WWW-Lemmie 2.0: User's Manual

Table of contents

1 Introduction

What Is Lemmie?

WWW-Lemmie 2.0 is a web-based tool for dynamic corpus work in the *Language Bank of Finland*. **WWW-Lemmie 2.0** is built on the *Lemmie API 2.0*, an object-oriented programmer's interface (written in Perl) to the lexical database of the Language Bank of Finland.

The fact that **WWW-Lemmie 2.0** is based on the *Lemmie API 2.0*, makes it possible for you to seamlessly combine results from your own programs that use the *Lemmie API 2.0* with the functionality of **WWW-Lemmie 2.0** and vice versa (mainly to use results fetched with **WWW-Lemmie 2.0** in your own programs). This interchangeability makes **WWW-Lemmie 2.0** a really dynamic tool; the basic work can be done with **WWW-Lemmie 2.0** and the real solving of a certain research problem can later be done with the help of the *Lemmie API 2.0* and regular Perl programming. Or putting it another way: Use **WWW-Lemmie 2.0** to test your hypotheses, and the *Lemmie API 2.0* to prove them right.

WWW-Lemmie 2.0 relies on the notion of *a token as a unique occurence of a word type with a collection of identical features*. These features are e.g. the *word form* (or *surface form*), the *base form*, the *part-of-speech*, a set of morphological features (as *case*, *definitiveness*, *modality*, *voice* etc.) and possible user-defined features (which might be semantic class, syntactic function etc.). As far as **WWW-Lemmie 2.0** is concerned, the word form is just a feature of a token among other features.

Lemmie as a Corpus Query Tool

The primary function of **WWW-Lemmie 2.0** is to be a corpus **query** tool, i.e. a tool with which the user can test assumptions about language against real texts.

In order to make a search, you need to compose a *query expression*. A query expression consists of a *search parameter* or a *combination of such* search parameters. A search parameter consists of features that a token in the targeted corpora must have to be considered a match. In a sense, the search parameter is thus an *underspecified type* (to connect to traditional linguistic terminology of word types and word tokens), which is hopefully matched by certain *fully specified tokens* in the corpora in use. (This is not the full truth though, as a search parameter can contain negation, truncation and lists too, which are not cases of trivial underspecification.)

The notion of *underspecification* is central in **WWW-Lemmie 2.0** and cannot be stressed enough; the search parameters in a query expression contain such features that the tokens in the corpora must match if they are to be returned, but the search parameters do not necessarily contain *all* the features that the tokens in the corpora are associated with. When tokens matching the features in a search parameter are returned, all features the tokens carries are returned, i.e. the tokens are *fully specified* in the sense that all data available in the corpus about these tokens are returned.

An example might be a query expression containing a search parameter containing the feature *word form* and requiring it to be "noutaja" (Finnish for "retriever") and nothing else. When this expression is used, **WWW-Lemmie 2.0** will return all such tokens in the corpora in use that match the requested word form feature. In practice all the tokens in the result will contain these features (as the word form itself is unambiguous): word form is "noutaja", base form is "noutaja", part-of-speech is "Noun", case is "Nominative" and number is "Singular". Depending on which corpora you have selected to be queried, the amount of matching tokens will vary.

Another example might be a query expression containing two consecutive search expressions, the first with the feature *base form* set to "presidentti" (Finnish for "president"), the second with the feature *part-of-speech* set to "Proper" (the tag for "proper noun"). When this expression is used, **WWW-Lemmie 2.0** will return two-word expressions like "presidentti Bush", "presidentti Gorbatshov", "presidentti Koivisto", "presidentti Koiviston", "presidentin Husseinin", "presidenttiä Jetsiniä" (showed here as word form combinations, though the tokens in the expressions in the result naturally contain other features as well, like *case* is "partitive" in the tokens in the two-word expression "presidenttiä Jeltsiniä").

In order for complex query expressions to be created, **WWW-Lemmie 2.0** offers an *advanced syntax* which is described in section 11, Search Syntax, below. Traditional queries (where the query expression consists of a word form, a set of word forms or the boolean operators AND and NOT) can be done with simplifed syntax, much like the syntax used at regular search engines at the web. See section 11, Search Syntax, for more information.

Read more about how the query expression is composed in section 11, Search Syntax, below.

Read about selecting the settings for a query in section 5, Setting dialog tab, below.

Different Views of the Result

After a search has been made in the targeted corpora using a certain query string, **WWW-Lemmie 2.0** returns a result which can be viewed in four different ways:

- as a KWIC **Concordance**
- as a **Joint Frequency (Full)** Table
- as a **Split Frequency (Quick)** Table
- as a **Collocation Table**

Read more about the result types in section 3, Search dialog tab, below.

Lemmie as a Corpus Comparison Tool

WWW-Lemmie 2.0 can also be used for comparing different aspects of language with each other. A typical example is comparing the use of a certain word or phrase in two different corpora in the Language Bank of Finland for finding out for instance regional differencies in language use. (One might expect the city name "Turku" to occur more often in the newspaper Turun Sanomat, which is published in Turku, than in the newspapers Karjalainen, which is published in Joensuu.)

Another example of comparison might be if one wanted to find out if two specific verbs occur with different or overlapping verb particles and if so, in what statistical relation these particles are represented between the two verbs.

The comparison function is described in more detail in section 4, Compare dialog tab, below.

Lemmie as a Corpus Result Manager

WWW-Lemmie 2.0 offers the possibility for the user to save the result of a query in both XML format for further processing outside **WWW-Lemmie 2.0** and in binary format for later use within **WWW-Lemmie 2.0** or any program using the *Lemmie API 2.0*.

The idea behind the binary save is that a user might want to return to a certain result at a later stage without having to remake it (which might require tampering with corpus settings or, in case of large queries, waiting for the result to be fully fetched from the targeted corpora). Also, if one wants to compare two results as described above, then these results need to be saved in binary format to disk first.

Read more about saving results in section 3, Search dialog tab, below.

Read more about managing and using saved binary results in section 6, My Results Dialog Tab, below.

Lemmie as a Corpus Compiler Tool

The Language Bank contains a number of predefined corpora, most of which are a full year of texts from a certain newspaper (e.g. "tusa1998", which contains all articles from Turun Sanomat 1998 or "karj1991", which contains all articles from Karjalainen 1991)

Sometimes, the predefined corpora are not good enough. They might be too large or contain texts from too many areas (e.g. sports articles as well as commentaries etc.). In such cases, **WWW-Lemmie 2.0** offers a possibility for you to select documents from the predefined corpora and compile a custom corpus of your own containing these documents only. The documents are selected according to the metadata associated with the documents in the predefined corpus, e.g. the writer's name, publication date, the document's title etc. Documents can also be added to custom corpora at any stage later.

The compilation of custom corpora is fully dependent on the metadata associated with the documents in the predefined corpora. Even though **WWW-Lemmie 2.0** and the *Lemmie API 2.0* support roughly a dozen metadata categories in the spirit of the Dublin Core Metadata Element Set 1.1, most of the predefined corpora still lack the essential metadata.

You cannot compile corpora of your own private documents! (If there is a demand for such functionality, please ask for it.)

Read more about compiling custom corpora in section 7, My Corpora dialog tab, below.

Using user-added (and user-corrected) features in corpora

Those documents in the Language Bank of Finland which can be used from **WWW-Lemmie 2.0** have been morphosyntactically annotated automatically. The annotation has not been manually checked and consequently not manually corrected either. Thus a few percent of the tokens contain annotation errors. For instance the numeral "kuusi" (Finnish for "six") might be annotated as a noun and not a numeral due to the homography of the word form ("kuusi" as a noun means "spruce").

It is also possible that you would like to add annotation of your own to certain tokens or combinations of tokens, for instance a feature for semantic class (e.g. the noun "kuusi" as *inanimate*) or syntactic function (e.g. "kuusi" as being the *subject* of the sentence "Kuusi kaatui auton eteen").

To cope with all cases where you want to change the annotation of a token or add annotation to a token, the *Lemmie API 2.0* offers something called **user addition files**. In short, a user addition file is an XML file which is linked to a specific (predefined or custom) corpus. When a query is made in that specific corpus, the query is extended to the user addition file too, and possible matches in this file will automatically override any in the static corpus to which the user addition file is linked. In this way corrected annotation errors and your own additions are searchable as if they where part of the static corpus.

WWW-Lemmie 2.0 does not offer the functionality to add annotation to or or change the annotation of tokens. To do this, you need to write your own Perl program using the *Lemmie API 2.0*. However, if you have created a user addition file, **WWW-Lemmie 2.0** will search it as soon as you have associated it with any of the corpora in use.

Read more about user addition files in section 7, My Corpora dialog tab, below.

2 Overview of Icons

You will notice a set of icons here and there on the **WWW-Lemmie 2.0** dialog tabs and popup windows. Some of the icons are clickable, some are only there to attract your attention. The icons are described below:

General Icons

If you click this icon, the interface language changes to **Finnish** (globally, on all dialog tabs).

If you click this icon, the interface language changes to **Swedish** (globally, on all dialog tabs).

If you click this icon, the interface language changes to **English** (globally, on all dialog tabs).

Clicking this icon will open the context-sensitive help with a description of the element in the immediate vicinity of the icon.

When this icon appears, **WWW-Lemmie 2.0** has an important notice for you.

When this icon appears, **WWW-Lemmie 2.0** has a tip for you.

Clicking this icon will take you to the top of the current dialog tab.

Clicking this icon will open your browsers Print dialog box, so you can print the page.

Clicking this icon will delete (*a*) the most recent result (*Search dialog tab*), (*b*) the selected results (*My Results dialog tab*), (*c*) the selected custom corpora (*My Corpora dialog tab*) or (*d*) the selected documents (custom corpus pop up window).

Clicking this icon will close the current window.

Icons on the Search dialog tab

This icon appears to the right of a KWIC concordance line. Clicking the icon will open up a new window containing the node (the highlighed **purple** part of the line) and surrounding paragraph (the <p> element in the corresponding XML file) as context.

This icon appears to the right of a KWIC concordance line. Clicking the icon will open up a new window containing the node (the highlighed **purple** part of the line) and surrounding "division of text" (the element in the corresponding XML file) as context. The division of text usually corresponds to one article in a newspaper or one chapter in a book.

Clicking this icon will save the result of your most recent query (the one showing in the window) to disk in XML format as a KWIC concordance, Frequency table or Collocation table depending on the current view type.

Clicking this icon will save the result of your most recent query (the one showing in the window) to disk in binary format for comparison or quick retrieval at a later stage.

Clicking this icon will open a download dialog, where you can download your most recent query to your local computer as plain text.

Icons on the My Results and My Corpora dialog tabs

Clicking this icon will (*a*) open a saved result and show it as a KWIC concordance in the *Search dialog tab* (*My Results dialog tab*), (*b*) open a new window and display a list of the documents that exist in a custom corpora (*My Corpora dialog tab*) or (*c*) open a new window and display the full document in structured format in it (custom corpus pop up window)

Icons in the Save as XML and Save Binary Popup Windows

Clicking this icon will change directory to the one selected in the directory menu to the left of the icon.

Clicking this icon will show a text field (instead of a file menu) in which you can write the name of the file in which the result is to be stored.

Clicking this icon will show a file menu (instead of a text field) with the files in the current directory from which you can choose the name of the file in which the result is to be stored.

3 Search Dialog Tab

Overview

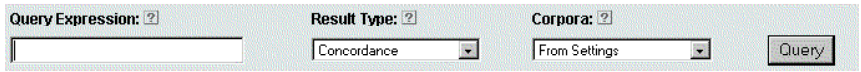
The *Search dialog tab* is used to query the targeted corpora of The Language Bank of Finland and view the result as one of the following result types:

- a KWIC **Concordance**
- a **Joint Frequency (Full)** Table
- a **Split Frequency (Quick)** Table
- a **Collocation Table**

The *Search dialog tab* is the most important tab. If you know how it works, you can use Lemmie to perform the most common operations.

The Elements of the Search Dialog Tab

The Query Bar - the Main User-Interaction Element



The Query Bar. *This bar is used for performing a query.*

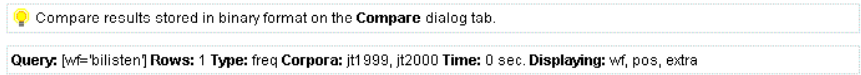
A query is created in the **Query Bar** by entering a query expression into the **Query Expression** text field, choosing a **Result Type** in the menu with the same name, selecting which corpora to query in the **Corpora** menu and pressing the **Query** button.

The **Query Expression** is written in either simple or advanced syntax, both described in section 11, Search Syntax, below.

The **Result Type** denotes the kind of result that will be shown at first. The result types *Concordance*, *Joint Frequency (Full)* and *Collocation Table* uses the *Lemmie API 2.0* in "full-flavoured fashion" and are therefore such that can be reviewed as another type later and saved to disk as either XML or in binary format. The result type *Split Frequency (Quick)* uses some "quick and dirty" functionality of the *Lemmie API 2.0*. Split Frequency tables cannot be reviewed as another result type, nor can they be saved to disk in any format or be retrieved with query expressions containing more than one search parameter. And also, if you are using a user addition file, be adviced that they are ignored when building a Split Frequency table.

If you choose the default menu element, *From Settings*, in the **Corpora** menu, then the corpora that will be queried are the ones selected in the *Settings dialog tab*. You can temporarily override the corpora selection from the *Settings dialog tab*, by selecting a specific corpus in the Corpora menu.

The Status Bar



The Status Bar. *Two examples of what the status bar can contains, the upper showing a typical welcome message, the lower showing the summary of a query resulting in a frequency table.*

The **Status Bar** contains information of immediate interest in the current context, as a summary of the result being viewed after a query has been made or the result of an action recently taken.

The Command Bar

The Command Bar. *The Command Bar appears when a result is showing.*

The **Command Bar** appears at the very bottom of the *Search dialog tab* when a query result of type *Concordance*, *Joint Frequency (Full)* or *Collocation Table* has been made (but not when a result of the type *Split Frequency (Quick)* is showing.)

Six different actions can be taken in the **Command Bar**:

- The result can be **reviewed** as another result type
- The result can be **printed**
- The result can be **saved** to disk **as XML**
- The result can be **saved** to disk **binary format**
- The result can be **downloaded** as plain text to your local computer
- The result can be **deleted** immediately

Select a view type from the menu in the **Command Bar** and press *Go!* to **review** the result as another result type than the one currently being displayed. This is regularly faster than remaking the query in the Query Bar, as the result is not fetched from the lexical database but read from a temporary file on disk.

Click the icon to open up your browser's **print** dialog box for printing the result of your current query (in the result type you are currently viewing).

Click the icon to **save** your query to disk **as XML**. A Save dialog window will open prompting for file name and location. The query will be saved as the result type you are currently viewing in the selected directory with the selected name. A file extension, `.xml`, will be appended to the file name.

Click the icon to **save** your query to disk **in binary format**. A Save dialog window will open prompting for file name. Binary results are always saved in your private Lemmie directory (`~/.lemmie`) and the file extension, `.lres`, will be appended to the file name (where `.lres` stands for *Lemmie result*).

Click the icon to **download** the result to your local computer in plain text format.

Click the icon to **delete** the current result immediately.

Result Chunks

More matches exist. *Large results are split into result chunks for efficiency.*

For efficiency reasons, **WWW-Lemmie 2.0** has a *memory buffer* in which a result in memory is stored. If the size of a result of a query exceeds the maximum size of the memory buffer, the result is fetched in chunks from the lexical database. This makes it possible for you to perform queries on common words or phrases without having to wait for several minutes before the result is shown or, in the worst case, without getting a result at all. (An example of a query on a common word is `[wf='ja']`, which will look up the word form "ja", which is Finnish for "and".)

You will notice that the memory buffer is full when a button labeled *Next Chunk* appears in the *Search dialog tab*. If you click the button, **WWW-Lemmie 2.0** will fetch the next chunk of data for you. *Next Chunk* buttons will appear for each chunk that is not exhaustive, i.e. for each chunk that in turn fills the memory buffer and leaves some data unfetched from the lexical database.

The maximum size of the memory buffer can be changed in the *Settings dialog tab*.

⚠ **Each chunk is sorted individually!**

⚠ **Each chunk is a result of its own** as far as **WWW-Lemmie 2.0** is concerned. Thus storing a chunk to disk will only store the current chunk, not the full result of the query.

⚠ When performing queries with query expressions that contain more than one search parameter (e.g. when performing multi-word or phrasal queries), **the result chunk might not contain as many hits as one would expect given the maximum memory buffer size**. The technical explanation to this is that, in these cases, the memory buffer has been filled up with the tokens that match the *least common* of the search parameters. When this intermediate result is narrowed down to the one's also matching the other search parameters, then the number of hits has most probably shrunk to below the maximum size of the memory buffer. (An example of this is the query `[pos='Conjunction'][wf='ehkä']`, which means looking up any conjunction followed by the word form "ehkä", which is Finnish for "maybe". Here, the search parameter `[wf='ehkä']` will match less tokens

than the search parameter [pos='Conjunction'], but there will probably be more hits on "ehkä" than there is space in the memory buffer. As a result of this, the result of the query is split into chunks. Then, the first chunk is narrowed down to the one's only containing a conjunction immediately to the left of "ehkä". Consequently, the final result chunk will contain less hits than the maximum size of the memory buffer.)

Read more about setting the size of the memory buffer in section 5, Settings dialog tab, below.

The Concordance Result Type

1	sedan FST lät nyhetsbomben	**brisera** . Plåstrets julkalender skulle	jt1999
2	I början av veckan	**briserade** en bomb , om	jt1999
3	bomben som på måndagen	**briserade** på Storgatan hade exploderat	jt1999
4	Måndag eftermiddag då bomben	**briserade** på Storgatan i Jakobstad	jt1999
5	Skenet bedrar , vardagen	**briserar** , hotande skuggor rycker	jt1999
6	skolgångens välsignelser ? Då	**briserar** den demografiska bomben och	jt2000
7	tidsinställd bomb som antigen	**briserar** eller måste desarmeras .	jt2000

The Concordance Result Type. *This is the result of the query [bf='brisera'] in Jakobstads Tidning 1999 and 2000 viewed as a KWIC concordance.*

The KWIC concordance contains a **line number**, the **KeyWord(s) In Context** (where the keyword(s) or the *node* is displayed in purple), the **icons** which when clicked fetch some more ( ) or much more ( ) context and the **ID** of the corpus from which the concordance line in question has been fetched.

The **displayed features** are set in the *Settings dialog tab* under *Concordance Settings*. In the example above, the default display feature is used, namely the word form of a token.

The **context size** is set in the *Settings dialog tab*. In the example above, it is set to four (4) tokens at both sides of the node. (Punctuation, quotation and parenthesis characters are treated as tokens in their own right.)

The **sort keys**, **sort features** and **sort directions** of the tokens in the concordance are set in the *Settings dialog tab*. The **sort key** denotes the tokens at a specific position in the concordance, e.g. the first tokens in the nodes on each concordance line. A **sort feature** is a feature of a token, e.g. the word form, which will be used for sorting a specific sort key. The **sort direction** is either *forward*, denoting that the value of the sort feature of a certain sort key is to be sorted from the left to the right, or *backward*, denoting that the value of the sort feature of a certain sort key is to be sorted from the right to the left.

⚠ Sorting is case-sensitive; uppercase charaters are sorted before lowercase charaters, e.g. "A" as well as "Z" come before "a" and "z".

In the concordance example above, the primary sort key is the token in the node, the secondary sort key is the first token in the context to the right of the node and the third sort key is the second token to the right of the node. For all three sort keys, the word form of the tokens is used as sort feature. The sort direction is forward for all sort keys.

A result being viewed as a KWIC concordance can be reviewed as a joint frequency table or collocation table by using the controls in the Command Bar. It can also be saved to disk as XML or in binary format.

Read more about concordance settings in section 5, Settings dialog tab, below.

The Joint Frequency (Full) Result Type

Joint Frequency (jt1999 + jt2000)		
Abs	**Rel**	**Token**
7	**0,0000021013**	**[bf='brisera']**
3	0,0000009005	briserade
3	0,0000009005	briserar
1	0,0000003002	brisera

The Joint Frequency (Full) Result Type. *This is the result of the query [bf='brisera'] in Jakobstads Tidning 1999 and 2000 viewed as a joint frequency table.*

The joint frequency table contains the absolute (the column labeled *Abs*) and relative (the column labeled *Rel*) frequency of all the selected display features of the matching tokens together (the first result row) and individually (the rest of the result rows) in the targeted corpora. No distinction between the distribution of the matching tokens between different targeted corpora is made, thus the frequency counts are "joint".

The **displayed features** are set in the *Settings dialog tab* under *Frequency Table Settings*. In the example above, the default display feature is used, namely the word form of a token. However, any feature of a token can be selected.

⚠ The frequency **data is counted according to the display features set**, e.g. if part-of-speech is set as display feature, then the frequency table will contain the frequencies of the different part-of-speech labels in the result.

The frequency table can be **sorted** according to the *descending frequency* (default) of the display features of the tokens or *alphabetically*.

The frequency table can be narrowed down by setting **minimum absolute frequency** and **minimum relative frequency** values. One might for instance exclude such rows of data that contain an abolute frequency count which is only one.

Clicking the absolute frequency value on a row in the joint frequency table will ask **WWW-Lemmie 2.0** to show the matches in question as a KWIC concordance.

A result being viewed as a joint frequency table can be reviewed as a KWIC concordance or collocation table. It can also be saved to disk as XML or in binary format.

Read more about frequency table settings in section 5, Settings dialog tab, below.

The Split Frequency (Quick) Result Type

Joint Frequency (jt1999 + jt2000)			jt1999		jt2000	
Abs	**Rel**	**Token**	**Abs**	**Rel**	**Abs**	**Rel**
7	**0,0000021013**	**[bf='brisera']**	**5**	**0,0000015009**	**2**	**0,0000006004**
3	0,0000009005	briserade	3	0,0000033945	0	0
3	0,0000009005	briserar	1	0,0000011315	2	0,0000008172
1	0,0000003002	brisera	1	0,0000011315	0	0

The Split Frequency (Quick) Result Type. *This is the result of the query [bf='brisera'] in Jakobstads Tidning 1999 and 2000 viewed as a split frequency table.*

The split frequency table contains the same information as the joint frequency table, *plus* the frequency information of the targeted corpora separately (or *split*). This makes is easy to compare language usage in different corpora manually.

The clearest advantage of **a split frequency table** in comparison with a joint frequency table is that it **is fetched and generated very fast**. For this promise to hold, a split frequency table **has some disadvantages**:

⚠ Split frequency tables will not be generated if the query expression contains more than one search parameter, i.e. **multi-word or phrasal queries are not done**.
⚠ Split frequency tables are generated from the lexical database only, i.e. possible **user addition files are not searched**.
⚠ Split frequency tables **cannot be reviewed** as any other result type.
⚠ Split frequency tables **cannot be saved** to disk.

Split frequency tables use the same settings as the joint frequency tables.

Read more about frequency table settings in section 5, Settings dialog tab, below.

The Collocation Result Type

You shall know a word by the company it keeps.

- Firth, J. (1957) "A Synopsis of Linguistic Theory 1930-1955"', *Studies in Linguistic Analysis*, Philological Society, Oxford.

Collocation Table

Collocation		z_score	loglikelihood	Abs (Left/Right)	Rel	Node Abs	Collocate Abs
[bf='spräng_laddning']	[bf='brisera']	150,1735002514	18,5342298699	2 (2 / 0)	0,0000002150	11	50
[bf='nyhet']	[bf='brisera']	31,7695599000	-420,3125659142	2 (2 / 0)	0,0000002150	11	1113
[bf='förhandling']	[bf='brisera']	25,4838526872	-667,7158229700	2 (2 / 0)	0,0000002150	11	1726

The Collocation Result Type. *This is the result of the query `[bf='brisera']` in Hufvudstadsbladet 1998 viewed as a collocation table where the collocate candidate used for calculus are any tokens at a maximum distance of three to the left of the node.*

In the perspective of **WWW-Lemmie 2.0**, a collocation is a combination of a node and a token (or a *collocate*) that occurs more frequently together (in respect of some defined features) than one would expect given the individual frequencies (of the defined features) of the node and the collocate. The collocation table is a listing of such collocations, ranked according to the collocations' *statistic relevance*, calculated with a given formula.

In the example above, the collocate candidates (such tokens that are passed to collocation calculus) are such tokens at a position of one, two or three tokens (the *collocation window*) to the left of the node that have base form candidates that occur at least twice in the targeted corpora combined or that have a relative frequency of at least 0.00000002.

For a collocate candidate to be considered a collocate in the example (and thus for the node and collocate to be considered a collocation), the statistic relevance score when applied on the frequencies of the base form features of the node and collocate candidates has been required to be at least 10 when using the *Z-score* test, which has been selected as the primary formula in the *Settings dialog tab*. The primary formula is also used as primary sort key, therefore, the collocation with the highest Z-score is displayed first in the table. For the sake of manual comparison, also the *Loglikelihood* test has been applied to the node and token combinations. The scores of this formula are displayed in the column to the right of the Z-score values.

The absolute frequency of the collocations in the targeted corpora combined is shown in the column labeled *Abs*. Within parenthesis, the absolute frequencies of the collocations, where the collocate is found to the left and to the right of the node, are given separately. By clicking such a frequency value, a concordance with the collocations in context are generated.

The relative frequency of the collocations in the targeted corpora combined is shown in the column labeled *Rel*. The columns labeled *Node Abs* and *Collocate Abs* contains the absolute frequencies of the nodes and the collocates respectively (or more precisly: the absolute frequencies of the nodes that share the same base form feature and the absolute frequency of the collocates that share the same base form feature).

The collocation itself is shown in the left-most column in the collocation table. The collocate is displayed on both sides of the node, if it occurs an equal amount of times on both sides of the node. If the collocate is displayed on only one side of the node, then the collocate is found more often on that side of the node than on the other. If the collocate on one side of the node is displayed in bold face, then it is found at least twice as many times on that side of the node than on the other.

The following parameters can be set in the *Settings dialog tab* under *Collocation Table Settings*:

- The **Minimum Absolute Frequency** a collocation must have to be displayed, i.e the frequency of the combination of (the selected display features of) a node and a collocate candidate.
- The **Minimum Relative Frequency** (in respect of the targeted corpora combined) a collocation must have to be displayed.
- The **Minimum Score** the collocation must show when calculating its relevance with the *primary formula*.
- The **size of the left-side window** (as viewed from the node), i.e. which tokens to the left of the node, if any, that are to be considered as collocate candidates.
- The **size of the right-side window** (as viewed from the node), i.e. which tokens to the right of the node, if any, that are to be considered as collocate candidates.
- The **statistic formulae** to use for calculus.
- Which of the formulae that is the **primary formula** which will be used as primary sort key and for testing if the score beats the *minumum score*.
- The **feature(s) of the node** that is/are to be used for calculus and display.
- The **feature(s) of the collocate candidates** that is/are to be used for calculus and display.
- Whether or not the node is to be treated as a single unit indifferent of the different values of its selected features.
- Whether or not the collocate candidates must exist in the same sentence as the node. (Regularly, you want to keep this box checked.)

The Statistic Formulae

In the following the available statistics are explained *very* briefly. If nothing else is said, then the book *Statistics for Corpus Linguistics* by Michael P. Oakes (Edinburgh University Press, 1998) can be used as the main reference of the collocation calculus formulae below.

Here are some good web sites on collocation calculus and multi-word extraction from corpora (the links open in new windows):

http://mwe.stanford.edu/collocations.html
http://www.collocations.de/EK/Articles/about-AMs.pdf

In the formulae below, fx stands for the total frequency of the node, fy the total frequency of the collocate and fxy the total (in the given collocation window) co-occurence frequency of the node and the collocate. Similarly, p denotes relative frequency and ξ expected frequency (under the null hypothesis). Further *ws* denotes the total amount of collocate candidates (i.e. the collocation window width × the amount of nodes, if the window width is fixed), *N* is the total amount of possible collocation windows of a given width in the corpus (= the size of the corpus - collocation window width + 1, if the collocation window width is fixed). Finally, a variable with a bar on top, means the complement of that variable. So if the *x* in fx had a bar on top, it would mean the complement (in respect of the corpus size) to the frequency of *x* in a corpus, i.e. the frequency of *not x*, i.e. the corpus size - fx.

$$\frac{2 \times fxy}{fx + fy}$$

Dice coefficient is a similiarity metric which has a value between zero (0) and one (1). Employed to collocation calculus it gives an estimate on how often a node and a collocate co-occur in contrast with the times they occur separately of each other. If the Dice coefficient is 1, this means that the node and the collocate *always* co-occur (in the given collocation window), if the value is 0.5, then the the node and the collocate co-occur every second time (well, roughly), if the value is 0, then the node and the collocate never co-occur in the given collocation window.

Reference: Dice, L. R. (1945), "Measures of the Amount of Ecologic Association Between Species", *Geology*, 26, pp. 297-302.

$$\frac{\log(fxy / ws \times pxy)}{\log(2)}$$

Mutual Information. Regularly, Mutual Information tends to emphasize such collocations that contain a rare constituent (i.e. a rare node or a rare collocate), e.g. foreign names or the such. See the referenced paper for detailed information.

Reference: Church, K. W. & Hanks, P. (1989), "Word association norms, mutual information and lexicography", *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada, pp. 76-83.

$$\frac{fxy - \xi xy}{\sqrt{fxy\,(1 - fxy / N)}}$$

T-score. The t-score measure compares the observed frequency of a collocation with the expected one (under the null hypothesis) and gives a measure on how much they differ. See the referenced paper for detailed information.

Reference: Church, K., Gale, W., Hanks, P. & Hindle, D. (1991), "Using Statistics in Lexical Analysis", Zernik, U. (ed.) *Lexical Acquisition: Exploiting On-line Resources to Build a Lexicon, Hillsdale, New Jersey, USA.*

$$\sum_{\substack{i \in (X,\bar{X}) \\ j \in (Y,\bar{Y})}} \frac{(f_{ij} - \xi_{ij})^2}{\sqrt{\xi_{ij}}}$$

Chi-square. The chi-squared measure allows an estimation of whether the frequencies *in a table* differ significantly from each other. Thus, it can serve as a measure of evenness of distribution. The chi-square measure is good for relatively large occurence counts (co-occurence frequency > 5).

$$\frac{pxy^2}{px \times py}$$

Symmetric Conditional Probability. See the referenced paper for detailed information.

Reference: Ferreira da Silva, J. & Pereira Lopes, G. (1999), "A local maxima method and a fair dispersion normalization for extracting multi-word units from corpora", *Sixth Meeting on Mathematics of Language*, pp. 369-381.

$$\frac{fxy - \xi xy}{\sqrt{\xi xy\,(1 - \xi xy / N)}}$$

Z-score. Z-score measures the unrandomness of the co-occurence of a node and a collocate in respect of the expected frequency. See the referenced paper for detailed information.

Reference: Smadja, F. (1993), "Retrieving collocations from text: Xtract", *Computational Linguistics*, 19, pp. 143-177.

Reference: Berry-Roghe, G. L. M. (1973), "The Computation of Collocations and their Relevance in Lexical Studies", Aitken, A. J., Bailey, R. & Hamilton-Smith, N. (ed.), *The Computer and Literary Studies*, Edinburgh: Edinburgh University Press.

$$\frac{f(x) - f(xy)}{f(x)}$$

Betabinomial.

$$-2\left(\begin{array}{l} f(x) \times \log(p_x \times p_y \times p_{\bar{x}} \times p_{\bar{y}}) \\ -\left(\begin{array}{l} f(xy) \times \log(p_{xy} \times p_{\bar{x}\bar{y}}) \\ + f(\bar{x}y) \times \log(p_{\bar{x}y} \times p_{x\bar{y}}) \end{array}\right) \end{array}\right)$$

Log-likelihood Ratio. The log-likelihood ratio is an alternative to the chi-square measure, taking account the co-occurences of rather low frequency counts too. See the referenced paper for detailed information.

Reference: Dunning, T. (1993), "Accurate Methods for the Statistics of Surprise and Coincidence", *Computational Linguistics*, 19(1), pp. 61-74.

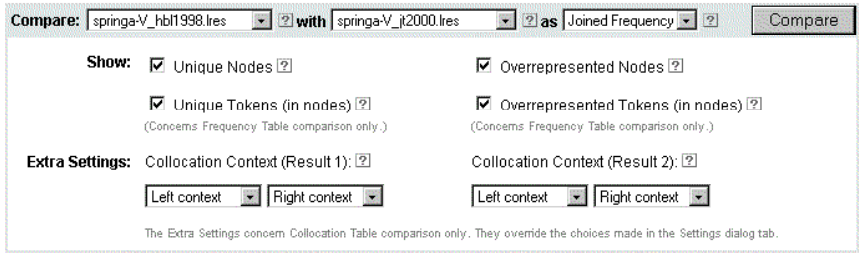
⚠ Only the collocation values between the node and *one* collocate at a time is calculated, thus *n*-collocate collocations (where *n* > 1) are not calculated. (Use *Lemmie API 2.0* to do that.)

⚠ Collocation calculus *within* the node is not done. (Use *Lemmie API 2.0* to do that.)

⚠ When calculating collocations in corpora associated with **user addition files:**

- Be prepared that the calculus will take a considerable amount of time.
- Collocates with an joint overall frequency in the targeted corpora *higher* than the maximum size of the memory buffer will be calculated as the joint frequency was the same as this maximum size limit. Collocation scores of high-frequency collocates are thus error-prone when using user addition files.

💡 Read more about collocation table settings in section 5, Settings dialog tab, below.



The main interaction element on the Compare dialog tab. *On the Compare dialog tab, you can set some comparison details and override selections you have made on the Settings dialog tab.*

Overview

The *Compare dialog tab* is used for comparing previous results with each other, i.e. for viewing the statistical differencies between two queries. In order to compare results, you will need to have done the corresponding queries earlier on the *Search dialog tab*, and then selected to store the results in binary format.

The differencies can either be measured directly on the *frequencies* of the different nodes in the results, or on the frequencies or other scores of the *collocations* the nodes appear in.

In the main interaction element on the *Compare dialog tab* (see screenshot above), you can select which primary (the left-most menu) and which secondary (to the right of the left-most menu) result to use. The hits in the primary result are going to be

compared against the hits in the secondary result, and any statistically significant hits in the primary result will be returned. In a way, the primary result is a test case, which is tested against a benchmark or reference result, i.e. the secondary result.

What "statistically significant" denotes is defined by you on the *Settings dialog tab* in the sections marked *Frequency Table Settings* and *Collocation Table Settings* respectively. (In the case of comparing collocation tables, some settings can be overridden directly on the *Compare dialog tab*. More on this topic further down.)

You must select which type of comparison you want to do (frequency of collocation table) in the menu to the left of the button marked *Compare*. Naturally, pushing the *Compare* button will launch a comparison.

Before launching the comparison, you will want to select the kind of result you want to have. To do this you check or uncheck the checkboxes to the right of the **Show** label. (The semantics of the checkboxes are explained further down in conjunction with the definition of the comparison types.)

Last, there are some **Extra Settings** that you can fiddle with if you are going to do a comparison of collocation tables.

Comparing Joint Frequencies

UNIQUE - Full nodes only found in frequency table of springa_ADV_hbl1998.lres

Abs	Rel	Node
6	0,0000006449	sprang omkring
6	0,0000006449	springer fram
4	0,0000004299	springa iväg
3	0,0000003224	sprang ut
3	0,0000003224	springa till
3	0,0000003224	springer också
3	0,0000003224	sprungit fram

Unique nodes. *Here two queries with the query expression* `[bf='springa' pos='V']` *in* Hufvudstadsbladet 1998 *and* Jakobstads Tidning 1999 *are compared and the unique (in respect of word forms) nodes in* Hufvudstadsbladet 1998 *are shown.*

When comparing joint frequencies between a primary (test) result and a secondary (benchmark) result, the main idea is to extract such nodes (or token in nodes) that, in respect of the selected display features in the primary result

1. do not exist *at all* in the secondary result
2. are *significantly overrepresented* in comparison with the secondary result

If all comparison results are requested, **WWW-Lemmie 2.0** will return four (4) comparison tables. You can select the kind of comparison results you want by checking or unchecking the checkboxes to the right of the **Show** label. The alternatives are:

1. Unique nodes
2. Unique tokens (in nodes)
3. Overrepresented nodes
4. Overrepresented tokens (in nodes)

Unique Nodes. If this box is checked, a table with all such (combinations of selected display features of) nodes in the primary result that do not exist *at all* in the secondary result will be shown.

Example: The query expression `[pos='Verb'][wf='lujaa']` matches some nodes consisting of two tokens each. If the word form is the only selected display feature, then e.g. *juosta lujaa*, *mennä lujaa* might be two displays of two nodes matching the query expression. If *juosta lujaa* only exists in the primary result and thus not at all in the secondary result, then it is returned as a unique node.

💡 The *selected display features* of a node, is the features set on the *Settings dialog tab* in the section labeled *Frequency Table Settings*. The default is the word form, but you can set other display values, forcing **WWW-Lemmie 2.0** to compare other features of tokens than just the word forms.

Unique tokens (in nodes). If this box is checked, a table with all such tokens *within* the nodes in the primary result that have selected display features that do not exist *at all* in the secondary result will be shown. Given the example above, *juosta* and *lujaa* are word form features of tokens in the node *juosta lujaa*, so e.g. *juosta* might be (a unique word form feature of) a token within the node *juosta lujaa*.

Overrepresented nodes. If this box is checked, then the result will be a table containing all such nodes in the primary result that,

in respect of the selected display feature(s), have a frequency which is *N* times higher than the frequency of the same selected display features in the node in the secondary result. The value of *N* is set on the *Settings dialog tab* under *Frequency Table Settings*. The setting is marked *Minimum score (when comparing frequencies)*.

Overrepresented token (in nodes). If this box is checked, a table with all such tokens in the nodes in the primary result will be built, where the frequencies of the tokens, in respect of their selected display features, are *N* times higher than the frequency of the tokens with the same features in the secondary result.

The comparison tables with unique nodes or tokens are interpreted as the regular joint frequency tables, i.e. they contain three columns marked *Abs*, *Rel* and *Node* (or *Token*, depending on the table). The *Abs* column contains the absolute frequency (in the primary result) of the selected display features in the nodes or tokens that are unique. The *Rel* column contains the relative frequency and the *Node* or *Token* column contains the unique display features of the node or token in question.

The comparison tables with overrepresented nodes or tokens contain two columns, *Quotient* with the comparison quotient (= denoting how much more frequent the display features in the overrepresented nodes or tokens are in the primary result in comparison with the secondary result) and *Node* or *Token* containing the overrepresented display features of the nodes or tokens.

Read more about the display features of the tokens to be examined in section 5, Settings dialog tab. You can select one or more of such features. If several features are selected, then frequency table comparison is done on the combination of those selected display features. The name of the setting is *Features to display*. By default the word form is used as display feature.

Comparing Collocation Tables

Compared: spriga-V_hbl1998.lres with spriga-V_jt2000.lres **Type:** coll **Min. Quot.:** 10 **Compared Value(s):** collocationRelativeFreq **Time:** 9 sec. **Min. Abs. Freq.:** 2 **Min. Rel. Freq.:** 0.00000002 **Min. Score:** 50 **Formula:** chi_square, dice **Node Features:** bf **Collocate Features:** bf **Collocation Within Same Element:** yes **Context (First Result):** 0 - 1 **Context (Second Result):** 0 - 1

UNIQUE - Collocations only found in collocation table of spriga-V_hbl1998.lres

Collocation	chi_square	dice	Abs (Left/Right)	Rel	Node Abs	Collocate Abs
[bf='spriga'] [bf='gatu_stafett']	16277,3124430157	0,0103359173	2 (0 / 2)	0,0000002150	381	6
[bf='spriga'] [bf='nerför']	7510,4578765122	0,0101522843	2 (0 / 2)	0,0000002150	381	13
[bf='spriga'] [bf='av_och_an']	2709,5621048522	0,0095923261	2 (0 / 2)	0,0000002150	381	36
[bf='spriga'] [bf='fram_och_tillbaka']	1709,8338939490	0,0091324201	2 (0 / 2)	0,0000002150	381	57
[bf='spriga'] [bf='iväg']	1134,6028810594	0,0110650069	4 (0 / 4)	0,0000004299	381	342
[bf='spriga'] [bf='show']	465,6717023424	0,0067911715	2 (0 / 2)	0,0000002150	381	208
[bf='spriga'] [bf='hem']	74,0026420624	0,0019150974	3 (0 / 3)	0,0000003224	381	2752
[bf='spriga'] [bf='genom']	69,8703038506	0,0012405409	5 (0 / 5)	0,0000005374	381	7680
[bf='spriga'] [bf='ner']	69,7763277856	0,0018253727	3 (0 / 3)	0,0000003224	381	2906

Unique collocations. *In this example two results of the queries* `[bf='spriga' pos='V']` *in* Hufvudstadsbladet 1998 *and* Jakobstads Tidning 2000 *are compared in respect of which collocations they take in the position immediately to the right of the node. The base form feature has been used as display value, so the collocations and all calculated scores are based on this feature. The collocations in the table are such that exist only in* Hufvudstadsbladet 1998 *and not at all in* Jakobstads Tidning 2000. *The table is sorted according to the collocations' chi-square scores and only collocations with scores over 50 have qualified. Collocations with an absolute frequency less than 2 have also been discarded. (As you can see, some multi-word units, such as* av och an *and* fram och tillbaka *have been joined into one base form. This is a side-effect of the morphosyntactic annotator used for Swedish texts.)*

When comparing collocation tables, **WWW-Lemmie 2.0** first calculates the collocations around the nodes in the primary and secondary results. The calculations are done on the selected display features of the node and the selected display features of the collocate candidates. (The formulae and features to be used are set on the *Settings dialog tab* in the *Collocation Table Settings* section.) When the collocations have been calculated, such collocations from the primary collocation table that do no beat the minimum scores (*Minimum Absolute Frequency*, *Minimum Relative Frequency* and *Minimum Score (with primary formula)*) are filtered out. Finally, the collocation table calculated using the primary result is compared against the table calculated using the secondary result. By default, the relative frequencies of the collocations are compared, but this can be changed on the *Settings dialog tab.*

Keep in mind that the collocation tables contain the collocation scores of the selected display features in the node and the collocate, thus it is not the association of tokens or nodes as units in their own right that has been calculated, but the association

between some feature(s) of the tokens and nodes. (To compare collocation tables of tokens or nodes as "full" units, you need to select *all* features as display features on the *Settings dialog tab.*)

As when comparing frequency tables, collocation table comparison will also return a few comparison tables. Which these tables are, can be selected or unselected by checking or unchecking the boxes to the right of the **Show** label. You can ask **WWW-Lemmie 2.0** to show such collocations that only exist in the collocation table of the primary result (in respect of the selected display features) and such collocations that are statistically overrepresented in the collocation table of the primary result (in respect of the selected display features). As with frequency table comparison, the meaning of "statistically overrepresented" is defined by you yourself on the *Settings dialog tab*. The default is that a collocation is regarded as overrepresented if the value being compared is at least the double in the test result in comparison with the reference result.

In contrast to frequency table comparison, collocation tables cannot be compared according to the display features of the individual tokens *within* the collocations. (You cannot for instance disregard of the node and simply compare the collocates in the two collocation tables.) In practice, this means that the nodes in the primary and secondary results need to be more or less the same if there is to be a point in doing the comparison. (If you compare the collocations of two word forms that are totally different and select the word form to be the display feature of the nodes, then naturally all collocations based on the primary result will be unique to that result as the nodes are totally different from the ones present in the secondary result.) However, you can work around this problem, by choosing such a feature as node display feature that is common to both the primary and the secondary result (e.g. part-of-speech). Then in practice, the comparison is done on the *collocates* and not the *collocations*, as the nodes are the same. But, keep in mind that the collocation scores will differ when using for instance part-of-speech as the selected display feature of the node! You might be calculating something totally different from what you expected.

When comparing two collocation tables, you might want to set different collocation windows for the two result being compared. This is done by altering the values of the menus to the right of the **Extra Settings** label. If you do not change the values in these menus, the collocation window size is set according to the settings on the *Settings dialog tab*.

The result tables of collocation table comparison are read as the regular collocation tables, which are described above in section 3, Search dialog tab. There is one exception: In tables with overrepresented collocations, an additional column is generated containing the comparison quotient. The column is at the very left of the table and the data is sorted on this quotient. By default, the comparison quotient is calculated by dividing the relative frequencies of a collocation in the primary and the secondary result. This can be changed on the *Settings dialog tab.*

Here are some use cases to get you started with comparing collocation tables:

- Use a query expression to fetch results from two different corpora or sets of corpora, then compare these results in respect of word form, base form or whatever display feature you want. The result will show such collocations that are significant for the primary result in contrast with the secondary result.
- Compare the collocations of two related words, as two synonyms, two antonyms etc. (e.g. `[bf='ostaa' pos='Verb']` and `[bf='myydä' pos='Verb']`). Select the part-of-speech as display feature for the node and base form as display feature for the collocates. Admire the differences in lexical context.
- Compare a result with itself using different collocation windows. This will give you an estimate on what positions, in respect of the node, the collocates tend to prefer.

5 Settings Dialog Tab

Overview

The *Settings dialog tab* is the command center of **WWW-Lemmie 2.0**. On this dialog tab most of the user's query and display settings are managed. The settings have been placed on one dialog tab especially in order for the *Search dialog tab* to be as clean and simple as possible. Moreover, as many settings (e.g. which corpora are queried) are common to many functions (across dialog tabs), it is justified to keep the settings in one place.

The *Settings dialog tab* is divided into these five sections (each described in more detail further down):

1. **Common Settings**
2. **Corpora In Use**
3. **Concordance Settings**
4. **Frequency Table Settings**
5. **Collocation Table Settings**

Common Settings

The following settings are available in the **Common settings** section:

- Memory buffer size
- Query restriction
- Interface language

Memory buffer size

When asked to perform a query, **WWW-Lemmie 2.0** reads data into memory up to a certain maximum level after which the reading stops and the rest of the data is left on disk for later retrieval (upon request) in a new *result chunk*. the size of the memory buffer determines how many hits **WWW-Lemmie 2.0** is entitled to hold in memory at a maximum. the value is set in the **Memory buffer size** menu. Keeping the value low, will speed up queries that result in several hits, but then not all the results will be fetched at once, which might have some negative consequences. If the memory buffer is big, common queries will take a considerable amount of time to finish, but the result chunks will also be bigger and possibly more reliable.

See section 3, Search dialog tab, for more information about the memory buffer.

Query restriction

When performing a query, **WWW-Lemmie 2.0** generally does not care within which elements the matching tokens occur. In other words, matches from headings, bylines, text openers, captions etc. are mixed. Setting a specific **query restriction** in the menu with the same name will force **WWW-Lemmie 2.0** to only return such matches that are found somewhere below the required element.

The elements correspond directly to the element names in the underlying XML-formatted corpus files:

- <s> - sentence
- <p> - paragraph
- <head> - title
- <opener> - text lead
- <closer> - text closer
- <caption> - caption (generally associated with a picture)
- <q> - subheading
- <byline> - byline

When making a phrasal query containing empty search parameters ([] that match any arbitrary token), it is generally a good idea to set the **query restriction** to *only within nearest element*. By doing this, you ensure that all parts of a match come from the same element (regularly, the same sentence).

Normally, a restricted query takes a bit more time to finish than an unrestricted one.

Interface language

WWW-Lemmie 2.0 talks Finnish, Swedish and English. Select which language you want to use in the **Interface language** menu.

Corpora in use

The **Corpora in use** section contains the available **predefined corpora** and **custom corpora**. (There will be no custom corpora available, unless you have compiled custom corpora of your own either on the *My corpora dialog tab* or using the *Lemmie API 2.0*.)

Check the corpora which you want to use by default. Whenever you perform a query on the *Search dialog tab* and use "from settings" as targeted corpora, the corpora you have selected here will be queried.

Concordance settings

The following settings are available in the **Concodrance settings** section:

- Left context
- Right context
- Primary sort key and direction
- 2nd sort key and direction
- 3rd sort key and direction
- 4th sort key and direction
- 5th sort key and direction
- 6th sort key and direction
- Primary sort feature
- 2nd sort feature
- 3rd sort feature
- 4th sort feature
- Features to display

Left context

Use this menu to set the amount of tokens in the context to the *left* of the node.

Right context

Use this menu to set the amount of tokens in the context to the *right* of the node.

Primary sort key and direction

WWW-Lemmie 2.0 can sort a concordance on any feature of any token anywhere in the concordance. Use the **primary sort keys and direction** menus to select

1. which position on the lines in the concordance that are to be used as primary sort key
2. which sort direction is to be used on the selected position, *forward* denoting from the left to the right and *backward* denoting from the right to the left

2nd thru 6th sort key and direction

If the sort is not finished after sorting the concordance according to the primary sort key, then **WWW-Lemmie** will sort the concordance according to the 2nd all the way thru the 6th sort key if necessary.

Primary sort feature

If the sort key corresponds to a position in the concordance, then the sort feature is a feature of a token at such a position. Thus, the primary sort feature is used when sorting the tokens at the position of the primary sort key. By default, the primary sort feature is set to *word form*, but using the **primary sort feature** menu, you can change the default to something else.

2nd thru 4th sort feature

Use these menus, if you would like to sort tokens according to other criteria than just the primary sort feature.

Features to display

This menu is used for setting which features are displayed of all the features in the tokens in a concordance. By default, only the word form is used as display feature, but you can select any that you like.

Frequency Table Settings

The following settings are available in the **Frequency table settings** section:

- Sort order
- Minimum absolute frequency
- Minimum relative frequency
- Minimum score
- Features to display

Sort order

A frequency table can be sorted either *alphabetically* on the display feature(s) or by the *descending frequency* of the tokens with the selected display feature(s). Default is to sort by descending frequency.

Minimum absolute frequency

Use this menu to cut off low-frequency tokens from the frequency table.

Minimum relative frequency

Use this menu to cut off low-frequency (relative to the full size of the corpus) tokens from the frequency table.

Minimum score

The **minimum score** value is used when comparing frequency tables (on the *Compare dialog tab*). The quotient, when dividing the relative frequency of an element in a test table with the relative frequency of the same element in a reference table, must beat this minimum score if the element is to be regarded as overrepresented in the test table.

Features to display

This menu is used for setting which token features are to be displayed and used when calculating the frequency table. By default, only the word form is used as display feature, but you can select any that you like. (Selecting, for instance, *part-of-speech* as the only display feature, would calculate and display a frequency table of different part-of-speech labels found in the targeted corpora with the query expression in use.)

Collocation Table Settings

The following settings are available in the **Collocation table settings** section:

- Minimum absolute frequency
- Minimum relative frequency
- Minimum score (with primary formula)
- Minimum score (when comparing collocations)
- Which numeric value to use for comparation
- The size of the left-side window
- The size of the right-side window
- Which statistic formulae to use
- The primary formula
- Which node features are used for calculus and display
- Which collocation features are for calculus and display
- Whether the node is to be treated as a single unit or not
- Whether the collocations mustn't contain a sentence break

Minimum absolute frequency

Collocations with absolute frequencies lower than the **minimum absolute frequency** will be discarded from the collocation table.

Minimum relative frequency

Collocations with relative frequencies lower than the **minimum relative frequency** will be discarded from the collocation table.

Minimum score (with primary formula)

Collocations with a lower score (using the primary formula) than the value of this setting will be discarded from the collocation table.

Minimum score (when comparing collocations)

When looking for overrepresented collocations in two collocation tables (on the *Compare dialog tab*), such collocations that have a quotient lower than the value of this setting will be discarded from the collocation table. The quotient is calculated according to

one of these values: the *relative frequency of collocation*, the *collocation score (with the primary formula)*, the *absolute frequency of collocation*, the *absolute frequency of node*, the *absolute frequency of the collocate*, the *absolute frequency of the collocation when the collocate is on the left of the node* or the *absolute frequency of the collocation when the collocate is on the right of the node*.

Which numeric value to use for comparation

This setting is used for selecting which type of value is used when calculating the comparation quotients between two collocation tables.

Left-side window

Modify this setting to define how many tokens to the left of the node in the running text is used as collocate candidates.

Right-side window

Modify this setting to define how many tokens to the right of the node in the running text is used as collocate candidates.

Formulae in use

Select the statistic formulae that will be used for collocation calculus.

Primary formula

Select the primary formula for collocation calculus. The collocation table will be sorted according to these primary scores (in descending order).

Node features used for calculus and display

Select those token features of the tokens in the node that will be used for collocation calculus and display. By default, the word form is used.

Collocation features used for calculus and display

Select those token features of the collocate candidates that will be used for collocation calculus and display. By default, the word form is used.

Whether the node is to be treated as a single unit or not

If this checkbox is checked, the node will be treated as a single unit. This means that even if there are several different occurences of the selected node features, these will be regarded as parts of the same unit. (Example: given that the word form is the selected node feature, a query `[bf="kello"]` (Fi. for "watch" or "clock") might contain these three different matches: *kellon* (Genitive Singular), *kellot* (Nominative Plural) and *kellojen* (Genitive Plural). If the nodes are treated as a single unit, then these three different nodes will still be regarded as a single unit.)

Whether the collocations mustn't contain a sentence break

If this checkbox is checked, the nodes and collocates must occur within the same sentence.

Update, Reset and Fetch Defaults

Before any changes take effect, they must be stored. This is done by clicking the **Update All Settings** button.

 If you flip to another dialog tab from the *Settings dialog tab* without saving first, all changes will be lost.

Press the button marked **Reset To Previous** if you want to reset the settings to what they where when you loaded the dialog tab.

If you want to load the "factory defaults", i.e. override *all* your private settings with the default values, then press the **Fetch Defaults** button. As a security precaution, you must then press the **Update All Settings** button for the defaults to actually be stored as your own settings.

⚠ All buttons effect all five sections on the *Settings dialog tab*.

6 My Results Dialog Tab

The *My Results dialog tab* is used for managing saved results of past queries.

(Results are saved by clicking the button on the *Search dialog tab* after a query has been made.)

The *My Results dialog tab* contains a table with one saved result for each row. On the tab you can **delete** results, **add or modify descriptions** and **open** results (as concordances).

To **delete** results, check the boxes to the left on the lines of the results you want to delete (in the column marked *Delete?*) and click the trashcan ( ) icon to proceed.

To **add or modify descriptions**, write or modify a description in the text box on the row of the result in question in the column marked *Description* and click the button marked *Update Descriptions* when finished.

Click the icon to open the result on the row in question. **WWW-Lemmie 2.0** will switch to the *Search dialog tab* and open the result there as a concordance.

Click the *Reset To Previous* button to reset the dialog tab to what it was when it was opened, removing all recent changes.

7 My Corpora Dialog Tab

Overview

The *My Corpora dialog tab* contains information about the available predefined and custom-made corpora, such as the corpus sizes (in tokens and documents) as well as possible associated user addition files. You can also create new custom corpora on this tab.

User addition files

You can add a user addition file to any of the available corpora. A user addition file is an XML file containing tagging corrections and additions of your own to a certain corpus. When a user addition file is in use, any information in it regarding a specific token in a corpus will override the information given for that specific token in the lexical database.

⚠ User addition files can make the queries somewhat slower.

The precondition for using a user addition file is that you have one to add; **WWW-Lemmie 2.0** will query user addition files created with *Lemmie API 2.0*, but it will not create them for you, nor write to them. In other words, you cannot use **WWW-Lemmie 2.0** as a tag editor, but if you have written such a program yourself using the API or somebody has supplied you with a file of tagging corrections/additions, then **WWW-Lemmie 2.0** will obey these corrections/additions upon request.

For each corpus on the *My Corpora dialog tab* there is a menu containing available user addition files (XML files with the file extension `.luaf`) in your private Lemmie directory (`~/.lemmie`). Select the file you want to use in the menu and click *Update User Addition Files* (predefined corpora) or *Update User Addition Files and Descriptions* (custom corpora). If the only available option is *None*, then there are no user addition files in your private Lemmie directory.

Managing custom corpora

Creating a custom corpus from scratch

Besides supplying support for the management of user addition files, the *My Corpora dialog tab* also contains a form with which

you can create your own custom corpora given the documents in the predefined corpora.

You select which predefined corpora the documents may come from originally in the select box labeled *Select corpora*. Then you supply a list of options that the metadata of a document must comply to in order for the document to be stored in your new custom corpus. You can leave some elements empty, in which case the values of these elements do not matter. You can also add a descriptive text to the custom corpora.

In order for **WWW-Lemmie 2.0** to start compiling the new custom corpus, you must click the button labeled *Compile New Custom Corpus*. Make sure you have entered a unique id in the text field with that name.

⚠ A custom corpus is allowed to contain 1-1000 documents. Empty custom corpora will not be generated and a custom corpus will finish compiling immediately when the maximum number of documents have been stored in it.

Managing custom corpora that already exist

The *My Corpora dialog tab* contains a table of existing custom corpora (if you have created any). By clicking the icon for any of these corpora, **WWW-Lemmie 2.0** will open a new browser window and list the documents in the custom corpus in question in this new window. You can delete documents from the custom corpora in the window and look at the full contents of the documents. To delete a document from a custom corpus, check the box in front of the title of the document and then click the icon. To view the full contents of the document, click the icon on the same line as the document. Yet another window will open displaying the text of the selected document.

Deleting a document from a custom corpus, does not delete the original copy of the document in the predefined corpus from which the document was originally retrieved. Thus, you can always get a deleted document back. Read more below.

Hover with the mouse pointer above the icon to see more information about the document on the same row.

Click a column's title to sort the document list according to the values of that column.

Adding documents to custom corpora that already exist

A custom corpus is not static. As stated in the section above, documents can be dropped from a custom corpus. In addition to this, documents can be added to custom corpora at any time. When you have made a query on the *Search dialog tab* and got a concordance as result, you can open up each of the concordance lines with more context by clicking the or icons. In the window that opens, you can choose to save the document, from which the snippet of text is fetched, to a custom corpus.

⚠ You cannot add documents of your own to a custom corpus, only such documents that exist in some predefined corpus. If there is a need for such functionality, please send a request to ling@csc.fi and we will see what we can do.

8 Manual Dialog Tab

You are reading the manual.

9 FAQ Dialog Tab

The *FAQ dialog tab* contains a list of the most common questions about the **functionality** and **interface** of **WWW-Lemmie 2.0** as well as of the **content** of the Language Bank.

10 About Dialog Tab

The *About dialog tab* contains a short technical description of **WWW-Lemmie 2.0** as well as credits and a change log.

11 Search Syntax

Overview

WWW-Lemmie 2.0 is foremost a corpus *query* tool. In order for the system to understand what query you want it to perform, you must compose a *query expression* using either **simple syntax** or **advanced syntax**. The query expression is written in the text field labeled *Query Expression* on the *Search dialog tab*.

Simple Syntax

If you are only interested in looking up word forms in your query, and the queries you want to do consist of a single word form or a combination of word forms (phrases), then query expressions in **simple syntax** are enough. With simple syntax you can also express *negation* (that some word form may not occur) and *conjunction* (that a word form must occur in conjunction with another). Moreover, you can use `.` (period) and * (star) for truncation. Period matches one arbitrary character and the star matches zero to infinite arbitrary characters.

⚠ Truncation is not allowed in prefix or infix position if you are using a user addition file.

⚠ Truncation never exceeds word form boundaries.

⚠ Queries are always treated case-insensitively.

Operation	Example	Note
Single word form	`tarmokas`	

Single word form (with truncation)	`tarmo*`
	`t.rmo`
	`t.rmo*`	

Fixed phrase	`kaunis ja ylpeä`	

Fixed phrase (with truncation)	`kauni* ja ylpeä`
	`*kaunis ja *ylpeä`
	`kauni* ja ylpe*`	

Flexible word form combination	`kaunis AND ylpeä`
	`kaunis + ylpeä`
	`ylpeä AND kaunis`
	`ylpeä + kaunis`
		1. The conjunction operator (*AND* or *+*) allows zero to five arbitrary words to appear between the word forms in the query.
		
		2. The order of the word forms is important, thus *AND* is not a regular boolean operator

Flexible word form combination (with truncation)	`kauni* AND ylpe*` `kauni* + ylpe*`	
Negation	`kaunis AND NOT ylpeä` `kaunis + ! ylpeä`	
Negation (with truncation)	`kauni* AND NOT ylpe*` `kauni* + ! ylpe*`	

Advanced Syntax

You can do a lot more than just look up simple word forms or word form combinations with **WWW-Lemmie 2.0**. In order for this to be possible, there is an **advanced syntax** for you to use.

Advanced syntax is based on the idea of a **query expression** consisting of one or more **search parameters**. These search

parameters contain **required features** and **disqualifying features** which are named key-value pairs where the key is a word token feature name and the value is the required or disallowed value of that feature in a word token in a targeted corpus.

Beside these basic concepts, advanced syntax also covers truncation, iterators and feature lists.

The search parameter

```
[]
```

The primary element in the query expression if the **search parameter**. There can be one or more search parameters in a query expression. A search parameter corresponds to a token in the targeted corpora. Therefore a query expression only containing one search parameter will generate a result with one-token-nodes (unless an iterator is used). Consequently, a query expression consisting of more than one search parameter, will commense a phrasal query.

A search parameter is denoted by **square brackets** (`[` and `]`). A special case is the empty search parameter, `[]`, matching any token whatsoever.

A required feature

```
[key="value"]
```

A **required feature** in a search parameter is such a feature that a token in a targeted corpus must have if it is to match the search parameter. A required feature is denoted by an **equals sign** (=). The feature's name is given to the left of the sign and the required value to the right of the sign within citation marks.

Example: `[bf="hakea"]` Looking up tokens matching the base form feature "hakea" (Fi. "fetch").

A disqualifying feature

```
[key!="value"]
```

A **disqualifying feature** or **disqualifier** is such a feature that a token in a targeted corpus must *not* have if it is to match the search parameter. A disqualifier is denoted by the string `!=`. The feature's name is given to the left of the string and the disallowed value to the right of it within citation marks.

Example: `[bf="kuusi" pos!="Numeral"]` Looking up tokens matching the base form feature "kuusi" (Fi. Noun "spruce" or Numeral "six") and not matching the part-of-speech label "Numeral".

Truncation

```
[key="val*"]
```

Any value in the key-value pair of a a required or disqualifying feature can be truncated with a `.` (period) or a * (star). A period matches one arbitrary character, the star zero to infinite arbitrary characters. The truncation marks can be used anywhere in a feature value, unless user addition files are in use, in which case only truncations marks in suffix position are allowed.

Example: `[bf="kuusi*" pos="Numeral"]` Looking up tokens with base form features that contain the prefix "kuusi" and with the part-of-speech label "Numeral".

Feature lists

```
[key="value1" key="value2"]
```

Any required feature or disqualifier can be given more than one value. This is done by adding required features or disqualifiers with the same key but different values to the search parameter. Regularly, you want to create such feature lists with disqualifiers.

Example: `[wf="tuli*" pos!="Verb" pos!="Common"]` Looking up tokens that match the word form prefix "tuli" and that do not carry the part-of-speech labels "Verb" or "Common".

Iterator

```
{n,m}
```

A search parameter can be iterated. This means that the search parameter in question will be duplicated a defined number of times. The **iterator** is denoted with curly braces (`{` and `}`). The iterator is appended to the search parameter and it contains two integers separated by a comma character, `{n,m}`. The first integer (`n`) denotes the *minimum* number of times the search expression must match tokens in the targeted corpora. The latter (`m`) denotes the *maximum* number of times the search expression may match tokens in the targeted corpora. The following holds: $0 \leq n \leq 9$, $0 \leq m \leq 9$ and $n \leq m$.

Regularly, the iterator will be used in conjunction with an empty search expression, simulating that there can be a vaguely specified number of arbitrary tokens in the middle of a node.

Example: `[bf="juosta"][]{0,2}[wf="lujaa"]` Looking up tokens matching the base form feature "juosta" (Fi. "run" Verb) followed by zero, one or two arbitrary tokens followed by a token matching the word form feature "lujaa" (Fi. "fast" Adverb).

Available token features

Above, required and disqualifying features have been mentioned. It has been said that these are key-value pairs where the value of a feature can vary. But which are then the keys?

The keys in the required and disqualifying features are a defined set of token feature names. The word form feature, for instance, has been given the name `wf`, the base form feature the name `bf` etc. These names, or keys, are used for pin-pointing which word token feature one is defining a value for.

The following table contains the word token features that exist in the corpora by default (more can be used in user addition files if needed). Note that some of the word token features have a defined set of values. The *tag sets* vary for the Finnish and Finland-Swedish documents, as different taggers have been used for the two languages (Kielikone's Textmorfo for Finnish and Lingsoft's SWECG for Swedish). Note all query expressions are handled case-insensitively.

Key	Full name or description	Allowed values
wf	Word form	Any string
bf	Base form	Any string
case	Case	**Textmorfo:** Ab (Abessive), Abl (Ablative), Acc (Accusative), Ad (Adessive), All (Allative), Com (Comitative), El (Elative), Ess (Essive), Gen (Genitive), Genom (nominalized word in Genitive), Ill (Illative), In (Inessive), Instr (Instructive), Nom (Nominative), Part (Partitive), Transl (Translative) **SWECG:** ACC (Accusative), GEN (Genitive), NOM (Nominative), NOM/ACC (Nominative/Accusative), NOM/GEN (Nominative/Genitive), NOM/SG (Nominative/Singular)
definitiveness	Definitiveness	**SWECG:** DEF (Definite), DEF/INDEF (Indefinite/Definite), INDEF (Indefinite)
deponent		**SWECG:** DEP (deponent)
derivationSuff	Derivational suffix.	**SWECG:** DER-are, DER-arinna, DER-else, DER-erska, DER-het, DER/-nde, DER/-ning, DER-bar, DER-ig, DER-isk, DER-lig, DER-ligg
extra	Default feature for various additions in user addition files.	Any string
gender		**SWECG:** NEU (Neuter), UTR (Utrum), UTR-MASC (Utrum-masculine), UTR/NEU (Utrum/Neuter)
grade		**Textmorfo:** Comp (Comparative), Sup (Superlative)
id	Unique identifier of word token occurence.	A number
modality		**Textmorfo:** Cond (Conditional mood), Iinf (1st Infinitive), IIinf (2nd Infinitive), IIIinf (3rd Infinitive), IVinf (4th Infinitive), Vinf (5th Infinitive), Ipartic (Present Participle), IIpartic (Perfect Participle), Ind (Indicative), Imper (Imperative), Pot (Potential mood) **SWECG:** CNJV (Conjunctive), IMP (Imperative), INF (Infinitive)
msd	Morhosyntactic description, the *tag* as it is written in the XML files on `corpus.csc.fi`. The cell to the right contains patterns for composing such tags.	**Textmorfo:** *[case][number][possSuff][person][grade] [case][number][voice][modal][possSuff][person][grade] [tense][voice][modal][possSuff][person]* **SWECG:** *[derivationSuff][gender][definitiveness][number][case] [voice/deponent][tense/modality/infinitive][number]*
number		**Textmorfo:** SG (Singular), PL (Plural) **SWECG:** PL (Plural), SG (Singular), SG/PL (Singular/Plural)
person		**Textmorfo:** 1P (1st person), 2P (2nd person), 3P (3rd person)
pos	Part-of-speech	**Textmorfo:** Abbrev, Adjective, Adjective-Noun, Adverb, Code, CompPart, Conjunction, Delimiter, Interjection, Noun, Noun-Noun, Numeral, Preposition, Pronoun, Proper, Verb **SWECG:** A (Adjective), A/N (Adjective/Noun), ABBR (Abbreviation), ADV (Adverb), ADV/PREP (Adverb/Preposition), CC (Co-ordinating conjuntion), DET (Determiner), INFMARK (Infinitive marker), INTERJ (Interjection), N (Noun), PREP (Preposition), PREP/ADV (Preposition/Adverb), PRON (Pronoun), SC (Subordinating conjunction), V (Verb)
possSuff	Possessive suffix.	**Textmorfo:** S (Singular possessive suffix), P (Plural possessive suffix)
tense		**Textmorfo:** Pr (Present), Imp (Imperfect) **SWECG:** PAST, PRES (Present), SUPINE
voice		**Textmorfo:** Act (Active), Pass (Passive) **SWECG:** ACT (Active), PASS (Passive)

⚠ Note that not all word token features can be combined with each other.

⚠ The cell in the column with allowed values for the word token feature *msd* above contains patterns for how this value is created. Note that not all realizations of those patterns are sensible.

⚠ Queries are always treated case-insensitively.

⚠ In documents tagged with SWECG (= all Swedish texts), the base form feature values will contain underscore characters marking out the boundaries between the words in a compound baseform. These underscores are needed in the query expressions too. (Example: **`[bf='akter_snurra']`**, not ~~`[bf='aktersnurra']`~~)

12 Usage examples

Overview

These usage examples are written using advanced syntax. Click on any query expression to commence a search.

Examples in Finnish

Example	Description
`[wf='kuusen']`	Matches tokens with the word form *kuusen* (Fi. "spruce" genitive singular) in Turun Sanomat 1998.
`[bf='kuusi' pos='Numeral']`	Matches tokens with the base form *kuusi* as a numeral (Fi. "six") in Karjalainen 1991.
`[bf='kuusi' case='Gen' pos!='Noun']`	Matches tokens with the base form *kuusi* and that are in genitive case but not nouns. Using Demari 1999.
`[bf='koristella' pos='Verb'][bf='kuusi' pos='Noun']`	Matches tokens that are verbs and have the base form feature value *koristella* and that are followed by tokens with the base form *kuusi* and that are nouns. Using Keskisuomalainen 1999.
`[bf='koristella' pos='Verb'][]{0,2}[bf='kuusi' pos='Noun']`	Matches tokens that are verbs and have the base form feature value *ostaa* and that are followed by zero to two arbitrary tokens and then a token which is a noun and has the base form *kuusi* . Using Keskisuomalainen 1999.
`[pos='Verb'][bf='kuusi' pos='Noun']`	Matches any verb followed by a token which has the base form *kuusi* and which is a noun. Using Demari 1999.
`[pos='Verb'][pos='Adjective'][bf='kuusi' pos='Noun']`	Matches any verb followed by any adjective followed by a token which has the base form *kuusi* and which is a noun. Using Demari 1999.
`[bf='kuusi' pos='Numeral'][pos='Noun']`	Matches tokens with the base form *kuusi* and the part-of-speech label "Numeral" followed by any noun. Using Karjalainen 1997.

Examples in Swedish

Example	Description
`[wf='jordgubbar']`	Matches tokens with the word form *jordgubbar* in Jakobstads Tidning 2000.
`[bf='jord_gubbe']`	Matches tokens with the base form *jordgubbe* (the underscore marks the word boundary in the compound base form) in Hufvudstadsbladet 1998.
`[pos='A'][bf='jord_gubbe']`	Matches all adjectives that are followed by tokens with the base form *jordgubbe*. Using Hufvudstadbladet 1998.
`[pos='V'][]{0,1}[bf='*gubbe']`	Matches all verbs that are followed by zero or one arbitrary token(s) followed by any token which has *gubbe* as suffix in the base form token feature value. Using Jakobstads Tidning 1999.

Terminology

In the following, some terms used in this manual in particular and to some extent in other parts of **WWW-Lemmie 2.0 are described in more detail.**

Term	Definition
Absolute Frequency	The number of occurences of some word token feature or combination of word token features in the targeted corpora.
Advanced syntax	The query language used in **WWW-Lemmie 2.0**. (☐*search parameter*)
Chunk	= *result chunk*
Collocate	A word token that occurs together with a specific node in a collocation (in respect of some selected display features). (☐*collocation*, *node*)
Collocate candidate	A word token that is used when calculating collocations. The *collocates* are a subset of the collocate candidates (☐*collocation*, *collocate*)
Collocation	A combination of a node and a collocate that occurs more frequently together (in respect of some selected display features) than one would expect given the individual frequencies (of the selected display features) of the node and the collocate.
Collocation context	= *collocation window*
Collocation score	A measure of the co-occurence strength of a collocation calculated with some statistic formula.
Collocation table	A table of collocations ordered accoring to their collocation scores. (☐*collocation*, *collocation score*)
Collocation window	The span to the left and to the right of the node in the running text in which all word tokens will be considered as collocate candidates. (☐*collocate candidate*, *node*)
Command bar	The user interaction element on the *Search dialog tab* where one can continue working with a result just fetched, e.g. view the result in another result type or save it to disk. (☐*result type*)
Common query	A query expression generating thousands of hits.
Comparison table	A table with results of a comparison between two frequency tables or collocation tables. (☐*frequency table*, *collocation table*)
Concordance	A table with nodes with neighbouring context. (☐*context size*)
Context size	The number of word tokens to the right and to the left of the nodes that will be shown in a concordance. (☐*concordance*)
Custom corpus	A corpus compiled by the user containing documents from the predefined corpora in the Language Bank of Finland. (☐*predefined corpus*)
Dialog tab	A user-interaction form (web page) where some specific function can be performed. **WWW-Lemmie 2.0** has eight dialog tabs: • *Search* - for quering the lexical database and displaying results • *Compare* - for comparing results from queries • *Settings* - for managing user-defined settings • *My Results* - for managing such results from queries that have been stored to disk in binary format • *My Corpora* - for creating, deleting and managing the corpora in use • *Manual* - for getting extensive help • *FAQ* - for getting answers on the most frequent questions • *About* - for getting an overview of **WWW-Lemmie 2.0**
Display feature(s)	A set of word token features used for calculating frequencies and collocations and for displaying tokens on screen.
Disqualifier	= *disqualifying feature*
Disqualifying feature	A word token feature in a search parameter that may not occur in a possible matching token in the targeted corpora, if that token is to be a match. (☐*token feature*, *required feature*)
Feature	= *token feature*
Feature list	= *token feature list*
Frequency	(☐*absolute frequency*, *relative frequency*)

Frequency table A table of frequencies of display features of word tokens in the targeted corpora. (☐*display feature(s)*)

Joint frequency table A frequency table, where the frequencies in the individual corpora of the targeted corpora have been joint. (☐*frequency table*)

Hit = *node*

Iterator A special construct in advanced syntax with which search parameters can be required to match a defined number of times in a row in the targeted corpora. (☐*advanced syntax*, *search parameter*)

KWIC = *concordance*

Match = *node*

Memory buffer size The maximum number of matches **WWW-Lemmie 2.0** will hold in memory at once. (☐*result chunk*)

Morphosyntactic annotation The (process and) result of making linguistic information in a word token explicit given the word form feature of the token and the context of the word token. E.g. the case feature *partitive* for the word form *autoa*.

Node A unique occurence (among possibly many) of one or more word tokens matching a query expression. (= *match*, *hit*)

Predefined corpus A corpus in the lexical database of the Language Bank of Finland. (☐*custom corpus*)

Primary formula A statistic formula which will be used for collocation calculus and which results will be used for sorting a collocation table. (☐*collocation table*)

Relative Frequency The number of occurences of some word token feature or combination of word token features in the targeted corpora divided with the total number of word tokens in the targeted corpora.

Required feature A word token feature in a search parameter that must occur in a possible matching token in the targeted corpora, if that token is to be a match. (☐*token feature*, *disqualified feature*)

Query bar The user interaction element on the *Search dialog tab* where a query is done.

Query expression A string used for querying, either written in advanced or simple syntax and consisting of one or more *search parameters*. (☐*advanced syntax*, *simple syntax*, *search parameter*)

Query restriction A restriction which forces (all word tokens in) a match to be found within a certain element, e.g. a sentence or heading.

Result chunk A set of nodes matching a given query expression. (☐*memory buffer size*)

Result type The kind of result requested by the user: concordance, joint frequency table, split frequency table or collocation table. (☐*concordance*, *joint frequency table*, *split frequency table*, *collocation table*)

Status bar A area surrounded by borders containing a short notice with the current status of **WWW-Lemmie 2.0**.

Search parameter A part of the query expression matching single tokens. (☐*query expression*, *required feature*, *disqualifying feature*, *iterator*, *truncation*)

Simple syntax The simplified query language used in **WWW-Lemmie 2.0**. Maps internally to advanced syntax. (☐*advanced syntax*)

Split frequency table A frequency table, where the frequencies in the individual corpora of the targeted corpora are displayed separately. (☐*frequency table*)

Sort direction The direction in which the selected sort feature(s) of word token is sorted, forward or backward. (☐*sort feature*)

Sort feature A word token feature used for sorting a set of word tokens in a selected sort direction. (☐*sort direction*)

Sort key A token at a certain position in a concordance, frequency table or collocation table used for sorting. (☐*sort feature*)

Targeted corpus One or a set of predefined and custom corpora that will be used for looking up a certain query expression.

Token An unique occurence of a word, associated with a set of token features. (☐*token features*)

Token feature A pre- or userdefined feature of a word token with a name, such as the word form, part-of-speech or case, and a value such as "nukkea", noun, partitive.

Token feature list A list of token features with the same name but different values. (☐*token feature*)

Truncation A method for allowing a certain character in a search parameter to to match one or any number of arbitrary characters in the targeted corpora.

Type An abstract entity describing a possible realization of a word, in contrast to a token. (☐*token*)

User addition file An XML-file where the user's private corrections of and additions to the features of tokens in a certain corpus are stored and which can be used in query.

Window = *collocation window*

Word token = *token*

Word token feature = *token feature*

Word type = *type*

 Last modified: Wed Sep 1 16:30:45 2004