

# Leipzig Corpora Collection User Manual

Version 1.0

## Table of Contents

Table of Contents .....	1
Introduction .....	2
Installation .....	2
Install on Microsoft Windows .....	2
Step 1: .....	3
Step 2: .....	3
Step 3: .....	4
Step 4: .....	4
Step 5: .....	5
Installation on other platforms .....	5
Walk-through Example .....	6
Using the Corpus Browser .....	8
Description of the Corpora .....	12
Sources of the corpora .....	12
Selection of Sentences .....	13
Word numbers .....	13
Significance Measures .....	14
Technical description of the database schema (for advanced users) .....	15
Database schema .....	15
words .....	15
sentences .....	15
inv_w .....	15
sources and inv_so .....	16
co_n and co_s .....	16
meta .....	16
Using the tables .....	17
Plain text tables .....	17
Additional tweaking .....	17
Speeding things up .....	17
Using alternative fonts .....	18
Creating presentation models for the Corpus Browser (for advanced users) .....	18
Conditions of use .....	22
Disclaimer .....	22
References .....	23
About .....	24

## Introduction

The Leipzig Corpora Collection presents corpora in different languages using the same format and comparable sources, as described in (Quasthoff et al. 06). The corpora are ready to use with the Corpus Browser. Moreover, all data are available as plain text and as MySQL database tables for various applications. They are intended both for scientific use by the corpus linguist as well as for applications such as knowledge extraction programs.

The corpora are identical in format and similar in size and content. They contain randomly selected sentences in the language of the corpus and are available in sizes of 100,000 sentences, 300,000 sentences, 1 million sentences etc. The sources are either newspaper texts or texts randomly collected from the web. The texts are split into sentences. Non-sentences and foreign language material was removed. Because the information which words co-occur with each other is useful for many applications, these data were precomputed and included as well. For each word, the most significant words appearing

- a) as immediate left neighbour
- b) as immediate right neighbour
- c) anywhere within the same sentence

are given. The quality of such co-occurrence increases with the corpus size, so we refer to forthcoming larger corpora.

The authors will add larger corpora and new languages soon. Online access and updates are available at <http://corpora.uni-leipzig.de/>. The Leipzig Corpora Collection will also include other existing corpora in collaboration with the corresponding owners.

This documentation describes the installation process and usage. For advanced users it also describes the database schema used. It also lays out how the corpus browser can be modified to produce a new user-specific view. This browser was designed to allow such changes without any programming skills.

## Installation

This section describes how to install all tools and corpora-databases of the Leipzig Corpora Collection. If you run into any problems regarding the installation or use of the Corpus browser, please refer to our online-FAQ: <http://corpora.uni-leipzig.de/faq.html>. If you plan to install on a Microsoft Windows platform, please proceed with the following section. In any other case please proceed with "Install on other platforms".

In every case you will need a running distribution of MySQL and SUN Java Runtime Environment (JRE, version 1.5 or above). Microsoft Windows users will find these tools on this DVD.

### Install on Microsoft Windows

Please insert the Corpora-DVD into your DVD-Drive. An autorun-screen should appear (fig.1). If not, please run "Autorun.exe" located in the root directory of your Corpora-DVD.

There are five steps for installation:

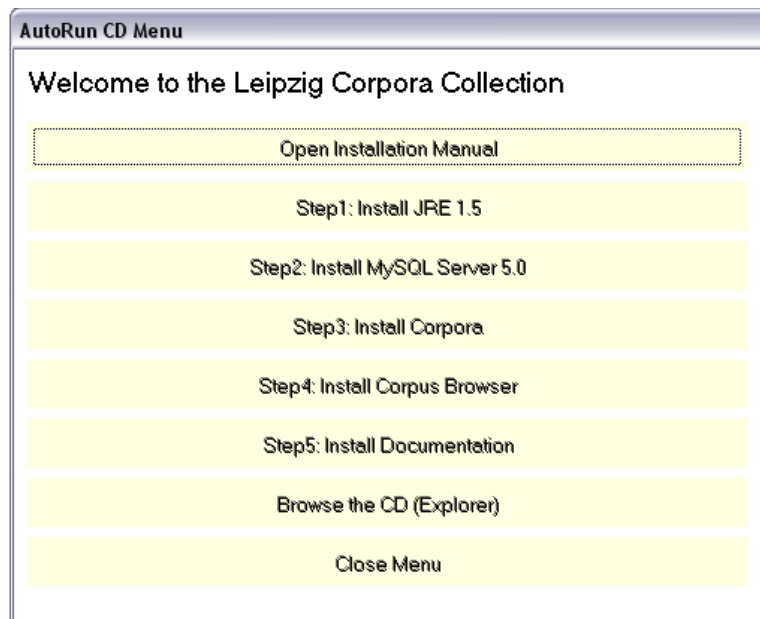


fig. 1

**Step 1:**

This will install SUN JRE 1.5 on your machine if not present. An assistant will guide you through the setup process. If you know there is a working distribution of SUN JRE in version 1.5 or above, you may skip this step. However, make sure that the `PATH` is set to the SUN Java installation, not the Windows native Java executable.

**Step 2:**

In this step MySQL Server 5.0 will be installed. Just follow the instructions of the setup assistant. Please keep in mind where your SQL-data directory is located (fig. 2) and which username (if no additional users are specified this is "root") and password (fig. 3) has to be used in order to connect to the MySQL Server. You will need these data later. You may skip this step if a MySQL Server 5.0 (or higher) is already present. Also make sure to not install the SQL-data directory to a partition that has at least 6GB of free space.



fig. 2



fig. 3

In some cases we experienced problems when upgrading from an older version of MySQL Server. In this case we suggest to uninstall your old version of MySQL Server and to manually delete the old program folder (for example `C:\Programs\MySQL\MySQL Server 4.1`). Please be sure to backup your databases (MySQL data-directory usually located at `C:\Programs\MySQL\MySQL Server 4.1\data`) before doing this.

### Step 3:

This will install all corpora databases from your Corpora-DVD on your machine. An assistant will start up to guide you through this process. It is important that you install the databases into your MySQL data-directory (fig. 4). Usually this directory is located in a subfolder named "data" in your MySQL Server program-directory (see "Step 2" for details). For example: If you installed MySQL Server in `C:\Programs\MySQL\MySQL Server 5.0`, your MySQL data-directory is located at `C:\Programs\MySQL\MySQL Server 5.0\data`.

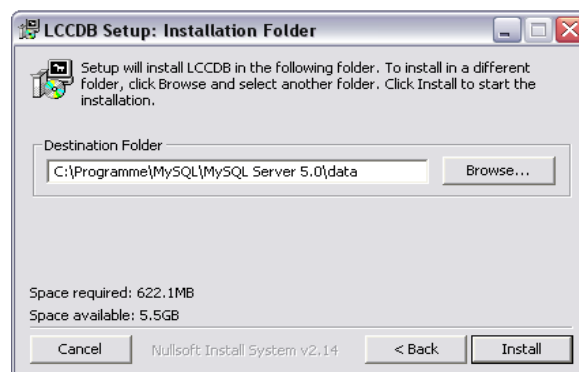


fig. 4

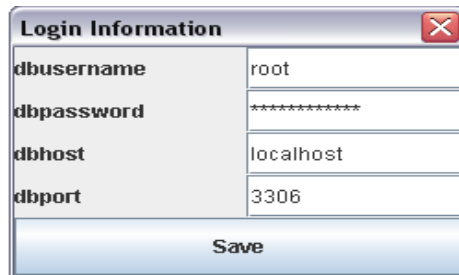
A subfolder named "LeipzigCC/LCCDB" will be created in your programs folder. If you decide to uninstall the corpora databases, you will find an uninstall routine there to do so.

### Step 4:

In this step the Corpus Browser will be installed. An assistant will start up and guide you through the setup process. Afterwards you will find a new subfolder named "LeipzigCC/CorpusBrowser" in your programs folder and in the start menu. There you will find an icon to run the application and an uninstaller. You may also run the Corpus Browser by double-clicking the Corpus Browser icon on your desktop.

When the Corpus Browser starts up for the first time, you will be asked to enter a *db-username*, *db-password*, *db-host* and *db-port* (fig. 5). Usually, only the first two fields have to be altered.

Please enter your MySQL username and password combination as entered in Step 2.



Login Information	
dbusername	root
dbpassword	*****
dbhost	localhost
dbport	3306
Save	

fig. 5

You may change these settings later by using the Corpus Browser settings dialog (see *"Using the Corpus Browser"* for details).

### Step 5:

This last step will install this documentation. Afterwards you find it in a new subfolder "LeipzigCC/LCCDoc" on your start menu.

## Installation on other platforms

First you will need a working distribution of SUN JRE 1.5 (or above) and MySQL Server 5.0 (or newer). You may obtain versions for your operating system from <http://java.sun.com/> (see also: <http://java.sun.com/j2se/1.5.0/download.jsp>) and <http://www.mysql.com/> (see also: <http://dev.mysql.com/downloads/>).

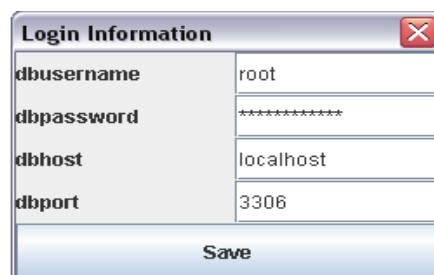
After installing SUN JRE and MySQL Server, please insert your Corpora-DVD into your DVD-Drive and change to its root directory. Change into directory "LCCDB/LCCDBFiles" and copy all of its contents into your MySQL data directory.

Go back to DVD root directory and change to "CorpusBrowser/CorpusBrowserFiles". Copy all of its content into a new directory (for example named "CorpusBrowser") on your harddrive.

At last you will have to register all TrueType fonts (\*.ttf) located in your install directory (for example: on Windows this means to copy these files to Windows\Fonts).

You may now start the Corpus Browser by changing into its directory on your hard drive and running "java -jar CorpusBrowser.jar" (Linux/Unix/MacOS users may use CorpusBrowser.sh, which is also located there).

When the CorpusBrowser is starting up for the first time you will be asked to enter a *db-username*, *db-password*, *db-host* and *db-port* (fig. 6). Usually, only the first two fields have to be altered. Please enter your MySQL username and password combination.



Login Information	
dbusername	root
dbpassword	*****
dbhost	localhost
dbport	3306
Save	

fig. 6

You may change these settings later by using the CorpusBrowser settings dialog (see *"Using CorpusBrowser"* for details).

This procedure was successfully tested on Microsoft Windows XP, Linux and Sun Solaris 10.

## Walk-through Example

To get to know the Corpus Browser and to exemplify some features of this tool, please carry out the following steps in this run-through example that guides you through the most important functionality by means of an example.

In the following figure 7, the sample page for English "King" is given. To arrive at it, please undertake the following steps (numbers in brackets refer to GUI elements as indicated in figure 7):

Open the corpus browser and switch to the English 300k corpus (1). Then enter "King" into the input field (2) and press the "Search"-Button (3).

The screenshot shows the Leipzig Corpora Collection Corpus Browser interface. The search bar (1) contains the word "King". The search button (3) is highlighted. The results page (4) displays the following information:

- Word:** King
- Frequency:** 555
- Frequency class:** 9 (*the* is seen about 2<sup>9</sup> times more frequently than *King*)
- Examples:** [10](#), [25](#), [50](#)
- Significant left neighbours for King:** [All...](#)  
[Burger](#) (891.43), [Luther](#) (794.89), [Lord](#) (71.32), [Stephen](#) (36.62), [Carole](#) (27.36), [Scott](#) (25.32), [Tom](#) (23.86), [Greene](#) (19.42), [Don](#) (14.78), [Miss](#) (12.53)
- Significant right neighbours for King:** [All...](#)  
[Hussein](#) (727.95), [Fahd](#) (688.82), [Jr](#) (386.01), [Birendra](#) (149.94), [Hassan](#) (142.34), [Mahendra](#) (114.55), [County](#) (111), [!](#) (109.48), [holiday](#) (75.13), [World](#) (64.94)
- Significant cooccurrences for King:** [All...](#)  
[Burger](#) (522.3), [Luther](#) (507.15), [Fahd](#) (462.19), [Hussein](#) (404.38), [Martin](#) (291.17), [Jordan](#) (208.63), [Saudi](#) (187.68), [Jr](#) (184.2), [Birendra](#) (183.17), [Arabia](#) (144.68)
- Cooccurrence graph:** [8](#), [16](#), [32](#), [64](#)

The cooccurrence graph (9) shows the word "King" at the center, connected to various other words. The graph is a network of nodes and edges, with "King" as the central node. Other nodes include "Luther", "Jordan", "holiday", "Pillsbury", "Mahendra", "Birendra", "Nepali", "Burger", "Hussein", "Hassan", "Arabia", "Fahd", and "Martin".

fig. 7

In the browser window (4) you obtain the following information: The word "King" was observed 555 times in the en300k corpus. Its frequency class is 9, which means that the most frequent word in the corpus, here "the", is about 29 times more frequent than "King". Some example sentences (5) show the usage of "King". Click on the numbers next to "examples" to see more examples. Words that typically are found left to "King" can be found in the section significant left neighbours (6), e.g. first names. Numbers in brackets are significance values. The significant

right neighbours (7) consist mostly of names of kings. Words that frequently and significantly come up together with "King" in sentences are listed as significant co-occurrences (8). Clicking on "All.." in the latter three sections leads to displaying not only the 10 most significant, but all co-occurrences. The co-occurrence graph (9) relates the sentence-based co-occurrences by drawing them on a plane, connecting words if they are significantly co-occurring. In this graph, three usages of "King" are visually perceivable. Click on the numbers above the graph to set the graph's granularity.

Until now, information regarding a single word was presented. But the Corpus browser can also be configured to display how several words are related. To practise this, please switch the presentation model to "complex\_mode.ini" using the drop down box (10).

The input field (2) will split into three fields. Enter "Olympics" into the first field and press "Search". You will obtain the same information as before with the simple model. Now select the radio button next to the second input field (11) as shown in figure 8 and click on "Summer" in the co-occurrences. Alternatively, enter "Summer" in the second input field (12). The display will look like in figure 8:

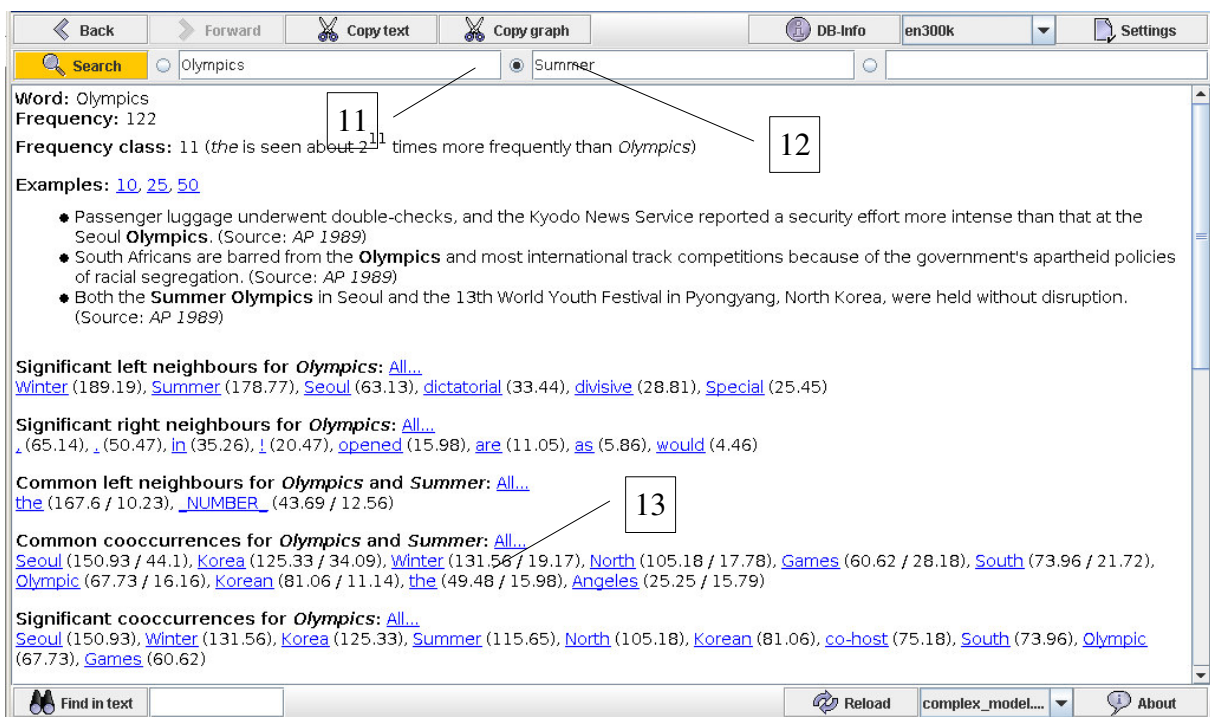


fig. 8

As common co-occurrences for "Olympics" and "Summer" (13), only places related to Summer Olympics are shown. In brackets, significances with respect to "Olympics" and "Summer" are given. Try to query the browser for Winter Olympics by entering "Winter" in the input field (2) or to narrow the topic by selecting the third input field and entering "Korea".

Please note that complex queries result in longer response times. They also result in non-empty result sets only for comparably frequent words.

The Corpus Browser is a flexible tool that allows you to define your own presentation models. A short introduction to this will be given in a later section.

## Using the Corpus Browser

In this section you will learn how to use the Corpus Browser in order to work with the Leipzig Corpora Collection databases.

The Corpus Browser was inspired by other web-browsing tools. You may use it in a similar way to explore the databases. Figure 9 shows a screenshot of the main application window.

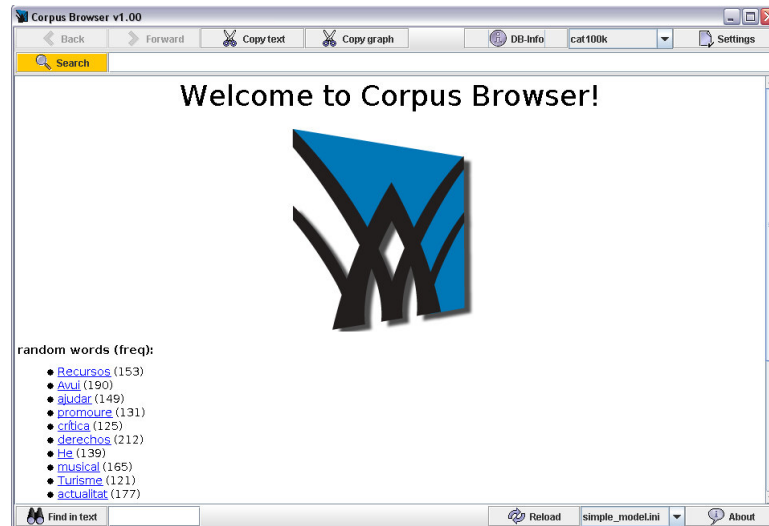


fig. 9

First you will have to choose the database you want to work with. To do so, select it in the drop-down box in the upper right corner of the application window (see fig. 10). Be sure your MySQL server is up and running. If you click on the button "DB-Info" left of the drop-down box, a new window showing some statistical information about the current database will open up (see fig. 11).

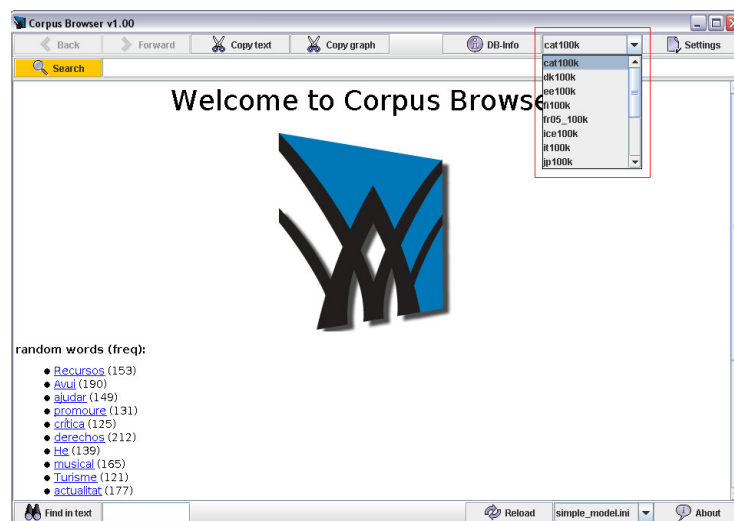
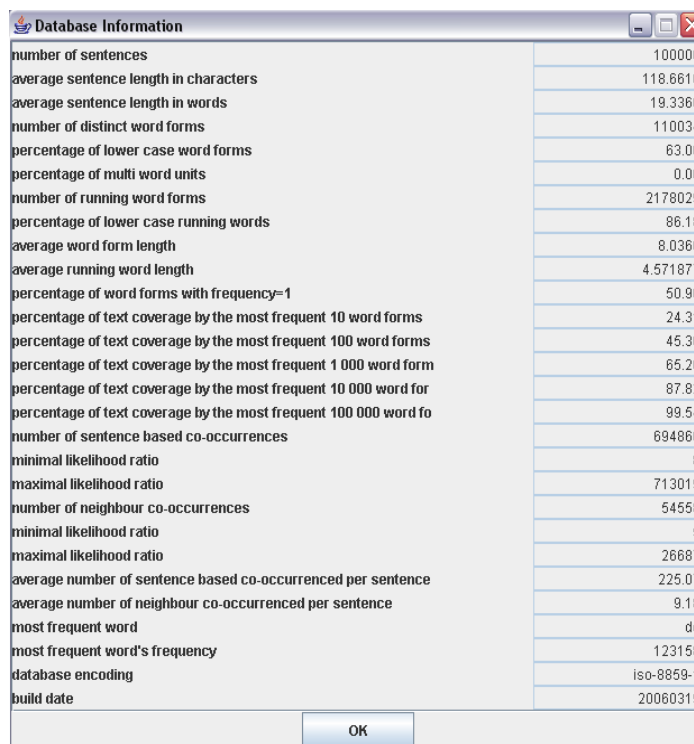


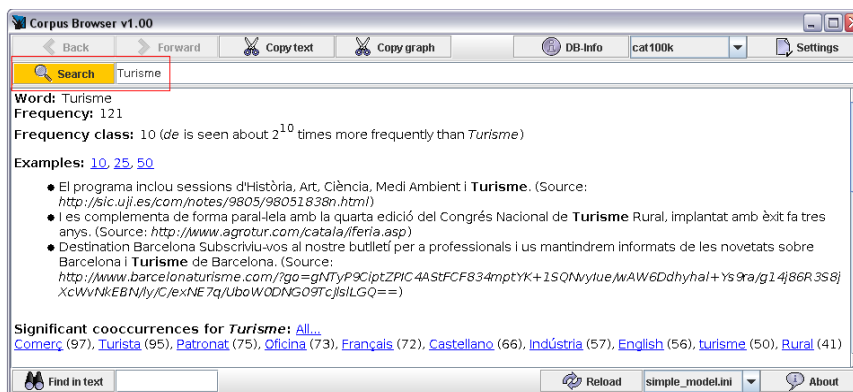
fig. 10



Statistic	Value
number of sentences	100000
average sentence length in characters	118.6616
average sentence length in words	19.3360
number of distinct word forms	110034
percentage of lower case word forms	63.00
percentage of multi word units	0.00
number of running word forms	2178029
percentage of lower case running words	86.18
average word form length	8.0360
average running word length	4.571877
percentage of word forms with frequency=1	50.96
percentage of text coverage by the most frequent 10 word forms	24.31
percentage of text coverage by the most frequent 100 word forms	45.30
percentage of text coverage by the most frequent 1 000 word form	65.20
percentage of text coverage by the most frequent 10 000 word for	87.82
percentage of text coverage by the most frequent 100 000 word fo	98.54
number of sentence based co-occurrences	694860
minimal likelihood ratio	8
maximal likelihood ratio	713019
number of neighbour co-occurrences	54558
minimal likelihood ratio	5
maximal likelihood ratio	26687
average number of sentence based co-occurred per sentence	225.07
average number of neighbour co-occurred per sentence	9.18
most frequent word	de
most frequent word's frequency	123158
database encoding	iso-8859-1
build date	20060319

fig. 11

To query this database for a word, enter it into the text field right of the yellow search-button and click on "Search" or click on one of the word-links in the result page. The Corpus Browser will collect all necessary data and present the result (see fig. 12). While the Corpus Browser is searching, the search-button will turn red and its new caption will be "Stop". Press it, if you want to cancel your query.



Corpus Browser v1.00

Back Forward Copy text Copy graph DB-Info cat100k Settings

Search Turisme

Word: Turisme  
 Frequency: 121  
 Frequency class: 10 (de is seen about 2<sup>10</sup> times more frequently than Turisme)

Examples: [10](#), [25](#), [50](#)

- El programa inclou sessions d'Història, Art, Ciència, Medi Ambient i **Turisme**. (Source: <http://sic.uji.es/com/notes/9805/98051838n.html>)
- I es complementa de forma paral·lela amb la quarta edició del Congrés Nacional de **Turisme** Rural, implantat amb èxit fa tres anys. (Source: <http://www.agrotur.com/catala/iferia.asp>)
- Destination Barcelona Subscriu-vos al nostre butlletí per a professionals i us mantindrem informats de les novetats sobre Barcelona i **Turisme** de Barcelona. (Source: <http://www.barcelonaturisme.com/?go=gNTyP9CjptZPIC4AStFCF834mpTYK+1SQNyyIue/wAW6Ddhyhal+Ys9ra/g14j86R3S8jXcWvNkEBNjlyjCjexNE7q/UboWODINGO9TCjSjILGQ==>)

Significant cooccurrences for Turisme: [All...](#)  
[Comerc](#) (97), [Turista](#) (95), [Patronat](#) (75), [Oficina](#) (73), [Français](#) (72), [Castellano](#) (66), [Indústria](#) (57), [English](#) (56), [turisme](#) (50), [Rural](#) (41)

Find in text

Reload simple\_model.ini About

fig. 12

Use the "Back" and "Forward" buttons in the upper left corner of the application window to browse through the recent result pages. You may also copy text out of the result page (mark specific areas using your mouse or copy the complete text) using the "Copy" button. If you want to search for any occurrence of a specific string in the result page, enter it into the text field at the bottom left of the application window. After you clicked on "Find in text" all occurrences in the result page will be marked in red (see fig. 13).

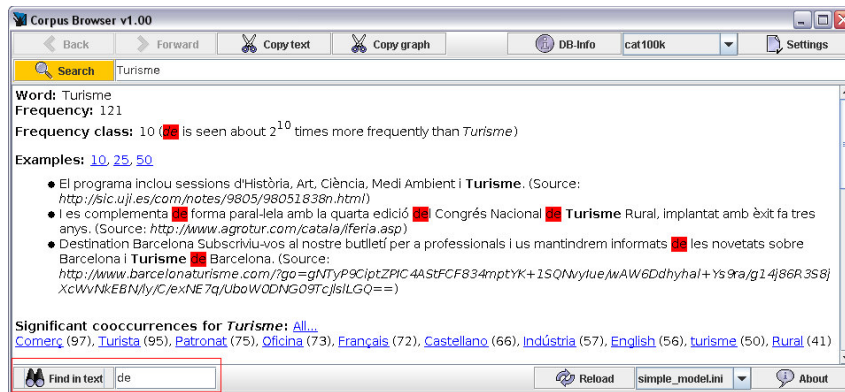


fig. 13

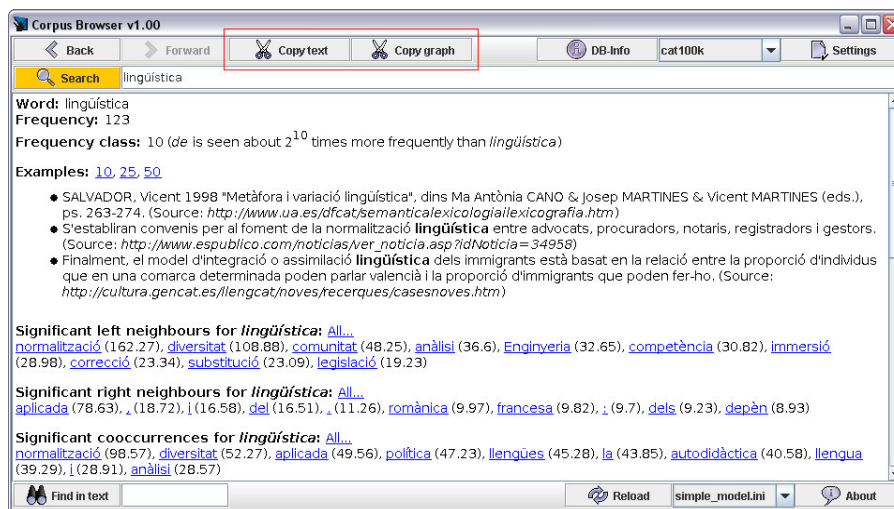


fig. 14

It is also possible to switch between several presentation models. Each model provides different information on your query. Change the current presentation model by selecting it in the drop-down box at the bottom right of the application window (see fig. 15).



fig. 15

Some additional information on the Corpus Browser and its developers is available by clicking on the "About" button at the bottom right of the application window. By clicking on the "Settings" button in the upper right corner of the application window, you will open the Corpus Browser settings dialog (see fig. 16).

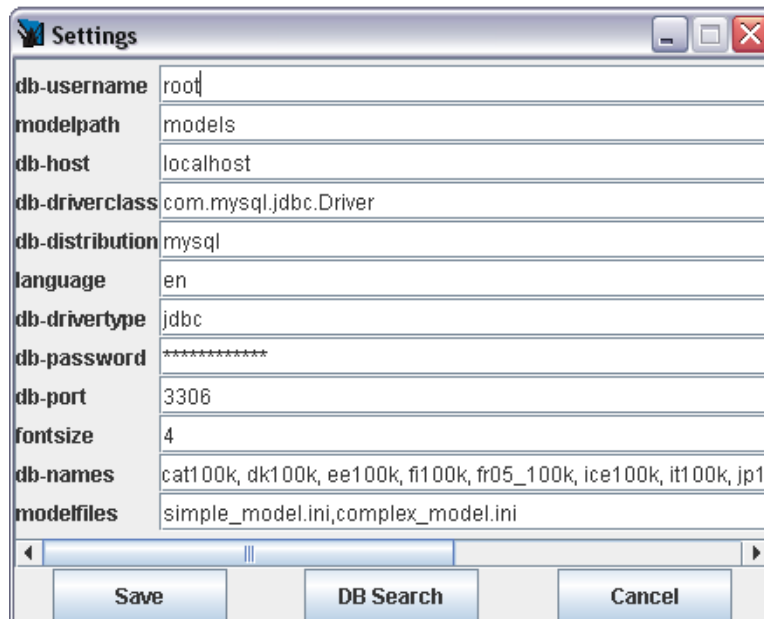


fig. 16

The options are to be used as follows:

- db-username : specifies the username used to connect to the MySQL server
- db-host : specifies the host of the MySQL server (for example: localhost or mypc.mydomain.com)
- modelpath : specifies the location where the Corpus Browser searches for presentation model files
- db-drivertype : specifies the drivertype used to connect to the MySQL server
- db-password : specifies the password used to connect to the MySQL server
- db-port : specifies the port where MySQL is listening for requests
- language : specifies the language of the Corpus Browser GUI ("en" for English, "de" for German)
- db-distribution : specifies the distribution of your database server
- fontsize : specifies the font size used in the result page (1-8, other values are ignored)
- modelfiles : specifies the names of the available Corpus Browser presentation model files
- db-driverclass : specifies the driver used to connect to your database server
- db-names : specifies the names of the available corpora databases

Use the "DB Search" button in order to automatically scan for compatible databases. The db-names field will be filled with the result. Please remember to click "Save" in order to store changes.

## Description of the Corpora

The sizes of the corpora were chosen according to several factors. One is that the corpus should be big enough to allow useful statistics on the language of that corpus. Thus, the ranking of the most frequent words, the typical co-occurrences of the frequent words and other information should be helpful. Additionally for each language we want to have corpora of comparable size.

On the other hand, however, for most languages it is not easy to find publicly available material for a corpus of maximal size. Therefore we decided to release several sizes. The smallest size was decided to be 100,000 (in short 100k) sentences. Thus, once we acquire enough text data for a given language to extract 100k sentences from this (see below, selection of sentences), we can release at least this smallest sized corpus. The next larger size is 300k. All other sizes are tenfold of their respective last variant, thus 100k, 300k, 1M (one million), 3M, 10M, 30M and 100M. All released languages are released in at least the 100k variant, many in the 300k and German in 1M:

size	Released Languages
100k	Danish, Dutch, Finnish, French, Japanese, Sorbian, Swedish, Turkish
300k	Catalan, English, Estonian, Italian, Korean, Norwegian
1M	German

## Sources of the corpora

Because of copyright restrictions we made sure that we either used publicly available text such as website. Or we included material where the copyright holders gave us explicit permission to use their texts for these purposes. Additionally in all cases we do not make the original texts available. As seen in the next chapter, a random selection of sentences is chosen and included in the final corpus in a way that even if by chance (due to the random choosing) all the sentences of an original text were put into the final selection, it would still be very hard to reconstruct the original text.

However, we do store the necessary information for each sentence where it was obtained. If it was obtained from the web, the web server domain address (not the full URL) is stored. If it was obtained from other sources, then the name of the organisation is stored. For an overview see the following table:

Language	Most prominent sources
de	German Newspapers
en	English Newspapers
ee	Estonian Newspaper
fr	French Newspaper
it	Italian Newspapers
kr	KAIST corpus
nl	Newspapers
cat, dk, fi, jp, no, se, tr	Randomly selected Web pages

## Selection of Sentences

The procedure to create a corpus for a given language consists of the following steps: First extract all sentences using a tokenizer. Then assign each sentence a unique random natural number from 1 to the total number of sentences. Then sort the sentences according to their numbers. The last step consists of taking the first sentences of this collection to build a corpus. Thus, the first 100,000 sentences of the en300k corpus are the same as all the sentences from the en100k corpus and the sentence numbers range from 1 to 300,000.

We also try to make sure that all selected sentences are well-formed sentences. Well-formedness is ensured by a set of scripts that filter out all sentences which are either too long (mostly due to a mistake by the tokenizer), contain too much repetitive information, contain too many words which are not from the language of the given corpus and other such rules. Additionally we remove all duplicate sentences. To some extent we also exclude too similar sentences (where for example only one word is different). However, it is still possible that nonsensical or otherwise malformed sentences remain in the corpora.

The original sizes of the corpora from which the smaller public corpora were made can be drawn from the following table:

Language	Original size
de	39M
en	13M
cat	10M
dk	7.2M
it	9M
se	6.9M
fi	4M
fr	3.2M
ee	2.7M
kr	2.3M
nl	1.4M
tr	1M
jp	0.4M
sorb	0.3M

## Word numbers

Each word in each corpus has its own unique number. The numbers are assigned according to the frequency of the corresponding words. However, the first 100 numbers are reserved for special characters such as comma or exclamation mark. These first 100 numbers are the same for all languages. Thus, the most frequent word "the" in the English corpus has the word number 101 and the second most frequent word "of" has the word number 102.

All numbers in the corpus, as long as they are comprised only of digits, are replaced by the same string "\_NUMBER\_" both for co-occurrence computations (see below) as well as for frequency counts. Moreover, there are the special words `%^%` and `$$%` marking the beginning and the end of each sentence.

The inclusion of word groups for the German corpus does not change the frequencies of the

single words that comprise the word groups. Thus the sentence "Michael Schumacher fährt Auto" which contains the first two words "Michael Schumacher" as a multiword still generates an increase in frequency for the word "Michael". Hence, for a comparison of frequencies between different languages these multi words units can be ignored.

## Significance Measures

In order to find out, for example, whether it is correct to say "powerful chip" or "strong chip" it is possible to measure which combination occurs more often. However, since both combinations do occur in the English corpus, it is possible to measure the significance of a particular construction.

In order to measure whether the co-occurrence of "powerful chip" in 4 sentences or the co-occurrence of "strong chip" in 3 sentences in the English en100k corpus is statistically significant we use the log-likelihood measure (Dunning 93). First a hypothesis is formulated which states that the two words are statistically independent. Based on this assumption it computes the amount of "surprise" to observe the two words so frequently together in sentences.

Based on the degrees of freedom (always one in these cases) and the desired significance level of 5%, the value must be larger than 3.84 in order to be counted as significant. In this particular case, the significance is 22.72 for "powerful chip" and only 8.17 for "strong chip". This means that both constructions are possible, but the first one is more appropriate. When using the en300k corpus the usefulness of such significance computations becomes more evident since even though "strong chip" occurs 5 times as opposed to only 4 times of "powerful chip", the significances are only 11.91 for the former and 20.11 for the latter case.

The information gained from co-occurrence measurements can be used in various other ways. These include direct visualizations of them, as is being done by our Corpus Browser, or the computation of word similarity. It is important to know, that if a word occurs several times in a sentence, that it still produces only one co-occurrence observation with all the other words in that sentence.

For the German corpus we also included information about word groups that were created by a supervised algorithm. However, this means that if the words "in a" are known as a word group, that any other word such as "thought" is observed to co-occur with the entire word group, instead of the parts. But "thought" might still end up having co-occurrences with "in" and "a" if the parts of the word group co-occur with the word sufficiently frequently.

Since we also included a special character for sentence begin and sentence end, co-occurrences of certain words will reflect this. Typical question words such as "When" do have the sentence begin as a strong left neighbour, according to the neighbour co-occurrences.

## Technical description of the database schema (for advanced users)

When deciding the database schema we had three aims in our mind:

- It should store the information in a way to facilitate access for the various views defined in the corpus browser
- It should be simple, yet avoid redundancies
- It should be easy to add new information

For every item in the database there are identification numbers. Thus every word has a word number and every sentence has a sentence number. Relations between words are then stored as pairs of word numbers. Different relations are stored in different tables, instead of adding a further column to a general relations table. This is because there are only very few different relations (such as sentence co-occurrences, neighbour co-occurrences) but many items that stand in these relations. If put into one single table that would result in a prohibitively large table for the larger corpora.

While the tables that use identification numbers facilitate access for programs, they are rather hard to read for humans. Therefore we provide additional plain-text tables in which all numbers were resolved into their corresponding strings (with the exception of the inverse list). This allows viewing the underlying databases without any browser, just by using any text viewer.

This section is therefore divided into two parts: One that describes the identification number based table schema and another that describes the plain text files.

### Database schema

#### words

The most important table of the data base schema is the word list, called **words**. It consists of the following fields:

- `w_id` – An integer with at most 10 digits which uniquely identifies any word string
- `word` – A string with maximally 255 characters
- `freq` – A number with at most 8 digits

This table is used in all select statements to resolve the identification numbers of words into their strings for final output. If it contains, e.g. two rows such as

509	strong	552
1764	powerful	156

then that means that the corpus has at least two different words and the 509<sup>th</sup> one is “strong” with a frequency of 552, while the 1764<sup>th</sup> word is “powerful” with a frequency of 156.

#### sentences

The actual corpus is a collection of sentences stored in the table with the name **sentences**. It consists of the following fields:

- `s_id` – A 10-digit number which uniquely identifies any sentence
- `sentence` – A string with (theoretically) unlimited length

Sentences are stored directly. A row like

95015	And you don't conclude that we don't need a strong defence because of the actions of some faithless employees.
-------	--

means that the sentence with the number 95015 is about a strong defence.

#### inv\_w

In order to efficiently find out in which sentences a given word occurred, the table **inv\_w** has to be accessed. It stores relations between word numbers and sentence numbers:

- `w_id` – A 10-digit number from the words table
- `s_id` – A 10-digit number from the sentences table
- `pos` – A three-digit number that expresses the position of the word (`w_id`) in the sentence (`s_id`). The position is based on words as defined in the words table, not on characters.

Thus, if it has two rows like

1	1	1
1	2	16

then that means that word 1 appeared both in sentence 1 and 2. Once it appeared in the first position, the other time as the 16<sup>th</sup> word of the sentence.

### sources and `inv_so`

Sometimes it is interesting to know from where a particularly peculiar example was drawn. For research purposes it is also important to know that it is not an artificial example. Therefore the table `sources` stores from which websites or other sources a given sentence was obtained and table `inv_so` allows to look this information up conveniently. The table `sources` has the following structure:

- `so_id` – An 8-digit number which uniquely identifies any source
- `source` – A string with at most 255 characters. Usually an URL.
- `date` – A date that tells when that source was visited to obtain the information, which is stored in this corpus.

For each sentence the table `inv_so` stores, where that sentence was obtained:

- `so_id` – The 8-digit number which identifies the source
- `s_id` – The 10-digit number which identifies the sentence

### `co_n` and `co_s`

As mentioned in the introduction, information about which words co-occur with each other is very useful. The two tables `co_n` and `co_s` store this information. `co_n` stores, which words co-occurred directly next to each other (bigrams). This expresses mostly typical uses of words with each other. `co_s` on the other hand stores, which words co-occurred anywhere within sentences. This expresses typically related or associated words. The structure of both tables is exactly the same:

- `w1_id` – A 10-digit number which is any of the `w_id` numbers from the words table
- `w2_id` – Equivalent to `w1_id`
- `freq` – How often the word `w1` was observed to co-occur with `w2`
- `sig` – How significant that co-occurrence is, according to the log-likelihood significance measure

### meta

The last table to be described is the meta table. Any additional information or statistics is stored here. It is useful for referencing as well as general information. Its structure does not need identification numbers, because the entries are purely for viewing purposes:

- `attribute` – A string which describes the entry
- `value` – A string as well, which describes the value

This table stores for example the number of sentence in the corpus, the number of word types and tokens, text coverage of the most frequent 10, 100, etc. words and other metadata. Please note that without the entry with attribute “LCCBrowser”, the database will not be displayed in the Corpus Browser.

## Using the tables

These tables can be used to access information in various ways. This is done by using simple SQL select statements to the MySQL server, see also the online MySQL manual for further reference (<http://dev.mysql.com/doc/mysql/en/>). For example it is possible to print out all words that have a frequency of exactly 50:

```
select word from words where freq = 50;
```

It is a bit trickier to combine information from several tables together and for example print out all sentences in which 'strong' appears:

```
select s.sentences from sentences s, inv_w I, words w where
w.word = 'strong' and w.w_id = i.w_id and i.s_id = s.s_id;
```

Finally a much more complex statement would be to print out all words that co-occur with the input word "strong". The statement:

```
select w1.word, w2.word, s.sig from words w1, words w2, co_s s
where w1.w_id = s.w1_id and w2.w_id = s.w2_id and w1.word =
'strong' order by s.sig desc limit 10;
```

produces on the en300k corpus the following output:

word	word	sig
strong	demand	112.84
strong	growth	94.91
strong	earnings	70.25
strong	gains	66.82
strong	very	58.59
strong	market	52.28
strong	dollar	44.32
strong	markets	43.79
strong	despite	37.02
strong	performances	33.91

## Plain text tables

The tables `co_n` and `co_s` are both additionally provided in a way where the word IDs were replaced by their respective word string. The files containing these tables are located on the DVD in the separate directory `Flatfiles`.

## Additional tweaking

### Speeding things up

If the Corpus browser queries take too long, it might be helpful to tune the MySQL Server. One useful way to tune the server is to edit the key buffer size and the sort buffer size to utilize more RAM, if your computer possesses a lot (i.e. 1GB).

For these purposes stop the server and then edit the server configuration `my.cnf`.

Change in the `[mysqld]` section of the file the following values:

`key_buffer = 256M` (change this to about one third of the RAM of your computer)

`sort_buffer = 32M`

## Using alternative fonts

The installation routine will install fonts for Japanese and Korean that were released under a free license. These might not be the best fonts and therefore if you happen to possess other fonts, feel free to use the fonts that fit best your needs. For Korean a better font might be Arial Unicode MS, whereas for Japanese a better font might be MS Mincho.

If you choose to use alternative fonts, you have to have them installed both on your system as well as place them in the installation folder of the Corpus Browser. Then login to the MySQL server and change the entries in the meta table in the following way:

```
update meta set value="msmincho.ttf" where attribute="fontfile";

update meta set value="MS Mincho" where attribute="fontname";
```

## Creating presentation models for the Corpus Browser (for advanced users)

Using the internal presentation model description language (PMDL), the Corpus Browser becomes a much more powerful tool for you to work with. You can change existing models to show only the data you are interested in, change the way data is presented and create new presentation models with new queries and data.

This walk-through exemplifies how to create or alter a presentation model using PMDL. To create a new presentation model for testing please follow the instructions below:

Change to the Corpus Browser main directory on your hard drive and open the directory named *models*. There you will find at least two files named *simple\_model.ini* and *complex\_model.ini*. Please make a copy of *simple\_model.ini* and rename it to *my\_model.ini*. Afterwards open *my\_model.ini* with an editor of your choice. It should look like figure 17.

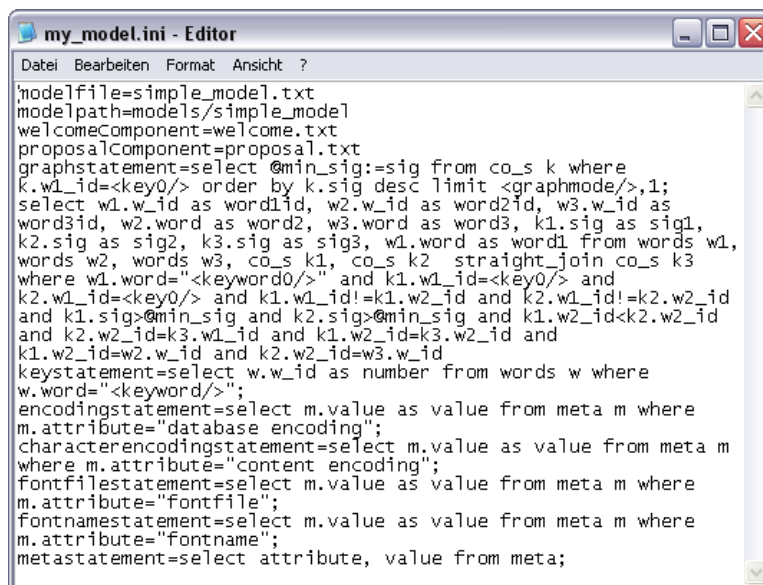


fig. 17

As you can see, each *\*.ini* file has 11 settings you may alter. Usually you will only change *modelfile*, *modelpath*, *welcomeComponent* and *proposalComponent*:

- *modelfile*: defines the name of the main presentation model file
- *modelpath*: specifies the directory where every presentations model component will be found
- *welcomeComponent*: the component to be shown when no valid keyword is entered

- *proposalComponent*: the component to be shown when the keyword was not found
- *graphstatement*: SQL statement to use to get the cooccurrence graph data
- *keystatement*: SQL statement to derive the key number from any keyword
- *encodingstatement*: SQL statement to get the encoding of the current database
- *characterencodingstatement*: SQL statement to get the character encoding of the current database
- *fontfilestatement*: statement to fetch the filename of the TrueType fontfile to be used
- *fontnamestatement*: statement to fetch the name of the TrueType font to be used
- *metastatement*: SQL statement used to show the database meta information

To go on with this test, please set *modelfile* = *my\_model.txt* and switch modelpath to *models/my\_model*. Save changes and close *my\_model.ini*. Please make a copy of the directory *simple\_model* in *CorpusBrowser/models/* place it at the same location and rename it to *my\_model*. Also rename *simple\_model.txt* in *CorpusBrowser/models/my\_model* to *my\_model.txt*. Now check if the new model is working. Run the Corpus Browser and switch to the settings dialog and change modelfiles from *simple\_model.ini,complex\_model.ini* to *simple\_model.ini,complex\_model.ini,my\_model.ini* (see fig. 18) and click "Save".

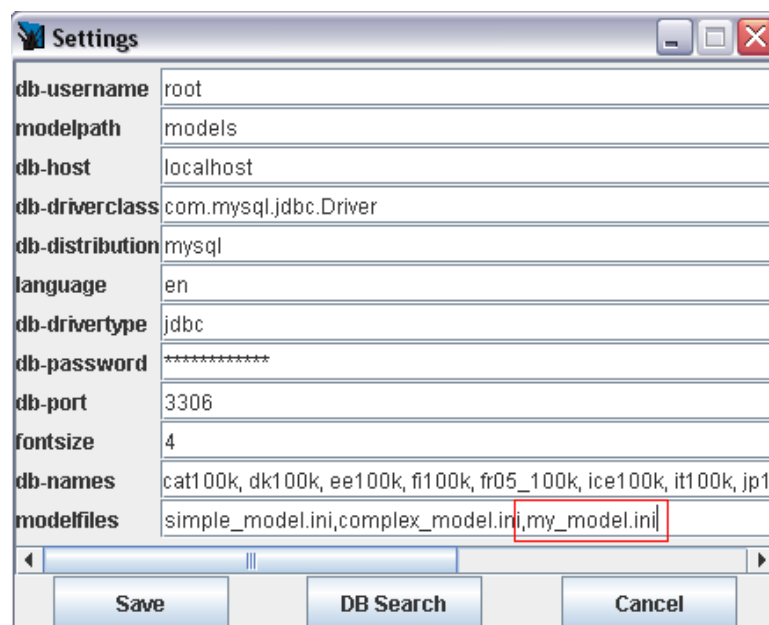


fig. 18

You may change the current model in the model dropdown box to *my\_model.ini* (figure 19). The presentation model should look like the old *simple\_model.ini*. Now we are going to change this new presentation model. Change into *CorpusBrowser/models/my\_model* and open *my\_model.txt*. It should look like in figure 20.

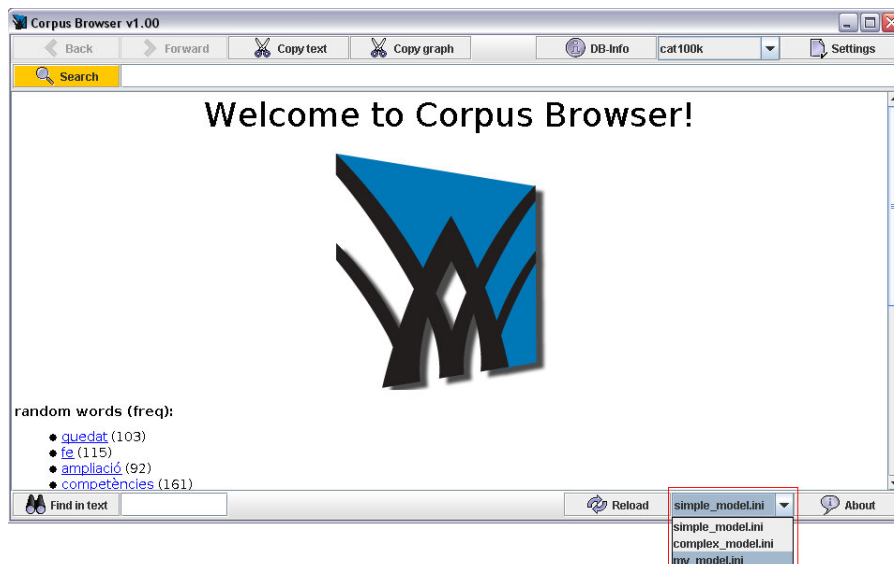


fig. 19

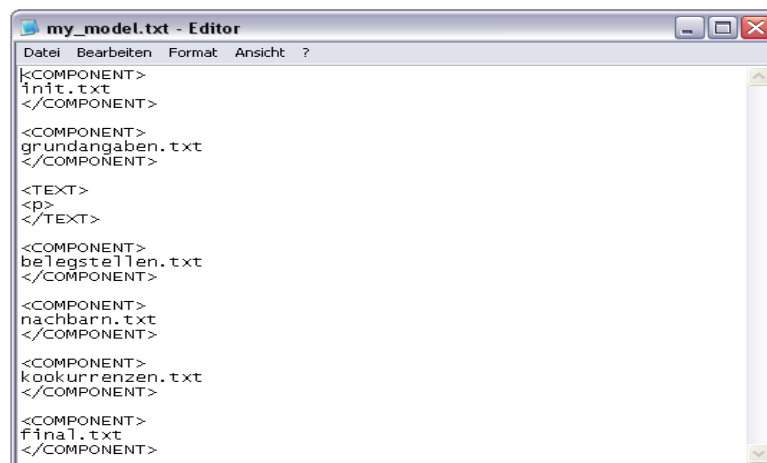


fig. 20

As you can see, the Corpus Browser presentation model description language (PMDL) was inspired by XML. PMDL describes how the result page of the Corpus Browser has to be built. Corpus Browser parses the current presentation model written in PMDL and creates a HTML-document to show as result.

PMDL supports the following tags:

- `<COMPONENT>` and `</COMPONENT>`: Text between these tags is interpreted as the path to another PMDL component. Corpus Browser will load and interpret it and all content of this component will be inserted on this position. The file will be searched relatively to the directory specified by `modelpath` in the current `model.ini`-file.
- `<DYNAMIC_COMPONENT>` and `</DYNAMIC_COMPONENT>`: Same as `<COMPONENT>` but the file will not be cached in memory, so Corpus Browser will load it from your hard drive every time it's needed.
- `<TEXT>` and `</TEXT>`: Text between these tags will be copied directly to the result-document (except interpretation of some tags like `<KEYWORDx/>`, `<KEYx/>`). You may use HTML 3.2 conform code to manipulate presentation.
- `<SQL>` and `</SQL>`: The Corpus Browser treats text between these tags as an SQL statement. The current active corpora database will be queried.
- `<SQL_NOT_NULL>` and `</SQL_NOT_NULL>`: Elements between these tags will only be interpreted if the last SQL statement was successful and the result was not empty.

- `<LIST>` and `</LIST>`: All text and elements between these tags will be interpreted as often as there are rows in the result of the last SQL query (see `<SQL>`-tag). While HTML text will be directly copied to the result-document, all tags of the `<PARAM>`-family will be treated according to the current row-number (iteration).
- `<LIST_START>` and `</LIST_START>`: Same as `<LIST>`, but only *n-1* of *n* rows are computed.
- `<LIST_END>` and `</LIST_END>`: Same as `<LIST>`, but only the *n-th* of *n* rows will be computed.
- `<PARAM>` and `</PARAM>`: These tags are only allowed/interpreted in between tags of the `<LIST>`-family. The parameter named like the text between these tags will be inserted at this position.
- `<PARAM_KEYWORD_BOLD>` and `</PARAM_KEYWORD_BOLD>`: Same as `<PARAM>` but all occurrences of keywords in the result will be marked bold (`<b>xyz</b>`).
- `<KEYWORDx>`: Inserts keyword number *x* at this position.
- `<KEYx>`: Inserts the key-number of keyword *x* at this position.
- `<KEY_NOT_NULLx>` and `</KEY_NOT_NULLx>`: All elements in between these tags will only be interpreted when key number *x* is not null, which means the corresponding keyword is present in the currently selected database.
- `<ISFILE_xyz>`: Content between these tags will only be interpreted if there is a file at the location specified by *xyz* relatively to Corpus Browser main-directory.
- `<FONTSIZE/>`: The currently selected font size will be inserted at this position (possible values are 1-8).
- `<FONTNAME/>`: The currently selected font name will be entered at this position.
- `<ISMODE_xyz>` and `</ISMODE_xyz>`: Elements between these tags are only interpreted if this component is in mode *xyz*. Every component is initialised in mode "simple". See the links-tag family on how to change a components mode.
- `<MODEL_PATH/>`: The model path specified in the model.ini-file will be inserted at this position.
- `<LINK_TEXT>` and `</LINK_TEXT>`: Specifies the text/caption of a link.
- `<LINK_URL>` and `</LINK_URL>`: Specifies the target of a link. For example, this may be a keyword on internal links or an Internet-URL when dealing with external links. It is also possible to refer to a component-mode or a graph-mode when working with `<LINK_SWITCH>` or `<GRAPH_SWITCH>`.
- `<LINK_EXTERN>` and `</LINK_EXTERN>`: Describes an external link. Use `<LINK_TEXT>` and `<LINK_URL>` in between these tags. A browser will be opened to show the contents of this URL.
- `<LINK_INTERN>` and `</LINK_INTERN>`: Describes an internal link. The active keyword will be replaced by the text in `<LINK_URL>` when the user clicks on the text specified by `<LINK_TEXT>`.
- `<LINK_SWITCH>` and `</LINK_SWITCH>`: This link changes the mode of this component to `<LINK_URL>mymode</LINK_URL>` when `<LINK_TEXT>change to mymode</LINK_TEXT>` is clicked.
- `<GRAPH_SWITCH>` and `</GRAPH_SWITCH>`: Switches the graphmode to `<LINK_URL>10</LINK_URL>` when `<LINK_TEXT>switch to 10</LINK_TEXT>` is clicked.
- `<GRAPHMODE/>`: This tag is only interpreted in the graph statement line of every model.ini-file. It is used to switch between different steps of graph-complexity. Graph mode is initialised with the value 4.

Feel free to experiment with your new model. Create new or change existing components.

## Conditions of use

The Leipzig Corpora Collection contains text from publicly accessible sources. All data have been processed automatically so that it is not possible to reconstruct the original source texts.

The corpora are protected by copyright. They are made available on the condition that they may be used for scientific purposes only and not passed on to third parties. Any use of the data must be duly documented and referenced. Commercial use of the data requires the prior written consent of the Leipzig University department for Natural Language Processing.

## Disclaimer

The Leipzig Corpora Collection has been processed automatically from publicly accessible sources based on the outlined methodology without considering in detail the content of the contained text. No responsibility is taken for the content of the data. In particular, the views and opinions expressed in specific parts of the data remain exclusively with the authors.

For each word, the list of words that significantly co-occur with that word are computed on the basis of the available text and neither express a general fact of language nor the particular view of the Leipzig University department for Natural Language Processing.

## References

Dunning, T. E. 1993: Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61-74.

Quasthoff, U., Biemann, C. and Richter, M. 2006: Corpus Portal for Search in Monolingual Corpora. Proceedings of LREC-06, Genoa, Italy

MySQL Reference Documentation: <http://dev.mysql.com/doc/mysql/en/>

Corpus Browser frequently asked questions: <http://corpora.uni-leipzig.de/faq.html>

## About

Natural Language Processing Department  
University of Leipzig  
Website: <http://corpora.uni-leipzig.de>

Editors:	Uwe Quasthoff, Gerhard Heyer
Corpus Collection:	Uwe Quasthoff, Sebastian Gottwald
Corpus Processing:	Sebastian Gottwald, Matthias Richter
Corpus Processing Software:	Chris Biemann, Fabian Schmidt, Matthias Richter, Sebastian Gottwald
Language Cleaning:	Sven Teresniak, Sebastian Gottwald
Browser Programming:	Volker Boehlke
Testing:	Chris Biemann, Matthias Richter, Markus Ackermann, Thomas Eckart, Stefan Bordag
Graphics Design:	Gunnar Boldhaus
Documentation:	Stefan Bordag, Volker Boehlke, Chris Biemann
Special thanks to:	Key-Sun Choi (KAIST Institute ( <a href="http://www.kaist.edu/">http://www.kaist.edu/</a> )), Heiki-Jaan Kaalep (University of Tartu), Gemma Boleda (University of Catalonia)
Language Consultants:	Hans Jörg Bibiko, Christer Johansson, Alice Renault, Ron van Kesteren, Unni Cathrine Eiken, Anders Nøklestad, Henrik Pedersen, Peter Walde, Mustapha Özbek