susan
8:2	On 77th Day, the Voters Speak ..	1	12	2	Susan
8:3	In an emphatic end to an extra..	1	20	2	Susan
8:4	Californians voted overwhelmin..	1	20	1	Susan
8:5	the Austrian-born bodybuilder ..	1	20	1	Susan
8:6	With nearly 100 percent of the..	1	21	1	Susan

Figure 228: Restricting a search to a sub group of data

Figure 228 above shows a simply query for the code “lang_direct quote” (use of direct quotes in newspapers articles).

The question of interest is whether this writing style is used equally across the various newspapers. The two countries compared are Germany and the USA. The frequency for all documents is 25 (see the code in the code list on the left hand side).

The scope is set to the PD family “Country::USA”. This means that the result pane is filtered only displaying quotations from US newspapers. This results in 16 quotations, which can be seen at the bottom left of the query tool window.

Double-clicking on the PD Family “Country:: Germany” in turn yields the remaining 9 quotations (16 + 9 =25). Further comparisons could be by circulation, local versus national papers, or by political orientation comparing papers that are more politically right or left oriented.

For the results always look at the result pane of the query tool window. The bottom pane of the Scope of Query window shows all quotations from the selected document group. In mathematical terms, the result pane in the query tool window shows the intersection between the code query and all quotations from a particular document group.

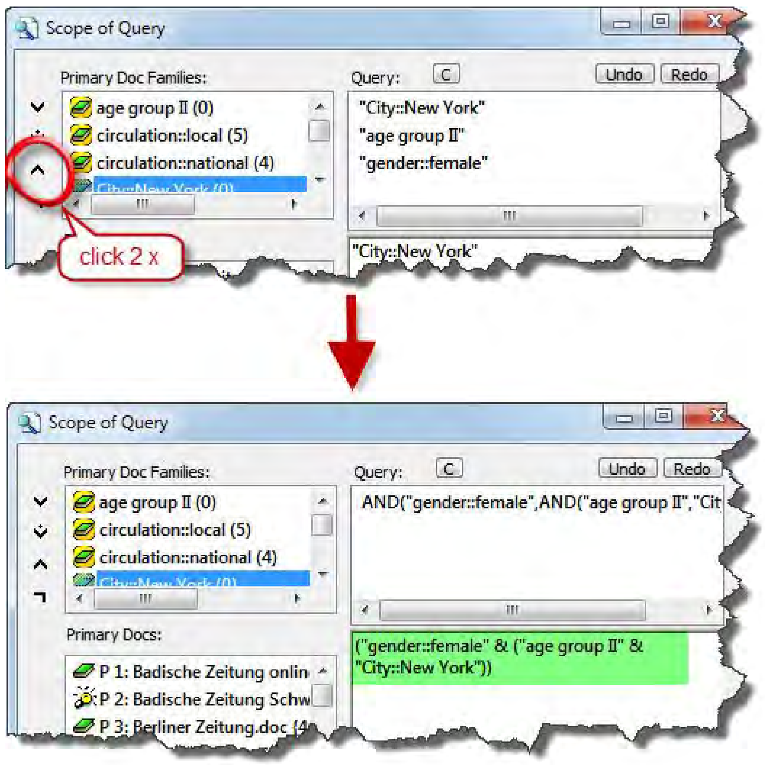
You can also combine a number of PD families or individual PDs using Boolean operators in the Scope of Query Window (see “Combining group of documents to restrict searchers“ below).

A scope is not stored as part of a Super Code's query specification. When you process the query of a Super Code later, the complete data base is queried by default.

Combining Group Of Documents To Restrict Searchers

A combination of document families like all females in age group II (31-40) from New York is clicked in the same way as you click a code query. You first select two or more PD families and then you select one or more operaators:

- Create a query.
- Click on the **Scope** button.
- To create the above described subgroup, you would double click on the three families: female, age group II and City::New York. Then you click on the **AND** operator twice.



age group II (0)
circulation::local (5)
circulation::national (4)
City::New York (0)

"City::New York"
"age group II"
"gender::female"

"City::New York"

AND("gender::female",AND("age group II","Cit

("gender::female" & ("age group II" & "City::New York"))

P 1: Badische Zeitung onlin
P 2: Badische Zeitung Schw
P 3: Berliner Zeitung.doc {4

Figure 229: Combining document families to set the scope

If you need certain combinations of document families more often, then you can create so called **Super Families**. How this works is described in the next section.

Super Families

Just like Super Codes, Super Families re-calculate their members "on demand."

Super Families follow the same underlying logic as Super Codes (cf. "Super Codes" on page 267). They are constructed by combining families or already existing super families.

Their members are determined *dynamically* whenever you activate a Super Family. Super families can be created based on primary document families, code families and memo families. Below an example is shown for primary document families. Creating super code or super memo families works in the same way.

Using Super Families

Example

You are working in the Customer Department of an airline and have been given the task to analyze customer complaints. As basis for your analysis, you have a set of documents dealing with customer complaints about domestic flights and a set of documents dealing with customer complaints about international flights. Your company is particularly interested in differences between domestic and international flights and differences between business and leisure travelers. Important factors to analyze might be gender, level of income, and frequent-flyer status.

The matrix below is based on the four customer groups:

Domestic Flights	**International Flights**
Business traveler	Business traveler
Leisure traveler	Leisure traveler

To represent these four groups in ATLAS.ti, you create primary document families. Next, you code the data. For example, you may use codes such as "punctuality," "general service," "in-flight services," and "human interaction" to describe complaints customers had.

After coding, you can use the query tool in combination with the scope function to find out how the four groups differ in terms of their complaint behavior.

Open the Query Tool.