IMPACT CORPUS BASED LEXICON TOOL (COBALT)

CONTENT

INSTALLATION

SHORT FUNCTIONAL DESCRIPTION

CoBaLT is an application in which a corpus of texts can be loaded so as to be able to annotate its tokens (lemmatize and more). The annotation work in CoBaLT gives two products: an annotated corpus, and a lexicon consisting of word forms and their corresponding lemmata and such assigned to them.

In terms of workflow, working with CoBaLT will mean:

[1] Loading a corpus into the tool,
[2] Using the tool to annotate the corpus,
[3] Finally exporting the result of this work in the form of an enriched version of the corpus files originally loaded into the tool. The resulting lexicon can be exported to XML, or one can just re-use the lexicon database built during the CoBaLT work.

TECHNICAL OVERVIEW

CoBaLT is an AJAX application designed for Mozilla Firefox (other browsers may work as well, but are discouraged). It uses a MySQL database, and is written in PHP, Perl and Javascript.

During the installation, some PHP, MySql and Apache variables need to be set to ensure proper functioning of the tool. Check the 'Apache settings', 'PHP settings' and 'MySql settings' sections for that. Some settings discussed below aim to facilitate working with very large data sets: give those settings special attention if you need to process lots of data.

HARDWARE REQUIREMENTS

CoBaLT consists of an application and a database. It is considered good practise to install the two components on separate servers. The first one (which will house the PHP/Perl code) is the application server, and the second one will act as the (MariaDB/MySql) database server.

As a reference for hardware requirements, here are the specifications of the servers on which CoBaLT is developed and maintained at the moment, with good performance as a result:

Application server:
CPU : Intel Xeon e5 2,3 GHz FIO
Mem : 4Gb
Storage : 8Gb iscsi storage
Application-storage : 30 MB
OS : Debian squeeze
Software : Apache 2.2.16 / PHP 5.3.3 / Perl v5.10.1 (*) built for x86_64-linux-gnu-thread-multi

Database server:
OS : Linux

Software: MariaDB (community-developed fork of MySQL)
CPU : Intel Xeon e5 2,3 GHz FIO
Mem : 4Gb
Storage : 60Gb
Application-storage: 700MB

INSTALLATION PACKAGE

The installation package consists of a zip file. The unpacked archive should contains two components: a PHP directory and a Perl directory. We will see how to deal with it in the 'PHP/Perl' section.

APACHE

A webserver is needed to handle the PHP request. Apache is by no means the only suitable web server around, but it is very widely spread and the tool was developed and tested on it.

APACHE SETTINGS

[1] To ensure the tool will process diacritics and special characters properly, the right character set has to be set.

With Apache this can be done by setting the 'AddDefaultCharset' directive in the server's httpd.conf file:

```
AddDefaultCharset UTF-8
```

[2] Sometimes POST requests can be very large due to very big amounts of data being sent to and from the server. Therefore it might be necessary to increase the limit the web server imposes on such requests.

This can be done by setting the 'LimitRequestLine' directive in the server's httpd.conf file:

```
LimitRequestLine 100000
```

[3] With Apache it is necessary to restart the web server for this changes to have effect.

PHP/PERL

As noted above, a webserver should be running that can handle PHP requests. The source code of the tool should be placed in a directory for which PHP support is turned on.
The tool also needs to run some Perl code, so the webserver needs to support that as well.

The tool was tested on PHP 5.1.6 running on Red Hat Enterprise Linux Server release 5.2. It was running Perl v5.10.1 (*) built for x86_64-linux-gnu-thread-multi.

SOURCE CODE

The source code is usually delivered in a zip file. The unpacked archive should contains two components: a PHP directory and a Perl directory.

The PHP directory contains the main code. Copy the whole directory to a location at which PHP support is turned on. This is the location users will have to go to if they want to work with CoBaLT.

The Perl directory contains the code of the tokenizer. Copy the whole directory to a location at which Perl can be run, and make sure that the PHP code can access this Perl directory (as the PHP application will need to run the Perl tokenizer from the command line).
NOTE that later on, you will have to declare the location of the tokenizer in the CoBaLT configuration, in such a way that the tool can find and use it. We will see that in the 'CoBaLT configuration' section.

PHP SETTINGS

[1] First of all, we need to enable output buffering in PHP. It decreases the amount of time it takes to render information in the tool, among others. This can be done by setting the 'output_buffering' variable in the php.ini file:

```
php_value output_buffering 4096
```

[2] Second, we need to set the maximum size allowed for files to be uploaded, as working with CoBaLT starts with uploading a corpus, which might consist of large files. For some systems the maximum size allowed for a file to be uploaded is only 2 Mb. If you have files larger than this, you might want to increase this limit somewhat. This can be done by setting the 'upload_max_filesize' in the php.ini file:

```
php_value upload_max_filesize = 30M
```

Of course, the 30M can be any value you like.

To make this work, the php.ini variable 'post_max_size' must be set higher than 'upload_max_filesize' (or at least equal):

```
php_value post_max_size 30M
```

This setting is on its turn depending on another setting. The 'memory_limit' variable in the php.ini file must generally speaking be set higher than the 'post_max_size':

```
php_value memory_limit = 256M
```

Of course, the 256M can be any value you like.

[3] Third, it can be convenient to set an error log, for debug purposes in case some problem occurs. This also can be done in the php.ini file:

```
  php_value error_log /some_directory/some_subdirectory/php-scripts.log
```

[4] Finally, it may be necessary to restart the web server for this changes to have effect.

NB: full information about php.ini directives can be found at http://php.net/manual/en/ini.core.php

UPLOADING MULTIPLE FILES, PHP ZIP SUPPORT

PHP can not read directory listings, so it is not possible to upload an entire directory in one go. As a workaround, zip files can be used. When a directory is zipped, it can be uploaded at once to CoBaLT.

In order for this to work, PHP has to have its zip functionality enabled. Please refer to the PHP documentation for your platform and version.

As a reference, our `phpinfo()` shows this:

zip

Zip	enabled
Extension Version	$Id: php_zip.c,v 1.95.2.6 2007/05/19 22:35:49 pajoye Exp $
Zip version	1.8.10
Libzip version	0.7.1

MYSQL SUPPORT FOR PHP

The PHP installation needs to be configured with MySQL support. Usually the standard installations provides for this. If not, please refer to the PHP documentation on your system.

MYSQL (MARIADB)

In order to set up the MySQL database for the tool, root access is needed to the MySQL server.

The tool was tested on MySQL 5.0.45 (and following versions), running on Red Hat Enterprise Linux Server release 5.2. Recent installations (2017) with MariaDB proved to function better.

MYSQL SETTINGS

[1] The tool runs quite some heavy queries, most notably the ones that gives the overview of the type-frequency list, that make extensive use of MySQL's built in GROUP_CONCAT() function. The result of this function can quite easily become larger than the default length that is allowed. There are two MySQL server variables that control how much data can be sent by the server: @@max_allowed_packet and @@group_concat_max_len. To see what they are currently set to, the following queries can be used:

```
mysql> SELECT @@group_concat_max_len;
mysql> SELECT @@max_allowed_packet;
```

It is advisable to set these values to 64 Mb or higher. This can be done by these statements:

```
mysql> SET GLOBAL max_allowed_packet = 67108864;
mysql> SET GLOBAL group_concat_max_len = @@max_allowed_packet;
```

[2] Additionally, we need to set the ‘group_concat_max_len’ and ‘max_allowed_packet’ variables in the my.cnf file. Otherwise large query results concatenated by the GROUP_CONCAT function will be trunked and cause malfunction of the tool:

```
group_concat_max_len=102400;

max_allowed_packet=300000;     # max_allowed_packet must be
                                 higher than group_concat_max_len
```

Restart the MySql server after that.

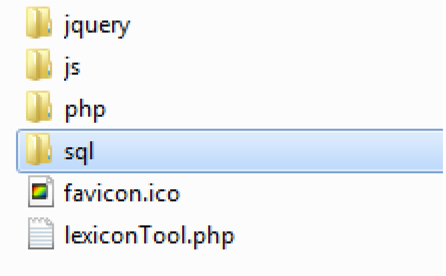
CREATE DATABASES

The tool expects at least two MySQL databases to be there with the right table structure. The first one is used as an index to the word forms in the corpora and is called the ‘token database’. The second one is used for storing all the lexicon data and is called ‘lexicon database’.

In a normal installation, each CoBaLT project has its own lexicon database and token database instances, distinct from the databases instances of other CoBaLT projects.

NOTE that it is possible to configure the tool in such a way that the token database has to be created only once per database host (so usually once in the lifetime of a distribution of CoBaLT). In such an installation, each project will normally get its own lexicon database instance, but one single token database instance will be shared among all projects.
However, this type of installation (which was originally the default one) is nowadays heavily discouraged since it makes moving a CoBaLT project to another server/computer more difficult, so we won’t describe this type of installation here.

The distribution of the tool comes with some SQL files in the 'sql' directory of the installation.



These files, called 'emptyLexiconDatabase.sql' and 'emptyLexiconTokenDatabase.sql' contain the data structures of both databases. This files need to be loaded into MySQL.

Before this can be done, the databases need to be created. This is done in MySQL by running these queries:

```
mysql> CREATE DATABASE myLexiconDb;

mysql> CREATE DATABASE myLexiconTokenDb;
```

In this queries the italic part should be replaced by your own database name.

NOTE that you'll later on need to declare in the configuration of CoBaLT which token database belongs to a given lexicon database. We will see how to do that in the 'CoBaLT configuration' section.

For now, the SQL files that come with the distribution (in the 'sql' directory) should be loaded into the databases just created. On the command line you can do this by executing the following commands:

```
mysql -h<server> -u<user> -p<password> myLexiconTokenDb <
emptyLexiconTokenDatabase.sql

mysql -h<server> -u<user> -p<password> myLexiconDb <
emptyLexiconDatabase.sql
```

CREATE A MYSQL USER

When a database has been created, a MySQL user should be added. This can be done by running the following statements:

```
mysql> GRANT ALL ON myLexiconDb.* TO 'newUser'@localhost IDENTIFIED BY
'password';

mysql> GRANT ALL ON myLexiconDb.* TO 'newUser'@'%' IDENTIFIED BY 'password';

mysql> FLUSH PRIVILEGES;
```

Again, the italic parts in the queries should be set to your own desired values.

NOTE that later on you'll have to declare the username and password of the MySql user in the CoBaLT configuration, in such a way that the tool can access the database. We will see that in the 'CoBaLT configuration' section.

FILL THE USERS TABLE (REDACTORS)

The entire database can be empty at start up, but for the users table which needs at least one row (that is: one redactor entitled to log in). New users can be added with the following statement:

```
mysql> INSERT INTO users (name)
       VALUES ('Amy'), ('Billy'), ('Duffy'), ('Ella');
```

Now Amy, Billy, Duffy and Ella will be able to log in just by typing their names in the Log-in screen of the tool (see the 'User interface' section about that).

PRE-LOAD A LEXICON

The tool allows a user to start annotating a corpus by means of a pre-loaded lexicon (but this is not required). If you'd like to use a lexicon in your annotation work, you'll need to load it before you upload your document.

Use the 'loadLexicon.sql' file (in the 'sql' directory of the installation) to achieve that.

NOTE you have to edit this file to specify the location of your lexicon file:

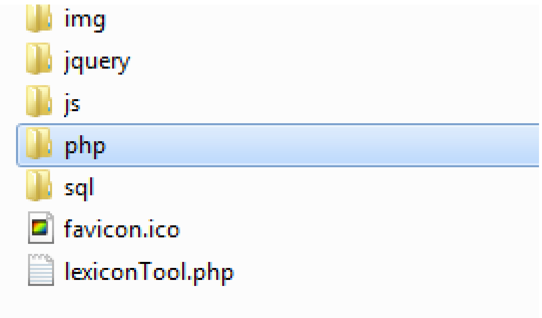
```
load data local infile "put_full_document_path_here" into table upload;
```

THE LANGUAGE TABLE

If analyses should allow for languages (see below), the different options should be listed in this table.

COBALT CONFIGURATION

The tool needs a bit of configuration before you can use it. The configuration is set in the globals.php file, which can be found in the PHP directory the your installation:



img
jquery
js
php
sql
favicon.ico
lexiconTool.php

NOTE that in previous versions of the tool, some configuration was also stored in the lexicontool.php file in the root directory of the application, but this is no longer the case. All configuration is now exclusively stored in php/globals.php.

GLOBALS.PHP

[1] The first thing to set in this file is the database host and the username and password needed to access it. Put here the username and password of the MySql user you created before in the 'Create a MySql user' section.

```
$sDbHostName = "yourhost.com";
$sDbUserName = "username_of_your_host";
$sDbPassword = "password_of_your_host";
```

[2] Secondly, you have to declare at least one lexicon project to be run in the tool (which is why you actually use the tool for).

Say you have some Dutch lexicon building project. As stated in the 'Create databases' section, each project consists of a lexicon database and a token database. Let's say that the lexicon database of your project is called 'LexiconTool_NL' and its token database is called 'LexiconToolTokenDb_NL'.

You first have to declare the lexicon database, together with a user friendly project description (this description will be shown to users in the log-in screen).

```
$asProject['LexiconTool_NL'] = 'My Dutch lexicon building project';
```

Then you need to declare which token database belongs to this lexicon database:

```
$asTokenDbName['LexiconTool_NL'] = 'LexiconToolTokenDb_NL';
```

For any new project to be run in the tool, just repeat the two lines above with the relevant database names:

```
$asProject['LexiconTool_NL'] = 'My Dutch lexicon building project';
$asProject['LexiconTool_DE'] = 'Some German project is cool too';
```

```
$asTokenDbName['LexiconTool_NL'] = 'LexiconToolTokenDb_NL';
$asTokenDbName['LexiconTool_DE'] = 'LexiconToolTokenDb_DE';
```

If more projects are indeed declared, it can be convenient to have the project being mostly used at the moment selected by default in the log-in screen. This can be done by giving the $sChecked variable the name of the relevant lexicon database as a value:

```
$sChecked = 'LexiconTool_DE'; // the German project will be
                                 selected by default
```

If no project must be preselected, give this variable NULL as a value:

```
$sChecked = NULL;
```

[3] Third, the corpus files uploaded to the tool can be stored in two ways:
- on a file server, or
- in the lexicon database (in the 'documents' table).

Saving the file in the 'documents' table of the lexicon database has a strong advantage. If somehow you need to move your CoBaLT project to another server, having the files stored in the lexicon database makes moving the project much easier (because only the database will need to be moved, and no separate files). So, choosing for files storage in the lexicon database is recommended.

This has to be set in the $aFullDatabaseMode array:

```
// save the German files into the lexicon database
$aFullDatabaseMode['LexiconTool_DE'] = true; // default and recommended

// save the Dutch files on the file server
$aFullDatabaseMode['LexiconTool_NL'] = false;
```

BEWARE if you choose for file storage on the file server! When some project run in CoBaLT has reached its end and you want to export the results by enriching your original (uploaded) corpus files with the analyses created in CoBaLT, you'll need to have these original upload files at your disposal. So don't throw these files away or you won't be able to export the results of your work.

[4] Whatever mode you choose for the file storage, you will need to state some document root CoBaLT will (at least temporarily) put the documents into when those are uploaded into the tool.

This document root can be set to any available directory, as long as this directory is readable and writable for the tool. If you've chosen for file storage on the file system ('false' setting in section **[3]**), it is advisable to avoid having this directory in the same directory the application is in, because it might in that case accidentally be thrown away when a new version of the tool is installed. The document root can be set this way: (*use an absolute path*)

```
$sDocumentRoot = "/servername/some_directory/uploadedDocuments";
```

If you choose to upload your documents in a zip file, it is needed to declare a subdirectory in which CoBaLT will unpack the archive. This subdirectory will be created automatically if it doesn't exist yet, but must declare a name for it:

```
$sZipExtractDir = 'zipExtractDir'; // subdirectory in the document root
```

[5] When corpus files are being uploaded into CoBaLT, the tool will need to tokenize them as a first step before anything else (see chapter about tokenizing). To be able to do that, the tool needs to know where to find the tokenizer: (*use an absolute path*)

```
$sTokenizerDir = '/servername/some_directory/Tokenizer_directoryname';
```

The default tokenizer can do things differently depending on the language it deals with. The proper language can be declared the following way:

```
$sTokenizerLanguage = 'ned';     # Also possible are 'eng', 'esp', etc.
```

If this setting gives unexpected results in your language, the tokenizer.pm file of the default tokenizer will need to be given some extra specific code for proper processing (check the section about ‘Tokenizing’ if you wish to configure the tokenizer for your own language anyway).

[6] For the tool to be able to call the tokenizer the right permissions should be set. The web server runs as a certain user (e.g. 'www-data'). It is necessary for this user to have execute permissions in the tokenizer directory and for the tokenizer executable itself.

For Ubuntu it has proven necessary to add the web server user to the sudoers file and to preprend the call to the tokenizer with a 'sudo' command.
There is a setting for this in the php/globals.php file, called $sSudo. Just give the variable an empty string as value if this not required.

```
$sSudo = '';
```

[7] In some projects you might need to use a different character set. To do so, you need to follow the following steps:

- if you’re using a special font type stored in .eot and .otf files, copy those files to the ‘fonts’ directory of the application (*if you’re not, just skip to the next step*).
- now go to the ‘css’ directory of the application.
- make a copy of the ‘lexiconTool.css’ file.
- open your copy of the file in a text editor.
- now, if you’ve copied .eot and .otf files to the ‘fonts’ directory as a first step, you’ll need to declare those files here (*if you’re not using such files, just skip to the next step*). To do so, add the following lines on top of the file:

```
@font-face {
    font-family: YourFontNameHere;
    src: url('../fonts/YourFontNameHere.eot');
}
@font-face {
    font-family: YourFontNameHere;
    src: url('../fonts/YourFontNameHere.otf');
}
```

- now look for blocks of lines starting with ‘td.wordCol’, ‘.contextWord’, ‘.contextWord_’ and ‘.matchedPart’. If you don’t find some of them, add those with curly brackets like this:

```
td.wordCol {   }
```

- in each of the blocks of lines just quoted, make sure that the curly brackets contain a declaration of your font type, in such a way that it looks like this:

```
td.wordCol {
    font: 8px YourFontNameHere;
}
```

NOTE: If the curly brackets already contain a font declaration, remove the font names declared on those lines and put the font name you need instead.

Now, tell the tool to use your copy of the 'lexiconTool.css' file in the relevant project by adding the following line to the globals.php file:

```
$asCustomCss['your_lexicon_database_name'] = 'your_copy_of_lexiconTool.css';
```

Other options of the php/globals.php file are less likely to need customization, so we won't discuss them here.

TOKENIZING

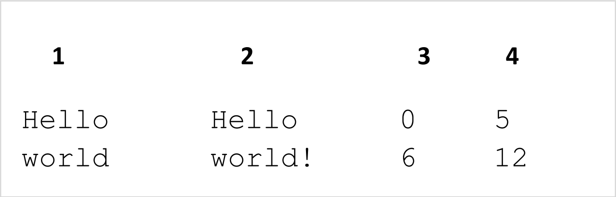
TOKENIZER

A lexicon building project in CoBaLT will always start with uploading a corpus. The tool can without difficulties be fed with untokenized documents. The only condition is that those are saved as UTF-8 encoded plain texts in .txt format. During the upload, those untokenized document will be tokenized by the tokenizer integrated in the tool.

However, it is also possible to upload documents that are tokenized already, which will be recognized as such automatically. In order to determine whether or not a file is tokenized, the tool looks at the first 10 lines to see if they look 'tokenized' like this:

```
canonicalForm<TAB>wordForm<TAB>onset<TAB>offset<TAB>....
```

So, e.g. the next bit could be the output of a tokenizer for the text "Hello world!":



```
  1         2         3    4

Hello     Hello      0    5
world     world!     6    12
```

helloWorld_tokenized.tab

Some character sequences might not be interesting for lexical processing in the tool even though it would be nice to see them displayed for ease of reading. This goes e.g. for punctuation marks that appear separately in a text.

Text like this, that should be displayed but which should not be treated as a wordform in the tool should have the string 'isNotAWordformInDb' in the fifth column of the tokenized file. Other words (as in the example above) should have an empty fifth column. So PLEASE NOTE that in that case a tab will end the line:

```
  1            2            3    4     5

Zamensp      Zamensp.     744  751
over         over         753  757
de           de           758  760
Deenemarken  Deenemarken  801  812
. .                       821  822   isNotAWordformInDb
Bl.          Bl.          844  847
210          210          848  851
```

POSITION INFORMATION FOR OCR'ED MATERIAL

When working with OCR'ed material, there is a relation between the text appearing in the tool and the original image. The tool supports the relation between the two (see section 'Viewing a word form in the original image' below).

The 'documents' table in the MySQL database has a column 'image_location' where a path to the image can be set. Also, the coordinates of a word occurrence in its original document image can be set in the tokenized file in its sixth to ninth column. These columns should specify x-coordinate, y-coordinate, height and width, respectively:

1	2	3	4	5	6	7	8	9
DE	DE	2268	2270		**365**	**1700**	**105**	**1130**
KAARTEN	KAARTEN	2271	2278		**365**	**1700**	**105**	**1130**
MOETEN	MOETEN	2279	2285		**365**	**1700**	**105**	**1130**
GEPLAATST	GEPLAATST	2286	2295		**365**	**1700**	**105**	**1130**
WORDEN	WORDEN.	2296	2302		**365**	**1700**	**105**	**1130**

TOKENIZING XML

The default tokenizer can also handle XML files in certain formats. Currently the supported formats are IGT XML (the IMPACT ground truth format) and TEI (as provided by JSI).

A TEI file should look the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<TEI.2>
<teiHeader type="letter"></teiHeader>
<text>
<body>
<div type="letter" xmlns="http://www.tei-c.org/ns/1.0" xml:lang="nl"
n="filename">
<p>
   <w>Desen</w>
   <c> </c>
   <w>brijef</w>
   <c> </c>
   <w>sal</w>

   (…)

   <c> </c>
   <w>ul</w>
   <c> </c>
   <w>huijsvrou</w>
</p>
</div>
</body>
</text>
</TEI.2>
```

TEI files need a TEI-header, within which the document type is stated. The actual text content is enclosed within a text-tag, which in turn has a body-tag.

A body-tag has div-tags enclosed. Within the div-tag, each paragraph of a text must be enclosed within a p-tag. Single words are enclosed within w-tags, and the space in-between in encoded by c-tags. The example above shows all that.

If a word has some punctuation attached, a normalized form (without punctuation) is added to the w-tag as an attribute:

```
<w nform="pas">pas,</w>
<c> </c>
<w nform="an">[…]an</w>
```

A w-tag isn't allowed to contain any other tag. So, if any property of a word needs to be given, it must always be set as an attribute of the w-tag.

So, do absolutely NEVER write something like:

```
<w><abbr>ver</abbr>staen</w>
```

But do instead write this:

```
<w original="&lt;abbr&gt;ver&lt;/abbr&gt;staen">verstaen</w>
```

The above example shows that the original word with inside-tags has to be escaped and put into an 'original' attribute of the w-tag.

CHOOSING THE RIGHT FORMAT: .TXT, .TAB OR .XML (TIE)?

Before choosing to feed CoBaLT with a corpus in a given format, it is important to know the consequences of one choice above the other.

If you want to upload a corpus into CoBaLT so as to be able to annotate it, at the end of the project you'll also probably want to get your original corpus files back enriched with those annotations. This is can only be achieved if you choose for uploading in .xml format (TIE). At the end of a project, the scripts described in the 'Data export' section with actually take your original files (stored in the database) and add your annotations to those before exporting them to disk. CoBaLT won't be able to do so if you originally loaded something else than .xml.

If you especially want to use CoBaLT to build a lexicon, but don't feel the need to enrich your original corpus files with annotations, then loading your corpus as .txt files (plain text) or as .tab (tokenized) files is good enough. As the project is being worked on in CoBaLT, the lexicon will be gradually built or enriched in the so-called lexicon database (which you have created in the 'Create databases' section). All you'll have to do at the end of a project will be copying the lexicon database to a suitable location for your own needs; or you can export this database to .xml (check the 'Data export' section about that).

Of course, .tab has the advantage above .txt format that you can add position information for OCR'ed material and so on, which is impossible in .txt format. Unfortunately adding position information is not supported yet in .xml (TIE) format at the moment.

CUSTOMIZE THE TOKENIZER FOR YOUR OWN LANGUAGE

For a tokenizer, being able to deal with a specific language means knowing its common abbreviations, so as to be able to distinguish an ‘abbreviation dot’ from a ‘sentence end dot’. It also means knowing words commonly bordered by a comma (like *’em* in English), so as to be able to distinguish such comma’s from the ones used in quotations. It is very important that the tokenizer actually knows these special cases, because every dot it encounters will otherwise be interpreted as a sentence end, and a comma as the beginning or end of a quote.

So, you’ll need to build two files containing abbreviations and comma words or your specific language.

Let’s start with the filenames. Let’s say you’d like the tokenizer to be able to process English. Choose an short name for this language, say ‘eng’. This is the name you’ll have to use throughout the tool configuration to refer to this language. Now create two text files with this short name as an extension:

```
abbr.eng
apostrof.eng
```

Now you’re ready to tell the tool which abbreviations and comma words it should be able to recognize. You must declare those abbreviations in the ‘abbr.eng’ file, and the comma words in the ‘apostrof.eng’ file. Open the files in a text editor, and make sure both files contain just one word per line:

```
app.
Arab.
Aram.
arbitr.
```

abbr.eng

```
'em
'til
'cause
```

apostrof.eng

When you’re finished, put both files into the tokenizer directory, so the tokenizer will be able to find them. If you’re not sure where this directory is to be found, look it up in the php/globals.php file, as the tokenizer location must have been set there (check the ‘CoBaLT configuration’ section about this file):

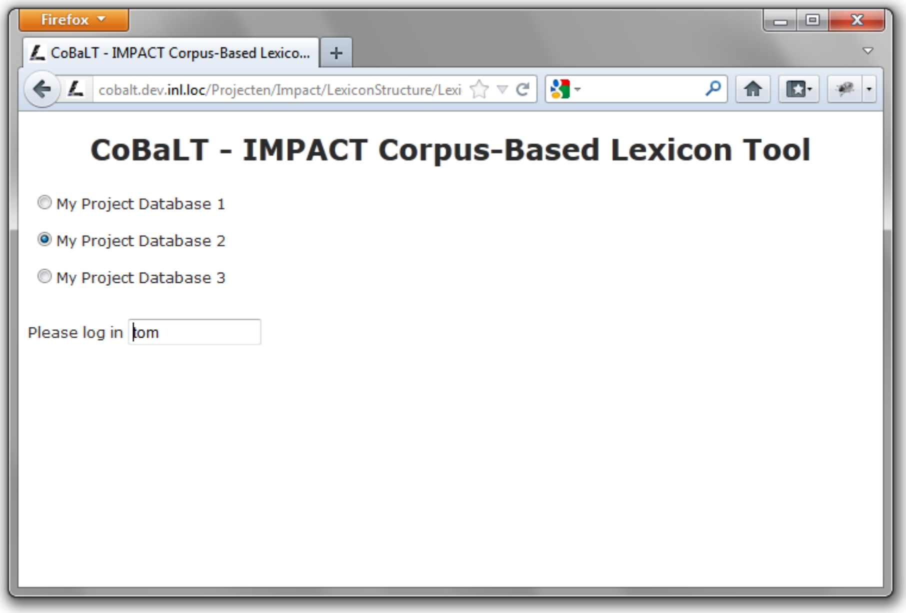
```
$sTokenizerDir = '/servername/some_directory/Tokenizer_directoryname';
```

As a last step, you’ll need to tell the tool you want it to tokenize English now. You can achieve that by setting the following in the php/globals.php file:

```
$sTokenizerLanguage = 'eng';
```

THE USER INTERFACE

LOG-IN SCREEN


Firefox
CoBaLT - IMPACT Corpus-Based Lexico...
cobalt.dev.inl.loc/Projecten/Impact/LexiconStructure/Lexi

CoBaLT - IMPACT Corpus-Based Lexicon Tool

○ My Project Database 1

● My Project Database 2

○ My Project Database 3

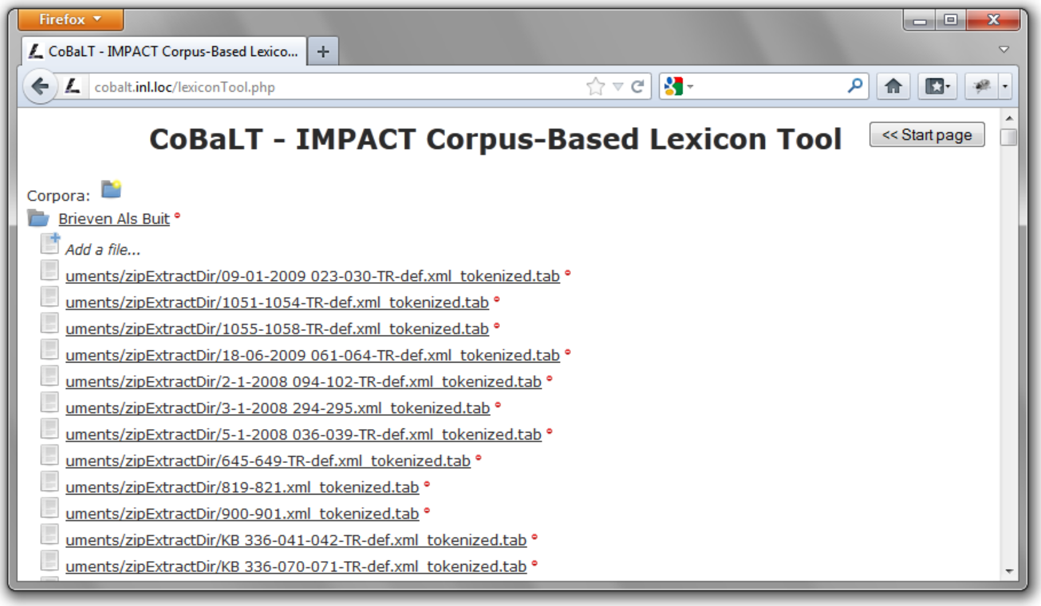
Please log in tom

Screenshot of CoBaLT – log-in screen

The first screen is a log-in screen. The different lexicon projects listed here should have been set in the 'CoBaLT configuration' section. The user logs in by simply typing in his/her name and chooses a project to work on.

CORPUS SCREEN

In the next screen the user can choose a corpus to work on, corpora can be deleted, new corpora can be added and files can be added to or deleted from corpora.



Firefox
CoBaLT - IMPACT Corpus-Based Lexico...
cobalt.inl.loc/lexiconTool.php
<< Start page

CoBaLT - IMPACT Corpus-Based Lexicon Tool

Corpora:

Brieven Als Buit

Add a file...

uments/zipExtractDir/09-01-2009 023-030-TR-def.xml tokenized.tab
uments/zipExtractDir/1051-1054-TR-def.xml tokenized.tab
uments/zipExtractDir/1055-1058-TR-def.xml tokenized.tab
uments/zipExtractDir/18-06-2009 061-064-TR-def.xml tokenized.tab
uments/zipExtractDir/2-1-2008 094-102-TR-def.xml tokenized.tab
uments/zipExtractDir/3-1-2008 294-295.xml tokenized.tab
uments/zipExtractDir/5-1-2008 036-039-TR-def.xml tokenized.tab
uments/zipExtractDir/645-649-TR-def.xml tokenized.tab
uments/zipExtractDir/819-821.xml tokenized.tab
uments/zipExtractDir/900-901.xml tokenized.tab
uments/zipExtractDir/KB 336-041-042-TR-def.xml tokenized.tab
uments/zipExtractDir/KB 336-070-071-TR-def.xml tokenized.tab

Screenshot of CoBaLT – corpora screen

CORPUS MODE VS DOCUMENT MODE

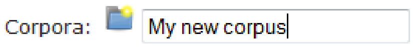
The tool can work in two modes: corpus mode and document mode.

By clicking on a corpus name the tool will load in corpus mode (i.e. the user will work on a list of word forms for an entire corpus). By clicking on a document name instead, the tool will load in document mode, which means that you will be exclusively working with the tokens of that particular document.

UPLOADING DOCUMENTS

Documents can be uploaded into the tool one by one, or a lot of them together in a zip file. The tool checks filename extensions. Only .txt files, .tab files and .xml files are processed (and of course .zip for a collection of such files).

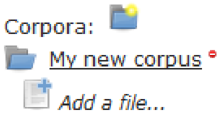
First click on the directory icon to make a new corpus directory:



Corpora: My new corpus

Then tell the tool where the corpus files are to be found (remember collections of files must be zipped in advance). **Please note** that every single file should be utf-8 encoded. No other encoding format is supported.

Click on the 'Add a file…' icon, and you'll be able to browse to the right location to upload your file from:



Corpora:

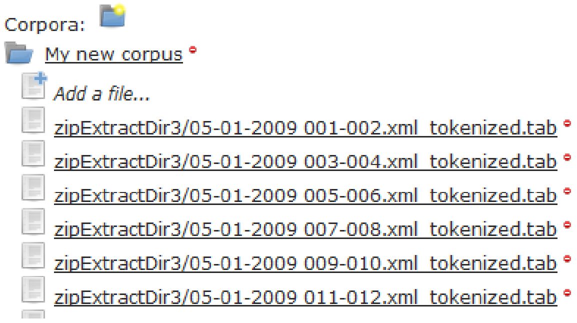
My new corpus

Add a file...

Finally, click onto the 'Upload file' button and wait till the file is processed. If you've been choosing a large zip file, the processing can take quite a while (a few seconds after uploading started, a progress indicator will be shown in the page tab on top of the browser window).

Any untokenized document will be tokenized by the tokenizer integrated in the tool. However, it is also possible to upload tokenized documents (which will be recognized as such automatically). This is particularly handy if you need your tokenization to be different from what the default tokenizer provides. Please refer to the 'Tokenizing' section to see what a tokenized file should look like.

When the file upload has completed, the screen should look like this:



Corpora:

My new corpus

Add a file...

zipExtractDir3/05-01-2009 001-002.xml_tokenized.tab

zipExtractDir3/05-01-2009 003-004.xml_tokenized.tab

zipExtractDir3/05-01-2009 005-006.xml_tokenized.tab

zipExtractDir3/05-01-2009 007-008.xml_tokenized.tab

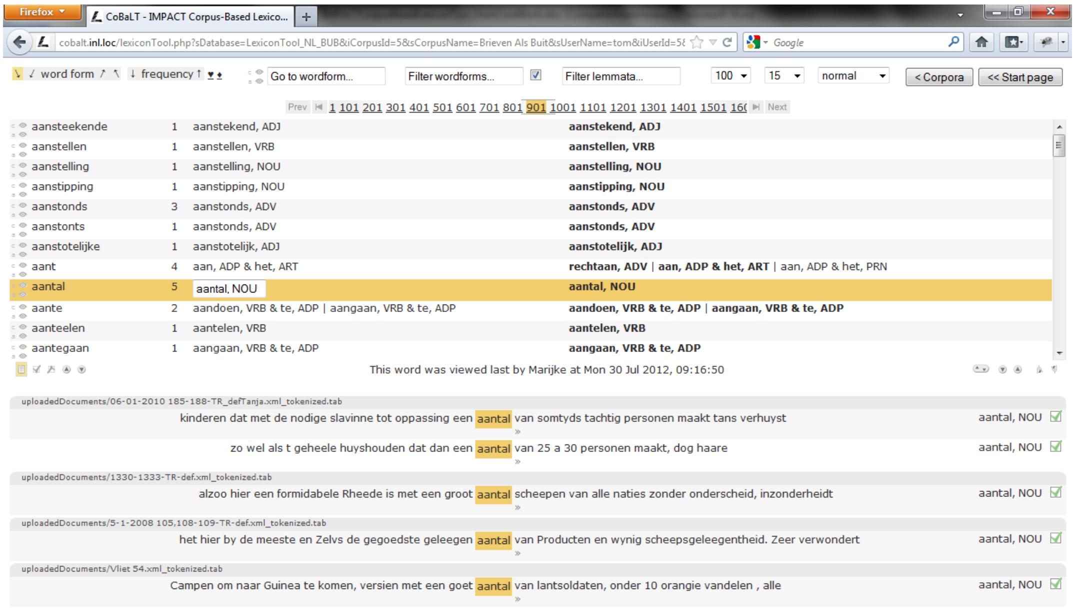
zipExtractDir3/05-01-2009 009-010.xml_tokenized.tab

zipExtractDir3/05-01-2009 011-012.xml_tokenized.tab

Now you're ready to go and work with your files. Click on the corpus name if you wish to see and/or process the whole corpus at once, or click on a single file name if you'd like to work only with that particular document.

MAIN SCREEN

As soon as you've loaded files in the Corpus screen, you'll be able to access the Main screen. This is the place where you'll be able to work on your data:


Firefox
CoBaLT - IMPACT Corpus-Based Lexico...
cobalt.inl.loc/lexiconTool.php?sDatabase=LexiconTool_NL_BUB&iCorpusId=5&sCorpusName=Brieven Als Buit&sUserName=tom&iUserId=5
Google
word form
frequency
Go to wordform...
Filter wordforms...
Filter lemmata...
100
15
normal
< Corpora
<< Start page
Prev 1 101 201 301 401 501 601 701 801 901 1001 1101 1201 1301 1401 1501 160 Next

aansteekende	1	aanstekend, ADJ	**aanstekend, ADJ**
aanstellen	1	aanstellen, VRB	**aanstellen, VRB**
aanstelling	1	aanstelling, NOU	**aanstelling, NOU**
aanstipping	1	aanstipping, NOU	**aanstipping, NOU**
aanstonds	3	aanstonds, ADV	**aanstonds, ADV**
aanstonts	1	aanstonds, ADV	**aanstonds, ADV**
aanstotelijke	1	aanstotelijk, ADJ	**aanstotelijk, ADJ**
aant	4	aan, ADP & het, ART	**rechtaan, ADV** | **aan, ADP & het, ART** | aan, ADP & het, PRN
aantal	5	aantal, NOU	**aantal, NOU**
aante	2	aandoen, VRB & te, ADP | aangaan, VRB & te, ADP	**aandoen, VRB & te, ADP** | **aangaan, VRB & te, ADP**
aanteelen	1	aantelen, VRB	**aantelen, VRB**
aantegaan	1	aangaan, VRB & te, ADP	**aangaan, VRB & te, ADP**

This word was viewed last by Marijke at Mon 30 Jul 2012, 09:16:50

uploadedDocuments/06-01-2010 185-188-TR_defTanja.xml_tokenized.tab

kinderen dat met de nodige slavinne tot oppassing een aantal van somtyds tachtig personen maakt tans verhuyst — aantal, NOU ☑

zo wel als t geheele huyshouden dat dan een aantal van 25 a 30 personen maakt, dog haare — aantal, NOU ☑

uploadedDocuments/1330-1333-TR-def.xml_tokenized.tab

alzoo hier een formidabele Rheede is met een groot aantal scheepen van alle naties zonder onderscheid, inzonderheidt — aantal, NOU ☑

uploadedDocuments/5-1-2008 105,108-109-TR-def.xml_tokenized.tab

het hier by de meeste en Zelvs de gegoedste geleegen aantal van Producten en wynig scheepsgeleegentheid. Zeer verwondert — aantal, NOU ☑

uploadedDocuments/Vliet 54.xml_tokenized.tab

Campen om naar Guinea te komen, versien met een goet aantal van lantsoldaten, onder 10 orangie vandelen , alle — aantal, NOU ☑

Screenshot of the CoBaLT – main screen

In the screenshot above we see CoBaLT in action. It is running in corpus mode, which means that the user selected a collection of documents to work on ('corpus mode'), rather than just one single document ('document mode').

The interface is divided in three main parts.

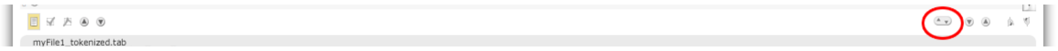
- Top bar
  For sorting and filtering word forms and lemmas, and adjusting the views on the data.

- Middle part
  Shows the type-frequency list plus analyses in the corpus/document and database.

- Lower part
  Shows the individual occurrences of word forms ('tokens') in context plus their analyses.

ADJUSTING THE SIZE OF THE WINDOW PARTS

The ratio between the middle part and the lower part can be altered by dragging the resize icon (circled in red in the partial screenshot below).

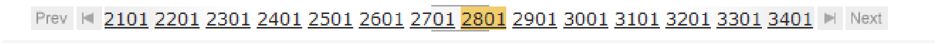


Partial screenshot of CoBaLT – resize window parts

NAVIGATION

CoBaLT offers two different way to navigating through the data. The first one is a scroller bar, the second one is a ‘Go to wordform’ box:

SCROLLER BAR



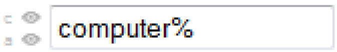
Partial screenshot of CoBaLT – top scroller bar

When there is a large amount of pages available a scroller bar will appear at the top of the screen. Hovering the mouse cursor a the right side of the middle will scroll the list of pages to the left and vice versa. To stop the list from scrolling, move the mouse cursor to the middle of the bar. Click onto the highlighted number to get to that page number.

‘GO TO WORDFORM…’ BOX

When a data set consists of hundreds of pages, using the scroller bar to access the right page means scrolling for quite a while before the right page gets in sight.

If you do know exactly which wordform you’d like to get to, just type it into the ‘Go to wordform…’ box and press ‘Enter’. CoBaLT will then look up the right page and load it right away. This also works with only the prefix of a wordform (just enter a prefix followed by a %-sign to indicate some unknows letters should follow). The following example catches both ‘computer’ and ‘computers’:

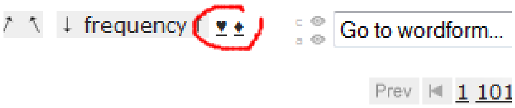


computer%

Partial screenshot of CoBaLT – ‘Go to wordform…’ box

GETTING STATISTICS

As a project is getting on, one might want to know how much work still has to be done. The tool offers two possibilities:
- click onto the small heart icon to get a list of all word forms that are not yet lemmatized (at all)
- click onto the small diamond icon to get a list of all word forms there were but partially lemmatized.



Partial screenshot of CoBaLT – heart and diamond icons

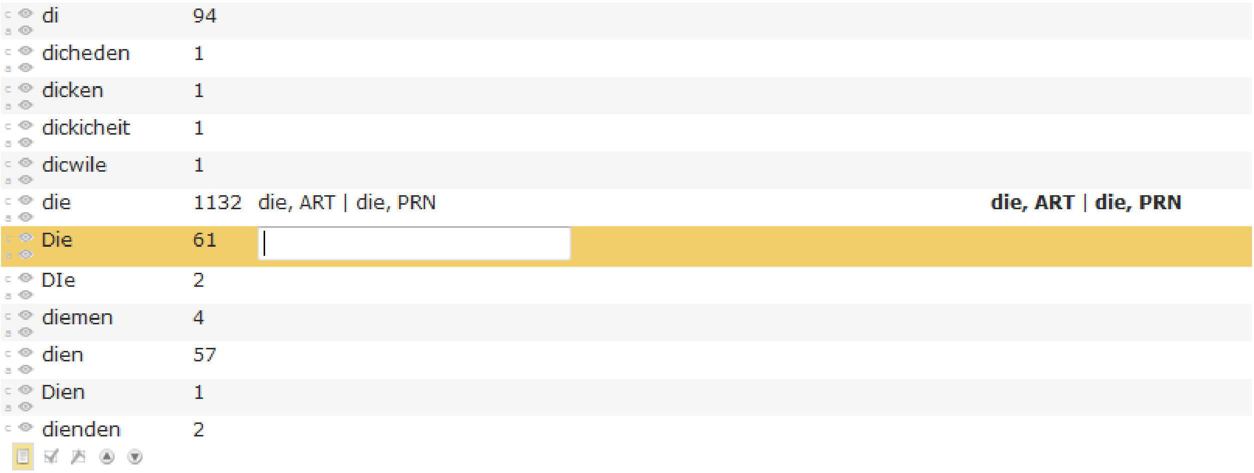
In the following section it is described how analyses can be edited in the lower as well as in the middle part, what analyses may consist of, and how to customize the ways in which the data are presented.
An overview of key functions and mouse actions can be found below. Note that with most buttons and also with the analyses in the right-hand column in the middle part, an explanatory box appears when the mouse cursor is hovered over it.

THE MIDDLE PART: WORD FORM TYPES AND THEIR ANALYSES IN CORPUS/DOCUMENT AND DATABASE

Type-frequency list

Analyses
in loaded corpus
or document

Analyses in the
entire database



di	94		
dicheden	1		
dicken	1		
dickicheit	1		
dicwile	1		
die	1132	die, ART | die, PRN	**die, ART | die, PRN**
Die	61		
DIe	2		
diemen	4		
dien	57		
Dien	1		
dienden	2		

Partial screenshot of CoBaLT – middle part

In the middle part of the screen a type-frequency list is displayed with some additional options and information. A selected row is editable, allowing the user to add/delete analyses. A row is selected by clicking on it; it will be highlighted yellow (unless it is hidden, in which case it is displayed pink; see below). In this part only one row can be selected at a time; by using arrows the previous or the next row becomes selected.

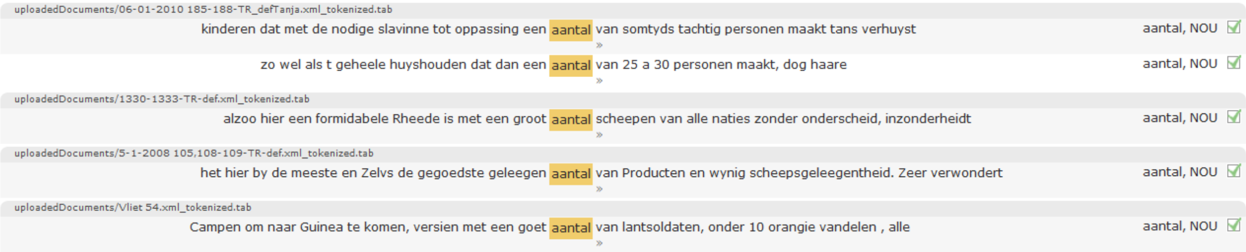
The middle column displays the analyses a word form has in the loaded corpus or document (depending on the mode).

The rightmost column in the middle part shows all the analyses linked to a word form in the entire database; those in bold are validated (see below).

When a word form is selected (like *Die* in the example screenshot), its occurrences are loaded in the lower part of the screen together with some context (see next section).

As actions performed in the middle part apply to the tokens listed in the lower part, the functioning of the latter will be explained first.

THE LOWER PART: OCCURRENCES IN CONTEXT



uploadedDocuments/06-01-2010 185-188-TR_defTanja.xml_tokenized.tab

kinderen dat met de nodige slavinne tot oppassing een aantal van somtyds tachtig personen maakt tans verhuyst » aantal, NOU ☑

zo wel als t geheele huyshouden dat dan een aantal van 25 a 30 personen maakt, dog haare » aantal, NOU ☑

uploadedDocuments/1330-1333-TR-def.xml_tokenized.tab

alzoo hier een formidabele Rheede is met een groot aantal scheepen van alle naties zonder onderscheid, inzonderheidt » aantal, NOU ☑

uploadedDocuments/5-1-2008 105,108-109-TR-def.xml_tokenized.tab

het hier by de meeste en Zelvs de gegoedste geleegen aantal van Producten en wynig scheepsgeleegentheid. Zeer verwondert » aantal, NOU ☑

uploadedDocuments/Vliet 54.xml_tokenized.tab

Campen om naar Guinea te komen, versien met een goet aantal van lantsoldaten, onder 10 orangie vandelen , alle » aantal, NOU ☑

Partial screenshot of CoBaLT – lower part

The lower part of the screen shows the occurrences (tokens) in context of the word form selected in the middle part. The right-hand column lists the analyses of the word forms in each context.

Each occurrence is presented with some context (see below on how to adjust the amount of context in view, or the number of context rows per page).
Token plus context can be selected and deselected by clicking anywhere in the corresponding row[1]. You can select a group of adjacent rows by holding down the mouse button while moving the mouse from the first row of the group down to the last row you wish to select. Non-adjacent rows can be selected by Ctrl-clicking them one at the time (t.i. clicking while holding down the Ctrl key). Select all rows at once by double clicking one single row. In order to deselect a single row from a group selection, Ctrl-click that row.

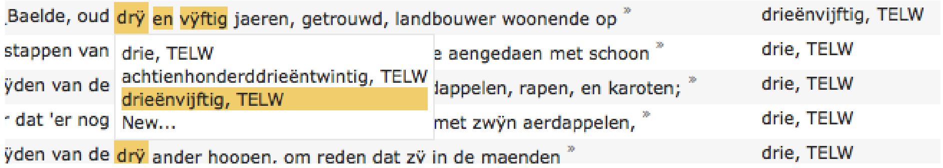
In this part of the tool, any action that is performed is applied to all rows that happen to be selected. Note that not every row that is selected is necessarily in view; see below on the number of context rows per page.

[1] You might want to avoid clicking on the analyses on the right though as these will be deleted when you do that.

EDITING ANALYSES ON THE TOKEN LEVEL

Analyses can be added, removed, and validated in various ways on interdependent levels. The most specific level is that of a token in context.

By clicking on a yellow-highlighted word in its context in the lower part of the screen, a drop-down menu appears which lists all corresponding analyses in the database, with those analyses already assigned to the token in question highlighted. In the following screenshot, the word group *drÿ en vÿftig* appears to have the analysis *drieënvijftig, TELW* assigned.



.Baelde, oud drÿ en vÿftig jaeren, getrouwd, landbouwer woonende op »	drieënvijftig, TELW
stappen van ... e aengedaen met schoon »	drie, TELW
ÿden van de ... dappelen, rapen, en karoten; »	drie, TELW
· dat 'er nog ... met zwÿn aerdappelen, »	drie, TELW
ÿden van de drÿ ander hoopen, om reden dat zÿ in de maenden »	drie, TELW

Partial screenshot of CoBaLT – close up of analysis drop-down menu

By switching analyses 'on' or 'off', they will be (de)assigned to the token(s) in the selected row(s). The option *New…* at the bottom of the menu provides the possibility to type in a new analysis in a text box, with existing analyses being suggested as you type.

The rightmost column shows the analyses for the occurrence in this context. Clicking on one of them will result in this analysis being deleted from the row it is in, as well as from any other selected rows it featured in.

ONE ANALYSIS FOR A GROUP OF WORDS

In the example screenshot the words *drÿ*, *en,* and *vÿftig* are marked together to make up the analysis *drieënvijftig, TELW*. You can add words to a group, or delete them from it by clicking on them while holding down the F2 or F9 key. If a word you just added to the group already has analyses of its own in this particular context, these will show in the right-hand column.

MORE LEMMATA FOR ONE WORD FORM

Conversely, a word form may occasionally consist of more than one lemma. Multiple analyses can be assigned by using an ampersand (&); e.g., *l'équipe* could be analysed as *la, DET & équipe, N*.

VALIDATION

To indicate whether attestations and analyses are verified by a user, they can be validated.

Word forms in their context can be validated (regardless of whether they have an analysis or not) by checking a validation box at the right of the context row in the lower part of the screen. This can be interpreted as 'the user saw the word form in this context and approves of the listed analyses'. As with any other action in this part of the screen, (de)validation is applied to any sentences that happen to be selected.

By assigning an analysis to a token attestation in the lower part of the screen (either by choosing an option from the drop-down list as described above or by an action in the middle part as will be described below), the token plus its analyses become validated automatically.

When the validation checkbox is greyed out a bit, this means that this occurrence of the word form in this context was validated by a different user.

When a token is attested in its context or is validated, the analyses associated with the word form in this context automatically become validated. Validated analyses are displayed in **bold** in the right column of the middle part in the tool (in the example screenshot nearly all analyses are validated).
Analyses can also be (un)validated 'by hand' by Shift-clicking them (clicking while holding down the Shift key).

NOTE that it is not necessary for a validated type analysis to have a token attestation. E.g., the analysis *duizend, TELW* for the word form *duijsent* in the example screenshot is currently unvalidated. It only occurs once in the corpus, apparently as part of a group *duizendachthonderdrieëntwintig*. A user might decide to validate the analysis *duizend, TELW* nevertheless, even though there is no attestation to support it.

EDITING ANALYSES IN THE MIDDLE PART

The rightmost column in the middle part of the screen shows all the analyses a word form (type) has in the entire database. As mentioned above, those in bold are validated; they can be (un)validated by Shift-clicking on them.

By clicking on an analysis it is applied to the rows that are selected in the lower part of the tool, which will in turn become validated (shown by the checkbox at the right being checked). When no row is selected, the analysis will be assigned to all tokens in the lower part, but these will NOT become validated.
The idea behind this is that in this way it is easy/fast to, e.g., when a word has more than one analysis, assign these to all occurrences of a word form without further disambiguation. If the user goes through the trouble of selecting one or more rows (s)he must be pretty sure about it, and the analyses become validated.

An analysis for a certain word form can be deleted altogether by Ctrl-clicking it in the right-hand column. As a consequence of course, the analysis disappears as well for any token attestation for the word form in question it featured in.

As with the analyses of the entire database, the corpus/document analyses filled in in the text box in the middle part will be applied to the selected row(s) in the lower part of the tool which then will also become validated. Again, when no row is selected in the lower part, the analyses assigned in the middle part will be applied to all token attestations without them being validated.

WHICH ANALYSES APPEAR IN WHICH COLUMN?

The lower part of the screen shows occurrences in context from the corpus/document. By assigning analyses to these occurrences, the word forms become attested.

As said earlier, the middle column in the middle part of the screen shows the analyses a word form has in the current corpus/document. So these will match with the ones in the lower part of the screen.

The analyses in the rightmost column however do not necessarily show anywhere, which may be a source of confusion. This can be the case if these analyses are for word forms in a document in the same database but in a different corpus, or because they are not associated with any word form in context at all (i.e. the analysis *is* associated with the word form, or it wouldn't show in the first place, but there is no attestation in context anywhere yet). The latter may e.g. be the case when a database comes preloaded with information from an external lexicon/dictionary.

ON WORD FORM ANALYSES

WHAT DO ANALYSES LOOK LIKE?

An analysis as it is used in the tool (and this manual) refers to a tuple that can be described as:

modern lemma (, <modern_equivalent>) (, [set of patterns]) , part of speech (, language) (, gloss)

The parts between round brackets are optional. Only the lemma and the part of speech are obligatory, so a typical, simple lemma would be e.g. *the, DET*.

MODERN EQUIVALENT AND PATTERNS

Optionally, a modern equivalent word form and some patterns can be specified in an analyses.

E.g. for the German word form *theyle* an analysis could be *teil, <teile>, [(t_th, 0), (ei_ey, 2)], N*. The part between angled brackets (<>) is the modern equivalent word form (which is possibly inflected) and the part in between the square brackets represents a series of pattern substitutions to get from the modern equivalent to the historical one. In the example, the substitution *th* for *t* should be applied at position *0* (the first character).

Neither modern equivalent nor patterns are obligatory. Nor are they required to be specified both. In other words

teil, [(t_th, 0), (ei_ey, 2)], N and *teil, <teile>, N*

are valid lemmas as well.

LANGUAGE

Next to the part of speech, a language may be specified. This could be used e.g. to keep Latin phrases apart from other text.

GLOSS

To be able to keep similar lemmas apart, a gloss may be added. E.g. *paper, N, material* versus *paper, N, essay* or *paper, N, newspaper*.

NOTE that if language names are to be used the table 'languages' in the database should be filled with the various options. It is only when the first word after the part of speech in the analyses matches one of these options that it is treated as a language. If it doesn't match it is treated as (part of) the gloss.

ONE WORD FORM, MORE LEMMAS

Sometimes a word form might better be analysed as consisting of two or more lemmas rather than one. In the tool this can be done by separating analyses by ampersands (&'s). E.g. *l'Afrique* could be analysed as *le, DET & Africa, NELOC*.

NOTE that in these cases the analysis cannot contain modern equivalents or patterns.

ONE ANALYSIS FOR SEVERAL WORDS

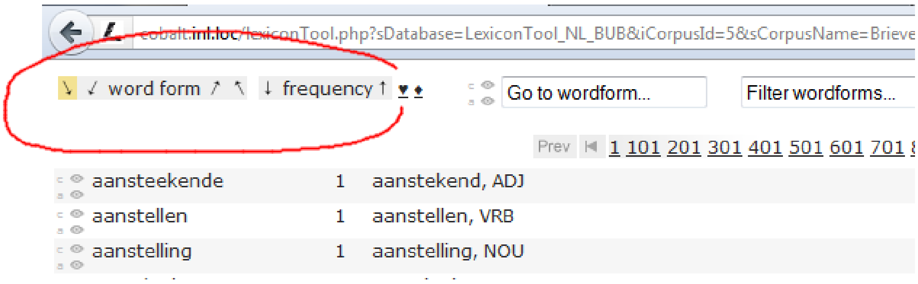
Sometimes, what can be thought of as one word appears as two or more. Consider e.g. separable verbs, which are very common in Dutch. *Meedoen* means to participate, and *"I participate"* would translate to *"Ik doe mee"*. In this phrase, the word forms *doe* and *mee* together make up the analysis *meedoen, V*.

The word forms forming one lemma do not have to be next to each other. *"Nina en Ella doen morgen mee"* (*"Nina and Ella will participate tomorrow"*) can be analysed as well by clicking on them while holding down the F2 or F9 key.

CUSTOMIZING THE VIEWS ON THE DATA

SORTING

The type-frequency list in the middle part of the tool can be sorted by using the arrows at the left side of the top bar. The list can be sorted alphabetically by word form, or by frequency. The alphabetical ordering can also be done from right to left (so e.g. *blueish, reddish* and *greyish* will appear near each other).



cobalt.inl.loc/lexiconTool.php?sDatabase=LexiconTool_NL_BUB&iCorpusId=5&sCorpusName=Brieve

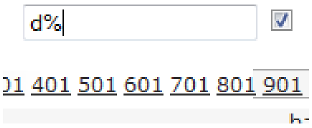
aansteekende	1	aanstekend, ADJ
aanstellen	1	aanstellen, VRB
aanstelling	1	aanstelling, NOU

Partial screenshot of CoBaLT – top bar

FILTER ON WORD FORMS

There are two filterboxes in the top bar. The filter in the left-hand one applies to the word forms in the type-frequency list. You can filter on word form beginnings, endings or, in fact, anything in between. The filter is directly passed to the MySQL LIKE operator.

For not-MySQL-guru's: the most frequently used wildcard is % which means any sequence of characters. So %a% means: any word form matching an arbitrary sequence of characters, then an *a* and then possibly some more characters. So *ball* would match, as would *dance* or *pizza* or indeed any word form with an *a* in it (including the word *a* itself). In the screenshot d% means all the word forms starting with a *d* (so *dance* would match again, but e.g. *adorable* wouldn't).



d% ☑

For further documentation please refer to the MySQL documentation.

The filter is case-insensitive by default, but unchecking the box next to it will make it case-sensitive.

To de-activate a filter, empty the filterbox and apply it (by hitting 'Enter').

FILTER BY LEMMA

To the right of the box for filtering word forms is a box for filtering by lemma. In this box you can type in a lemma and its part of speech, separated by a comma (e.g., *lemma, NOU*) and only word forms that have this lemma assigned to them will be shown.

Please note that patterns are not supported in this box, only complete analyses.

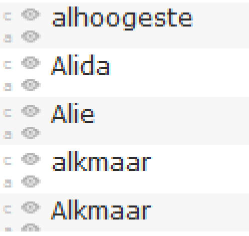
EDIT A LEMMA OR DELETE IT FROM THE DATABASE

When a lemma filter is applied and matches a lemma, an additional icon appears next to the lemma filter box. By clicking on this icon a new sub window opens in which a lemma can be edited or deleted. Please note that editing or deleting a lemma will apply to that particular lemma in the entire database (not just the corpus selected).

HIDING/SHOWING WORD FORMS

There can be various reasons for hiding word forms. It could be, e.g., that one feels that words in certain closed categories (let's say determiners like *the* and *a*) have been dealt with sufficiently and there is no need to analyse them time and time again for every new document or corpus. Or it could be convenient for a particular task to temporarily hide all word forms that have no characters in them (so e.g. cipher words will not show in the list).

At the left side of each row in the middle part of the screen there are two buttons labeled *c* and *a* in corpus mode (for "don't show in the corpus" and "don't show at all", respectively), or *d* ("don't show for this document) and *a* in document mode.



alhoogeste
Alida
Alie
alkmaar
Alkmaar

The small c- and a-buttons
to the left of the middle part.

The *d* button (only visible in document mode) is for hiding the word form of that row for the current document. The *c* button (only visible in corpus mode) is for hiding the word form in that row for the current corpus.

The *a* button is for hiding the word form for the entire database regardless of corpus or document. “Never show me this word again!”.

By switching on one of these buttons the row will be shown as hidden and displayed in pink. When the type-frequency list is reloaded, e.g., when a new filter is applied, or the user logs in again, the word form in question will not show again, unless the relevant show/hide button in the top bar is switched on.

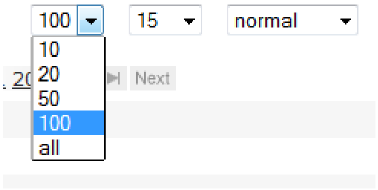
In the top bar, just left of the filtering box, there are two buttons for showing, or hiding again, hidden word forms. They too are labelled *c* and *a* in corpus mode, or *d* and *a* in document mode. By default, these buttons are switched off (i.e., hidden word forms are not shown).

Word forms may be marked — by using the buttons in the middle part, or by means of a script — to be e.g. "hidden for this corpus". If the *c* button at the top is inactive, the word form will not be shown. The word forms are 'unhidden' if the *c* button is activated.

E.g., the row for *dra* in the example screenshots above is hidden for the current corpus, but is shown nevertheless because also the *c* button is switched on in the top bar, which means, "do show all word forms that were hidden for this corpus".

NUMBER OF WORD FORMS PER PAGE

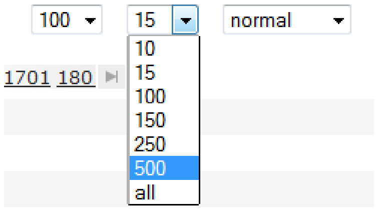
The number of word forms displayed per page in the middle part is set on 100 by default. In the top bar, to the right of the filtering boxes, this can be changed into 10, 20, 50, 100 or 'all'. The latter means that all word forms that match the filter are displayed on a single page. In a large corpus this set can be very large, resulting in a long loading time.



Your choice will determine the number of pages the whole corpus/document is divided into. The horizontal bar with the page numbers will be shown only if there is more than one page to be displayed. By clicking on a number in this bar you jump to the corresponding page and the number will be highlighted. If there are a lot of word forms, the bar becomes scrollable.

NUMBER OF CONTEXT ROWS PER PAGE

When working with large corpora some word forms might occur very often. By clicking on them in the middle part of the screen, all the contexts they occur in will be loaded into the lower part, which may slow down things considerably. Because of this, the number of context rows shown per word form per page is 15 by default. Do note however that the speed advantage is most striking when the sentences are not sorted, which is the default (‘sorted by document’). If they are sorted, the tool has to actually collect *all* the sentences in the background, sort them, and then show a subset of them, so this is somewhat slower.

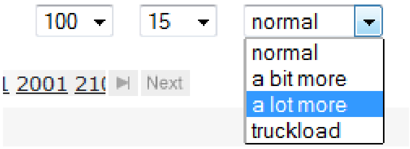


The number of context rows to be loaded in one page can be adjusted in the top bar, from 10 up to ‘all’. If there are more rows than fit on one page, a horizontal bar appears with clickable numbers to go to other pages.

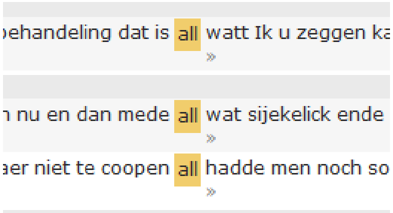
Beware that with more rows per page than visible on the screen, rows may happen to be selected, and thus affected by actions, without actually being in view.

AMOUNT OF CONTEXT

The amount of context surrounding the word form occurrences in the lower part of the tool, by default set to ‘normal’, can be increased by a drop-down menu in the top bar.



The user can also see more context for a particular token by clicking the » (guillemet) at the right of the context row in question:

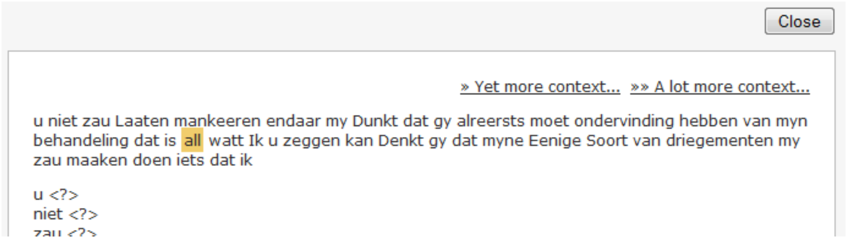


ɔehandeling dat is all watt Ik u zeggen ka »

ı nu en dan mede all wat sijekelick ende »

ɜer niet te coopen all hadde men noch so »

A pop-up window will appear in which more context is shown. In this window the user can be shown even more context. Or even more, as long as there is more context to show.



u niet zau Laaten mankeeren endaar my Dunkt dat gy alreersts moet ondervinding hebben van myn behandeling dat is all watt Ik u zeggen kan Denkt gy dat myne Eenige Soort van driegementen my zau maaken doen iets dat ik

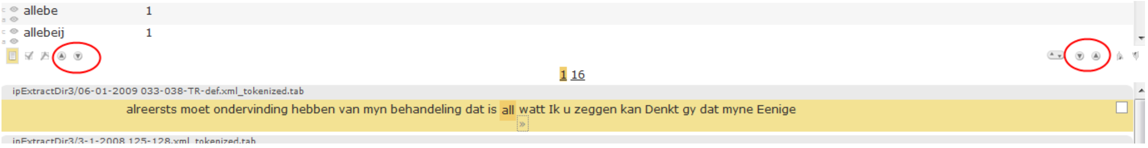
u <?>
niet <?>
zau <?>

SORTING THE TOKENS IN CONTEXT

By default, the context rows in the lower part of the tool are displayed in the order in which they appear in the documents (‘sorted by document’).

SORT ROWS BY CONTEXT

It can be convenient however to sort the rows by the immediate context of the occurring word forms. By clicking on the small arrow buttons at the left side of the screen, the rows will be sorted according to the words to the left of the token (either ascending or descending).



allebe 1
allebeij 1

1 16

ipExtractDir3/06-01-2009 033-038-TR-def.xml_tokenized.tab

alreersts moet ondervinding hebben van myn behandeling dat is all watt Ik u zeggen kan Denkt gy dat myne Eenige

ipExtractDir3/3-1-2008 125-128.xml_tokenized.tab

For example:

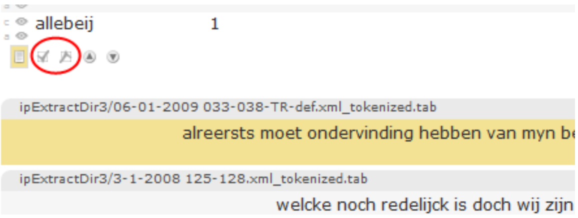
benevens virginie Desmandryl jonge dogter alle	drÿ	[…]
hoop tisten wel hadde konnen onderscheÿden van de	drÿ	[…]
lagen en desen hoop wel onderscheÿden van de	drÿ	[…]
maand meert in het iaar een duizend acht honderd	drÿ	[…]
Pro justitia Ten jaere agtien honderd	drÿ	[…]
van tisten en was gemaekt, maer dat 'er nog	drÿ	[…]
verklaerd genaemd te zÿn pieter Baelde, oud	drÿ	[…]
distinctelijk kwam te remarquéren de voetstappen van	drÿ	[…]

By using the arrow buttons at the right side of the screen the rows can be sorted by the right-hand side of the context. For the example screenshot this would be:

[…]	drÿ	ander hoopen, om reden dat zÿ in de maenden
[…]	drÿ	andere hoopen waeren, vervold met zwÿn aerdappelen,
[…]	drÿ	andere inhoudende verkens-aerdappelen, rapen, en karoten;
[…]	drÿ	diversche persoonen, wanof twee aengedaen met schoon
[…]	drÿ	en twintig Wÿ ondergeteekende joannes
[…]	drÿ	en twÿntig den tiensten februarius, zÿn voor ons
[…]	drÿ	en vÿftig jaeren, getrouwd, landbouwer woonende op
[…]	drÿ	woonende op t' gezegde Langemarck, welke

SORT ROWS BY VALIDATION

Rows can be sorted by validation by clicking on one of the checkbox buttons next to the 'sort by left context' buttons. This way all validated rows are grouped at the top/bottom, so it is e.g. easy to see what has been done and still needs to be done.



allebeij 1

ipExtractDir3/06-01-2009 033-038-TR-def.xml_tokenized.tab

alreersts moet ondervinding hebben van myn be

ipExtractDir3/3-1-2008 125-128.xml_tokenized.tab

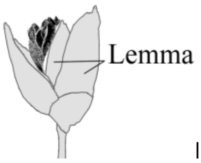
welcke noch redelijck is doch wij zijn

SORT ROWS BY LEMMA

At the right-hand side, there are two buttons for sorting by lemma.[2] Clicking on of these buttons results in the rows being sorted by their analyses.

Again, for the sample screenshot this would amount to:

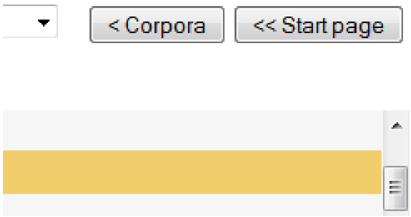
[…]	drÿ	[…]	achtienhonderddrieëntwintig, TELW
[…]	drÿ	[…]	achtienhonderddrieëntwintig, TELW
[…]	drÿ	[…]	drieënvijftig, TELW
[…]	drÿ	[…]	drie, TELW
[…]	drÿ	[…]	drie, TELW
[…]	drÿ	[…]	drie, TELW
[…]	drÿ	[…]	drie, TELW
[…]	drÿ	[…]	drie, TELW



[2] In case you are wondering what the little icons are supposed to look like… they are schematic representations of 'lemmas'. A lemma (also) is a "phytomorphological term used in botany referring to a part of the spikelet of grasses (Poaceae). It is the lowermost of two chaff-like bracts enclosing the grass floret". (Wikipedia: http://en.wikipedia.org/wiki/Lemma_%28botany%29)

BACK TO CORPUS PAGE/START PAGE

At the right-hand side of the top bar there are two buttons. One, labelled *Corpora*, for going back to the available corpora in the current database. The other one, labelled *Start page*, gets you back to the start screen where a database can be selected.



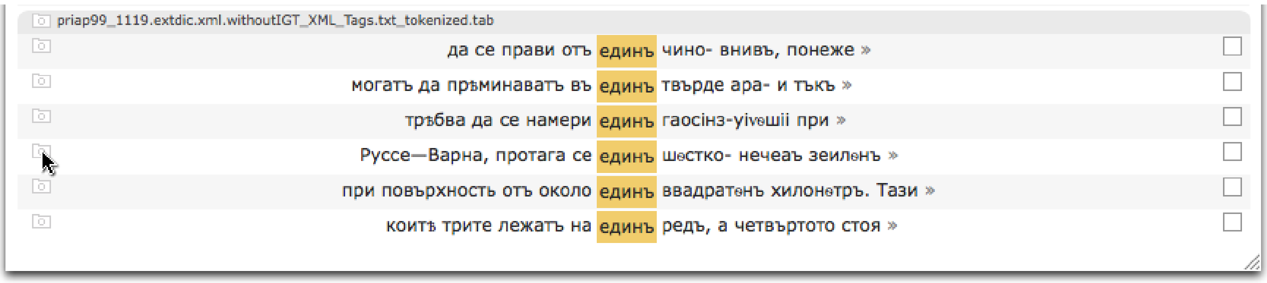
WORKING WITH OCR'ED MATERIAL

CoBaLT is a word form based tool. Word forms are grouped together and can be processed together. It is not always the case though that the word forms themselves are definite. Especially when working with OCR'ed material it could be the case that an OCR error occurred and that a particular word form was mistaken for another.

If the right data is available, the word form can be viewed in the tool in the original image it was scanned from. This way it is very easy, with one mouse click, to see whether or not the word was scanned correctly.

VIEWING A WORD FORM IN THE ORIGINAL IMAGE

If the right data is provided in the database some extra icons will show in the tool.

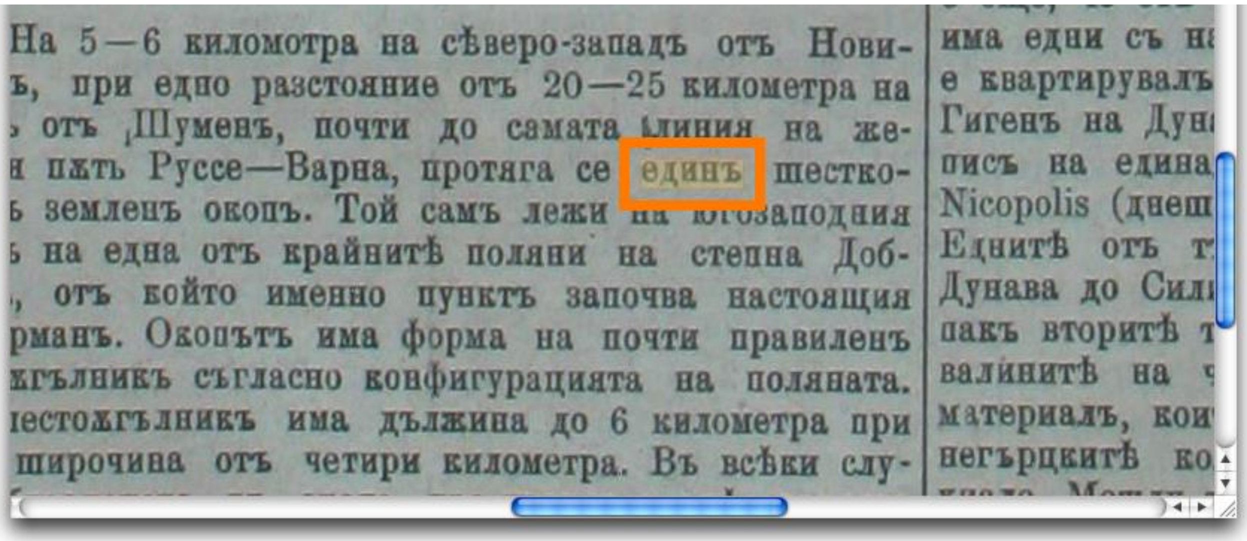


priap99_1119.extdic.xml.withoutIGT_XML_Tags.txt_tokenized.tab

да се прави отъ	единъ	чино- внивъ, понеже »	□
могатъ да прѣминаватъ въ	единъ	твърде ара- и тъкъ »	□
трѣбва да се намери	единъ	гаосінз-уіvѳшіі при »	□
Руссе—Варна, протага се	единъ	шѳстко- нечеаъ зеилѳнъ »	□
при повърхность отъ около	единъ	ввадратѳнъ хилонѳтръ. Тази »	□
коитѣ трите лежатъ на	единъ	редъ, а четвъртото стоя »	□

Partial screenshot of CoBaLT – display image

Clicking on a 'view image' icon in the document title bar will open a new window with the image. If an icon in a word form row is clicked on, the same image will load and the word form in question will be highlighted.



На 5—6 киломотра на сѣверо-западъ отъ Нови-
ъ, при едно разстояние отъ 20—25 километра на
, отъ Шуменъ, почти до самата линия на же-
я пѫть Руссе—Варна, протяга се единъ шестко-
ъ земленъ окопъ. Той самъ лежи на югозападния
ъ на една отъ крайнитѣ поляни на степна Доб-
, отъ който именно пунктъ започва настоящия
рманъ. Окопътъ има форма на почти правиленъ
ѫгълникъ съгласно конфигурацията на поляната.
шестоѫгълникъ има дължина до 6 километра при
широчина отъ четири километра. Въ всѣки слу-

има едни съ н
е квартирувалъ
Гигенъ на Дун
писъ на едина
Nicopolis (днеш
Еднитѣ отъ т
Дунава до Сил
пакъ вторитѣ т
валинитѣ на ч
материалъ, кои
негърцкитѣ ко

Partial screenshot of CoBaLT – word highlighted in image

Note that the border/highlight can be removed by pressing a key (any key will do). Releasing the key will restore the highlighting again. This can be handy because the highlighting border sometimes covers surrounding characters.

OVERVIEW OF KEYS AND MOUSE CLICKS

Below is a table that lists key and mouse behaviour in the tool. Please note that if you hold the mouse cursor over any clickable item in the tool a short message will be displayed about its function.

Lower part	Select multiple rows	**Click and drag**	Moving the mouse cursor over any rows in the lower part of the screen (word forms in their contexts) while holding down the mouse button will result in all these rows being selected. Please NOTE that if you do this very quickly, some rows might be left out of the selection, so go slow (enough).
	Add to/subtract from selection	**Ctrl-click**	Clicking on a row in the lower part of the screen while holding down the Ctrl key will cause this row to be added to/subtracted from any existing selection (while if you don't hold down the Ctrl key you will start making a new selection).
	Select all rows	**Double click**	Double clicking in the lower part of the screen will result in all rows being selected.
	Add analysis	**Click form**	Clicking on a word form in context gives the available analyses in a drop-down-menu, or the option to type a new one.
	Delete analysis	**Click analysis**	Clicking on an analysis in the lower part will result in this analysis being deleted for this row and for any other rows selected.
	Toggle validation	**F8**	Pressing the F8 button will cause any selected rows to be validated/unvalidated.
	Make word form group	**F2/F9**	Clicking on a word form in the context of another word form while holding down F2 or F9 will result in a word form group.
	Go to middle part	**Ctrl-m**	Pressing the 'm' button while holding down the Ctrl key will put the focus on the middle part (so, e.g. the up/down arrows will select the row above/below again).
Middle part	Previous/next word form	**↑ ↓**	If the focus is on the middle part, the arrow keys will select the previous/next row.
	Add analysis	**Click row**	Clicking in the left-hand part of a row in the middle part gives the available analyses in a drop-down-menu, or the option to type a new one.
	Add analysis to forms in lower part	**Click analysis**	Clicking on an analysis in the right-hand column of the middle part will result in this analysis being assigned to all selected rows in the lower part.
	Delete analysis	**Ctrl-click**	Clicking on an analysis in the right-hand column of the middle part while holding down the Ctrl key will cause this analysis to be deleted for the word form.
	Toggle validation	**Shift-click**	Clicking on an analysis in the right-hand column in the middle part while holding down the Shift key will (un)validate it.
Drop-down menu	Browse through menu	**↑ ↓**	When a drop-down menu is active, the arrow keys allow you to browse through it. Hit 'Enter' to select the highlighted option.

NOTE: pay attention to where you click!

Clicking on a row in the middle or lower part will select this row. However, as described above, clicking on an analysis in the middle part causes it to be applied, and clicking on an analysis in the lower part causes it to be deleted. So, if you just want to select a row, you will usually want to avoid doing so by clicking on any analysis it contains. You would better click anywhere else in the row.

Another thing to pay attention to is that some behaviour can make it appear to the tool as if the Ctrl key is held down when it isn't. This situation is particularly perilous if you click on an analysis in the middle part in order to assign it to some rows below, because it will be deleted instead!
There is an indicator next to the 'sort by left context' buttons in the middle of the screen that will say whether or not the tool thinks the Ctrl key is held down or not (to see it in action, start the tool and press/release the Ctrl-key).

TROUBLESHOOTING

Usually if something isn't working, it has to do with permissions. The tool should have read/write access to the right folders (e.g. the tokenizer location).
A good place to start looking when something appears not to work is the error log file of the web server. For Apache it is simply called 'error_log' and on Red Hat Linux it is located in the directory /var/log/httpd/. Also it might be a good idea to set the display_errors directive in the php.ini file to 'On' (it is 'Off' by default).

MOVING AN EXISTING INSTALLATION

During the lifetime of a CoBaLT project, it might happen that for some reason (e.g. hardware difficulties, etc.) you'll have to move your CoBaLT installation to another server.
If this is the case, here is how you'll need to move the tool:

[1] At first, you'll need to dump the content of your project databases . Each project consists of two databases, a lexicon database and a token database (as we explained in the 'Create databases' section). You can determine which token database belongs to a given lexicon database by checking the $asTokenDbName array in the CoBaLT configuration file php/globals.php. It must state something like:

```
$asTokenDbName['myLexiconDb'] = 'myTokenDb';
```

The lexicon and token databases of a project can be dumped that way:

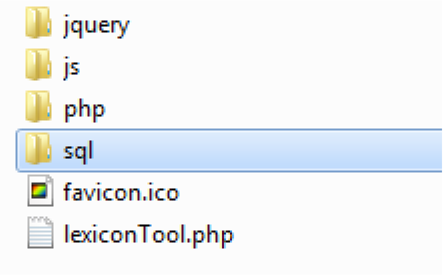
```
mysqldump -h<server> -u<user> -p<password> myLexiconDb > lexiconDumpFile.sql

mysqldump -h<server> -u<user> -p<password> myTokenDb > tokenDumpFile.sql
```

[2] Second, you'll need to perform a new installation of the software on your new server. Refer to the 'Installation' section for that.

[3] When the software is installed, you can load the dump files of step **[1]** into your new database installation. First create a lexicon and token database with the right names:

```
mysql> CREATE DATABASE myLexiconDb;
mysql> CREATE DATABASE myTokenDb;
```

Then import the right database structure into those new databases. The sql-files needed can be found in the sql directory of your CoBaLT installation:



```
mysql -h<server> -u<user> -p<password> myLexiconDb < emptyLexiconDatabase.sql
mysql -h<server> -u<user> -p<password> myTokenDb < emptyLexiconTokenDatabase.sql
```

When this is done, you're ready to load the databases you've been dumping at step **[1]**:

```
mysql -h<server> -u<user> -p<password> myLexiconDb < lexiconDumpFile.sql
mysql -h<server> -u<user> -p<password> myTokenDb < tokenDumpFile.sql
```

[4] Now you need to tell the tool where to find your databases and how there are called. Please refer to the 'CoBaLT configuration' section for that.

IMPORT AND EXPORT

TEI XML

The TEI format used by the scripts below is based on TEI P5[3]. Some IMPACT partners (the Jožef Stefan Institute in Slovenia in particular) make extensive use of this format .The software described below expects and puts out TEI in this format.

DATA IMPORT

IMPORT TEXT BASED LEXICON

If information is available about word forms and their lemmata these can be loaded into the MySQL database CoBaLT uses by a command line script.

```
$ ./loadLexicon.pl

 Imports word forms plus lemmata into a Lexicon Tool database.

 ./loadLexicon.pl [OPTIONS] -d DATABASE FILE [FILE2 [FILE3 ...]]

 -d DATABASE

    Database to be imported to.

 OPTIONS:

 -f SEPARATOR

    Separator used to separate columns in the input file (default: tab).

 -h DATABASE HOST

    Host the database is on (default: localhost).

 -l POS

   Default part of speech, in case the input file lacks them (for some
   entries).

 -m SEPARATOR

   Separator used to indicate multiple lemmata analyses in lemma headword,
   pos, and gloss.

 -p DATABASE_PASSWORD

    Password for the database user (default: INL setting).

 -u DATABASE_USER

    Database user (default: INL setting).
```

[3] http://www.tei-c.org/release/doc/tei-p5-doc/en/html/DI.html

```
 Input files are expected to have four columns:

   Column 1: word form
   Column 2: lemma_headword
   Column 3: pos
   Column 4: gloss (is just left empty if not provided).
```

IMPORT TEI XML

Usually, corpora are loaded just by using the tool interface as described above. A special script has been made for importing TEI XML (as used by e.g. the Jožef Stefan Institute in Slovenia). With this script both the documents and the lemma information in it will be loaded to the database in one go.

```
$ ./loadXmlInfo.pl

 ./loadXmlInfo.pl [DB_OPTIONS] -d DATABASE DIR1|FILE1 [DIR2|FILE2 [...]]

 DB_OPTIONS:

 -h HOST

    Database host.

 -u USER

    Database user.

 -p PASSWORD

    Database password.

 File names in absolute paths.

 Directories (also full paths) can also be passed as arguments, in which case

 all XML files in them will be processed (recognized by their extension).
```

DATA EXPORT

The data the tool uses is stored in a MySQL database. Several ways of exporting are supported.

EXPORT CORPUS AS VERTICAL TEXT

Every file that is loaded in the tool can be exported as vertical text. This means the every word form is printed on a separate line with additional information in tab separated columns.

```
./exportAsVerticalText.pl -c CORPUS -d DATABASE -t TOKEN_DB -o OUTPUT_DIR [OPTIONS] FILE
[FILE2 [FILE3 ...]]

  If the name of the original is abc.xml the output will end up in
  OUTPUT_DIR/abc.exported.xml.

  OPTIONS:

  -e
```

```
      Print explicitely as utf-8 (somehow this is not always needed).
   -n
      Print hader with column names
   -h HOSTNAME
      Defaults to localhost.
   -u USERNAME
      Defaults to the Lexicon Tool setting.
   -p PASSWORD
      Defaults to the Lexicon Tool setting.
```

The columns in the output are:

wordform	The word form as it appears in the left column in the middle screen of CoBaLT.
analysesInCorpus	All single analyses the word has in the corpus (the middle column in the middle part of the CoBaLT interface).
multipleLemmataAnalysesInCorpus	All multiple analyses the word has in the corpus (the middle column in the middle part of the CoBaLT interface).
analysesInDb	All single analyses the word has in the entire database (the right column in the middle part of the CoBaLT interface).
multipleLemmataAnalysesInDb	All multiple analyses the word has in the entire database (the right column in the middle part of the CoBaLT interface).
wordform_group_id	If a word form is part of a group this column has a non NULL value. This value is the same for all word forms belonging to the same group.
dontShowForDocument	If a word is hidden in the tool for this document this column has a non NULL value.
dontShowForCorpus	If a word is hidden in the tool for this corpus this column has a non NULL value.
dontShowAtAll	If a word is hidden for all any document or corpus in the tool this column has a non NULL value.

EXPORT CORPUS AS TEI XML

When a corpus has been imported as TEI (see above) it can also be exported as TEI. The exported files will contain the updated lemma information as edited in CoBaLT. The punctuation and further mark up will be restored form the original XML files.

```
$ ./exportTei.pl

  ./exportTei.pl -d DATABASE -o OUTPUT_DIR [DB_OPTIONS] FILE [FILE2 [FILE3 ...]]


  Export one or more files that were uploaded to CoBaLT as TEI XML.
```

```
The original files should have been in TEI XML format as well.

If the name of the original is abc.xml the output will end up in
OUTPUT_DIR/abc.exported.xml.

-f

  If this option is on, the last folder in the path of the output directory
  will be created if it doesn't exist already.

-t

   Create at a temporary table when running the scripts (increases speed
   considerably with large databases).

-a

   Export all documents in the database

DB_OPTIONS:

-h HOSTNAME

   Defaults to the JSI setting.

-u USERNAME

   Defaults to the JSI setting.

-p PASSWORD

   Defaults to the JSI setting.
```

EXPORT LEXICON AS TEI XML

Lexicon databases can be exported as TEI XML. When a corpus has been imported as TEI (see above) the lexicon can also be exported as TEI. The exported files will contain the lemma information as edited in CoBaLT including the citations. The punctuation and further mark up of the citations will be restored form the original XML files. If the originally uploaded files were not in TEI XML the quotations can be exported as well. In that case the are rebuild from the tokenized files that CoBaLT uses, or they vcan be restored from the database itself.

```
./exportLexiconAsTei.pl -d DATABASE [OPTIONS]

OPTIONS:

-a
   Only export validated analyses.

-b
   Only export validated attestations.

-c
   Only export lemmata that have attestations.

-e
   Encode output as utf-8 explicitely.

-E
   Decode the data from the database (use this if the lemma and part of
   speeched look garbled).
```

```
 -f
    Cache the content of all the files (don't open and close then all the
    time).

 -F
    Cache the parsed pieces of XML.

 -g
    Get the context of the attestations from the tokenized files (rather than
    from the original XML which is default).

 -G
     Use the german way of linking derivations

 -h HOSTNAME
    Defaults to the JSI setting.

 -l LIMIT
    Specify a limit (e.g. only print the first 100 entries).

 -m
    Don't use JSI mapping of part of speeches.

 -M
    Try to insert metadata from the documents table into the <bibl> tag

 -n
    Use underscores in stead of spaces in the \@n attribute.

 -o FILE
    File to write the output to.

 -p PASSWORD
    Defaults to the JSI setting.

 -q
    Use the quote field in the token_attestations table for context

 -s CHUNK_SIZE

    For large lexica (databases) it is considerably faster to get the lemmata
    in chucks rather than all in one go. This options allows you to set the
    chuck size.
    Default is 1000.

 -t
    Make temporary table. This can speed things up considerably when a large
   database is being exported.

 -u USERNAME
    Defaults to the JSI setting.


NOTE that options a/b can have rather subtle effects. Analyses can be
validated while none of their attestations are. All attestations of a
certain analysis can be validated even while the analysis itself is not.
Usually though this isn't the case and the data will be straightforward.

ALSO NOTE that in case of groups of word forms, the analysis of one member
can be validated even if the analysis of the other member isn't.
This leads to unexpected results when using the -a option. So you might opt
not to.
```

LICENSING

The tool is produced at the Instituut voor Nederlandse Lexicologie (INL) in Leiden, Netherlands.
It is licensed under the Apache License, Version 2.0 (http://www.apache.org/licenses/LICENSE-2.0).