TXM User Manual

Version 0.7 ALPHA
February 2018
(translated from Manuel de TXM Version 0.7 Juillet 2015 by Sara Pullin)
[cURRENT REFERENCE MANUAL IS MANUEL DE TXM VERSION 0.7.9 Février 2018]

Copyright © 2013-2018 – ENS, Lyon & Université de Franche-Comté –
http://textometrie.ens-lyon.fr
Copyright © 2008-2010 – ANR project Textométrie

In the spirit of open-source programs, this manual is provided free of charge in exchange for feedback on corrections and possible improvements that could benefit the TXM user community. **We need you in order to improve this manual.**

See the TXM's FAQ section to find out how you can contribute: https://groupes.renater.fr/wiki/txm-users/public/faq.

![Creative Commons BY-NC-SA](https://creativecommons.org/licenses/by-nc-sa/3.0/fr)

**This document is published under the Creative Commons BY-NC-SA license:**
http://creativecommons.org/licenses/by-nc-sa/3.0/fr

Updates table (older changes at the end of the document)

<table>
<thead>
<tr>
<th>Date</th>
<th>Author</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>16/05/2015</td>
<td>SH</td>
<td>Updated macro, import, Groovy and R sections</td>
</tr>
<tr>
<td>17/06/2015</td>
<td>SJ</td>
<td>Updates for TXM 0.7.7: graphics</td>
</tr>
<tr>
<td>27/07/2015</td>
<td>SH</td>
<td>Finalised for delivery 0.7.7, additions from MD&amp;</td>
</tr>
</tbody>
</table>

Edition number: 2
Contents: 197 pp., 39774 occ., 130 ill., 40 tab.
Date of Edition: 27/02/18, 15:27:07
Contents Page

1 Preface ................................................................................................................................. 10
1.1 Who Should Use This Document ..................................................................................... 10
1.2 How Is This Document Organized? .................................................................................. 10
1.3 Additional documentation ............................................................................................... 10
1.3.1 The TXM User's Wiki .................................................................................................. 10
1.3.2 TXM user's mailing list ............................................................................................... 11
1.3.3 The Textometry project website .................................................................................. 11
1.3.4 The TXM software developers' website ..................................................................... 11
1.3.5 TXM presentation brochures ...................................................................................... 12
1.3.6 TXM on social networks ............................................................................................. 12
1.3.7 TXM Training Workshops .......................................................................................... 13
1.4 Accessing TXM documentation online ............................................................................ 13
1.5 Typographic Conventions ................................................................................................. 13

2 Installing TXM on your machine ......................................................................................... 14
2.1 System requirements ......................................................................................................... 14
2.2 Installing on Windows ....................................................................................................... 15
2.2.1 Prior warnings before installation (Windows 7 and 8) ................................................ 15
2.2.2 Running the installation program ............................................................................... 15
2.2.3 Initial TXM program launch ....................................................................................... 17
2.3 Installation on Mac OS X .................................................................................................. 17
2.3.1 Stage 1: System requirements ..................................................................................... 17
2.3.2 Stage 2: Running the installation program .................................................................. 18
2.3.2.1 Homepage ............................................................................................................... 18
2.3.2.2 Installation drive ..................................................................................................... 18
2.3.2.3 Authorisation ......................................................................................................... 18
2.3.2.4 Installation ............................................................................................................. 18
2.3.2.5 Installing the R statistical libraries ....................................................................... 19
2.3.2.6 Finalising the installation ....................................................................................... 19
2.3.3 Initial TXM program launch ........................................................................................ 19
2.4 Linux Ubuntu .................................................................................................................... 19
2.4.1 Installation with Ubuntu software centre .................................................................... 20
2.4.1.1 Opening TXM_0.7.7_LinuxXX.deb ....................................................................... 20
2.4.1.2 Launching the installation ....................................................................................... 20
2.4.1.3 After installation ..................................................................................................... 20
2.4.2 Installation with Gdebi ................................................................................................ 20
2.4.2.1 Opening TXM_0.7.7_LinuxXX.deb ....................................................................... 21
2.4.2.2 After installation ..................................................................................................... 21
2.4.3 Installing through the command line ......................................................................... 21
2.4.3.1 Accepting the licence ............................................................................................ 21
2.4.3.2 Installing the R statistical libraries ....................................................................... 22
2.4.3.3 Installation Progress ............................................................................................... 23
10.2.1.4 Main Menus........................................................................................................49
  File Menu..................................................................................................................49
  Corpus Menu............................................................................................................51
  Tools Menu...............................................................................................................51
  View Menu...............................................................................................................52
  Help Menu..............................................................................................................52
10.2.1.5 Displaying the results.....................................................................................54
10.2.1.6 Displaying messages in the Console...............................................................55
10.2.1.7 Changing the interface using perspectives......................................................55
10.2.1.8 Resetting the user interface............................................................................56
10.2.2 The Windows Manager......................................................................................56
10.2.3 The Text Editor................................................................................................57
  The Text Editor's context menu................................................................................57
10.3 Working with Corpora..........................................................................................59
  10.3.1 Importing text sources into TXM....................................................................59
  10.3.2 Parameters form for import modules..............................................................60
  10.3.3 Import module example: Importing with CNR+CSV.................................63
  10.3.4 Other import modules.....................................................................................64
  10.3.5 Installing TreeTagger to automatically add morphosyntactic properties and
     lemmas to words....................................................................................................65
    10.3.5.1 Using the browser and your file explorer...................................................65
    10.3.5.2 In TXM.....................................................................................................67
  10.3.6 Exporting or uploading a binary corpus..........................................................68
  10.3.7 Exporting corpus source files in a standard XML-TEI P5 format..................69
  10.3.8 Corpus Examples..........................................................................................69
    10.3.8.1 The VOEUX corpus..................................................................................69
    10.3.8.2 The GRAAL corpus.................................................................................70
11 Using TXM commands.............................................................................................71
  11.1 Corpus description..............................................................................................71
    11.1.1 Of a corpus....................................................................................................71
    11.1.2 Of a partition.................................................................................................72
  11.2 Bibliographic records.........................................................................................73
  11.3 Edition.................................................................................................................73
    11.3.1 Corpus..........................................................................................................73
    11.3.2 Partition........................................................................................................74
  11.4 Constructing a sub-corpus....................................................................................75
    11.4.1 Constructing a sub-corpus: 'simple' mode......................................................75
    11.4.2 Constructing a sub-corpus: 'assisted' mode.................................................77
    11.4.3 Constructing a sub-corpus: 'advanced' mode.............................................78
  11.5 Building a partition............................................................................................79
    11.5.1 Building a partition: 'simple' mode..............................................................79
    11.5.2 Building a partition: 'assisted' mode...........................................................80
    11.5.3 Building a partition: 'advanced' mode.........................................................83
11.6 Building a concordance ................................................................. 83
  11.6.1 Queries ............................................................................. 85
  11.6.2 Browsing .......................................................................... 88
  11.6.3 Return to the text ............................................................... 89
  11.6.4 Sorting ............................................................................. 89
  11.6.5 Word properties ................................................................. 89
  11.6.6 References ....................................................................... 89
  11.6.7 Exporting .......................................................................... 90
11.7 Co-occurrences ............................................................................ 90
11.8 Lexicon and Index ........................................................................ 92
  11.8.1 Lexicon ........................................................................... 92
  11.8.2 Index ............................................................................... 93
    11.8.2.1 Choosing the set of word properties to be listed ............ 94
    11.8.2.2 Queries ...................................................................... 95
    11.8.2.3 Filtering the results ...................................................... 96
    11.8.2.4 Browsing through the results ..................................... 96
    11.8.2.5 Using the results to activate other commands ............ 97
11.9 Specificities ................................................................................ 97
  11.9.1 Specificity score .................................................................. 97
  11.9.2 Directly calculating the specificity score .............................. 99
  11.9.3 Presentation of the results ................................................... 101
  11.9.4 Specificities of a partition .................................................... 101
    11.9.4.1 Sorting the results ...................................................... 104
    11.9.4.2 Graphical visualization of the specificity scores ............ 104
  11.9.5 Specificities in a lexical table .............................................. 105
  11.9.6 Specificities in a sub-corpus .............................................. 106
11.10 Progression visualisation .............................................................. 107
11.11 References ............................................................................... 109
11.12 CFA ....................................................................................... 110
11.13 Clustering ................................................................................ 114
11.14 Lexical Table ............................................................................ 114
  11.14.1 Saving a lexical table ......................................................... 117
    11.14.1.1 Exporting a lexical table ........................................... 117
    11.14.1.2 Importing a lexical table .......................................... 117
  11.15 Graphical visualization of the results ..................................... 117
    11.15.1.1 Interactive manipulation ......................................... 118
    11.15.1.2 Displaying additional data ....................................... 118
    11.15.1.3 Exporting graphs .................................................... 118
11.16 Saving and Exporting results ...................................................... 118
11.17 Results processing with R ......................................................... 118
11.18 TXM Settings .......................................................................... 120
11.19 Summary table of the relations between TXM commands and results ............................................. 123
12 Queries for the CQP full text search engine ................................... 125
  12.1 Introduction ........................................................................... 125
12.1.1 CQL, CQP.................................................................125
12.1.2 Queries in TXM: simple queries, assisted queries, advanced queries........125
12.1.3 The process of constructing a query..............................................126
12.1.4 Examples for learning purposes....................................................126
12.2 Simple search [level 1 (infralexical): values].................................127
12.2.1 Searching for a word.................................................................127
12.2.2 Writing variations.....................................................................127
12.2.3 Truncating and wild-card..........................................................128
12.2.4 Punctuation..............................................................................128
12.2.5 Character classes..................................................................128
12.2.6 Expressing alternations.............................................................129
12.3 Conducting a properties search [level 2 (lexical): the properties]......129
12.3.1 Introduction...........................................................................129
12.3.2 Searching for a property............................................................130
12.3.3 Expressing alternations (2).........................................................130
12.3.4 Combining information..............................................................130
12.4 Searching for patterns of several words [level 3 (supralexical): lexical unit sequences].........................................................................................131
12.4.1 Word sequences..................................................................131
12.4.2 Expressing insertions.................................................................131
12.4.3 Distributional study.................................................................132
12.4.4 Expressing alternations (3).........................................................132
12.5 Contextual information................................................................133
12.5.1 Using structures..................................................................133
12.5.2 Using a property on a structure..................................................133
12.5.3 Relation between two words......................................................133
12.5.4 Alignment links between parallel corpora..................................133
12.5.5 Additional documentation........................................................134

13 Regular expressions.........................................................................135
   Memorised character references......................................................137

14 Using the results graphs in other programs.......................................139
14.1 Directly importing a vector graphic in SVG format to LibreOffice Writer..140
14.2 Directly importing a bitmap image in JPEG format to LibreOffice Writer.141
14.3 Using Inkscape to pre-edit a graph in SVG format........................142

15 Running TXM software using Groovy macros and scripts or with R scripts.145
15.1 Using macros...........................................................................145
   15.1.1 Launching a macro.................................................................145
   15.1.2 Installing a macro................................................................146
   15.1.3 Modifying a macro...............................................................146
   15.1.4 Creating a macro.................................................................147
       15.1.4.1 Predefined global variables........................................148
   15.1.5 Predefined TXM Macros........................................................148
       15.1.5.1 Assistance with preparing source files for a corpus........149

- 6 -
15.1.5.2 Assistance with configuring text editions..................................................149
15.1.5.3 Assistance with correcting the annotation of words in a corpus..............150
15.1.5.4 Running R scripts..................................................................................150
15.1.5.5 Assistance with calling up CQP commands..........................................150
15.1.5.6 Assistance with repetitively calling up TXM commands......................150
15.1.6 Share your macros with the TXM user community....................................151

15.2 Using Groovy scripts..................................................................................151
15.2.1 Running a script.....................................................................................152
15.2.2 Using third party libraries (.jar files).......................................................153
15.2.3 How to use TXM objects with Groovy.....................................................153

15.3 Using R scripts..........................................................................................153
15.3.1 Running a script.....................................................................................153
15.3.2 Using TXM results and objects from within R.........................................154
15.3.3 Using the R perspective to organise access to R......................................154
15.3.4 The R environment in TXM....................................................................155
15.3.5 Sample work session using R.................................................................156
15.3.5.1 Displaying the frequency histogram for a lemma index calculated using R
...............................................................................................................................156
15.3.5.2 Displaying the histogram directly in TXM using a Groovy script..........157
15.3.5.3 Run the R script and display the histogram using one single Groovy script
...............................................................................................................................157
15.3.5.4 Enter the name of the index from within a dialogue box with assistance from
a TXM macro........................................................................................................158
15.3.5.5 Recovering the name of the index directly from the macro..................160
15.3.6 Descriptions of the main TXM objects transferred to R.........................161

16 Import modules for textual sources................................................................162
16.1 Metadata file 'metadata.csv'.......................................................................162
16.1.1 'metadata.csv' sample file....................................................................163

16.2 Clipboard module......................................................................................163
16.2.1 Input.......................................................................................................163
16.2.2 Output....................................................................................................163
16.2.3 Annotation.............................................................................................163
16.2.4 Edition...................................................................................................164

16.3 TXT+CSV module.....................................................................................164
16.3.1 Input.......................................................................................................164
16.3.2 Output....................................................................................................164
16.3.3 Annotation.............................................................................................164
16.3.4 Edition...................................................................................................164

16.4 CWB module............................................................................................165
16.4.1 Input.......................................................................................................165
16.4.2 Output....................................................................................................165
16.4.3 Edition...................................................................................................165

16.5 XML/w+CSV module................................................................................165
16.5.1 Input.......................................................................................................165
16.5.2. Output.................................................................166
16.5.3. Edition..............................................................166
  16.5.3.1. Using XML components to build the edition...166
  16.5.3.2. Styling the editions with CSS......................167
16.6. XML-PPS..................................................................168
16.6.1. Input.................................................................168
16.7. Transcriber+CSV module.........................................168
  16.7.1. Input.............................................................168
  16.7.2. Output..........................................................169
  16.7.3. Annotation......................................................169
  16.7.4. Edition..........................................................169
16.8. XML-TEI BFM module.............................................170
  16.8.1. Input.............................................................170
  16.8.2. Annotation.....................................................171
  16.8.3. Edition..........................................................171
16.9. XML-TEI Frantext Module........................................171
16.10. XML-TMX............................................................171
  16.10.1. Input...........................................................171
  16.10.2. Output........................................................171
  16.10.3. Edition........................................................172
16.11. XML-TXM module................................................172
  16.11.1. Input...........................................................172
  16.11.2. Output........................................................172
  16.11.3. Annotation....................................................172
  16.11.4. Edition........................................................172
16.12. CNR+CSV module................................................172
  16.12.1. Input...........................................................172
  16.12.2. Output........................................................173
  16.12.3. Annotation....................................................173
16.13. Alceste module.....................................................173
  16.13.1. Input...........................................................173
  16.13.2. Output........................................................174
  16.13.3. Annotation....................................................174
  16.13.4. Edition........................................................174
  16.14.1. Input...........................................................174
  16.14.2. Annotation....................................................174
16.15. Factiva TXT module..............................................175
  16.15.1. Input...........................................................175
16.16. Factiva XML module.............................................175
  16.16.1. Input...........................................................175
17. Keyboard Shortcuts....................................................176
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.1</td>
<td>Results tables</td>
<td>176</td>
</tr>
<tr>
<td>17.2</td>
<td>Text Editor</td>
<td>176</td>
</tr>
<tr>
<td>17.3</td>
<td>Graphic Shortcuts</td>
<td>180</td>
</tr>
<tr>
<td>17.4</td>
<td>Windows</td>
<td>180</td>
</tr>
<tr>
<td>17.5</td>
<td>Miscellaneous Shortcuts</td>
<td>181</td>
</tr>
<tr>
<td>18</td>
<td>TXM Settings</td>
<td>181</td>
</tr>
<tr>
<td>18.1</td>
<td>TXM</td>
<td>181</td>
</tr>
<tr>
<td>18.2</td>
<td>Advanced Settings</td>
<td>181</td>
</tr>
<tr>
<td>18.2.1</td>
<td>Corpus Engine</td>
<td>182</td>
</tr>
<tr>
<td>18.2.2</td>
<td>Graphics Engine</td>
<td>182</td>
</tr>
<tr>
<td>18.2.3</td>
<td>Statistics engine</td>
<td>183</td>
</tr>
<tr>
<td>18.2.4</td>
<td>NLP/TreeTagger</td>
<td>183</td>
</tr>
<tr>
<td>18.3</td>
<td>User</td>
<td>183</td>
</tr>
<tr>
<td>18.3.1</td>
<td>Clustering</td>
<td>183</td>
</tr>
<tr>
<td>18.3.2</td>
<td>Concordances</td>
<td>184</td>
</tr>
<tr>
<td>18.3.3</td>
<td>Co-occurrences</td>
<td>184</td>
</tr>
<tr>
<td>18.3.4</td>
<td>Description</td>
<td>184</td>
</tr>
<tr>
<td>18.3.5</td>
<td>File browser</td>
<td>184</td>
</tr>
<tr>
<td>18.3.6</td>
<td>Exporting</td>
<td>185</td>
</tr>
<tr>
<td>18.3.7</td>
<td>Importation</td>
<td>185</td>
</tr>
<tr>
<td>18.3.8</td>
<td>Progression</td>
<td>185</td>
</tr>
<tr>
<td>18.3.9</td>
<td>References</td>
<td>186</td>
</tr>
<tr>
<td>18.3.10</td>
<td>Scripts</td>
<td>186</td>
</tr>
<tr>
<td>18.3.11</td>
<td>Specificities</td>
<td>186</td>
</tr>
<tr>
<td>18.3.12</td>
<td>Lexical Table</td>
<td>186</td>
</tr>
<tr>
<td>18.3.13</td>
<td>Correspondence Analysis</td>
<td>186</td>
</tr>
<tr>
<td>18.3.13.1</td>
<td>Defining the display formatting for real or whole numbers</td>
<td>187</td>
</tr>
<tr>
<td>19</td>
<td>Morphosyntactic tagset used for French</td>
<td>188</td>
</tr>
<tr>
<td>20</td>
<td>TXM Glossary</td>
<td>190</td>
</tr>
<tr>
<td>21</td>
<td>Bibliography (not complete)</td>
<td>196</td>
</tr>
<tr>
<td>22</td>
<td>Index</td>
<td>197</td>
</tr>
</tbody>
</table>
1 Preface

1.1 Who Should Use This Document

For those interested in using the TXM software, this document explains step-by-step the various concepts and tools featured within that are useful for analysing diverse textual corpora.

If you wish to modify TXM software for specific corpora, this document also enables you to customise the scripting environment by adapting the importation system.

1.2 How Is This Document Organized?

To begin, this document describes how to install and launch the software on various operating systems. In following, it describes how the user interface is organized and how to import new corpora into the program. The next section goes onto describe the various tools and how to use them to analyse a corpus. Finally, instructions on piloting the software from within the scripting environment are given. The document concludes with a reference appendix containing a glossary of terms and an index.

1.3 Additional documentation

For further information in addition to this manual, we advise you to first of all consult the TXM user's wiki, which is the best source for up-to-date information and can be found at the following address: https://listes.cru.fr/wiki/txm-users/index

1.3.1 The TXM User's Wiki

The wiki is structured as follows:

- F.A.Q: provides answers to the most frequently asked questions in a more practical - and up-to-date - form than this reference manual;
- Bug reports from previous TXM versions: this page includes bug reports noted in various meetings we have held and emails received;
- Bug reports from the online TXM portal version
- Feature requests: inventory of various requests made for new TXM features
- The wiki also allows you to participate in improving the available documentation (including this manual) or translating it into German, Spanish, Russian, etc.
1.3.2 TXM user's mailing list

You are equally invited to sign up to the TXM user's mailing list (in French) at the following address: [https://listes.cru.fr/sympa/subscribe/txm-users](https://listes.cru.fr/sympa/subscribe/txm-users).

This inscription is necessary if you want to contribute to the previously mentioned wiki. It also allows you to directly get in contact with other TXM users (experienced or not) and software designers. You also receive regular information updates concerning new versions of the software.

We invite you to consult the mailing list's archives to ensure that a subject linked to your particular question has not already been discussed in the past, as several answers may already be available. A useful way of conducting this search is to use the full text search engine, which searches both the body of articles and their subject lines. The archive can be found at the following address: [https://listes.cru.fr/sympa/archives/txm-users](https://listes.cru.fr/sympa/archives/txm-users).

1.3.3 The Textometry project website

All the official documents linked to the TXM software can be found on the Textometry website: [http://textometrie.ens-lyon.fr/spip.php?article98&lang=fr](http://textometrie.ens-lyon.fr/spip.php?article98&lang=fr) (a video tutorial, manuals for users and developers, key documents on the textometry methodology, documentation related to text encoding, search engines, statistics engines and script interpreters), in addition to a list of scientific publications linked to the software's development and use: [http://textometrie.ens-lyon.fr/spip.php?article82&lang=fr](http://textometrie.ens-lyon.fr/spip.php?article82&lang=fr).

1.3.4 The TXM software developers' website

Finally, those people interested in the development of TXM's open-source software can consult the developers' wiki (in French) at the following address: [https://groupes.renater.fr/wiki/txm-info](https://groupes.renater.fr/wiki/txm-info) (the English wiki [http://txm.sourceforge.net/wiki](http://txm.sourceforge.net/wiki) is currently unavailable).

In order to contribute to editing this wiki, you must:

1) either have an account with an institution registered with Renater or create a CRU (Comité Reseaux des Universités/Universities Networking Committee) account with Renater: [https://cru.renater.fr/sac/faces/casRedirect.jsp?target=create](https://cru.renater.fr/sac/faces/casRedirect.jsp?target=create); and

2) join the developers' mailing list ‘txm-info’ using the email address linked to your account: [https://groupes.renater.fr/sympa/subscribe/txm-info](https://groupes.renater.fr/sympa/subscribe/txm-info)

You are also invited to join the TXM developers' (English) mailing list at the following address: [https://lists.sourceforge.net/lists/listinfo/txm-open](https://lists.sourceforge.net/lists/listinfo/txm-open)

---

1 This wiki is written entirely in English, as is the TXM source code and its comments.
TXM is provided to you free of charge. In return, and in the spirit of open-source software, you are invited to take part in its adaptation and improvement. You are not required to be a computer developer to help out. For example, you can:

- send us your publications or other research documents demonstrating your use of TXM
- add a link on your website towards the Textometry project home page: http://textometrie.ens-lyon.fr
- inform us of any malfunctions you notice in TXM, preferably directly via the TXM user's wiki: https://groupes.renater.fr/wiki/txm-users/public/retours_de_bugs_logiciel
  Though you can of course always use the TXM user's mailing list to send any feedback or reports: https://groupes.renater.fr/sympa//info/txm-users
- help us to translate the software's user interface or documentation into other languages (German, Spanish, Russian, etc.)
- share your course materials, as well as your corpora examples, with the user community
- invite your developer colleagues to adapt TXM for your own needs and then share these developments with us. Such developments can occur either at the program's core (Java and C languages for professional developers), or at its perimeter/margins/borders/outskirts (Groovy script language much more accessible)
- develop a project using TXM (e.g. funded through the ANR (French), NEH (US) or DFG (Germany)), for which we can provide you with advice on its implementation or how to adapt the software
- propose ways in which to improve the documentation or its distribution

### 1.3.5 TXM presentation brochures

- the PLUME file: [https://www.projet-plume.org/relier/txm](https://www.projet-plume.org/relier/txm)

### 1.3.6 TXM on social networks

- Twitter: [https://twitter.com/txm](https://twitter.com/txm)
1.3.7 TXM Training Workshops

TXM Workshops consist of training days that are free to attend and accessible to all: https://groupes.renater.fr/wiki/txm-users/public/ateliers_txm

1.4 Accessing TXM documentation online

This manual is available online:

- as a TXM corpus: https://sourceforge.net/projects/txm/files/corpora/refman

This English translation, and the original French version, are also available at the following address: http://sourceforge.net/projects/txm/files/documentation

1.5 Typographic Conventions

Within this manual, certain items are highlighted using various typographies:

- literal expressions use Courier font, for example directory paths, file names, sample queries and character strings, as well as hyper-links.
- Arial font is used for section headings
- Arial font is used for the application's commands
2 Installing TXM on your machine

2.1 System requirements

In order to install TXM you need to have administrator privileges to install programs on your machine.

You will also need Internet access.

This version of the software is compatible with the following operating systems:

- Windows XP, Vista, 7 and 8: consult the installation instructions in section 2.2 below;
- Mac OS X 10.6 (Snow Leopard) to 10.9 (Mavericks): consult the installation instructions in section 2.3 on page 17;
- Linux Ubuntu 12.04 and higher, plus variants (Xubuntu, Kubuntu, Lubuntu) and Debian: consult the installation instructions in section 2.4 on page 19.

Required disk space for installing TXM:

- Windows: 250 Mo
- Mac OS X: 335 Mo
- Linux: 200 Mo
- Allow 120 Mo for the DISCOURS and GRAAL corpora examples regardless of the system

Required resources for using TXM:

- disk: allow space for each additional corpus, for example:
  - DISCOURS: 50 Mo (100,000 words, 4 word properties)
  - Base de français médiéval (Medieval French base) (BFM): 1.4Go for 5M words, 14 word properties and 60 structural properties
- memory: 1 Go (see section 8 on page 33 on how to consult the current memory use)
2.2 Installing on Windows

2.2.1 Prior warnings before installation (Windows 7 and 8)

The TXM installation program is not certified by a recognised Microsoft body, therefore it triggers a pop-up security warning that appears in the middle of the screen. To continue the installation, you just need to select 'OK' then 'Run quand même'.

If you have doubts concerning the TXM installation program you have downloaded, the only website officially distributing TXM is the project's sourceforge website:
https://sourceforge.net/projects/txm/files/software/0.7.7

2.2.2 Running the installation program

- Download the set-up file 'TXM_0.7.7_WinXX.exe' from the address below, whereby XX signifies your machine's processor (32 = 32-bit ou 64 = 64-bit):
  https://sourceforge.net/projects/txm/files/software/0.7.7
- Run the set-up file by double-clicking its icon:

  Depending on the security level installed on your Windows version, it is possible that a pop-up window (see Image 2.1) may appear. If this occurs, click 'Run'.

  Image 2.1: Security window

---

2 To find out whether your Windows system runs on a 32 or 64 bit processor: see the Microsoft documentation.
In the pop-up window displayed in Image 2.2 click 'Install' (before installing it is possible to nominate a different installation folder if necessary).

- The installation should take around a minute.

- During the installation, if the following message appears (see Image 2.3: Error opening file 'R.dll') Error opening file 'R.dll', this means the Rserve process (the TXM statistics engine) is still running on the machine and that the installation process cannot update its binary files. You must therefore first quit TXM or terminate the Rserve.exe process using the Windows Task Manager, and then click 'Restart' to resume the installation process.
If the pop-up window in Image 2.4 appears, the operation has been successful and the installation is complete. You can now close the installation window by clicking 'Close'.

If the installation has not been successful, please follow the problem reporting instructions in section 9 on page 33.

### 2.2.3 Initial TXM program launch

You need to conduct an initial launch of TXM in order to finalise the installation process, see section 10.1 on page 37.

Note: automatic updates and extensions do not become available until the second launch of TXM, see section 3 on page 25.

### 2.3 Installation on Mac OS X

#### 2.3.1 Stage 1: System requirements

TXM functions on Mac OS X with version 10.6 (Snow Leopard)\(^3\).

Specificities for other versions:

- using Mac OS X 10.7 (Lion): you will need to modify the 'Security' parameters in order to correctly install TXM (if you don't complete this modification the related issue will not appear during installation, but will appear during the initial launch). To do this, you need to go into 'System Preferences > Personal > Security' and authorise application downloads from the internet;

---

\(^3\) To find out the version of your Mac's operating system, you will need to consult the System Information using the Apple menu > 'About this Mac'.
for a Mac OS X 10.9 (Maverick): you need to install a preliminary application called XQuartz. From the page https://xquartz.macosforge.org/landing, download and install the application 'XQuartz-x.x.x.dmg'.

2.3.2 Stage 2: Running the installation program

Installation procedure for all Mac OS X:

- Download the setup file 'TXM_0.7.7_MacOSX.pkg' from the following address: https://sourceforge.net/projects/txm/files/software/0.7.7
- While connected to the internet, double-click the file icon 'TXM_0.7.7_MacOSX.pkg'. This will launch the installation program for Mac software which will run across several screens:

2.3.2.1 Homepage

When on the homepage (Image 2.5), click 'Continue'.

2.3.2.2 Installation drive

If necessary (Image 2.6), you can choose the disk on which you wish to install TXM (the disk 'Macintosh HD' will be used by default), click 'Continue'.

2.3.2.3 Authorisation

To install TXM (Image 2.8), you need to validate your administrator privileges for your Mac OS X using your Apple login and password.

2.3.2.4 Installation

TXM will copy the files into the installation folder and prepare the work environment (Image Erreur : source de la référence non trouvée).
2.3.2.5 Installing the R statistical libraries

TXM automatically installs the R statistical libraries by default, which it needs to calculate Specificities, CFA, Co-occurrences and Progression.

For advanced users: it is possible to install the R statistical libraries yourself by modifying the installation options found in section 2.3.2.2.

http://txm.sourceforge.net/wiki/index.php/Build_the_toolbox_or_the_application#Install_R_yourself

2.3.2.6 Finalising the installation

If the message displayed in Image 2.9 appears, the operation has been successful and the installation is complete. Click 'Close'.

If the installation has not succeeded, please follow the problem reporting instructions in section 9 on page 33.

You can access the complete installation logbook using the keyboard short-cut 'Apple + L' before quitting the installation program. This short-cut will display the installation logbook which you can then copy+paste.

2.3.3 Initial TXM program launch

You need to conduct an initial launch of TXM in order to finalise the installation process, see section 10.1 on page 37.

Note: automatic updates and extensions do not become available until the second launch of TXM, see section 3 on page 25.

2.4 Linux Ubuntu

To install TXM on Linux Ubuntu (version 12.04 and higher):

- Download the setup file 'TXM_0.7.7_LinuxXX.deb' at the following address, whereby XX signifies your machine's processor ((32-bit or 64-bit): https://sourceforge.net/projects/txm/files/software/0.7.7

5 To find out if your Linux system's processor is either 32- or 64-bit: In Ubuntu (tested using Ubuntu 12.04 LTS), in the menu in the upper right-hand corner (small indented wheel), System settings… > System: Details > Type of OS (= 32- or 64-bit).
Next, install using either the software centre (Ubuntu), Gdebi or through the command line.

2.4.1 Installation with Ubuntu software centre

2.4.1.1 Opening TXM_0.7.7_LinuxXX.deb

To launch the software centre using the Linux installation program, double-click on .deb. Alternatively, you can right-click on the installation icon, then select 'Open with another application...', and in the list of available applications select 'Ubuntu software centre'.

2.4.1.2 Launching the installation

If this is your first TXM installation, you will need to click on the button 'Install', otherwise click on 'Update' (or 'Reinstall' if you are reinstalling the same TXM version).

2.4.1.3 After installation

The steps to follow after installation can be found in sections 2.4.3.1 and 2.4.3.2.

2.4.2 Installation with Gdebi

There is a second method for installing TXM that may be useful in the event that there is a problem with the software centre.

Gdebi is installed by default in the Debian releases. If it is not installed, you can install it using the software centre or via a Terminal using the following command line:
sudo apt-get install gdebi

2.4.2.1 Opening TXM_0.7.7_LinuxXX.deb

Once the Home screen opens, click 'Install the package'.

If the field line 'Status' is red, this means Gdebi has not been able to find the packages needed to install TXM. In this case, the simplest solution is to contact us to verify that your Ubuntu version has been properly tested. In Image 2.10, we can see that all the dependencies are available.

2.4.2.2 After installation

The steps to follow after installation can be found in sections 2.4.3.1 and 2.4.3.2.

2.4.3 Installing through the command line

There is a third method for installing TXM which may prove useful if any problems arise with the previous two methods. In a Terminal, launch the following command line:

```
sudo dpkg -i TXM_0.7.7_LinuxXX.deb
```

2.4.3.1 Accepting the licence

- The first installation step consists of accepting the terms of the software's licensing agreement:
• Click 'I agree' to continue.

• Attention: Accepting these terms of agreement also means you accept the software's GNU General Public Licence.

  • The full version of the GNU General Public Licence can be consulted at the following address: http://www.april.org/gnu/gpl_french.html;
  
  • In addition, an introduction to the notion of Free software can be found here: http://www.april.org/sites/default/files/documents/html/logiciel-libre.html

### 2.4.3.2 Installing the R statistical libraries

• After validating the licence, TXM will automatically propose to install the R statistical libraries:
For a standard TXM installation, click 'Yes'. For more advanced users, it is possible to install the R statistical libraries yourself. For more information regarding this procedure, please refer to the following page: 
http://sourceforge.net/apps/mediawiki/txm/index.php?title=Build_the_toolbox_or_the_application#Install_R_yourself

2.4.3.3 Installation Progress

During the installation, the following dependencies will be installed:

- r-base
- r-recommended
- zenity
- default-jre
- libwebkitgtk-1.0-0
- debconf
- libc6 (>= 2.15)
To obtain details concerning the download progress, click 'Terminal'.

2.4.3.4 Finalising the package installation

In Gdebi, if the window in Image 2.14 appears then the operation has been successful and the installation is complete. You may therefore close the Gdebi windows.

![Image 2.14: Finalising the installation](image)

If the installation has not been successful, please follow the problem reporting instructions found in section 9 on page 33.

2.4.4 Initial TXM program launch

You need to conduct an initial launch of TXM in order to finalise the installation process, see section 10.1 on page 37.

Note: automatic updates and extensions do not become available until the second launch of TXM, see section 3 on page 25.

2.4.5 Reconnecting following the initial installation

The first time you install TXM on your machine, you will need to exit your work session (disconnect) and then reconnect a second time.

2.5 Checking the installation of R packages

TXM can verify the correct installation of the R packages, which it needs for its statistical calculations. To do so, use the command 'File > Check R packages' in the main menu.

The verification process happens as follows:

- TXM tests the presence of each package;
if a package is present, TXM verifies the version of the package and updates it if necessary;

- if the package is absent, TXM installs it.

Attention, TXM must be connected to the Internet for all package installations or updates.

### 3 Automatic TXM updates

From version 0.7.5 onwards, TXM automatically updates itself which means users no longer have to download a new TXM installation program for each version. When an update is available, a small window will appear in the bottom right-hand corner of the main TXM window and will propose to download and install the update (see the following section).

#### 3.1 Updated versions

Updates are organised according to four different levels that range from the simple TXM user to those more heavily involved in the software's development:

- the 'STABLE' level (default level) enables all users to benefit from ongoing improvements made to the software, as they become available;

- the 'BETA' level allows interested parties to test improvements suggested for the 'STABLE' level that have not yet been tested in certain environments (for example with certain versions and operating systems) or with all possible scenarios (compatibility with specific types of corpora, with certain external tools such as taggers, etc.). Volunteers testing the improvements on the 'BETA' level are invited to report their feedback to the development team, either by email or through the txm-users wiki: [https://groupes.renater.fr/wiki/txm-users/public/retours_de_bugs_logiciel](https://groupes.renater.fr/wiki/txm-users/public/retours_de_bugs_logiciel);
the 'ALPHA' level allows the TXM development team to test prototype improvements suggested for the 'BETA' level. These improvements are similar to preliminary drafts. Generally, it could be the case that the development team has not yet reached a full consensus regarding their implementation, or that their user interface is not yet developed, or that the drafting of their documentation has barely started, or perhaps that their use may provoke TXM to crash and leave the software in a working, yet unstable state;

– The 'DEV' level allows TXM developers to rapidly debug any changes made to TXM. We do not advise you to use this level, as the corresponding updates are not always compatible with the TXM public version and can provoke crashes, again leaving TXM in a working, yet unstable state.

To nominate the TXM update level, the user can change the settings via 'TXM / Advanced / Update level' to the desired value (STABLE, BETA, ALPHA, DEV).

3.2 Launching an update

Updates automatically occur by default when TXM is launched. If you wish to disable this, go to the program settings on the 'TXM' page and uncheck the option 'Automatically search for and inform me about updates'. You can later initiate manual updates by using the command 'Help / Check for updates'.

If automatic updates are selected and TXM is connected to the internet, it will automatically verify if there are any available updates when opened. If an update is available, TXM will display the update window (Image 3.3).

3.3 Updating the System

When an update is launched it displays a list of newly available components.
3.3.1 Step 1

Select the components to be updated then click 'Next' to advance.

![Image 3.3: Step 1: Available updates](image3.3)

3.3.2 Step 2

The next step will display a more detailed description of the updates, which in most cases can be ignored. Click 'Next' to advance to the following step.

![Image 3.4: Step 2: Update details](image3.4)

3.3.3 Step 3

This step involves accepting the distribution licenses for each component. These licenses are generally identical to the one used for TXM's distribution – the GNU General Public License
V3. Select the option 'I accept the terms of the license agreements', then click 'Finish' to download the updates.

![Image 3.5: Step 3: Agreeing to the distribution licenses](image)

### 3.3.4 Steps 4 to 6

A new window will open displaying the download progress.

![Image 3.6: Step 4: Downloading updates](image)

Once the download has finished, TXM will demand final confirmation for any unsigned updates without authenticity certificates guaranteeing TXM as their source of origin (this is the case for all updates at present).
Finally, TXM will request a restart to ensure that the updates take effect. Click 'Yes'.

4 Installing an extension in TXM

From version 0.7.5 onwards, it is now possible to install extensions in order to add new features to TXM. These extensions are proposed as optional features by both the TXM development project and our partners.

To install an extension in TXM, use the command 'Help / Add an extension'. This will open a window displaying the available extensions and their descriptions. Select the desired extensions then click 'Next'.

Image 3.7: Step 5: Final security confirmation before installation

Image 3.8: Step 6: Restart TXM to apply updates
The subsequent steps are similar to steps 2 to 6 for TXM updates, see section 3.3.

4.1 Documentation for extensions

Each extension will add an entry to the menu 'Help > Extensions' to provide access to the documentation for that extension.

5 Installing a third-party extension in TXM

From version 0.7.5 onwards, it is possible to install third-party extensions (or 'plug-ins') in TXM, which have been developed through compatible independent projects using TXM's Eclipse RCP architecture.

To find out what extensions are available, public extension portals can be used such as the Eclipse marketplace from the Eclipse consortium: http://marketplace.eclipse.org.

If you wish to install a third-party extension, it is necessary that TXM recognises the address of the update site. This address is provided either by the extension portals or through the development sites of these extensions.

---

6 Conform with the OSGi standards: http://en.wikipedia.org/wiki/OSGi
5.1.1 Step 1

To launch the installation, follow the command chain: 'File > Add a third-party extension'. This command opens a dialogue box in which (see Image 5.1):

- the field 'Work with:' allows you to copy or paste the address for the extension's update site or to choose one of the addresses from the list of recognised update sites (using the arrow to open the dropbox);
- you can filter the available extension names in the chosen update site by entering part of the name into the following field (for example 'XML');
- the extensions list will then display underneath;
- select the desired extensions and click 'Next' to proceed to the next step.

5.1.2 Following steps

The subsequent steps are similar to steps 2 to 6 for TXM updates, see section 3.3.
If at step 2 you receive the message 'The operation cannot be completed. See details', you will need to indicate additional update sites for TXM to be able to download other extensions that the principal extension depends upon. These extensions are displayed in the field 'Details'. You will therefore need to:

- abandon installing the extension;
- add the necessary update sites;
- restart the extension installation from the beginning.

### 6 Uninstalling an update, extension or third-party extension

You can uninstall extensions and third-party extensions using the same method:

- Open the window 'About' through the main menu 'Help > About'
- Select the button 'Installation Details'
- A new window will open and display the installation history related to the status of updates and TXM extensions
- Select the TXM version:
  - Clicking 'Delete' will delete files from a previous installation (that are not in use at the time);
  - Clicking 'Revert' will re-download and apply an update or extension (therefore deleting the most recent updates or extensions).

You will then need to restart TXM so that the uninstall takes effect.

### 7 By proxy network access settings

If you access your network through a proxy, you can configure TXM in the following way:

- go to the network preferences page 'General > Network Connections';
- change the field 'Active Provider' to 'Manual';

It may be worthwhile to add the update site 'http://download.eclipse.org/releases/indigo' in advance, as it contains all the base extensions for the Eclipse platform used by TXM. It is likely that these base extensions will be requested by the third-party extensions.
– double-click on the field 'HTTP' and modify the 'Host' and 'Port' values so they correspond to your proxy;
– double-click on the field 'HTTPS' and modify the 'Host' and 'Port' values so they correspond to your proxy;
– confirm by clicking 'OK.'

Note: when accessing by proxy, during the initial TXM start-up you may have to wait up to 10 minutes for the network access attempt to be unblocked by TXM before you are able to modify the proxy settings in Network Connections.

8 Consulting the memory space used by TXM

It is possible to consult the Java memory space used by TXM. To do so:

– go to the settings page via 'General';
– activate the field 'Show heap status';
– click 'OK' to confirm.

Note: this concerns only the Java memory use, it does not take into account the memory consumed by the search engines and statistics engines used by TXM. Though it nevertheless gives a proportional indication of the total memory being used by TXM at any given moment.

9 In the event of a problem

If you encounter a problem with TXM:

A) Verify that the problem continues to persist after restarting

B) Verify that the problem has not already been mentioned in the archives of the TXM user's mailing list: https://groupes.renater.fr/sympa/arc/txm-users
   (do not hesitate to use the search engine)

C) Check that the problem has not already been reported on the TXM user's wiki on the following pages:
   a) the F.A.Q. page: https://groupes.renater.fr/wiki/txm-users/public/faq
   b) the problem reports for your TXM version: https://groupes.renater.fr/wiki/txm-users/public/retours_de_bugs_logiciel/txm_0.7 (problems marked with an OK have already been resolved and their solutions will be available in the next version of TXM)
If the problem has already been reported but remains unresolved, you may add details to its description in the wiki or ask via email where we currently are in resolving it.

If the problem is new, we would welcome your feedback either via the TXM wiki or the user's mailing list.

To help us best diagnose the problem, it is important that you provide us with the greatest number of details possible regarding your configuration (TXM behaves differently depending on the specific operating system) and the TXM activity logs at the moment the problem occurred.

To indicate your configuration:

– in TXM, go to the menu 'Help / About TXM';
– click on 'Installation Details';
– open the tab 'Configuration';
– click on 'Copy to Clipboard' and paste the information into a text file or the body of an email.

To provide the TXM start-up log:

– in TXM, go to the menu 'Help / About TXM';
– click on 'Installation Details';
– open the tab 'Configuration';
– click on 'View Error Log' and save the information in a file or copy and paste it into the body of an email.

To provide the detailed activity log:

– Configure the messages being displayed in the console to the maximum number: in Settings, click Advanced, then adjust the Log Level to 'ALL';
– Save the message as a file: go to Settings, then Advanced, check the box 'Copy the Log to a file'. A file will be created in the TXM work folder;
– Next prompt the problem to occur and provide us with a copy of the resulting log file.
In the event of an installation problem, please provide us with a copy of the 'TXMPostInstallLogs.txt' and 'TXMPostInstallErrorLogs.txt' files, which can be found:

- in Windows: in the folder (cache) %APPDATA%:
  
  The path for this folder varies depending on your Windows version:

  - Windows XP:
    C:\Documents and Settings\<user>\Application Data
  - Windows Seven: C:\Users\<user>\Application Data\Roaming).
  
  The simplest way to get to this folder is to use the script BAT from the TXM installation folder named OpenStartUpLogsDirectory.bat which will directly open the browser containing this folder.

- in Mac OS X and Linux: in the folder 'HOME/TXM/.txm'

---

$HOME represents the user's personal folder path.
10 Getting to Know TXM

The TXM software helps you to build and analyse tagged and structured corpora:

- The software enables you to import textual resources in order to build a corpus either from various sources, or more directly from any text copied to the clipboard.
- It can build sub-corpora using the different properties of textual units.
- It can also build partitions using these properties.
- It can build a HTML edition for each textual unit of the corpus.
- It can calculate the entire vocabulary of a corpus or the list of values of a particular property.
- It can build lexical tables from either partitions or by using an index.
- It can search for complex lexical patterns constructed from properties of lexical units and it can produce Kwic concordances from the results. Each concordance line gives you access to the corresponding page in the HTML edition.
- It can calculate collocations located around a complex lexical pattern.
- It calculates the references of complex lexical patterns.
- It can calculate the word specificities model located within a partition or a sub-corpus.
- It calculates the correspondence factor analysis (CFA) of word properties in a partition.
- It calculates the clustering of a partition.

The program is composed of four modules that work together:

- the CQP full text search engine;
- the R statistics engine;
- the text import module for building corpora, and
- the script interpreter.

This manual will introduce each of these modules by way of the various commands available.
10.1 Starting TXM

10.1.1 In Windows

1. Menu 'Start / TXM / TXM' (place the TXM application icon on the Quick Launch bar to enable direct access)

2. Depending on the security level of your Windows operating system, when launching for the first time you may have to answer certain security alerts, which can be done in the following way:

   Attention though, you must have administrator permission in order to complete this step.

   a. If the window above appears:
      Click 'Unblock'⁹

---

⁹ The program 'cqpserver' is the full text search engine used by TXM.
b. Or if the following window appears:

![Window](image)

Click 'Unblock'\(^\text{10}\)

### 10.1.2 In Mac OS X

Using the Finder, navigate to the folder 'Applications / TXM' and double-click the TXM application icon (place the TXM application icon in the Dock to enable direct access).

### 10.1.3 In Linux

Using the Launchpad Unity, navigate to the section 'Installed Applications' and double-click the TXM application icon (to enable direct access, right-click on the TXM icon in the Dock and select 'Keep in the Dock'). Or via a Terminal by launching the command: 'TXM&'.

---

\(^\text{10}\)The program 'Rserve' is the statistic engine used by TXM.
10.2 Using the windows, menus, toolbars and keyboard shortcuts

10.2.1 General overview of the user interface

The TXM user interface is divided into four zones, as indicated in Image 10.1:

- The Browser: enables access to corpora and sub-corpora, to the command results and to script files and folders. In short, to all the objects managed by TXM to which commands are applied;
– Commands panel: displays buttons and menus that allow actions to be launched on objects selected in the browser;
– Results zone: contains output windows;
– Messages panel: contains comments related to each executed action.

All the zones are managed in the same window.¹¹

To begin, all the main zones will be introduced and, in following, an explanation will be given regarding how this interface is organised in the main window.

10.2.1.1 The Browser

The browser allows users to select the object to which they wish to apply their commands.

The browser has two default views:

– The 'Corpus' view: displays the icons of the corpora and sub-corpora available for analysis, in addition to the different results already calculated;
– The 'File' view: displays the files located in your folders, offering the possibility to edit them.

In this same zone, it is also possible to display other views such as:

– The 'R Variables' view: displays the TXM objects which have been sent to R
– The 'Query' view: displays all the CQL queries applied to each corpus.

¹¹ It is possible to open all the zones in a separate window.
The Corpus view

The 'Corpus' view presents the different corpora available for analysis in TXM, in addition to all the object icons generated by TXM during a session. This is the main TXM view. The corpora are created using either the Import or Load commands in the 'File' menu.

The 'Corpus' view is organised hierarchically. Each root icon represents an independent corpus. All the elements descending from the roots result from the application of TXM commands:

- Sub-corpus (has a 'C' icon, identical to that of the 'root' corpus) in 'Create a sub-corpus';
- Partitions ('P' icon) in 'Create a partition';
- Lexicon;
- Index;
- References;
- Concordance;
- Co-occurrences;
- Specificities;
- CFA;
- Clustering;
- Lexical Table
- Send to R

Each new command generates a branch on the tree diagram of results.

Each object type can have a specific collection of commands applied to it:

- all commands may be applied to a 'Corpus';
- the same commands may also be applied to a 'Sub-Corpus', and also to the 'Specificities' command.
- the Specificities, CFA and Lexical table commands can also be applied to a 'Partition'.

Double-clicking on a result will reopen the results window if it has closed or will display it if it is hidden.

**The File view**

The File view displays a classic tree structure of the folder and file icons on your hard drive.

From within this view, it is possible to open a file in the TXM Text Editor. As a result you can modify the source texts of a corpus or the scripts using TXM.

**Browsing**

The '^' button opens the parent folder of the folder currently being viewed.

In the text field you can edit the path of the folder currently in use, then click 'OK' or tap the 'ENTER' key to refresh the view.

The 'TXM' button sends you directly to the user's TXM folder.
A double-click on a folder will open its content.

A double-click on a file icon will open it in a new Text Editor window. A similar result is obtained using the command 'Open a file' in the 'File' menu.

Through the context menu, you can:

– create a new folder;
– create a new file;
– open a HTML file in the Web browser;
– open a file in the Text Editor;
– run a file as Groovy script
– run a file as R script

**The Console view**

The Console view is one of the most important views. It informs you of the correct (or incorrect) functioning of all TXM calculations. If it is closed, you can reopen it with the command 'View / Views / Console'. If nothing occurs when using this command, refresh the current TXM perspective to reopen it (general windows configuration) using the command 'View / Perspectives / Resetting the perspective'.

**The Text Editor window**

You can open as many text editor windows as needed to edit files in text formats such as TXT, XML, etc. See section 10.2.3 'The Text Editor' on page 57 to learn how it operates.

**The R Variables view**

This view is part of the R perspective. At any moment, it allows you to see all the TXM objects in the R environment by identifying the name of an object in the Corpus view to its symbol in the R workspace (if the object has been transmitted to R, see section < to do >).

The 'Refresh' button refreshes this list. The 'Show eval logs' button, generates a history of the user's R command lines dating back to TXM's start-up.

It is possible to copy an object icon in the R workspace using the context menu or the keyboard shortcut Ctrl+C, in order to paste it for example into an R command line.

**The R Console view**
This view displays the output from all the R commands executed by the user. This output is exactly the same as that from an R interpreter launched from within a Terminal.

**The Query view**

This view generates an inventory of all the CQL search queries in each corpus (not for sub-corpora).

It is possible to copy a query using the context menu, or the keyboard shortcut Ctrl+C, in order to paste it for example into a query field.

It is also possible to export the entire list of queries into a plain text file using the 'Export' button.

**10.2.1.2 Commands**

In TXM, the main commands can be launched in three different ways:

1) Toolbar: when an object's icon is selected in the Corpus view the user can apply a command to this object by clicking the corresponding command icon in the **tool bar**.

![Image 10.5: The Toolbars.](image)

2) Main menu: when an object's icon is selected in the Browser the user can apply a command to this object by selecting the corresponding action in either the 'File', 'Corpus' or 'Tools' **menus**.
a. The 'File' menu, where the Import command can be found:

b. The 'Corpus' menu, where Description and corpus manipulation commands are found:

   The configuration of this menu changes in accordance to the type of icon selected. The first menu below appears when a corpus is selected, while the second appears when it is a partition.
c. The 'Tools' menu gives access to Textometry tools:

![Image 10.8: The Tools menu for corpora (left) and for partitions (right).]

- 46 -

d. The 'View' menu enables access to the layout configurations for the TXM windows.
3) The user can open a context menu by right clicking on the object that they wish to apply a command to.

Image 10.9: Corpus context menu.

The different commands are described in detail in section 4 'Using TXM: Commands'.

Certain commands are equally accessible by double-clicking on a hyper-link in the results windows, depending on the object types contained in these results, for example the table row, concordance row, etc.
10.2.1.3 Icons

Below is a list of the icons found in TXM's visual/graphical interface alongside their names:

**Object icons**

<table>
<thead>
<tr>
<th>Icon</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>📜</td>
<td>Corpus</td>
</tr>
<tr>
<td>📜</td>
<td>Partition</td>
</tr>
<tr>
<td>📚</td>
<td>Bibliography</td>
</tr>
<tr>
<td>📚</td>
<td>Edition</td>
</tr>
<tr>
<td>📚</td>
<td>Lexical Table</td>
</tr>
<tr>
<td>🌡️</td>
<td>Internal view</td>
</tr>
</tbody>
</table>

**Command icons**

<table>
<thead>
<tr>
<th>Icon</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>🎯</td>
<td>CFA</td>
</tr>
<tr>
<td>📚</td>
<td>Bibliography</td>
</tr>
<tr>
<td>📚</td>
<td>Concordances</td>
</tr>
<tr>
<td>🇧🇷</td>
<td>Co-occurrence</td>
</tr>
<tr>
<td>🎫</td>
<td>Description</td>
</tr>
<tr>
<td>🎅</td>
<td>Send to R</td>
</tr>
<tr>
<td>📦</td>
<td>Export corpus</td>
</tr>
<tr>
<td>☐</td>
<td>Export (entry point for the context menu)</td>
</tr>
<tr>
<td>📚</td>
<td>Export data</td>
</tr>
</tbody>
</table>
10.2.1.4 Main Menus

All of the main menus in TXM, located in the upper left-hand corner of the interface, are described here below:

**File Menu**

- Export: exports the results of a command in various formats.
- Import: imports a new corpus from its source using the various available import modules (see section on 'Import modules' for more information about these):
  - Clipboard: imports text copied to the clipboard;
  - TXT+CSV: imports files in plain text format accompanied by a metadata file of CSV formatted texts 'metadata.csv';
  - XML/w+CSV: imports XML files with their words in plain text or marked up by a `<w>` tag that is accompanied by a 'metadata.csv' file;
TXM User Manual 0.7 - June 2015

- XML-TEI BFM: imports files in XML-TEI P5 format, including the encoding of texts and metadata, in accordance with the BFM project recommendations;
- XML-TEI Frantext: imports files in XML-TEI format from ATILF (Analyse et Traitement Informatique de la Langue Francaise - Computer Analysis and Processing of the French Language);
- XML-TEI TXM: imports files in the internal TXM format;
- XML-TRS+CSV: imports files in '.trs' format generated by the program Transcriber and accompanied by a 'metadata.csv' file;
- XML-PPS: imports files in XML format produced by the Factiva portal;
- XML-TMX: imports aligned corpora in TMX format;
- Factiva TXT: imports files in text format produced by the Factiva portal;
- CNR+CSV: imports files in CNR format (produced using the program Cordial) accompanied by a 'metadata.csv' file;
- Alceste: imports files formatted for the program Alceste;
- Hyperbase: imports files formatted for the older version of the program Hyperbase;
- CWB: imports corpora in the source format of the CQP engine.

- Load: loads a new corpus from its binary folder;
- New file: opens a new file in a Text Editor;
- Open...: opens a file in a new Text Editor;
- Save: saves the text file currently being edited;
- Save all: saves all the text files currently being edited;
- Close: closes the text file currently being edited;
- Close all: closes all the text files currently being edited;
- Open in browser: displays a file in a new Web browser;
- Change language: changes the language settings for the TXM interface;

12 http://bfm.ens-lyon.fr
13 http://www.cnrtl.fr/corpus/frantext
• Check for updates: checks if there are any updates available;
• Add an extension: adds a TXM extension;
• Add a third-party extension: adds a third-party extension to TXM;
• Restart engines: restarts the search and statistics engines;
• Quit: closes the TXM application.

**Corpus Menu**

• Open a bibliographic record: displays the bibliographic data for the available texts.
• Edition: displays the first page of the edition of the first text in the corpus
• Description: displays structures and their properties, as well as word properties found in the corpus
• Delete: deletes the selected object.
• Create a sub-corpus: generates a new sub-corpus
• Create a partition: generates a new partition
• Lexical table: creates a lexical table using a partition or the index of a partition table.

**Tools Menu**

• Lexicon: hierarchical list of the values of a word property throughout the whole text.
• Index: hierarchical list of value combinations of word properties for all the occurrences of a given CQL query
• Concordances: searches the occurrences of a pattern expressed using a CQL query and displays the contextualised results in the form of Kwic concordances
• References: displays the references of a CQL query
• Progression: displays the evolution of one or several patterns throughout the corpus
• Co-occurrences: calculates the co-occurrences of a CQL query
• Specificities: calculates the most specific property values for each part of a partition
• CFA: calculates the correspondence factor analysis of a partition for a given word property and displays the primary factorial plane.
• Classification: calculates a classification using a lexical table or an CFA.
• Send to R: sends the selected object's digital data to the R workspace.
Settings: opens the page to configure the command settings. [In this version, this menu is identical to the one accessed through File / Settings]

View Menu

This menu allows you to open the following views and perspectives:

- Perspectives (two different layouts for the TXM windows)
  - Corpus: this is the default layout that allows Textometry commands to be launched and the results to be manipulated.
  - R: this is a specialised layout for working with the R statistics engine.
- Views
  - Corpus: displays the icons of corpora and sub-corpora available for analysis, in addition to the different results already calculated;
  - File: displays the files located in your folders with the added possibility of editing them;
  - Query: a list of all the CQL queries that have been applied to each corpus;
  - Console: displays the TXM error messages or results
  - R Console: displays the output from all the R commands executed by the user.
  - R Variables: a list of the TXM objects that have been sent to R
- ... 

- Reset the Perspective: rearranges all the windows in the current perspectives to their default positions.
- New window: duplicates the entire TXM window as a new window.

Help Menu

- Keyboard shortcuts: displays all the available keyboard shortcuts;
- Graph keyboard shortcuts: displays all the shortcut keys to manipulate graphs;
- Online manual: opens the online version of this manual in a browser window;
- Report a bug: opens the 'Bug report' webpage;
- Features requests: opens the 'Features requests' webpage;
- Sign up to the TXM user's mailing list: opens the inscription form for the TXM user's mailing list;
TXM User Manual 0.7 - June 2015

- Install TreeTagger: opens the TreeTagger setup tutorial in TXM;
- Go to the TXM website: opens the TXM download page;
- Textometry project website: opens the home page for the Textometry project website;
- TXM manuals' folder: opens the manuals' page on the software's website;
- Textometry Documentation: opens the project's documentation page;
- About: displays details about the current TXM version, information about its licenses, in addition to a list of all the software components and their licences.
10.2.1.5 Displaying the results

All the command results are, by default, displayed in the results zone on the right\textsuperscript{15}.

The results of each new command are displayed as a new tab with its name matching the applied command and its parameters. A new icon is also added to the 'Corpus' view, following the same process.

The name of the window is displayed on the tab and in the icon's legend.

\textsuperscript{15} This zone can be moved anywhere you wish by using the windows manager.
This tab allows you to manage the window's display, as explained in the section on window management.

To increase the surface of the results display table, you can click the [Hide parameters] button.

If a window is inadvertently closed during a session, it can be reopened by double-clicking the corresponding icon in the 'Corpus' view.

**Searching in the results**

The shortcut 'Ctrl+F' (Find) launches a character string search in a results table.

**Copying & Pasting the rows of results**

Results rows in a table can be highlighted using the mouse and copied to the clipboard using the shortcut key 'Ctrl+C' (Copy). As a result, the clipboard will contain a tabulated version of the original table's rows encoded in UTF-8 plain text.

### 10.2.1.6 Displaying messages in the Console

![Image 10.11: The messages.](image)

The current messages panel displays temporary information such as the number of results generated by the last command.

The command comments zone gives more information related to these commands. Text here can be scanned, selected, copied and pasted. Error messages will also be displayed here.

### 10.2.1.7 Changing the interface using perspectives

The perspectives, which are accessible through the 'View > Perspectives' menu, coherently organise how commands, views and results windows are accessed, according to the type of work session:
– The 'Corpus' perspective: installed by default, displays the available corpora and provides access to the commands and results views enabling a Textometric analysis session to be conducted;

– The 'R' Perspective: specialised for work sessions favouring the use of the R statistics environment. It facilitates the transfer of TXM objects (lexical table, specificities table, corpora, sub-corpora, index, concordances, etc.) towards the R workspace, as well as the manipulation of these objects using R symbols and running R scripts with a specialised console.

10.2.1.8 Resetting the user interface

The perspectives memorise the arrangement of the windows in a particular TXM session. At any moment you can reset the window arrangement in the user interface by using the current perspective’s context menu (right-click on the call button of the perspective in the toolbar), by using the 'Reset' command in the context menu of the perspective's button.

10.2.2 The Windows Manager

Using the Windows Manager, you can effectively maximize, minimize, collapse, reopen, move and resize any interface window using the mouse.

The possible modifications that can be carried out are:

• maximize the window to full screen: double-click on the window tab;

• return the window to its original size: double-click on the window tab;

• move and resize a window by 'dragging-and-dropping': drag the window tab to the desired position. Before releasing the mouse button, a phantom window will appear demonstrating the proposed size of the window if positioned at this spot. Each ARRIVING WINDOW has four potential placement 'hot spots' along its borders:
  ◦ to the left: to split vertically and position the window on the left side;
  ◦ to the right: to split vertically and position the window on the right side;
  ◦ at the top: to split horizontally and position the window above;
  ◦ at the bottom: to split horizontally and position the window below.

• to minimize the window: click the “Minimize” icon;

• to close all the windows at once: in the context menu of the view tab, select 'Close all'

• To detach a window: in the view tab's context menu, select 'Detach'.

Each window containing objects and results zones is managed in a logical way.
The position of the windows is automatically saved by TXM and reinstalled the next time TXM is opened.

10.2.3 The Text Editor

TXM's integrated Text Editor allows you to modify, save, etc. any type of text file (TXT, XML, etc.). You can open as many files as you like using:

- the 'File' view (by double-clicking on the file icons to be edited);
- the 'File > New File' menu (to edit a new file);
- the 'File > Open...' menu.

The Text Editor commands can be accessed through:

- the toolbar (see Image 10.5);
- the context menu (right click the mouse);
- the keyboard shortcuts (see section 17 page 172 'Keyboard shortcuts').

The Text Editor Toolbar

Image 10.12 The Text Editor Toolbar:

The tools located on the toolbar are in the following order:

Settings manager
- TXM settings manager

File manager
- Edit new file
- Edit existing file
- Save current file
- Save current file under a new name

Edition
- Cut selected text
- Copy to clipboard
- Paste clipboard contents
– Delete selected text or character to the right of the cursor
– Undo previous edit
– Redo cancelled edit
– Select all text

**Find & Replace**
– Find and replace a character string or a regular expression:
  – Previous (in text above)
  – Next (in text below)
– Incremental search (the search is specified with each character entered) in text below
– Incremental search in text above

**General**
– Go to row number...
– Display row numbers
– Display non-printable characters
– Open link selected in a browser
– Re-open file with a specific character encoding
– Return to original version of the text

**Running scripts**
– Run the script (through either Groovy or R interpretors according to the file extension)
– Run selected rows (through Groovy or R interpretors)
– Run a Groovy script contained in a file
– Re-run last Groovy script or last macro executed

**The Text Editor's context menu**
– Cancel editing
– Return to the original version of the text
– Save
10.3 Working with Corpora

10.3.1 Importing text sources into TXM

The TXM software is designed to import and analyse three major groups of textual corpora:

- A. corpora based on written texts, potentially including numbered editions with facsimile images (like, for instance, images of medieval manuscripts or author's manuscripts or even copies created by students)
- B. corpora based on transcripts of recordings, possibly synchronized with an audio or video source
Importing corpora consists of analysing source documents to construct a corpus representation within TXM that corresponds to its processing model. This model is the basis for all the calculations that can be carried out in TXM. It is composed of the following key components:

- each corpus is composed of a collection of **texts** that have properties called 'metadata' (author, title, date, genre...)
- each text is composed in an undefined fashion of internally nested **structures** capable of generating properties
  - each text is composed of a **word** sequence also capable of possessing properties. These words can be nested in text structures and can form the smallest unit of a corpus.

Teaming each word in a text with its lemma and morphosyntactic category should be done when importing a corpus. It can be done using the automated tools provided (such as TreeTagger).

This is also the occasion to build an HTML edition for each text contained in the corpus, to enable the 'Back to text' analysis command action.

For corpora comprised of written texts, the edition's pagination can be aligned with facsimile image files (manuscript folios, manuscript pages or editions, etc.). This allows for a synoptic reading of the facsimile or edition page in TXM.

Transcript corpora comprised from recordings can be synchronised with video or audio files (i.e. the original sources of the transcripts). This allows corresponding video or audio passages to be played on request when a word occur in TXM.

Aligned corpora are aligned according to an internal structure (sentence, paragraph, etc.). This allows given words appearing in the aligned passages to be searched simultaneously in two languages or in two different versions of the same text.

The folder path for corpora source texts is systematically input into an import module, along with other parameters if necessary, via the importation parameters form, that is opened in advance using the import command (see section 10.3.2).

All these import modules result in new corpus objects being added to the 'Corpus' view. TXM commands, such as Information, Lexicon, Concordances, Edition, etc., can then be applied to these objects. A new internal TXM folder is created in the workspace: `$HOME/TXM/corpora/<nom du corpus>`. This contains the corpus in the format labelled 'binary'.

### 10.3.2 Parameters form for import modules

The parameters for the import modules are configured using a form (see Image 10.13):
Before beginning, you must select the folder containing the source files to be imported. To do so, click the 'folder' icon.

All the import parameters will be saved in a file named 'import.xml' located in this source folder. If the file does not already exist, it will be created.

Once the source folder is selected, you can modify the following parameters (open the various zones by clicking on their titles):

- Corpus name: this is the internal name given in TXM which will notably be displayed in the Corpus view. It must follow a very strict format. It must only contain non-accented capital letters or numbers, and it must also not begin with a number. If the name does not conform to this format, the importation can not begin;

- Description: the detailed corpus description is unformatted (full name, author, production date, distribution license, comments, etc.). HTML tags can be used if you wish to format it (bold print, italics, subtitles, etc.).

In the following sections, however, their display will depend upon the import module used:
Character encoding: to be specified if the character encoding of the source texts is not Unicode UTF-8\(^\text{16}\). The default character encoding system for texts varies according to the operating systems:

- Windows: it is normally 'windows-1252' or 'cp1252';
- Mac OS X: 'x-MacRoman' or 'MacRoman'

If, depending on the text, the encoding varies or if you don't know which to choose, you can select the option 'Guess', which will attempt to automatically determine the text's encoding\(^\text{17}\).

Display Font: select a particular character font for displaying the results and editions.

Main language\(^\text{18}\): used for lexicographic sorting and for selecting the linguistic model to be used by TreeTagger when the option 'Annotate' is selected. Like for the character encoding parameters, the option 'Guess' will attempt to automatically determine the language of the texts\(^\text{19}\).

Tokenizer: you can configure the tokenizer's behaviour by modifying several of its parameters. To find out the default names and values for these parameters, refer to the script: [https://txm.svn.sourceforge.net/svnroot/txm/trunk/Toolbox/trunk/org.textometrie.toolbox/src/groovy/org/txm/tokenizer/TokenizerClasses.groovy](https://txm.svn.sourceforge.net/svnroot/txm/trunk/Toolbox/trunk/org.textometrie.toolbox/src/groovy/org/txm/tokenizer/TokenizerClasses.groovy)

XSL input sheet: for import modules based on an XML format. Before reading any source files, TXM can apply an XSLT transformation sheet to them in advance\(^\text{20}\).

Editions: choose to generate or not to generate editions (practical for testing the importation of a large corpus source file), choose to generate the number of words per page during the pagination, in addition to the XML tag used to mark page limits, in the case of import modules based on XML format (default tag is <pb>).

Commands: allow you to configure the behaviour of certain commands.

\(^{16}\) The TXM macro 'ChangeEncoding' allows you to modify character encoding by lots from all the source files of a corpus located in a folder. It should be used from within TXM on the chosen source folder, before proceeding to import the corpus. This is documented on the TXM macros documentation page: [https://groupes.renater.fr/wiki/txm-users/public/macros#changeencoding](https://groupes.renater.fr/wiki/txm-users/public/macros#changeencoding). Once the sources files are encoded in Unicode UTF-8, it is no longer necessary to modify the 'Character encoding' importation parameter.

\(^{17}\) The search algorithm for encoding is initially launched across all texts to establish a general value. And is later applied text by text. If a text is too small then the general value will be used.


\(^{19}\) See 14

Configuring structures that delimit the concordance contexts. By default, this field only contains the structure 'text' (concordance contexts do not go beyond the limits of each text). For example, concordance contexts of transcript corpora can be limited to speech turns by using the structure 'sp'. In this case, this parameter is set to the value 'text,sp' in order to combine the limits of these two structures.

### 10.3.3 Import module example: Importing with CNR+CSV

The CNR+CSV module reads corpus source files that adhere to the following formats:

- each documentary unit must be contained in one individual file;
- the format of the documentary unit is 'CNR': this is the output format of 'Cordial' a commercially-owned French tagger and lemmatiser program. This format is similar to the CSV format (one word per row, followed by its properties that are separated by tab strokes). Cordial must be configured to provide each word with the following properties: paragraph number, sentence number, form, lemma, morphosyntax and function;
- all the metadata is stored in an Excel document exported in CSV format. All the metadata from one documentary unit is located on the same row. Each metadata element is defined in a column;
- the only structural unit recognised and encoded is the sentence, which is provided by the Cordial tagging program;
- the properties of lexical units are decoded using the columns of the CNR file (written form of word = property 'word', lemma = property 'lemma', and part of speech = property 'pos').

This module can be applied to the source files of the example corpus DISCOURS. Consequently, the results of this module are:

- a new corpus root added to the 'Corpus' view, to which all the TXM commands can be applied;
- two different HTML editions per documentary unit: one paginated every 200 words and the other one single page. You can hover over each word in these editions to see its properties temporarily displayed;
- the search engine indexes have been composed.

The following section offers a general overview of the different import modules, along with the information we recommend you provide in the dialogue box (Image 10.13).
10.3.4 Other import modules

All the formats that can be imported into TXM have been summarised in the following import modules table:

<table>
<thead>
<tr>
<th>Name</th>
<th>Documentary unit</th>
<th>Main format</th>
<th>Metadata</th>
<th>Structural units</th>
<th>Lexical properties</th>
<th>Editions</th>
<th>Recommended options</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNR+CSV</td>
<td>One text per file</td>
<td>Cordial CNR Multext tagset</td>
<td>'metadata, csv' file</td>
<td>s (sentence)</td>
<td>word, pos, func, lemma (FR)</td>
<td>One single page, paginated every 200 words</td>
<td>System encoding</td>
</tr>
<tr>
<td>Hyperbase</td>
<td>Several texts (in one individual file)</td>
<td>Hyperbase (previous format)</td>
<td>None</td>
<td>s</td>
<td>word, pos, lemma (FR)</td>
<td>One single page, paginated every 200 words</td>
<td>System encoding</td>
</tr>
<tr>
<td>Alceste</td>
<td>Several texts (in one individual file)</td>
<td>Alceste</td>
<td>Analytics</td>
<td>s</td>
<td>word, pos, lemma (FR)</td>
<td>One single page, paginated every 200 words</td>
<td>System encoding</td>
</tr>
<tr>
<td>Factiva</td>
<td>One text per file</td>
<td>Factiva XML</td>
<td>Analytics</td>
<td>s</td>
<td>word, pos, lemma (FR)</td>
<td>One single page, paginated every 200 words</td>
<td></td>
</tr>
<tr>
<td>Transcriber +CSV</td>
<td>One individual file per transcript</td>
<td>Transcriber</td>
<td>'metadata, csv' file</td>
<td>Speech turns, sections, utterances</td>
<td>word, pos, lemma, spk, event</td>
<td>Paginated every 200 words after a speech turn</td>
<td></td>
</tr>
<tr>
<td>XML-TEI BFM</td>
<td>One text per file</td>
<td>XML TEI P5</td>
<td>Bibliographical</td>
<td>s and other BFM units</td>
<td>word, pos (AFR)</td>
<td>paginated by &lt;pb/&gt;</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Documentary unit</td>
<td>Main format</td>
<td>Metadata</td>
<td>Structural units</td>
<td>Lexical properties</td>
<td>Editions</td>
<td>Recommended options</td>
</tr>
<tr>
<td>-------</td>
<td>------------------</td>
<td>-------------</td>
<td>----------</td>
<td>------------------</td>
<td>-------------------</td>
<td>----------</td>
<td>---------------------</td>
</tr>
<tr>
<td>FRANTEXT</td>
<td>One text per file</td>
<td>XML TEI P5</td>
<td>Bibliographical</td>
<td>s and other BFM units</td>
<td>word, pos (AFR)</td>
<td>paginated by &lt;pb/&gt;</td>
<td></td>
</tr>
<tr>
<td>XML-TXM</td>
<td>One text per file</td>
<td>XML TXM</td>
<td>None</td>
<td>All XML tags</td>
<td>Depends on the source</td>
<td>One single page, paginated every 200 words</td>
<td></td>
</tr>
<tr>
<td>TXT+CSV</td>
<td>One text per file</td>
<td>TXT: plain text</td>
<td>'metadata, csv' file</td>
<td>None</td>
<td>word, pos, lemma</td>
<td>One single page, paginated every 200 words</td>
<td></td>
</tr>
<tr>
<td>XML/w+C SV</td>
<td>One text per file</td>
<td>XML</td>
<td>'metadata, csv' file</td>
<td>All XML tags</td>
<td>word, pos, lemma</td>
<td>One single page, paginated every 200 words</td>
<td></td>
</tr>
</tbody>
</table>

10.3.5 Installing TreeTagger to automatically add morphosyntactic properties and lemmas to words

This tutorial will ensure the automatic functioning of your corpus' lemmatisation and morphosyntactic tagging, during its import into TXM. It will help you:

1. Recuperate the program TreeTagger and one or several of its linguistic models that we are unable to provide together with TXM²;
2. Indicate to TXM where your TreeTagger is located and what linguistic model has been chosen.

10.3.5.1 Using the browser and your file explorer

While connected to the internet:
1. Download the TreeTagger software archive that corresponds to your operating system from the TreeTagger website:
   - Windows (32bit and 64bit)
   - Mac OS X
   - Linux 64bit
   - Linux 32bit

2. Extract the contents of the compressed file (*.zip) into a folder named 'treetagger':
   - In Windows
     C:\Program Files\treetagger
   - In Windows XP
     C:\Program Files\treetagger
   - In Mac OS X
     /Applications/treetagger
   - In Linux
     /usr/lib/treetagger

   **Verification:** Once extracted this folder should contain the following folders and files:
   - bin
   - cmd
   - doc
   - FILES
   - LICENSE
   - README

3. Create a sub-folder entitled 'models' in your 'treetagger' folder that will contain the TreeTagger language models.

4. From the TreeTagger website, download the model (compressed file '*.gz') for each language you wish to use for lemmatising:
   - French: french-par-linux-3.2-utf8.bin.gz (fr)
   - English: english-par-linux-3.2-utf8.bin.gz (en)
   - German: german-par-linux-3.2-utf8.bin.gz (de)
   - Italian: italian-par-linux-3.2-utf8.bin.gz (it)
   - Spanish: spanish-par-linux-3.2-utf8.bin.gz (es)
   - Russian: russian-par-linux-3.2-utf8.bin.gz (ru)
   - Classical Latin: latin-par-linux-3.2.bin.gz (la)
   - Old French from the BFM project (without lemmas): fro.zip (fro)
   - Other languages: see the list of all the TreeTagger language models available (in the section 'Parameters files')

5. Extract each compressed model file into your 'models' folder.
   - In Windows, if you do not have archive software that is compatible with '*.gz' files, we recommend the free software 7-zip.

6. Rename each model file using the two-letter ISO 639-1 language codes.
   - For example:
     - 'french.par' becomes 'fr.par' for the French model file
     - 'english.par' becomes 'en.par' for the English model file
In Windows and Mac OS X: These systems hide from the user, by default, the file extensions for those file types they manage. As such, you may think you have renamed a 'fr.bin' file to 'fr.par', though instead the full name actually remains 'fr.par.bin'. If this is the case, you must access the full display for the file names and rename them once again:

- In Windows:
  1. To display an entire file name with its extension, follow this tutorial: [Afficher-les-extensions-et-les-fichiers-caches-sous-windows](#)
  2. Then modify and rename the full name.
- In Mac OS X:
  1. Right click on the file icon (Ctrl+mouse click or tap the Trackpad with two fingers)
  2. Launch the command 'Read information'
  3. Edit the field 'Name and extension': erase the extension '.bin'.
  4. Close the information window.

Verification: The 'models' folder should contain the file 'fr.par' which takes up about 17Mo, and possibly other language model files ('en.par', 'de.par', etc.).

10.3.5.2 In TXM

7. Go to the TreeTagger settings (see Image 10.14: TreeTagger settings):
   - Menu 'Tools / Settings'
   - Go to the page 'TXM / Advanced / TAL / TreeTagger'
   - Fill in the field 'TreeTagger installation folder path'. Click 'Browse...', then select your 'treetagger' folder (see step 2) and lastly click 'OK'

![Image 10.14: TreeTagger Settings](#)
8. **Verifying your installation**

In the event of a problem, you can find additional help in the FAQ section.

If for some reason you are unable to finalise the installation process above, please contact us using the TXM user's mailing list (txm-users@cru.fr) after having signed up as a user: txm-users.

### 10.3.6 Exporting or uploading a binary corpus

TXM can export a corpus into a format called a 'binary corpus', which is easily copied, transferred or uploaded into another TXM.

- **Exporting**: To transfer an already imported corpus to another TXM program (for example on another computer). In the 'Corpus' view, select the corpus and launch the command *(File / Export)*. The file will produce a '.txm' extension. This can then be transferred by email or copied to a USB stick, for example.

- **Uploading**: If you have transferred a TXM corpus file into a binary format (.txm) on your computer, you can rapidly upload it to TXM using the command *(File / Upload)*. This command will result in a new corpus appearing in your 'Corpus' view. This command is much faster than the import command, as it does not analyse the corpus source files. You also need only run it once for the corpus to be permanently uploaded into TXM.

### 10.3.7 Exporting corpus source files in a standard XML-TEI P5 format

During the import process, all the source files of a corpus (regardless of their original format) are encoded in pivot compatible TEI format *(XML-TXM)* in the transferred folder of the binary corpus. You can find this folder via the following path:

- In Windows:
  
  « C: \ Users \ <username> \ TXM \ corpora \ <corpusname> \ txm »
  
  or else:
  
  « C: \ Documents and Settings \ <username> \ TXM \ corpora \ <corpusname> \ txm »

- In Mac OS X:
  
  « /Users/<username>/TXM/corpora/<corpusname>/txm »

---

21 The XML-TXM format is an extension of the XML-TEI P5 format that effectively represents the corpus model processed by TXM <texts@metadata/structures@properties/words@properties>.
In Linux:

```
/home/<username>/TXM/corpora/<corpusname>/txm
```

These files, by-products of the import process, can be used to exchange source files between partners, to import sources into other programs or additionally to permanent store source files. They can equally be re-imported into TXM using the XML-TXM import module.

### 10.3.8 Corpus Examples

When TXM is installed it contains several corpus examples, encoded in formats representative of those capable of being processed by the software. They are all distributed under the Creative Commons BY-NC-SA licence \(^{22}\).

Other corpus examples are available online: [https://groupes.renater.fr/wiki/txm-users/public/corpus](https://groupes.renater.fr/wiki/txm-users/public/corpus).

#### 10.3.8.1 The VOEUX corpus

The 'VOEUX' corpus has been edited by Jean-Marc Leblanc from the CEDITEC research laboratory (Centre d’étude des discours, images, textes, écrits, communication) situated in Créteil Val de Marne, France. It is composed of 54 transcripts of presidential addresses containing the following characteristics:

- seven French presidents: Pompidou (5 speeches), de Gaulle (10 public addresses), Giscard (7 addresses), Mitterand (14 addresses), Chirac (12 addresses), Sarkozy (5 addresses) and Hollande (1 address);
- covering the period from 1959 to 2012.

Each transcript has been lemmatised by the TreeTagger program using the fr.par model. A description of the morphosyntactic tagset is available on the TreeTagger website: [http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/french-tagset.html](http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/french-tagset.html).

The corpus is composed of the following elements:

- structural units: text (address) / p (paragraph) / s (sentence)
- each 'text' is comprised of the following properties:
  - year: in the format 'yyyy'
  - loc: name of president
- each lexical unit contains the following properties:

\(^{22}\) If used, these examples must be cited and are not for commercial purposes, they are distributed using the same license as TXM.
10.3.8.2 The GRAAL corpus

The 'GRAAL' corpus has been edited by Christiane Marchello-Nizia and Alexei Lavrentiev, from the ICAR laboratory (UMR CNRS) situated in Lyon, France. It is based on an edition of the K manuscript (Lyon, Municipal Library, 77 Palais des arts) from the book La Queste del saint Graal (http://catalog.bfm-corpus.org/qgraal_cm).

Each word in the text is tagged using morphosyntactic tags from the CATTEX2009 tagset (an Old French tagset, its definition is available via this address: http://bfm.ens-lyon.fr/article.php3?id_article=176).

The following elements have been encoded into TXM with the importation of this corpus:

- main structural units (which include either all or a substantial part of the words in the text): p (paragraph) / q (direct speech) / s (sentence)
  - the p and s units are numbered with the attribute 'n'
- additional structural units (which includes particular text fragments): add (words or phrases added by scribes), corr (words or phrases corrected by editors), damage (damaged passages in the manuscript), orig (punctuation belonging to scribes), subst (words substituted by scribes), supplied (words or phrases added by editors). These structures correspond to the TEI tags used in the digital edition from which the corpus originates;
- each lexical unit contains the following properties:
  - word: the 'actual' written form;
  - dipl: the 'diplomatic' form;
  - n: the order number of each word in the text;
  - pos: the morphosyntactic tag;

---

The properties $\text{ana}$, $\text{rend}$ and $\text{type}$ represent obsolete or partial annotations in the current state of the corpus.

## 11 Using TXM commands

Quite often, TXM commands open a window that allows the user to configure, launch and view the result of a calculation. A calculation can be interrupted by clicking 'Cancel' in the progress window.

### 11.1 Corpus description

#### 11.1.1 Of a corpus

This command calculates a complete summary of the structure of the selected corpus: the structural elements, the lexical units and their properties:

- number of words: the total number of lexical units in the corpus
- number of word properties: the number of different annotations for each word
- for each type of annotation: for each annotation, its name and its total number of different values is given, along with several examples of these values.
- number of structural units: the number of the different structural units in the corpus
- for each type of structural unit: the name of the structure and a list of its properties along with their values
- for each property: the first n elements of the list of values

Image 11.1 provides an example of the DISCOURS corpus description.
11.1.2 Of a partition

Less comprehensive, the description of a partition produces a graph that presents the size of
the parts making up the partition. Using the settings, you can choose to sort and display these
parts by either size or name.

11.2 Bibliographic records

This command enables the bibliographic records of corpus texts to be displayed in TXM. To
do so, the binary file must be configured correctly during importation.

11.3 Edition

11.3.1 Corpus

This command displays the first page of the HTML edition of the selected corpus. The
preamble of this edition, located at the top of the first page, displays all the metadata for that
text.
In this edition, you can browse:

- to the next page [>] or back to the previous page [<];
- to the end of the text [>] or to the beginning of the text [<];
- to the next text in the corpus [>>] or back to the previous text in the corpus [<<].

Double clicking on a concordance row (see below) will take you directly to the corresponding page in the edition, where the concordance pivot will be highlighted in red (if several occurrences of the query appear on the same page, the additional occurrences will be highlighted in light red).

Image 11.2 shows the first page of the edition for the first text of the DISCOURS corpus:

- in this example the metadata are: id, file, loc, type, date
  - loc: name of speaker
  - type: type of speech
  - date

- you can hover over each word with the mouse and a hoverbox will display its properties: pos, func, lemma
  - in this example, the mouse is placed over the word 'équilibre', and the hoverbox shows:
    - pos = 'Ncms': common noun, masculine, singular (Multext tag);
    - func = '- ': none
    - lemma = 'équilibre'
11.3.2 Partition

The Edition command, when applied to a partition, allows you to browse within parts of the selected partition (see Image 11.3).
This browsing system is similar to that used for a corpus.

11.4 Constructing a sub-corpus

This command constructs a sub-corpus from the selected corpus. The sub-corpus is represented as a descendant of the main corpus in the 'Corpus' view.

This command opens a dialogue box entitled 'Create a sub-corpus'. It is composed of three tabs: these allow you to build sub-corpora in a simple mode, an assisted mode or an advanced mode.

11.4.1 Constructing a sub-corpus: 'simple' mode

Image 11.4 displays the dialogue box for the command 'Create a sub-corpus' as viewed in simple mode.

In this box, you can:

- enter the name of the new corpus (optional). This will be displayed in the 'Corpus' view
- select a structural unit of measure
- select the property for this unit
• select one or several values

The new corpus will contain all the lexical units located within the structural units as they are defined.
11.4.2 Constructing a sub-corpus: 'assisted' mode

Image 11.5 illustrates the input form used to create a sub-corpus in 'assisted' mode that allows you to build a query for the creation of the sub-corpus using the various properties of a structure.

Using this form, you can:

- Enter the sub-corpus name (optional)
- Select 'All criteria' to include all the search criteria entered or check 'Some criteria' to include only certain elements.
- Select the structure of the sub-corpus to be used
- Enter the selection criteria:
  - add criteria using the + button
  - delete criteria using the - button
  - choose the property used by the criteria:
    - which either contains or does not contain the selected value
- Click 'Refresh' to rebuild the query used to create the sub-corpus
- **Modify the query if necessary**
- Finally, click 'OK' to create the sub-corpus
Attention, adding selection criteria will add an 'AND' type limitation to the query logic. You can replace the '&' in the query with ' | ' if you wish instead to 'ADD'.

11.4.3 Constructing a sub-corpus: 'advanced' mode

Image 11.6 shows the dialogue box in advanced mode, this mode allows an expert-user to construct sub-corpora using CQL query language.

In this window, you can:

- enter the new corpus name that will appear in the 'Corpus' view (optional)
- write a CQL query that will select the lexical units for the new sub-corpus

The sub-corpus will include all the lexical units selected by the query.

---

24 The full regular expression is: `/region[text,a]: a.text_loc="Pompidou" & a.text_date=".*1970"`
11.5 Building a partition

This command builds a partition using a selected corpus. The new partition will appear as a descendant in the 'Corpus' view.

Using this command opens a dialogue box entitled 'Create a partition'. It is composed of three tabs: one for simple partition building, one for assisted partition building and one for advanced partition building.

11.5.1 Building a partition: 'simple' mode

Image 11.7 shows the window for the simple mode.

In this window, you can:

- enter the name of the new partition which will appear in the 'Corpus' view (optional)
- select a structural unit of measure
- chose the property of the structural unit selected.
The parts of the new partition will be built as sub-corpora, in accordance with the various values of the selected structural unit. You can not access the parts individually, however they are accessible via the partition object and certain commands that allow you to compare them, for example the Specificities and CFA commands.

11.5.2 Building a partition: 'assisted' mode

The assisted mode allows you to more accurately define part of the partition by offering the possibility of selecting various values for the structural property used to compose each part.

Image 11.8 shows the partition creation window in the assisted mode.

In this window, you can:

- enter the name of the partition that will appear in the 'Corpus' view (optional)
- select a structural unit, as well as one of its properties
- select the values that will constitute a part of the partition
- click 'New part' to create an additional part
  - enter the title of the part in the corresponding field
  - click 'Assign' to transfer the previously selected values to this part
  - click 'Remove' to delete specific values from this part
  - you can also click the cross to delete the part
  - click 'Remove all parts' to simultaneously delete all the parts at once
• click 'OK' to create the partition as configured.

Image 11.8: Assisted mode: building a partition based on the dates of the DISCOURS corpus
11.5.3 Building a partition: 'advanced' mode

Image 11.9 shows the partition creation window in advanced mode. In this window, you can:

- enter the name of the new corpus which will appear in the 'Corpus' view (optional)
- write the required CQL queries, with each one selecting lexical units to compose each part
  - use the + button to add a new part and enter the corresponding query
  - use the – button to delete a part

The new partition will be composed of all the created parts, with each one containing lexical units selected in the corresponding query.

Attention, users must verify that the sum of all the parts combined covers the entire corpus.

You can rename a part by clicking on its name in the Advanced mode.

11.6 Building a concordance

This command builds a kwic concordance from search results that correspond to a CQL query that has been applied to a specific corpus or sub-corpus.

The Concordance window is structured as follows:

25 The full queries displayed in Image 11.9 are:
- 
- 

TXM User Manual 0.7 - June 2015

- a field for entering the CQL query;
- a button for accessing the search history;
- a button to access the editor for the displayed properties of lexical units, in order to select which properties will be displayed in the pivot column;
- a 'Search' button to launch the calculation.
- a 'Hide parameters' button that hides the concordance parameters to improve reading quality.

![Image 11.10: The Concordance window]

Champ de requête CQP
Requête CQP assistée
Historique des requêtes
Lancer la recherche

Clé de tri des différentes colonnes
11.6.1 Queries

The CQP engine allows you to write queries using formal CQL language (see section 12 below on using the search engine).

TXM employs a simplified syntax based on CQL language to easily write queries. For example, to search for 'je', you only need to write 'je' in the Query field.

For more complex searches, you can use a whole variety of CQL language. For example, to search for:

the word 'je' followed by a verb

in the DISCOURS corpus, you can enter the follow query:

"je" [pos="V.*"]

This query can be broken down as such:

- 'je' represents the word 'je';
- [pos="V.*"] indicates that the verb will be to the right of the word 'je';
  - the square brackets [...] indicate that there should only be one lexical unit to the right of the word 'je';
  - pos="V.*" indicates that the occurrence should have the morphosyntactic tag "V.*". In the DISCOURS corpus, tagged using Cordial and the Multext tagset, this request selects all verbs (in this corpus all verbs have a tag beginning with 'V').

It is also possible to launch an assisted query. Clicking the 'Query Assistant' icon opens a window that allows you to build CQL queries more easily:
The 'Additional word' button allows you to add a word to the query.

The first drop-down menu allows you to select a word property.

The second drop-down menu allows you to define a search field that is more or less limited.

The final field allows you to enter a word or several letters.

The drop-down menu located between word no.1 and word no.2 allows you to specify if the words are consecutive or not.

Validate your query by clicking 'OK'. It will appear in the 'Query' field in its CQL form.

To launch the search, click 'Search'.

Before displaying the results of the concordance, both the Console and the status line will indicate the total number of results.

Image 11.12 displays the results:

- there are 206 occurrences;
- the result numbers 22 to 41 are displayed (the second page of results is shown);
- the 'pivot' column creates a list of all the words targeted by the query 'je' followed by a verb;
- the concordances are sorted alphabetically in the 'pivot' column by default;
• in this example the reference is formed from the speaker's name;

• you can open the context menu by right-clicking on a concordance:
  ◦ Define the reference pattern: configures the data in the reference column;
  ◦ Define the sorting property: defines the word property that will determine the initial sorting order;
  ◦ Multiple key sorting: defines several sorting keys/
  ◦ Define the context size: chooses the maximum number of words in the context fields on the right and left of the central column;
  ◦ Rows per page: defines the number of results displayed on each page
  ◦ Select the properties: chooses the word properties to be displayed in each column.

\[\text{Image 11.12: Concordance of the word 'je' followed by a verb in the DISCOURS corpus.}\]
11.6.2 Browsing

A concordance begins by displaying the first page of results.

The browser buttons allow you to navigate through the results:

- [|<|]: return to the first page;
- [<]: return to the previous page;
- [>]: forward to the next page;
- [>|]: forward to the last page.
- [Hide/Show parameters]: hides or displays the parameters of the concordance tool for better reading quality.

The number of rows per page can be configured via the 'File / Settings' menu, then through the window 'TXM > User > Concordances'. To configure uniquely the current concordance window, you must use the context menu of the concordance table.

11.6.3 Return to the text

By double-clicking on a concordance row, you return to the edition page containing the pivot. The edition opens in a new window.

On this page, the pivot is highlighted in red, while the rest of the pivots from the concordance, appearing on the same page, will be highlighted in a lighter shade of red.

If you double-click once more on a concordance row, the same window will be used. To browse within the edition, you can place the edition window next to the concordance window.

11.6.4 Sorting

The concordance rows can be sorted according to each column by clicking on their headings: 'References', 'Left context', 'Pivot' and 'Right context'. You can change the sort direction (highest to lowest or vice versa) by clicking a second time on the heading. You will note that the sorting keys change according to the heading selected. The default mode for sorting is based on the pivot. However, you can change the sorting properties by clicking 'Sort options' in the context menu. Lastly, you can conduct a multiple key sorting by configuring each sort key.
11.6.5 Word properties

Each column containing a word property can be customised in two different ways:

– properties displayed for the pivot can be configured by clicking the 'Edit' button, located next to the query field;
– in a concordance, select 'Define the properties' in the context menu.

11.6.6 References

You can choose what information will be displayed in the Reference column (on the left in each concordance row).

In the context menu, select 'Define the reference pattern'. A window will open like that in Image 11.13:

![Image 11.13: 'References pattern' dialogue box]

All the structural and lexical unit properties are located in the list on the left.

For example, text:loc represents the 'loc' property within the structure 'text'.

To choose a property, select it then click the '>' button to transfer it to the field on the left. The list forms on the right will correspond to what will be displayed in the reference column.

In order to remove a property, select it in the list on the right and click '<' to transfer it to the list on the left.

To change the order of the properties in the list on the right, use the up and down buttons 'v' and 'v'.

11.6.7 Exporting

Concordance rows can be exported in CSV format. Select the concordance icon in the 'Corpus' view and then either click on the icon in the toolbar or on the Export command in the context menu.
11.7 Co-occurrences

This command calculates the table of various co-occurrences for the occurrences of a CQL query. These are sorted by default according to the 'co-occurrence score'\(^{26}\) (an indicator determining the probability of association). It therefore enables co-occurrences of a form, a lemma, a combination of a lemma and a category, etc., to be calculated.

When activated, this command opens a window that has one section for the parameters and another displaying the co-occurrences (see Image 11.14).

The parameters zone allows you to:


*Image 11.14: Co-occurents of words beginning with 'j'.*
• Enter a CQL expression in the query field (you can also use the query assistant).
• Choose the lexical properties used to build the co-occurrences (form, lemma, etc.)
• Configure limits for the frequency, co-frequency and score to narrow down the results. The co-frequency is the number of encounters that occur between the co-occurrences and the occurrences of the query
• Choose the type and size to define the context of the encounter:
  ◦ For the structural context, select 'structure'
  ◦ For the word window context, select 'form'
  ◦ You can define the size of the context to the left and/or right of the pivot (by selecting the number of structures or words that follow the type of context).
  ◦ Or you can omit these contexts by unchecking 'Left context active' or 'Right context active'.
• You can sort the list of co-occurrences by clicking the top of a column.

To launch the calculation, click 'Calculate'.

11.8 Lexicon and Index

Word lists can be obtained via two additional commands:

– Lexicon: calculates a hierarchical list of all the values of a given word property in a corpus or sub-corpus (the frequency of each written form, lemma, etc.);

– Index: calculates a hierarchical list of value combinations for properties that correspond to the occurrences of a CQL query researched in a corpus or sub-corpus (the frequency of each substantive lemma, written forms of occurrences from the sequence 'Adj Subst', etc.).

11.8.1 Lexicon

The Lexicon command calculates a frequency list for all the lexical property values of a corpus or sub-corpus (for example, the word forms, morphosyntactic tags, lemmas, etc.). When activated, the default mode for this command calculates the lexicon of the lexical property 'word' (that of its written form).
The results are then presented in a table:

![Table Image](Image 11.15: Hierarchy list of the written forms of words from the DISCOURS corpus.)

The table can be sorted by column, by clicking the column headings (for example, you can sort by form or frequency). A second click will reverse the sorting direction.

You can export this table in CSV format by selecting the Lexicon icon in the 'Corpus' view.

### 11.8.2 Index

The Index command establishes a frequency list for the properties of occurrences based on a CQL query applied to a given corpus, sub-corpus or partition.
11.8.2.1 Choosing the set of word properties to be listed

Occurrences are calculated according to the selected word properties. By default, it is the written word form of the occurrences generated by the query that are listed and calculated ('word'). TXM also allows you to construct this list using grammatical categories of words, their lemmas or any other word property encoded into the corpus, in addition to using combinations of these properties.

You can select the set of properties to be compiled using the 'Edit' button:

---

27 In this example, the property 'word' represents the word's written form.
In the list on the left, select the properties that you wish to add. Transfer them using the arrows, which allow you to either add or remove properties:

- `>`: adds a property (you can also double-click a property in the left-hand list);
- `<`: removes a property (you can also double-click on it in the right-hand list);
- `^`: changes the order of a property, moving it towards the top of the list (the property that is located at the top of the list will be displayed first);
- `v`: moves the property towards the bottom of the list.

### 11.8.2.2 Queries

You can use the same CQL queries here as are used with the concordance rows (and also the query assistant).

![Index formed using the 'word' and 'pos' properties for the lemma 'pouvoir', in the DISCOURS corpus.](image)

---

28 Double-click on a word to transfer it directly to the right.
11.8.2.3 Filtering the results

You can narrow down the results by using:

- Fmin: minimum frequency for a result to be added to the list;
- Fmax: maximum frequency;
- Vmax: maximum number of results to be displayed. For example, if Vmax = 100, you will obtain the first 100 values sorted by frequency;
- Page size: number of results per page.

11.8.2.4 Browsing through the results

The index will initially display the first page of results.

You can browse through all the results using the following buttons:

- [ |< ]: return to the first page of results;
- [< ]: return to the previous page;
- [>] : forward to the next page;
- [>] : forward to the last page.

11.8.2.5 Using the results to activate other commands

The Index command is linked to the Concordance and Progression commands.

You can select specific rows in the index using the mouse\textsuperscript{29}, then through the context menu, choose the command you want to run:

- Send to concordance: a corresponding CQL query will be created in order to construct the concordance.
- Send to progression: the number of CQL queries created to construct a progression chart is based on the number of rows selected.

11.9 Specificities

The Specificities\textsuperscript{\textdegree} command calculates a statistic indicating whether in each part of a partition the occurrences of a word or CQL query appear in abundance (or in decline). This can also be calculated with a chosen sub-corpus (comparing it to its parent corpus).

\textsuperscript{29} Shift+left-click allows you to select adjoining rows. Ctrl+left click allows you to select several non-adjointing rows.
11.9.1 Specificity score

In order to analyse the occurrence specificity of a textual event in one part of a corpus rather than another, with an 'event' being defined as any appearance of a word or CQL expression, the most plausible number of occurrences can be progressively estimated in the following way:

• Calculating the occurrences of a CQL expression (or of a basic written form) in each part, in other words the frequency, allows you to develop an initial impression with regards to the contrasting nature of the parts.

• Dividing this frequency by the total number of occurrences located in the part in question (also known as the size of the part) allows you to exploit the 'relative frequencies' (like for example in the Stella engine in the FRANTEXT database). This thus 'normalises' the frequency or independently weighs it separately from the size of each part. This allows the frequencies found in each part to be compared more easily.

• Another approach which is even more precise, is to calculate the specificity score of the occurrence of an event within a specific part in TXM. In fact, normalising by dividing by the size of the part leads us to implicitly consider (or not) that relative frequencies are representative of original frequencies (before they are divided by the size). To do this, while erring as little as possible outside of all the additional information, we can consider the relative frequency to be the maximum likelihood of the number of occurrences in a part of any size, according to a normal distribution of occurrence. As such, relative frequency is considered to behave like the mode of a standard probability distribution (for example the middle of the Gaussian bell curve, where it is at its highest and most probable), or like a mean (refer to the properties of normal distribution: mean, standard deviation, etc.) Thus, as it so happens the occurrence probability of a written form - or more generally of a CQL expression - within a part has no reason to behave in accordance with normal deviation. In other words, its distribution would not necessarily resemble a neat Gaussian bell curve, with a mean, a standard deviation, etc. This is what Pierre Lafon has noted in his thesis, insisting on the distortion of

\[
\text{mode}(\{A \in V | A \in p \} = f) = \frac{(F+1)(t+1)}{T+2}
\]

Equation 11.19: Maximum likelihood/probability of occurrence in a part.

• A: the registered event;
• V: all possible events (the vocabulary for the words);
• p: the part in question;
• f: the frequency of the event in the part;
• F: the total frequency of the event in the corpus;
• t: the total number of events taking place in the part;
• T: the total number of events taking place in all the parts.
the distribution of small frequencies ($\ll 20$ for example), which do not resemble a Gaussian bell curve at all. He formalised this occurrence and observed that it instead resembled a type of hypergeometric distribution. This probability law is very general and appears in diverse forms. Though in those cases which are of interest to us, it most often resembles an asymmetrical Gaussian bell curve that leans to the right with a tail tapering increasingly towards the higher frequencies. Moreover, the mode of this distribution, that is the maximum likelihood of occurrence that we seek to estimate, is not obtained through an arithmetic mean, but rather by the Equation 11.19.

In TXM, the calculation of the probability that form $A$ will occur $f$ times in part $p$ of length $t$, where the form occurs $F$ times in total in the entire corpus, whose total length equals $T$ occurrences, was modelled by Pierre Lafon [Lafon80] and can be officially expressed by the Equation in 11.20 30.

\[
Prob_{spécf}(\text{card } \{A \in V | A \in p\} = f) = \frac{C_f^F \times C_{T-F}^{f-t}}{C_T^t}
\]

Equation 11.20: Probability for occurrence in a part

\[
C_n^k = \frac{n!}{k!(n-k)!}
\]

is the sample number of $k$ elements among $n$ elements, or the number of parts of $k$ elements in a set of $n$ elements.

\[
n! = 1 \times 2 \times 3 \times ... \times (n-1) \times n
\]

The exact calculation for the specificity score used in TXM is the calculation of the probability of the fact that the event occurs as many times as we effectively observe it in the part (in other words $f_{obs}$) or even more frequently as limited by the size of the part (following the hypergeometric rule expressed in the Equation 11.20 that relies on $f$, $t$, $F$ and $T$). Concretely, we obtain this score by adding together the values of the probability $Prob_{spécf}$ for each possible frequency of occurrence as demonstrated in the Equation 11.21.

\[
Prob_{spécf}(\text{card } \{A \in V | A \in p\} \geq f_{obs}) = \sum_{f = f_{obs}}^{|V|} Prob_{spécf}(\text{card } \{A \in V | A \in p\} = f)
\]

Equation 11.21: Specificity score

30 This equation can be obtained by proceeding roughly in the following manner. If there are $C_F^f$ ways to obtain $f$ elements among $F$ and $C_{T-F}^{f-t}$ ways of combining the remaining forms of the corpus, then there are $C_F^f \times C_{T-F}^{f-t}$ ways of obtaining $f$ times the form $A$ in a sample of $t$ occurrences. The quotient of this number divided by the number of ways to obtain different samples of $t$ occurrences among $T$ (mathematically speaking $C_T^t$) gives us the probability we seek.
11.9.2 Directly calculating the specificity score

The macro 'ExecR', provided with TXM, allows you to calculate the specificity score given the values defined its parameters\(^3\). By default, the R script example that it runs displays the distribution curve for the probability of specificity.

To use this macro:

- open the view 'Views / Macro';
- double-click the macro 'ExecR';

\(^3\) [https://groupes.renater.fr/wiki/txm-users/public/macros#execr](https://groupes.renater.fr/wiki/txm-users/public/macros#execr)
the parameters window opens (Image 11.22). The default parameters are based on the word 'peuple', present in the D9 speech of Robespierre, illustrated in [Lafon80] (see Figure 1, pp 140-141):

- \( f \) the frequency of the form in the part;
- \( F \) the total frequency of the form in the corpus;
- \( t \) the total number of occurrences in the part
- \( T \) the total number of occurrences in the corpus

- next, click 'Run' to display the probability density curve using these parameters (Image 11.23):

  - the most probable number of occurrences (the mode) is 5;
  - the probability of occurring exactly 11 times in the D9 speech is 0.01013% ;
  - the probability of occurring 11 times or more in the D9 speech (specificity score) is 0.01699%

To reproduce the distribution form so that it corresponds to a specific event in a specificities results table, you need only launch the ExecR macro ensuring that the \( f \), \( F \), \( t \) and \( T \) parameters correspond to the appropriate cells in the table.
11.9.3 Presentation of the results

In TXM, specificity is represented by the whole part of the base-10 logarithms (\( \log_{10} \)) for the estimations of specificity probability as, like the name hypergeometric suggests, the probabilities obtained through the calculations vary in an exponential field and the probability's order of magnitude is often enough to compare it to others. Thus the orders of magnitude are compared instead of the probabilities themselves.

As a rule, the representation of the sub-specificity (or sub-representation) is distinguished from the super-specificity (or super-representation) by a minus sign (-) located in front of the score. We will therefore focus on the weak probabilities (thus those significant values of \( \log_{10} \)) that take into account:

- either the number of appearances that are lower-than-expected if the observation is inferior to the theoretical distribution mode (that is, if the number of appearances of an event in a part is inferior to the maximum likelihood estimated by our hypergeometric distribution modelling (see Equation 11.19). In which case we would speak of sub-specificity or negative specificity;

- or the number of appearances that are higher-than-expected if the observation is superior to the theoretical distribution mode. In this case we would speak of super-specificity of positive specificity.

These should not be confused with strong probabilities (for example with more than a 5% chance), and therefore with weak \( \log_{10} \) values, which instead indicate the banality of the appearance in a part (as they can be predicted using the model of specificities).

For those individuals interested in the exact value of the calculated probability, rather than clustering events amongst themselves using this probability's order of magnitude (which is, our main use for probability estimations), a TXM macro allows you to not only directly calculate the specificity score according to the model parameters, but more importantly to situate this value in the probability density curve (see section 11.9.2 page 97 'Direct calculation of the specificity score').

11.9.4 Specificities of a partition

When the Specificities command is applied to a partition, it opens the following parameters window (Image 11.24):

- Property: the property that will be the subject of the calculation;
- Part focus: selects the parts for the calculation through a
selection of values pertaining to the property value of the structural unit on which the partition is based.

– the <v> button allows you to access all the available values;
– similarly, you can add filters by using the <+> button;
– if you do not choose any filters, then all the parts will be considered.

The results are presented in table form (see the example in Image 11.25)

– rows = the various values of the examined word property (for example the various word forms);
– columns =
  – the first column displays the property value that corresponds to the row (for example the form 'nous');
  – the second column displays 'F', the total frequency of this value in the entire corpus (for example 694 'nous' in the corpus). In the column heading, 'T' represents the total number of occurrences in the corpus (for example a total size of 100,810 words);
  – the other columns function in pairs:
    – the first column displays the value frequency in the part (for example 6 occurrences of 'nous' in the 'Allocution radiotélévisée' part). In the heading of this column, 't' represents the size of the part;
    – the second column displays the specificity score for the value of that part (for example specificity is 21.3 for 'nous' in the part).

Image 11.25 presents the results generated by the Specificities command based on the written form of all the words in the partition related to speech type in the DISCOURS corpus. The table is sorted in descending order according to the specificity score column in the 'Allocution radiotélévisée' part. We can see from this example that the most specific forms in the 'Allocution radiotélévisée' speech type are:

– 'nous' with a specificity score of 21.3 for 241 appearances/occurrences in this genre of text out of 694 appearances in the whole corpus;
– 'notre' with a specificity score of 13.6 for 124 appearances/occurrences in this genre of text out of a total of 335 appearances;
11.9.4.1 Sorting the results

You can sort the table by clicking on the column headings. Clicking the heading a second time reverses the sort order.

Sorting a score column in descending order allows you to rapidly access those words considered the most abundantly used when examining the whole corpus. The last words in the list are considered under-used and the words in the middle - around the score 0 - are considered standard (neither over- or under-represented).

11.9.4.2 Graphical visualization of the specificity scores

See also the shared documentation concerning all visual displays in section '11.15 Graphical visualization of the results', on page 114.

The specificity scores can be viewed in graph form. Using the mouse, select the rows you wish to view in the results table\(^{32}\) then, from the context menu, launch the command 'Calculate a graph using the selected rows'. This will produce a graph like the one illustrated in Image 11.26:

---

\(^{32}\) Holding down Shift+left mouse-click allows you to select several adjoining rows. Holding down Ctrl+left mouse click allows you to select several non-adjoining rows.
In the graph:

- each part is represented by a set of adjoining bars, organised in the same order as in the table;
- each word property (which in this example is the written form of the word) is represented by a bar that is the same colour for each part;
- there is a legend defining each colour in the lower right-hand corner of the graph;
- two red rows mark out the standardisation band on either side of the 0 score axis (bars that remain within this limit are considered to represent standard scores).

The graph can be exported as an image using the 'Export' button in the toolbar.

11.9.5 Specificities in a lexical table

You can apply the specificity calculation to a lexical table (originating from a partition). In this particular case, the word property to be considered would already have been selected, so the calculation is automatically launched.
11.9.6 Specificities in a sub-corpus

When used on a sub-corpus the Specificities command allows you to select what word property the calculations will be applied to using a parameters window similar to that of the Lexicon command. This window can be seen in Image 11.27.

The results are presented in table form (see the example in Image 11.27):

- rows = the various values of the examined word property (for example the various word forms);
- columns =
  - the first column displays the property value that corresponds to the row (for example the form < - >);
  - the second column displays 'F', the total frequency of this value in the entire corpus (for example 90 < - > in the corpus). In the column heading, 'T' represents the total number of occurrences in the corpus (for example a total size of 105,191 words);
  - the third column displays the frequency of the value in the sub-corpus (for example 16 occurrences of < - >). In the column heading, which includes the name of the sub-corpus, 't' represents the size of the part;
  - the fourth column displays the specificity score of the value for the part (for example -0.4 specificity for < - > in the sub-corpus);
  - the fifth column displays the frequency of the value for the complement of the sub-corpus (for example 74 occurrences of < - >). In the heading of this column, which includes the 'corpus or sub-corpus name', 't' represents the size of the complement;
  - the sixth column displays the specificity score of the value for the complement (for example -0.4 specificity for < - > in the complement).
11.10 Progression visualisation

See also the shared documentation concerning all visual displays in section '11.15 Graphical visualization of the results', on page 114.

A progress bar displays the evolution of one or several patterns throughout the corpus. This command is launched on a corpus. It produces either a cumulative or density graph and, on request, superimposes the position of the corpus structures. When launched this command opens a parameters dialogue box, as shown in Image 11.28:

- First select the type of progress graph: cumulative or density
- You can choose a structural unit and one of its properties: each unit limit for each property value will be represented on the graph in the form of a vertical bar. You can choose in the display options to have one bar displayed per structural unit.
- You can filter the values of a property by using a regular expression (to limit the number of bars, for instance).
- Next, you can add one or several CQL pattern queries to be displayed (possibly with the help of the assistant) by using the 'Add' button. You can delete a query with the 'Delete' button

If the 'density' mode has been selected, you can vary the density range using a scale factor. By default, the size of the window is based on the minimum range between two structural units (for example between two texts if the structure selected is 'text').

Several display options are available:

- Display the graph in black & white
- Repeat or do not repeat the property values of the structure.
- Use different styles for rows
By clicking 'OK' you obtain a progression graph as shown in Image 11.29. In this graph, the dates of the speeches are displayed at the beginning of each speech. The curves represent the respective progress of the words 'France' and 'Algérie'.

Image 11.28: Progression graph calculation for the words 'France' and 'Algérie' in the speeches of Pompidou and De Gaulle.
The graph can be exported as an image using the 'Export' button in the toolbar.

11.11 References

The References command displays a list of all the references for those values returned by a CQL query, using information from structural units containing these values.

Next to each reference, you will find the frequency of the reference in brackets. This is the number of times this reference appears in a pivot. The references can be sorted by frequency or alphabetically.

When the CQL query corresponds to a series of lexical units, the first unit will be taken into account.

Use:

- Enter a CQL query in the query field
- Choose the display property for the occurrences, as well as how you want them to be grouped.
- You can also choose the structural properties to be used. Just like with the concordance command, this command is based on a pattern.
Finally, launch the calculation by clicking 'Search'.

11.12 CFA

See also the shared documentation concerning all visual displays in section '11.15 Graphical visualization of the results', on page 114.

The CFA command calculates the correspondence factor analysis for a partition, in which each part is represented by a frequency vector for a word property (written word form, lemma, pos...).

This command must be applied to a partition consisting of at least four parts. It can equally be used with a lexical table. First, you must select a word property. Next, a window similar to that for the Lexicon command will open, as shown in the Image Error: source of reference non trouvée.

The results are displayed in two windows:

- the first window displays a graph of the factorial plane
- the second window assembles together the data tables that enable this graph to be interpreted. This window is subdivided into four tabs:
  - the eigenvalues
  - data about the rows
  - data about the columns
  - a histogram of the eigenvalues

---

The visualization window for the factorial planes allows you to choose what elements are displayed in the graph. To do so, click the 'Display columns' or 'Display rows' buttons on the graph's toolbar.

The scale of the graph can be adjusted using the mouse wheel and the graph's position using the right-hand mouse button.

The scale and position of the graph can be reset by clicking 'Return to original view'.

The current view of the graph can be exported using the command 'Export the view'. Various formats can be selected in the exportation dialogue box.

See also the shortcut keys for zooming, moving, etc. the graph in section 11.15.

By default, the CFA only displays the parts (columns) on the factorial plane.

This parameter can be modified in the CFA settings, in the section entitled 'Chart rendering':

- Show rows: displays the word properties;
Show columns: displays the parts.

In the right-hand panel, various data is available to help the user interpret the coordinates of the columns (variables) or rows (individuals).

The eigenvalues table indicates their rank, value, inertia percentage, in addition to the cumulative percentages.

The eigenvalues bar chart also provides a similar outline.

The data charts based on the columns and rows show:

- the quality of the planes (Q-): this is the representation of the point on each plane, calculated as the sum of the \( \cos^2 \) of the point on the two axes in question. The closer the quality is to 1, the less the point's position is distorted by the projection on the plane.

- the relative weight (Mass): the frequency is relative to the sum of the frequencies of other words (rows).

- the square of the distance from the point to the origin (Dist) (the origin represents the centre of gravity of the point cloud): the greater the distance, the more this point moves away from the mean profile, in other words the more unique it is in relation to the rest of the corpus.

- the participation of the point in the construction of the axis (Cont-): the sum of the contributions is equal to 100 and the points that present the strongest contributions for a given axis serve to interpret this axis.

- the \( \cos^2 \) of the point on each axis (Cos\(^2\)): the measurement of the angle between the vector representing the point and the axis. A \( \cos^2 \) near to 1 indicates that the point is well represented on the axis, while a \( \cos^2 \) near to 0 shows that the projection strongly distorts the position of the point in relation to the axis, and that it would therefore be better to avoid interpreting that point's position in comparison with others, according to its coordinate on this axis. In particular, a point that has a weak \( \cos^2 \) on both axes of the chosen representation has a misleading position. Its apparent proximity with other points should not be interpreted on this plane.

- the coordinates of the points in the space of the three initial axes (c-).

The results windows propose a multiple selection tool that simultaneously highlights points where they appear in both the graphs and the rows of a data table. Clicking on a point in the graph or in one of the tables will highlight it. The multiple selection tool works according to the operating system's usual tool: Ctrl+Left click (Windows and Linux) and Cmd+Left click (Mac OS X). This tool can switch to both add or remove a point from the current selection.

The regular expressions search command (shortcut key Ctrl+F) can be used in a data table together with the multiple selection highlighting tool.
Image 11.31: Example of highlighting points using the multiple selection tool in a CFA, created using a partition based on the presidents in the VOEUX corpus.

The 'Export the view' command conserves these multiple selections in files that it generates.

The CFA command algorithm is implemented though the FactoMineR package.

For more detailed information, notably concerning the R perspective, please consult the documentation for this package:

- official R documentation:  
  http://cran.r-project.org/web/packages/FactoMineR/index.html

- PDF manual:  
  http://cran.r-project.org/web/packages/FactoMineR/FactoMineR.pdf


- reference documentation (including monographs):  
  http://factominer.free.fr/docs/index.html
11.13 Clustering

See also the shared documentation concerning all visual displays in section '11.15 Graphical visualization of the results', on page 114.

After calculating the CFA, you can launch the clustering command on the CFA in question. The main options are:

– the number of clusters
– the dimension to be clustered: columns or rows

You can also find the following additional parameters in the clustering settings:

– tree diagram format (dendrogram): 2D or 3D
– default clustering dimension: rows or columns
– default number of clusters
– distance to be used for the calculation

The algorithm for this command is implemented using the FactoMineR package. Please consult the documentation for this package for more detailed information.

11.14 Lexical Table

A Lexical Table assembles together the different lexical units of a partition and displays them in table form.

This table can be generated out of a partition or from the index of a partition. Once the partition is selected, you need to choose the word property that the lexical table will be based on, as is illustrated in Image 11.32:
The table is presented in the following way: one entry per row, one part per column. This table can be edited, and the rows and columns can be merged or deleted. It is also possible to keep only certain rows according to their frequency, as the size of the table can be limited to a maximum number of rows.

Finally, a lexical table is automatically created as soon as a CFA or Specificities command is applied to a corpus and it will appear in the Corpus view as a descendant of this corpus.

In the Image above you can see the lexical table formed from the partition Dates in the DISCOURS corpus. It is possible to:

- Configure the row number and the minimum frequency. You must then validate this choice by clicking 'Apply'.
- Merge or delete columns: by clicking 'Merge or delete columns'. This opens a dialogue box (see Image 11.34):
In this window specific columns can be selected. This can be done either by using the search field (which filters by word or regular expression) at the top of the window, or by directly selecting a part.

• > allows you to add a specific column
• < allows you to remove a column

Next, you must then choose to either 'Merge' or 'Delete' depending on the desired result. If merging these selected columns, you need to specify the name of the new field.

– Merge or delete rows:
by clicking 'Merge or delete rows' a window similar to the previous one for columns will allow you to edit the number of rows in the table.

– directly select the rows that you wish to delete or merge, then access the context menu by right-clicking the mouse button.

– click 'OK' and you will return to the updated lexical table

– It is possible to export the table created by using the context menu.

– Finally, you can sort the columns by clicking on their headings.

11.14.1 Saving a lexical table

Lexical tables, like all other calculation results, are automatically erased when you quit TXM (only sub-corpora and partitions are conserved). If you wish to conserve a lexical table in order to reuse it in a later work session, you need to export it and save it as a file. You can then recover it at the beginning of your next TXM session by re-importing it.

11.14.1.1 Exporting a lexical table

Attention however, the usual 'Export' command is not used in this situation (this command is used to transfer a table and display it as an Excel or Calc table). Another exportation method is necessary (in R format), which can be accessed by right-clicking on a cell in the table in question, and then selecting 'Export the table'. Your file can then be saved with a '.csv' extension (the character encoding is UTF-8).

11.14.1.2 Importing a lexical table

Create any random lexical table in advance within the same partition. Right-click on a cell in this table and open the context menu, then select the command 'Import a table'. Select the file containing the table you exported. This file will then replace the contents of the table currently open.

Note: any changes you make to the table in TXM are not saved to the file from where you imported it. Therefore, if you wish to save any modifications made during a session, you will need to re-export the table again.

11.15 Graphical visualization of the results

Some TXM commands produce graphic representations in designated areas, which contain a toolbar specifically for viewing and manipulating these graphs.
11.15.1.1 Interactive manipulation

You can interact with the graphs by:

– changing the scale: using either the mouse wheel or Ctrl + and Ctrl - (Cmd + and Cmd - in Mac OS X)
– panning the view: right-click and move the mouse or keyboard arrows
– resetting the original view: use the button on the graph's toolbar

11.15.1.2 Displaying additional data

Various information is displayed in hoverboxes when the mouse cursor hovers over an element in the graph (for example a bar, point or line). The additional data displayed depends on the type of graph (for example CFA, partition dimensions, etc.).

11.15.1.3 Exporting graphs

Graphical representations can be exported to be used in other programs using:

– export the current view: use the button on the graph's toolbar. This command exports the graph as it appears in the visualisation area, in other words it includes any changes made to the scale or panning of the graph.
– the graph can equally be exported via the graph's result icon in the 'Corpus' view (for example CFA, HAC, etc.), by following 'Export → Graph...' through the context menu. In this case, the export is not linked to the graph's specific tab and therefore it will not include any scale changes or pans made to the graph.

11.16 Saving and Exporting results

All TXM command results (lists, tables, graphs) can be exported as a file to be processed using another software program (word processing for publication, spreadsheet program for additional analyses, etc.). In its most basic form, this file is made available in CSV format for lists and tables and in SVG format for graphs. The export command can be accessed either through the context menu, by clicking on the result icon in the 'Corpus' view or by clicking on the 'Export' button on the toolbar, if the object is already selected.

11.17 Results processing with R

For R users, it is possible to manipulate the results in an R work space. Certain results are made available by default in the R environment: Specificities, CFA, Classification,
Progression and Lexical Table. Others can be transferred on request with the command 'Send to R'. For example, the Lexicon, Index, Concordance or Corpus.

Advice on accessing these results through R has been documented in the section Using TXM results and objects from within R on page 150
The 'Tools / Settings' menu provides access to configuring all the default parameters for TXM commands and the TXM interface.

Settings overview:

- **TXM**
  - **Advanced**: advanced parameters for the TXM software
    - Corpus Engine: configurations for the CQP search engine integrated into TXM
    - Graphics Engine: selection of the motor used to generate the graphs, R/SVG or Java
    - Statistics Engine: configurations for the R statistics engine integrated into TXM
  - **NLP**: configurations for the natural language processing software
    - **TreeTagger**: morphosyntactic tagger and lemmatiser used by TXM
  - **User**: TXM's default parameters that allow you to modify certain commands
Correspondence analysis: configures the display parameter for columns or rows in a factorial plane, modifies the format of the results columns, defines the maximum number of rows or a minimum frequency.

Clustering: configures the number of clusters built and the choice of a 2D or 3D view.

Concordances: configures the number of rows per page, as well as the size of the right and left contexts.

Co-occurrences: defines the minimum frequency, the number of encounters and the minimum score.

Description: configures the number of different property values displayed, the sorting (or not) of parts and configures the view for the number of parts in the partition dimension bar plots.

File browser: configures the files to be displayed.

Export: configures the character encoding for exported results, the separator characters for columns and text, the display (or non-display) of the results being exported, and the default file format to be used when exporting a graph.

Import: configures the 'metadata.csv' file encoding.

Progression: configures the type of graph, colour, etc.

References: configures the order for displaying references.

Scripts: configures where scripts will be stored.

Specificities: configures the formatting for scores, the maximum score threshold, and options for graphs.

Lexical table: defines the minimum frequency for entries.

'General': the general settings for the interface.

'Appearance': select the 'theme' for the interface, the default positions for views, editors, etc.

Content Types: associations between file types and processing components.

'Editors': how editor windows function (display one editor only, limit the number of editor windows open).

'File associations': associations between file types and processing components.

'Text Editors': parameters for the Text Editor.

'Keys': configure all the TXM keyboard shortcuts, create your own shortcut keys.
'Perspectives': manage the TXM perspectives (Corpus and R)

'Workspace' (not yet operational)

Section 18 on page 177 details each settings page.
11.19 Summary table of the relations between TXM commands and results

These relations can normally be accessed through the context menu of either a result or an icon.

<table>
<thead>
<tr>
<th>COMMANDS</th>
<th>FROM</th>
<th>TOWARDS</th>
<th>USED BY</th>
</tr>
</thead>
</table>
| CFA      | Partition  
Lexical Table |         |         |
| Clustering | CFA |         | CFA     |
| Concordances | Corpus  
Index  
Lexicon  
Co-occurrences | Edition | Co-occurrences |
| Co-occurrences | Corpus | Concordances |         |
| Corpus    | Importation |         | Co-occurrences  
Concordances  
Corpus  
Description  
Edition  
Index  
Lexicon  
Partition  
Progression |
| Description | Corpus |         |         |
| Index     | Corpus  
Partition  
Progression | Concordances  
Progression | Lexical table for a partition |
| Lexicon   | Corpus  
Concordances  
Progression |         |         |
| Partition | Corpus |         | CFA  
Edition  
Index |
<table>
<thead>
<tr>
<th>COMMANDS</th>
<th>FROM</th>
<th>TOWARDS</th>
<th>USED BY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Progression</td>
<td>Corpus</td>
<td></td>
<td>Specificities</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Lexical table</td>
</tr>
<tr>
<td>References</td>
<td>Corpus</td>
<td>Concordances</td>
<td></td>
</tr>
<tr>
<td>Sub-corpus</td>
<td>Corpus</td>
<td></td>
<td>Same as Corpus + Specificities</td>
</tr>
<tr>
<td>Specificities</td>
<td>Partition</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Lexical Table</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sub-corpus</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lexical Table</td>
<td>Partition</td>
<td></td>
<td>CFA Specificities</td>
</tr>
<tr>
<td></td>
<td>Partition Index</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edition</td>
<td>Concordances</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Corpus</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sub-Corpus</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Partition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notice</td>
<td>Corpus</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
12 Queries for the CQP full text search engine

[by Bénédicte Pincemin, "mémo CQL ", 4 October 2012, TXM Workshops]

12.1 Introduction

12.1.1 CQL, CQP

CQL is an acronym for Corpus Query Language, which is a language used to express queries. A CQL expression (or equation) is a character string which expresses a linguistic pattern (a word, or a sequence of words) according to the values of their properties (such as the grammatical category, lemma or written form).

CQP is the acronym for Corpus Query Processor, this is a software component that processes queries. It is a search engine that allows you to find all the occurrences in a given corpus that correspond to a CQL equation.

The CQP engine was initially developed at Stuttgart University <http://www.ims.uni-stuttgart.de/projekte/CorpusWorkbench> and is now available as open-source software <http://cwb.sourceforge.net>. It has been embedded in TXM where it handles occurrence searches and, more generally, all the selection operations that occur within a corpus. It was chosen due to its excellent performance ratio in relation to the complexity of queries processed.

12.1.2 Queries in TXM: simple queries, assisted queries, advanced queries

CQL is a formal language with a lexicon and syntax that has been designed for its operators. These operators form a metalanguage that enables elements to be combined together in order to search structured patterns.

Learning CQL language is not a compulsory rite of passage for TXM users, however by using it you can access more complex ways of expressing patterns.

For example, if you enter a word into the query field it will be interpreted as a search for words presenting this exact written form in the corpus. This enables you to therefore conduct a certain number of simple searches. However two limitations can be quite quickly noted. Firstly, we remain at the 'surface' of the text, we don't draw on any other linguistic information encoded in the corpus (lemma, grammatical category, etc). And secondly, we are limited by the exact span of the word. The formulation of the search can not be based on different parts of the word (its beginning for example), or on compound word expressions - although this can become possible through the use of CQL.
TXM software contains an assistant to help with writing queries. The assistant can be accessed using the ‘magic wand’ icon to the left of the search query entry field. If you happen to be unfamiliar with query language, this assistant allows you to conduct a search thanks to its highly intuitive drop-down menus. However, it still does not allow you to write as complex expressions as CQL language does, which is much more flexible and comprehensive. It is therefore worthwhile having some knowledge of CQL to access the broader and more specific possibilities for expression.

In practice, we can acknowledge the value of combining the use of the assistant with knowledge of CQL language. The assistant can help make writing an initial version of the query easier, while having knowledge of CQL allows you to fully understand the equation and adjust or refine it if necessary.

### 12.1.3 The process of constructing a query

When building a query, there is often a gap between what you wish to search for (what you believe you have expressed in the query) and what you effectively find in the corpus, that needs to be corrected. It is therefore always wise to verify the actual scope of the query being applied to the chosen corpus, before using it for a statistical calculation.

Both learning and using CQL therefore figure prominently in TXM's Index function. The Index function allows you to list all the forms in the corpus that correspond to the pattern. You can examine the results either by descending quantitative importance (sorting by descending frequency, which is how they are presented by default), or by alphabetical order, which may facilitate their interpretation by grouping those with similar forms.

Examining this list of identified configurations may bring to light certain undesirable forms. On the other hand, it does nothing to indicate those forms which may be relevant but which, by not formally corresponding to the query, have not been located. Methodically, it is therefore recommended that when you have a linguistic pattern you wish to search for, begin with a fairly broad expression and pay attention to minimise the use of overly specific patterns that may lead to reductive results. Examining the corresponding occurrences thus identified will then govern the way in which you add certain limitations, little by little, allowing you to target relevant forms and eliminate undesirables.

### 12.1.4 Examples for learning purposes

The following examples have been chosen to illustrate the possibilities for CQL that we consider to be the most useful. However, you will need to input them in the Index function to really see their effect. They were created to be launched with the VOEUX corpus (http://sourceforge.net/projects/txm/files/corpora/voeux/voeux-bin-0.6.zip/download). Though the DISCOURS corpus is sometimes used in addition, when necessary. The examples displayed on a grey background are more complex and can be disregarded to begin with.
12.2 Simple search [level 1 (infralexical): values]

12.2.1 Searching for a word

bonheur

To search for any given word you need only enter its written form.

the friendship

time

because

frieg
friendsh

A part of a word will not yield any results, the expression must correspond to a whole word as attested within the corpus.

There are three similar ways of expressing a search for a written word form:

bonheur - writing the form as it is

"happiness" - the written form enclosed between double straight quotations marks

[word="happiness"] - by using square brackets along with the reserved operator word 'word'.

More complex methods of writing queries can prove useful for more complex cases.

[word="because" ]

[word="happiness "]

[ word = "happiness "] Blank spaces outside the quotations marks are not significant and can be used to make reading easier.

12.2.2 Writing variations

"government"%c Ignoring case differences (upper or lower case). Here the use of quotations marks is compulsory.

"State"%d Ignoring diacritical marks (accents, cedilla, etc.).

"franc.*"%cd These two elements can be ignored simultaneously in a query.
12.2.3 Truncating and wild-card

liberties?  
The question mark operates on the character that precedes it, signifying that this character is optional (0 or 1 times). It can be placed wherever you like in a word. It is especially useful when the corpus is not lemmatised, or when the quality of the lemmatisation is insufficient.

age%d%s?  

"premiere%s%"d  

The question mark operates on the character that precedes it, signifying that this character is optional (0 or 1 times). It can be placed wherever you like in a word. It is especially useful when the corpus is not lemmatised, or when the quality of the lemmatisation is insufficient.

nation.*  

Dot+asterisk (or star) at the end = 'word beginning with ...'. Dot = 'any character'.

."patri.*  

Star = '0 to n times, with n being as large as you wish'. Useful for searching for a stem.

.+"patri.*  

Plus sign = '1 to n times'. In this example, the query seeks to match a prefix.

."ables?  

These operators can be placed anywhere in the word, you can search for words that share the same affixes with stems that can vary freely.

in."ables?  

"i[mn]."ables?"  

Square brackets are practical for indicating a set of possible letters, out of which only one is needed to match.

.*  

0 to n characters, any character possible. This expression captures all words.

.* .*  

Written forms with at least one blank space.(in DISCOURS)

...  

Words of three letters.

12.2.4 Punctuation

\.  
The function of special characters (operators) needs to be 'switched off', which can be done by preceding them with a backslash,

\?  

if you wish to consider them as actual characters themselves in your search.

.*  

This is not the case with all the punctuation signs. For example, the backslash is not needed for words ending with an apostrophe.

12.2.5 Character classes

.+\{p\}  

Words ending with a punctuation symbol: allows you to also capture slanted apostrophes (often originating from Word and which cannot be easily entered into TXM using the keyboard).
12.2.6 Expressing alternations

\texttt{peace} | \texttt{war} \hspace{1cm} \textit{OR symbol ( | ) = non-exclusive alternation. Widens the search to include variations in the formulation.}

\texttt{(inter|supra)nation.*} \hspace{1cm} \textit{Can be used inside a word, with round brackets (parentheses) to delimit its scope.}

\texttt{(inter|supra)?nation.*} \hspace{1cm} \textit{Optional or repetitive operators can also be applied to the contents of the parentheses.}

12.3 Conducting a properties search [level 2 (lexical): the properties]

12.3.1 Introduction

At this point, all the searches demonstrated relate to the written form, which is recorded as the property \textit{word}. For example, \texttt{[word="bonheur"]} indicates that we are searching for the value \textit{bonheur} as the property \textit{word}, which in turn corresponds to the written form. However, when a corpus is enriched, words carry extra information in addition to their written form in the way of other properties. Queries can therefore be carried out for other word properties (and can also combine them).

As the written form is also a property (almost) like any other, everything discussed in the previous section also applies to other property values, regardless of the property, with the exception of simplified writing.

To examine the properties you need to know their name and their values. Effectively, property names depend on the importation of the corpus. In certain corpora, the property that records the lemma is called \textit{lemma}, in another corpora it may be \textit{frlemme}, and in another perhaps \textit{ttlemme}, and so on. In the same regard, values for the grammatical categories depend upon the tagset employed. In the TXM desktop version, the Description command shows which properties are present and, for each one, provides a snapshot of several attested values (using the initial occurrences found in the corpus as its examples). The Lexicon command allows you to comprehensively list the values of a known property found in the corpus. In the desktop version, double-clicking on one of these values allows you to see its use in context (in a concordance). It is useful to have these Description tables demonstrating the tagsets used in the corpus you are working on.
12.3.2 Searching for a property

Searching for a lemma allows you to refer to a word using its (many) variable forms. You need to be very explicit about what property you are working on, therefore the square bracket formulation is necessary.

In this example, the lemma 'je' includes also its split forms (j') or those beginning with a capital letter.

Similarly, you can search for other properties, like the grammatical category.

The value taken on by a property can use the same operators as previously mentioned in order to, for example, reconstruct categories by grouping together certain tags.

Here the vertical bar ( | ) is an integral part of the tag (ambiguities unresolved by TT).

12.3.3 Expressing alternations (2)

There are several, more or less factorised, ways to express an alternative.

The vertical bar is the most commonly used operator, its scope can be limited through the use of parentheses.

Square brackets are only used when alternating between single characters, though they also facilitate the expression of a wider selection (in DISCOURS) or range.

(in DISCOURS) The circumflex negates the chosen value. It excludes a given set of characters.

Using the vertical bar is the only way to indicate alternations within character sequences (either with or without identical lengths).

12.3.4 Combining information

Categorical disambiguation of a lemma.

Combining a category and a morphological feature
12.4 Searching for patterns of several words [level 3 (supralexical): lexical unit sequences]

12.4.1 Word sequences

[word="réduction"] [word="du"] [word="temps"]
[word="de"] [word="travail"]

"réduction" "du" "temps" "de" "travail"

[frlemma="réduction"] "du" "temps" "de" "travail"

[frpos="NOM"] [frlemma="de"] [frpos="NOM"]

[frpos="NOM"] [frlemma="de"] [frlemma="le"]? [frpos="NOM"]
[frpos="NOM"] ([frlemma="de"] [frlemma="le"] [frlemma="du"]) [frpos="NOM"]
[frpos="DET.*"] [frpos="ADV"]? [frpos="ADJ"]+ [frlemma="année"]

Level three uses the same operators to express variations as those used on the first and second levels.

12.4.2 Expressing insertions

[frlemma="il"][] [frlemma="y"]
[frlemma="avoir"]

[frlemma="il"][]? [frlemma="y"]
[frlemma="avoir"]

[frlemma="il"][][] [frlemma="y"]
[frlemma="avoir"]

[frlemma="il"][]{0,3} [frlemma="y"]
[frlemma="avoir"]

Any lexical unit (wildcard).

Optional lexical word.

With an interval of three lexical units.

With an interval of zero to three lexical units.
With an interval of 0 to 10. Two formulations are possible.

If [*] is used, it is imperative that the expansion is limited.

Intervals containing excluded words, increases the syntactical control.

Span based on the structure (if available) (in DISCOURS)

12.4.3 Distributional study

A position in a pattern can be made variable, either as a wildcard, or constrained by a word property.

Search for negation verbs.

Same search, more refined.

12.4.4 Expressing alternations (3)

Expressions.

Co-occurrences.
12.5 Contextual information

12.5.1 Using structures

\(<s>[\text{pos}="V.*"]\)\) (in DISCOURS) Verbs that begin a sentence.

\(<s>[\text{pos}="V.*"]\) expand to (in DISCOURS) Sentences that begin with a verb.

\(<s>[\text{pos}="Vmsm.*"]\) expand to (in DISCOURS) Sentences containing a specific lexical pattern (here imperfect subjunctive).

12.5.2 Using a property on a structure

\([\text{word}="\text{Algérie}"] \& \_.text\_loc!="dg"]\) Searches for 'Algérie' in a text whose speaker is not De Gaulle.

12.5.3 Relation between two words

\(a:\{\text{frpos}="\text{NAM}\mid\text{NOM}\mid\text{ADJ}\mid\text{VER.}\}* \& \text{word}!="\.*\p{P}\}][\]*\)[\text{word}=a.\text{word}]\) within 10

Repetition, agreement,...

12.5.4 Alignment links between parallel corpora

For this example CorpusLAT, a Latin corpus, has been aligned with CorpusFRO, an Old French corpus (the corpus texts exist in both languages as translated versions). The following queries have been carried out on CorpusLAT.

\([\text{lemme}="HIC"]\) :CorpusFRO [\text{lemme}="CIST"]\) Occurrences of the lemma HIC, for which the lemma CIST is found in the aligned passage in Old French.

\([\text{lemme}="HIC"]\) :CorpusFRO ! [\text{lemme}="CIST"]\) Occurrences of the lemma HIC, for which the lemma CIST is not found in the aligned passage in Old French.

\([\text{lemme}="HIC"]\) expand to seg :CorpusFRO [\text{lemme}="CIST"]\) Segments containing the lemma HIC, for which the lemma CIST is found in the aligned segment in Old French.

\([\text{lemme}="HIC"]\) expand to seg :CorpusFRO [\text{lemme}="CIST"]\) Latin segments aligned with those containing the lemma CIST in Old French (for constructing a sub-corpus for the calculation
of resonance).

<seg>[lemme!="HIC"]*<seg> :CorpusFRO [lemme="CIST"]

Segments not containing the lemma HIC, for which the lemma CIST is found in the aligned passage in Old French.

12.5.5 Additional documentation

For a more comprehensive description of CQL query language, you can consult (in English):

13 Regular expressions

When expressing character strings in CQL queries they must adhere to the syntax for regular PCRE expressions (Perl-Compatible Regular Expressions). A full description of the whole syntax is available in the section "Specification of the regular expressions supported by PCRE" in the PCRE manual, found here: http://regexkit.sourceforge.net/Documentation/pcre/pcrepattern.html.

Below is a summary of the most commonly used operators.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\</td>
<td>ignore operating function of the character to the right</td>
</tr>
<tr>
<td></td>
<td>alternation</td>
</tr>
<tr>
<td>()</td>
<td>grouping</td>
</tr>
<tr>
<td>[...]</td>
<td>defines a character class in square brackets (eg. [aeiou] for a vowel, or [a-z] for any lower case letter)</td>
</tr>
<tr>
<td>[^...]</td>
<td>complement for a character class in square brackets, with the ^ character acting as a negator (eg. [^aeiou] means a character that is not a vowel)</td>
</tr>
</tbody>
</table>

Table 1: Metacharacters (or Operators)

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>matches 0 or 1 times the expression located to the left</td>
</tr>
<tr>
<td>*</td>
<td>matches 0 or more times</td>
</tr>
<tr>
<td>+</td>
<td>matches 1 or more times</td>
</tr>
<tr>
<td>{n}</td>
<td>matches n times</td>
</tr>
<tr>
<td>{n,}</td>
<td>matches at least n times</td>
</tr>
<tr>
<td>{n,m}</td>
<td>matches between n and m times</td>
</tr>
</tbody>
</table>

Table 2: Quantifiers
| \x{CC} | character with the value CC (expressed as a hexadecimal numeral) (eg. \x{E9} "for" é) |
| \xC0 | character with the value CC (expressed as a hexadecimal numeral) |

*Table 3: Character codes*

| \d | a digit |
| \D | not a digit |
| \w | a 'word' character |
| \W | a 'non-word' character |
| \s | a space character |
| \S | a non-space character |
| \p{Class} | a Unicode 'Class' character (eg. \p{Lu} for an uppercase character) |
| \P{Class} | a non-Unicode 'Class' character |
| [[:ClassPOSIX:]] | a 'ClassPOSIX' character (eg. [[:upper:]] for an uppercase letter) |

*Table 4: Character classifications*
| \p{L}       | letter              |
| \p{Ll}     | lower case letter   |
| \p{Lu}     | upper case letter   |
| \p{N}      | digit               |
| \p{Xan}    | alphanumeric character |
| \p{Pd}     | dash character (eg ' - ', '—'...) |
| \p{P}      | punctuation character (eg ':', '.', '...'...) |
| \p{Ps}     | open punctuation character (eg (... |
| \p{Pe}     | close punctuation character (eg ... |
| \p{Sm}     | mathematical symbol character (eg ' ~ ') |
| \p{Cyrillic} | Russian alphabet character |
| \p{Arabic} | Arabic alphabet character |
| \p{Greek}  | Greek alphabet character |

**Table 5: Common Unicode Classes**

| **alpha** | alphabetic character (use: "[[[:alpha:]]"]) |
| **alnum**  | alphanumeric character |
| **ascii**  | character from the code ASCII |
| **digit**  | decimal number |
| **graph**  | printable character, without the space |
| **lower**  | lower case letter |
| **print**  | printable character, including the space |
| **punct**  | punctuation symbol |
| **space**  | space character |
| **upper**  | upper case letter |
| **word**   | word character |
| **xdigit** | hexadecimal number |

**Table 6: Common POSIX classes (an older and more basic system than the Unicode classes)**

**Memorised character references**

---

34 All the Unicode classes are described in the section 'Unicode character properties' of the PCRE manual.
\2 contents of the first recorded parenthesised group (assumes the prior presence of parenthesis '..(..)..' in the expression)

\3 contents of the second recorded parenthesised group

... contents of the first recorded parenthesised group named 'name'
(assumes the prior presence of '..(?<name>..)..' in the expression)

Examples:

- (\([[:lower:]]\)\([[:lower:]]\)).*m.*\3\2: two lowercase letters followed by 'm' followed by the two first lowercase letters in the opposite order;
- (.*\2: the same string twice in a row
- (?<groupe1>.*))\g{groupe1}: the same string twice in a row
14 Using the results graphs in other programs

TXM produces results graphs in the following formats:

**Vector graphics**
- SVG - Scalable Vector Graphics
  Open format of the vector graphic standardised by W3C
  [http://www.w3.org/Graphics/SVG](http://www.w3.org/Graphics/SVG),
- PS - PostScript
  Proprietary format of a vector graphic from the Adobe Systems group
  [http://fr.wikipedia.org/wiki/PostScript](http://fr.wikipedia.org/wiki/PostScript);
- PDF - Portable Document Format
  Proprietary format of the vector document from the Adobe Systems group

**Bitmaps**
- PNG - Portable Network Graphics
  Open format of the lossless compressed bitmap image, standardised by ISO
  [http://www.w3.org/TR/PNG](http://www.w3.org/TR/PNG),
- JPEG - Joint Photographic Experts Group
  Open format of the lossy compressed bitmap image, standardised by ISO/CEI 10918-1

The choice of export format can be configured in the Settings: Settings / User / Export / Default R graphics format.

Vector formats offer an advantage as they can be varied without the loss of any details (zoom = general enlarging or reduction of the image) and they can be edited using specialised software (for example to adjust the typography in accordance with editorial instructions, to improve readability by increasing or reducing the character size without changing the overall scale of the graph, to move the legend, etc.).

We recommend:
- the free and open-source program 'InkScape' for editing the SVG format
  [http://www.inkscape.org/fr](http://www.inkscape.org/fr);
TXM prioritises the exportation of results graphs in SVG format. The bitmap formats are recognised by a larger number of work tools and more importantly are easier to manipulate in word-processing programs. JPEG format is just slightly more recognised than PNG in Windows. It is therefore the easiest format to manipulate outside of TXM, even if it is not yet very practical to manipulate within TXM itself (the way in which it is exported from TXM still needs to be homogenised).

14.1 Directly importing a vector graphic in SVG format to LibreOffice Writer

- produce then export a graph from within TXM in SVG format. For example:
  - when in TXM, create a partition in the DISCOURS corpus, name it 'loc' and use the structure 'text' and its property 'loc';
  - configure the field 'Settings / User / Export / Default R graph format' to the value 'SVG';
  - launch the command 'Dimensions' on the partition 'loc';
  - in the graph's toolbar, click on the button, set the type to '*.svg' in the dialogue box and save the graph as a file;
- import the image into Writer:
  - in Writer, launch the command 'Insert / Image / From a file.,'
  - choose the SVG file by navigating to its folder;
  - the image is then inserted in the document wherever the cursor is placed. If necessary, you can modify the size of the image using the mouse:
click on the image → the green handles for manipulating the size will appear (see Image 14.1);

- 'shift+click' on one of the handles and move the mouse. This will modify the size of the image without distorting it;
- by clicking on the image and holding down the mouse button, you can move the position of the image on the page.

14.2 Directly importing a bitmap image in JPEG format to LibreOffice Writer

- produce then export a graph from within TXM in JPEG format. For example:
  - when in TXM, create a partition in the DISCOURS corpus, name it 'loc' and use the structure 'text' and its property 'loc';
  - launch the command 'Dimensions' on the partition 'loc';
  - in the graph's toolbar, click on the button, set the type to '*.jpeg' in the dialogue box and save the graph as a file;
- import the image into Writer:
  - in Writer, launch the command 'Insert / Image / From a file.';
choose the JPEG file by navigating to its folder;
the image is then inserted in the document wherever the cursor is placed. If necessary, you can modify the size of the image using the mouse:

- click on the image → the green handles for manipulating the size will appear;
- 'shift+click' on one of the handles to vary the size of the image without distorting it;

14.3 Using Inkscape to pre-edit a graph in SVG format

- produce then export a graph from within TXM in SVG format. For example:
  - when in TXM, create a partition in the DISCOURS corpus, name it 'loc' and use the structure 'text' and its property 'loc';
  - configure the field 'Settings / User / Export / Default R graph format' to the value 'SVG';
  - launch the command 'Dimensions' on the partition 'loc';
  - in the graph's toolbar, click on the button, set the type to '*.svg' in the dialogue box and save the graph as a file;
- edit the image in Inkscape:
  - open the SVG file using Inkscape ;
  - using the mouse, trace a rectangle around the y-axis title ('Number of words by part') to select it (see Image 14.2):

Image 14.2: Selecting the y-axis title
– Use the 'left arrow' key on the keyboard to transfer the axis title to the left (or 'click+drag' the selection using the mouse):

Image 14.3: The y-axis title has been moved to the left (indicated here in the red circle)

– You can then save your work to be subsequently imported into a word processor.
15 Running TXM software using Groovy macros and scripts or with R scripts

15.1 Using macros

Using macros is the easiest way to run the software using scripts. For Groovy scripts (see the following section) you can set the parameters using either a dialogue box or by using objects selected in the Corpus view. This is a practical way for users to run a script without needing to know Groovy programming language (and without needing to read scripting code).

Various macros are available:

- pre-installed macros in your TXM can be directly accessed through the 'Macro' view linked to the folder '$HOME/TXM/scripts/macro';
- macros published in the macros download folder located on the TXM software website hosted on Sourceforge: http://sourceforge.net/projects/txm/files/software/TXM%20macros (this folder contains the most up to date versions of TXM macros)
- macros from the TXM user's community listed in the wiki: https://groupes.renater.fr/wiki/txm-users/public/macros
- pre-installed experimental macros located in the folder '$HOME/TXM/scripts/sample/groovy/org/txm/macros'.

15.1.1 Launching a macro

Macros are launched either from the 'Macro' view, opened through the menu 'View > Views > Macro', or by double-clicking on their icon, or additionally, in the case of repeated activation, by using the keyboard shortcut 'F12' (this reruns the last executed script or macro).

There are two ways of providing the parameters for a macro:

- before launching: by selecting an object in advance in the Corpus view (corpus, sub-corpus, lexicon, index, concordance, etc.);
- at the start of the launch: by entering the given parameters into the dialogue box (the parameters are specific for each macro, for example a CQL query or a folder to be processed).

The parameters can be of the following types:
The macro has finished running when the message 'Done:' is displayed in the console, followed by its runtime in milliseconds.

The macro results vary a lot, much like the TXM commands (new sub-corpus or concordance, new files, messages in the console, etc.).

15.1.2 Installing a macro

New macros are available from several sources, see section 15.1.

A macro is a Groovy script file whose file name ends in '...Macro.groovy'. Certain macros can be accompanied by additional Groovy scripts (without the suffix '...Macro'). In this case the macro is downloaded in the form of a ZIP archive.

Installing a new macro in TXM:

– copy the macro's .groovy file or files into the folder $HOME/TXM/scripts/macro;
– open the Macro view (menu 'View > Views > Macro') or refresh it (↺ button in the Macro view) in order to access the macro.

15.1.3 Modifying a macro

In the 'Macro' view, right-click on the macro icon then launch the command 'Edit' in the context menu. Modifying the script requires knowledge of programming using Groovy language (see the following section).

A macro can be associated with several additional Groovy scripts which are not visible in the Macro view. To edit them, you must have the File view open on the folder containing the macros and scripts, which can be located using the following path: $HOME/TXM/scripts/macro.
15.1.4 Creating a macro

To add a macro, you need simply to place the corresponding Groovy script in the folder '$HOME/TXM/scripts/macro' and then refresh the 'Macro' view.

Writing this script requires some knowledge of Groovy programming (see the following section). The simplest way is to begin with a macro skeleton code, created using the 'New macro' button in the Macro view. This button opens a Text Editor in a new script file containing examples of Groovy code lines:

- in order to declare the different types of parameters (whole, character string, etc.), you need to uncomment them so they can be used. Example:

```groovy
@Field @Option(name="query", usage="an example query", widget="Query", required=true, def='[pos="V.*"]')
def query
```

Gloss: declaration of the mandatory 'query' parameter query for the 'CQL query' type using the default value `[pos="V.*"]`.

Detail:

- `@Field` - starts the declaration
- `@Option(...)` - contains the declaration
- `name="query"` - defines the parameter name as 'query'
- `usage="an example query"` - defines the documentation for the parameter
- `widget="Query"` - declares the type of parameter
- `required=true` - makes it mandatory to enter the parameter
- `def='[pos="V.*"]'` - declares the default value of the parameter (which serves to pre-fill the parameter's input field in the dialogue box)
- `def query` - declares the Groovy variable for the parameter

- to open the parameter input dialogue box and launch the macro's execution Example:

```groovy
if (!ParametersDialog.open(this)) return;
```

Gloss: after opening the parameter input dialogue box, you can then run the rest of the macro by clicking 'Run' or you can abort its execution by clicking 'Cancel'.

- to manipulate various objects currently selected in the user interface (elements from the Corpus view: corpus, sub-corpus, partition, results from within other views, text
editors, etc.). Example:

```groovy
println "Corpora selection: "+corpusViewSelection
```

Gloss: The name of the element currently selected in the Corpus view is displayed in the Console.

### 15.1.4.1 Predefined global variables

Macros can use the following predefined global variables:

<table>
<thead>
<tr>
<th>Name of the variable</th>
<th>Description</th>
<th>Groovy Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>corpusViewSelection</td>
<td>the element currently selected in the Corpus view (e.g. corpus, sub-corpus, partition, results)</td>
<td>Object</td>
</tr>
<tr>
<td>corpusViewSelections</td>
<td>a list of the elements currently selected in the Corpus view</td>
<td>List&lt;Object&gt;</td>
</tr>
<tr>
<td>selection</td>
<td>the element currently selected (e.g. selection of text in an editor, query icon in the Query view...)</td>
<td>Object</td>
</tr>
<tr>
<td>selections</td>
<td>a list of elements currently selected</td>
<td>List&lt;Object&gt;</td>
</tr>
<tr>
<td>monitor</td>
<td>internal object providing access to the user interface while running the macro</td>
<td>org.txm.rcpapplication.utils.JobHandler</td>
</tr>
<tr>
<td>gse</td>
<td>internal object allowing a mutual call between Groovy scripts</td>
<td>groovy.util.GroovyScriptEngine</td>
</tr>
<tr>
<td>editor</td>
<td>results window or text editor currently selected</td>
<td>org.eclipse.ui.IWorkbenchPart</td>
</tr>
</tbody>
</table>

### 15.1.5 Predefined TXM Macros

The Macro view proposes several useful macros by default.
15.1.5.1 Assistance with preparing source files for a corpus

- **Text2TXT**: Batch conversion of the format of all text files (MS Word, LibreOffice Writer, etc.) in a folder (.doc, .docx, .odt, .rtf, .html, etc.) to TXT format (plain text);
- **ChangeEncoding**: Batch conversion of the character encoding for all the files in a folder;
- **CharList**: Calculates a list of all the characters used by a Unicode UTF-8 encoded source file and indicates their frequency;
- **SearchInDirectory**: Batch search for a regular expression in all the TXT files in a folder;
- **SearchReplaceInDirectory**: Searches and replaces, by batch, a regular expression in all the TXT files in a folder;
- **TXT2XML**: Batch conversion of all the TXT files (plain text) in a folder to basic XML files;
- **XMLStatistics**: Calculates a frequency table for the XML tags used in files in a source folder (useful for conducting a quantitative overview of the use of tags in documents for which the encoding principles are unknown);
- **ExecXSL**: Applies a XSL Transformation Sheet to all the XML files in a folder, by batch;
- **TextTranscription2TRS**: Batch conversion of transcriptions of recordings, entered using a word processing program (.doc, .odt) or the program Transana (.rtf), to the XML-TRS format used by Transcriber, in order to import them into TXM using the Transcriber+CSV import module;
- **EuroPresse2XML**: Assists with the retrieval and transformation of HTML exports from the EuroPresse portal to be imported into TXM using the XML/w+CSV import module;
- **CSV2XML**: Transforms a source file containing a table in CSV format in which certain columns contain text (typically responses to open-ended questions in a survey) and other data (typically features of responders or responses to closed questions) in a XML file that can be imported in TXM with the XML/w+CSV import module (generally for conducting analyses that compare the characteristics of responders with the words used in their open responses).

15.1.5.2 Assistance with configuring text editions

- **AddFacsAttribute**: Updates the XML source files with links towards the image files containing a facsimile of each page;
15.1.5.3 Assistance with correcting the annotation of words in a corpus

- **BuildWordPropTable**: Exports a TXM concordance as a TSV table (with the columns separated by a TAB character) containing word properties to be correctioned or added using a spreadsheet program (such as Calc);

- **InjectWordPropTable**: Updates the XML-TXM pivot source files of a TXM corpus using a table that has been edited for re-importation with the XML-TXM import module.

15.1.5.4 Running R scripts

- **ExecR**: Macro example that calls up an R script that generates a graph, which is then displayed within TXM (calculates an Index histogram);

- **PlotSpecif**: Calls on the “specificities.distribution.plot” function in the 'textometry' R package to display the density curve for the statistical model of specificities by selecting the values for its T, t, F and f parameters.

15.1.5.5 Assistance with calling up CQP commands

- **CreateCQPList**: Defines a list of words (or more generally property values, such as a list of lemmas, list of categories, etc.) that can be used in CQL queries;

- **ExecCQP**: Runs any command line through the CQP engine;

- **SetMatchingStrategy**: Changes the CQP's “matchingstrategy” option (which influences the way word sequences of varying length are calculated) for the current work session.

15.1.5.6 Assistance with repetitively calling up TXM commands

- **CrossedPartitionBuilder**: Builds a partition by combining the values of several properties from the same structure. It is more flexible to use this macro than the CQL query input found in the Advanced mode of the Partition command.
15.1.6 Share your macros with the TXM user community

You can edit the page: https://groupes.renater.fr/wiki/txm-users/public/macros, in the TXM user's wiki, to document your macros and also post a link to them.

15.2 Using Groovy scripts

The TXM software can run Groovy scripts to:

- launch any TXM command: to launch a search using a CQL query, to apply a statistical model, to export and save results in a file, etc.
- set customised parameters for each of these commands;
- save and launch a command sequence for common searches.

This also allows the user to expand the software's functions thanks to scripts.\(^{35}\)

Scripts and macros are written in Groovy language (http://groovy.codehaus.org).

You can find a brief introduction on how to use this language at the following address: http://onjava.com/pub/a/onjava/2004/09/29/groovy.html.

These three reference books can provide you with more information:

- *Groovy in action*\(^{36}\)
- *Groovy programming: an introduction for Java developers*\(^{37}\)
- *Programming Groovy*\(^{38}\)

Text for the scripts to be run can be located either in a file or can be simply selected in a window (see the section 'Text editor').

The best way to begin writing your own Groovy script is to copy one of the script samples included with the TXM software and located in the folder: 'C:\Documents and Settings\<username>\TXM\scripts\groovy\samples'.\(^{39,40}\) For example, the script 'conc.groovy' automatically calculates a concordance for the word 'je' in the DISCOURS corpus, then exports the results in CSV format as a file named 'conc.txt'.

---

\(^{35}\) Similar to what occurs in MS Word through the use of Visual Basic macros.

\(^{36}\)Dierk König et al., *Groovy in action* (Greenwich: Manning, 2007).


In order to run your own Groovy scripts in TXM, they must be located in the folder: 'C:\Documents and Settings\<username>\TXM\scripts\user'.

To create a script, place a sample script into this folder or go to the File view (see section 3.2.1.1.2 The File view and Text Editor), and right-click on the User folder, then select 'Create a file', which you can then name (for example 'test.groovy'). Double-click its icon to modify this script.

**15.2.1 Running a script**

There are seven different ways to run a groovy script:

- through a Text Editor:
  - using the context menu 'Groovy / Run the text selection' runs text selected in the editor;
  - using the context menu 'Groovy / Run the Script' runs script located in the editor;
  - using the context menu 'Groovy / Run a Groovy file' runs a script located in a file;
  - using the keyboard shortcut 'F11' runs text selected in the editor;
  - using the keyboard shortcut 'Ctrl+F11' runs script located in the editor;

- through the File view:
  - using the context menu in a file's icon, clicking 'Run the Script' runs the script contained in the file;
  - the keyboard shortcut 'F12' runs the last script executed.

---

39 At present, no security protection is integrated into TXM software for the running of scripts, therefore it is necessary to pay close attention to the origin of the scripts being used.

40 In Linux: `/home/<username>/TXM/scripts/user`

41 You also have access to this script online, at the following address:


42 Open a Text Editor by opening a file in either the File view or through the main menu “File / New file” or “File / Open...”.
15.2.2 Using third party libraries (.jar files)

So that a third party library (a .jar file) can be accessed through a Groovy script, it needs to be placed in the 'lib' folder located here: 'C:\Documents and Settings\<username>\TXM\scripts'. Once this is completed, the corresponding packages are then able to be imported from the Groovy scripts.

15.2.3 How to use TXM objects with Groovy

As Groovy is based on Java language\(^{43}\), it provides access to all the Java modules in TXM.

The method for launching TXM commands through Java, and therefore Groovy, is documented in the TXM Javadoc located here: http://txm.sourceforge.net/javadoc.

For example, the parameters for the Concordance command are defined in the package 'org.txm.rcpapplication.editors.concordances', documented at the following address:

All the commands defined in this documentation can be run through Groovy script.

15.3 Using R scripts

15.3.1 Running a script

An R script can be run in five different ways:

- through a Text Editor\(^{44}\):
  - using the context menu 'R / Run the Script' runs script located in the editor;
  - using the context menu 'R / Run the selection as R script' runs text selected in the editor;
  - using the keyboard shortcut 'F11' runs text selected in the editor;
  - using the keyboard shortcut 'Ctrl+F11' runs script located in the editor;
- through the File view:

\(^{43}\)In order to be run, all Groovy scripts are first compiled in Java.

\(^{44}\)Open a Text Editor by opening a file in either the File view or through the main menu “File / New file” or “File / Open...”.
using the context menu in a file's icon, clicking 'R / Run the Script' runs the script contained in the file;

Sample session
Running the demo script 'HelloWorld.R':

- open the R scripts samples folder supplied with TXM via the File view: open the folder '$HOME/TXM/scripts/samples/R'
- open the script 'HelloWorld.R' in a Text Editor by double-clicking on its icon
- open the “R Console” to read the script output: open “View / Views / 'R Console' ” through the main menu
- in the context menu of the Text Editor containing the “HelloWorld.R” script, launch “R / Run the Script”

15.3.2 Using TXM results and objects from within R

All the TXM calculation results which have an icon displayed in the Corpus view can be transferred to R in the form of a data structure. To do so, use the command 'Send to R' located in the context menu of the corpus or results icon, via the Corpus view.

The console will then display the name of the created R object, allowing you to use it in R scripts. To confirm its transfer to the R workspace, a small red 'R' exponent is added to the icon of the TXM object. Certain transfers are automatically carried out by TXM for specific calculations (e.g. lexical tables, progression graphs, specificities).

The 'R Variables' view displays the list of data already transferred to the R workspace. To open the 'R Variables' view, open the menu 'View / Views / R Variables'. You can copy the name of the transferred data to the clipboard either by using 'Copy' in the icon's context menu or by using the keyboard shortcut Ctrl+C when the icon is selected. You can then paste the name of the object into an R script.

15.3.3 Using the R perspective to organise access to R

The R perspective allows you to quickly configure the TXM user interface with other views that are practical when working with R:

- a button labelled 'New session' is available on the toolbar to create script for an R session. This opens a Text Editor for a script file named 'sessionX.R' automatically created in the folder '$USER_HOME/TXM/scripts/R'.
- an 'R Variables' tab is available in the left hand panel, which gives quick access to the 'R Variables' view;
- three windows are stacked to the right of the interface:
Switching to the 'Corpus' perspective, to resume your corpus analysis session will close the 'R Variables' view and the R console.

You can switch from one perspective to the other whenever you like.

By default, the TXM user interface is configured in line with the 'Corpus' perspective. To change the perspective, you can use the perspectives toolbar located in the top right-hand corner of the interface, where the 'R' and 'Corpus' buttons are located. You can also use the main menu (above to the left) 'View / Perspectives / R or Corpus'.

### 15.3.4 The R environment in TXM

When launched, TXM makes several adjustments to the R statistics engine (Rserve):

- The default character encoding is set to Unicode UTF-8
- The directory path for the R workspace is set to:
  
  - In Windows:
    
    'C:\Users\<username>\TXM\R'
    
    or else:
    
    'C:\Documents and Settings\<username>\TXM\R'
  
  - In Mac OS X:
    
    '/Users/<username>/TXM/R'
  
  - In Linux:
    
    '/home/<username>/TXM/R'.

- The installation folder for R packages is set to:
  
  - In Windows:
    
    'C:\Users\<username>\TXM\R\libraries'
    
    or else:
    
    'C:\Documents and Settings\<username>\TXM\R\libraries'
  
  - In Mac OS X:
    
    '/Users/<username>/TXM/R\libraries'
  
  - In Linux:
    
    '/home/<username>/TXM/R\libraries'.
Options for configuring the launching of R (and Rserve) can be accessed through the Advanced settings page for the R workspace (TXM > Advanced > Statistics engine). For more information on R, see the following pages:

- R Options: http://www.r-project.org
- Rserve Options: http://rforge.net/Rserve/doc.html#start

15.3.5 Sample work session using R

15.3.5.1 Displaying the frequency histogram for a lemma index calculated using R

The aim of this sample session is to display the frequency histogram for the first 10 lemmas of the most common nouns found in the DISCOURS corpus. The instructions are as follows:

- build the Index using the noun lemmas:
  - configures the Index command with the parameters:
    - corpus: DISCOURS
    - query: [pos="N.*"]
    - property: lemma
    - Vmax: 10
- send the Index to R:
  - launch the command 'Send to R' from the context menu of the Index results icon, which should appear under the DISCOURS corpus icon in the Corpus view and be named "[pos="N.*"]:lemma"
  - a red letter R should appear next to the Index results icon to confirm the transfer
  - find and copy the name of the R object created in the Console: for example 'Index1'
  - open the R perspective by clicking on the 'R' in the perspectives toolbar
  - start a new session by clicking the 'New session' button in the toolbar
  - copy the following R script to the Text Editor labelled 'sessionX.R' (where 'X' represents the session number):

```
svg("/tmp/test.svg")
barplot(t(IndexN$data), space=c(1,35), horiz=F, las=2, beside=T)
dev.off()
```

- 152 -
- edit the 'IndexN' string by pasting the name of the R object created during the transfer to R (for example 'Index1')
- run the script by clicking the button 'Run' in the toolbar (green arrow)
- check the results by opening the file '/tmp/test.svg' in your browser using the file manager.

15.3.5.2 Displaying the histogram directly in TXM using a Groovy script

You can display the SVG file generated directly in TXM by double-clicking on its icon from within the 'File' view (its object is located by default next to that of the 'Corpus' view).

You can also directly display the SVG file by running the following Groovy script:

```groovy
import org.txm.rcpapplication.commands.*
monitor.syncExec(new Runnable() {
    @Override
    public void run() {
        OpenSVGGraph.OpenSVGFile("/tmp/test.svg", "histogram plot")
    }
});
```

- first copy the script into a Text Editor (from the menu 'File / New file');
- then select the script's text;
- lastly, launch 'Groovy / Run the text selection' from within the context menu of the Text Editor.

If the Text Editor is open on a file from the folder '$HOME/scripts/user' that has a '.groovy' extension, then you can run the script directly through the Editor's context menu: 'Groovy / Run the Script'.

15.3.5.3 Run the R script and display the histogram using one single Groovy script

It is possible to combine together the R script and the SVG display script into one single Groovy script:

- copy the following Groovy script into a Text Editor that is open on a file from the folder '$HOME/scripts/user' which has a '.groovy' extension:

```groovy
import org.txm.rcpapplication.commands.*
import org.txm.stat.engine.r.RWorkspace
```
def r = RWorkspace.getRWorkspaceInstance()

r.eval(""
##### beginning of the R script #####

svg("/tmp/test.svg")
barplot(t(IndexN$data), space=c(1,35), horiz=F, las=2, beside=T)
dev.off()

##### end of the R script #####
"")

import org.txm.rcpapplication.commands.*

monitor.syncExec(new Runnable() { @Override public void run() {
  OpenSVGGraph.OpenSVGFile("/tmp/test.svg", "histogram plot")
}});

- replace the name of the symbol 'IndexN' with that of the Index transferred to R (for example 'Index1');
- lastly, launch “Groovy / Run the Script” from the context menu of the Text Editor.

15.3.5.4 Enter the name of the index from within a dialogue box with assistance from a TXM macro

To make updating the name of the object in the script easier, it can be entered into a dialogue box by transforming the script to a TXM macro:

- in a Text Editor opened on the file 'testMacro.groovy' located in the folder '$HOME/scripts/macro', copy the following Groovy script:

```java
import org.kohsuke.args4j.*
import groovy.transform.Field
import org.txm.rcpapplication.swt.widget.parameters.ParametersDialog
```
import org.txm.rcpapplication.commands.*
import org.txm.stat.engine.r.RWorkspace

@Field @Option(name="symbol_name",usage="symbol name of the Index to use", widget="String", required=true, def="Index1")
def symbol_name
if (!ParametersDialog.open(this)) return;

def r = RWorkspace.getRWorkspaceInstance()

r.eval(""
##### beginning of the R script #####

svg("/tmp/test.svg")
barplot(t(${symbol_name}\$data), space=c(1,35), horiz=F, las=2, beside=T)
dev.off()

##### end of the R script #####
"")

monitor.syncExec(new Runnable() {
    @Override
    public void run() {
        OpenSVGGraph.OpenSVGFile("/tmp/test.svg", "histogram plot")
    }
});

- open the 'Macro' view (from the main menu 'View / Views / Macro')
- double-click on the 'Test' file icon in the Macro view to run it:
  - in the dialogue box, enter the desired symbol name into the field 'symbol_name' and click 'Run'.
15.3.5.5 Recovering the name of the index directly from the macro

To go even further to simplify the designation of the index in question, it is possible to obtain the name of the index transferred from within the macro by designating the index in the Corpus view:

– in a Text Editor opened on the file 'testMacro.groovy' located in the folder '$HOME/scripts/macro', copy the following Groovy script:

```groovy
import org.txm.rcpapplication.commands.*
import org.txm.stat.engine.r.RWorkspace
import org.txm.functions.vocabulary.*

if (!corpusViewSelection || !(corpusViewSelection instanceof Vocabulary)) {
    println "Error: this macro should be run with an Index selected"
    return
}
def symbol_name = corpusViewSelection.getSymbol()

def r = RWorkspace.getRWorkspaceInstance()

r.eval(""
##### beginning of the R script #####

svg("/tmp/test.svg")
barplot(t(${symbol_name}\$data), space=c(1,35), horiz=F, las=2, beside=T)
dev.off()

##### end of the R script #####
"")
```

- 156 -
monitor.syncExec(new Runnable() {
    @Override
    public void run() {
        OpenSVGGraph.OpenSVGFile("/tmp/test.svg",
        "histogram plot")
    }
});

– in the Corpus view, select the Index results icon to display its histogram (do not forget to transfer the Index to R beforehand);

– open the 'Macro' view (from the main menu 'View / Views / Macro');

– double-click on the macro file to run it.

To successively launch the same macro on several Index results, a practical setup for the interface is to display the Macro view in the panel on the left along with the Corpus view. For example, the upper half displays the Corpus view and the lower half the Macro view (to move the Macro view, drag its tab to the left-hand panel).

15.3.6 Descriptions of the main TXM objects transferred to R

a) a Lexicon object is transferred in the form of a matrix with a single column (vector) consisting of frequencies, with each row named after the value of the counted property.

b) an Index object generated by a corpus or a sub-corpus is transferred in the form of a list (data.frame) containing an object named data. The data object is a matrix with a single column of frequencies in which each row is named after the value of the counted property or after the concatenation of the values when there are several counted properties (in this case, the values are separated by the underscore character '_').

c) an Index object generated by a partition is transferred in the form of a list containing an object named data. This data object is a matrix made of as many columns of frequencies as there are parts. Each row is named either after the value of the counted property or after the concatenation of the values of these counted properties when there are several (in this case the values are separated by an underscore character: '_'). The first column contains the margin for each row and is named F.

d) a Lexical table object, generated by a partition or a partition index is transferred in the form of a matrix with one column of frequencies for every part. Each row is named either after the value of the counted property or after the concatenation of the values of these counted properties when there are several (in this case the values are separated by an underscore character: '_').

e) a Concordance object is transferred in the form of a list containing objects named data, leftcontext and rightcontext. The data object is a matrix of 4 columns of
strings named refs, leftcontext, keywords and rightcontext. They respectively contain the references, the left-hand context, the pivots and the right-hand context. The leftcontext and rightcontext objects contain the size (in words) for the left and right-hand contexts.

f) a Corpus object is transferred in the form of a list containing core objects that store codes for the word property values of each occurrence, following the order of the texts in the corpus (data), in addition to the dictionaries for these values (wordlex for the dictionary of written forms and idlex for the dictionary of word identifiers).

If the words in the corpus have been annotated, the values of these annotations are stored in as many additional dictionaries as is necessary, for example named enposlex, enlemmalex (for TreeTagger annotations this represents 'language code|property name|lex') or using any other additional annotation names. The data object is a matrix consisting of one column for each of the word properties. The columns are named after the names of properties. The start and end positions of a text are stored in the vectors $struct$text$start and $struct$text$end for the R object generated.

16 Import modules for textual sources

The import modules made available in the TXM desktop version are located in the folder 'scripts/import' in the TXM work folder ($USER_HOME/TXM). For the moment, only the launch scripts for each module (the xxxLoader.groovy files) can be accessed by the user.

16.1 Metadata file 'metadata.csv'

Modules named 'XXX+CSV' are modules that can link metadata defined in a CSV file with each text in the corpus. This file must be in the following format:

• the file is named 'metadata.csv';
• the field delimiter is ',', '
• the text delimiter is '"';
• the character encoding must be UTF-8;
• the first row of headers is used to name each metadata;
• the first cell of the first row must contain 'id' (in lowercase characters) - this is mandatory. This cell declares the 'id' metadata that name each text file, excluding their extensions;
• the rest of the first row cells declare other metadata and can be freely named, but must respect certain constraints:

---

45 The CSV files can be edited and exported using Calc or Excel spreadsheet programs.

16.1.1 'metadata.csv' sample file

Here you can see the first three rows of the file 'metadata.csv' for the DISCOURS corpus.

"id","loc","type","date"
01_DeGaulle,de Gaulle,Allocution radiotélévisée,27/06/1958
02_DeGaulle,de Gaulle,Allocution radiotélévisée,28/12/1958
03_DeGaulle,de Gaulle,Allocution radiotélévisée,30/01/1959

In order to link the metadata declared in this file with the texts - defined in the corpus source folder - the first three texts must be represented respectively by files named '01_DeGaulle.cnr', '02_DeGaulle.cnr' and '03_DeGaulle.cnr' (this corpus is imported using the 'CNR+CSV' module).

16.2 Clipboard module

16.2.1 Input

This module imports plain text copied to the system's clipboard. The property 'lb' is added to words to encode the number of the line.

16.2.2 Output

A text structure (text) is obtained as output, along with words tokenized by delimiters.

16.2.3 Annotation

Morphosyntactic annotations and the lemma are added using TreeTagger. The language model used by TreeTagger is specified through the general Settings menu > 'Default language'.

To modify the default annotation language for clipboard importation:
- go to the settings page, follow 'Settings > Import';
- enter the language code in the 'Default language' field. For example, 'en' for English, 'fr' for French;
- click 'OK'.

16.2.4 Edition
An edition of the text is created that includes the management of spaces and punctuation between words, and that is paginated by blocks of n words.

16.3 TXT+CSV module

16.3.1 Input

Text body
This module imports a folder of files containing all types of text (in plain text format). The corresponding default file extension is '.txt'.

Line breaks are interpreted and the line number for each word is encoded using the property 'lbid'.

Text metadata
Metadata for the text is encoded as a CSV formatted file named 'metadata.csv', which is located in the same folder as the source files.
The field delimiter is ',', The text delimiter is '"'.

The first column must be named 'id'. The others can be named at the user's discretion, though no special or accented characters may be used.
The first column must contain the name of the source file (excluding its extension) that corresponds to the metadata in the row.

16.3.2 Output

Text structures (text) are obtained as output, with properties corresponding to the metadata, in addition to words tokenized by delimiters.

16.3.3 Annotation

Morphosyntactic annotations and the lemma are added using TreeTagger.
16.3.4 Edition
There is one edition per text, paginated by blocks of n words. The first page of the edition for each text documents a list of metadata.

16.4 CWB module

16.4.1 Input
This module directly imports a WTC or VRT file\textsuperscript{48}. If a 'registry' file is present, it will be analysed to obtain the properties for the corpus.

16.4.2 Output
An 'id' is added to each word.

16.4.3 Edition
The edition produced is very simple. It is formatted in the following ways:

- a line break at the end of each 'p' paragraph structure
- a page break if an 'lb' or 'br' structure is encountered.

16.5 XML/w+CSV module

16.5.1 Input
Text body
This module imports a file folder\textsuperscript{49} containing text in XML format. By default, the corresponding file extension is '.xml'.
The tag <text> is reserved for this module. If words are delimited by <w> tags possibly containing attributes, they are interpreted as such. It is nevertheless necessary to pay attention that all the <w> tags have the same attribute names.


\textsuperscript{49}
Text metadata
Metadata for the text is encoded as a CSV formatted file named 'metadata.csv', which is located in the same folder as the source files.
The field delimiter is ',', . The text delimiter is '"'.

Additional parameters
The file 'import.properties' located in the source files folder allows you to configure the following parameters:
• stopifmalformed: interrupts the importation if one of the XML files is malformed.
• ignoredelements: regular expression for tag names that the tokenizer should ignore (the 'outside text/out-of-text' references). For example: 'note|teiHeader'
• normalizemetadata: normalises the values of structural properties = true/false. The text's metadata values will be converted to lowercase letters.
• sortmetadata: name of the metadata used to define the order of the texts

'import.properties' file example:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>stopifmalformed</td>
<td>false</td>
</tr>
<tr>
<td>ignoredelements</td>
<td>note</td>
</tr>
<tr>
<td>normalizemetadata</td>
<td>true</td>
</tr>
<tr>
<td>sortmetadata</td>
<td>true</td>
</tr>
</tbody>
</table>

16.5.2 Output
Each XML tag encodes a structural level using its properties.
If the <w> tags have a 'ref' attribute, then this will be used for concordance references.

16.5.3 Edition

WARNING
The contents of this section is temporary and will be noticeably different to the TXM versions 0.7.8 and onwards.

There is one edition created per text, paginated by default in blocks of n words.

16.5.3.1 Using XML components to build the edition
This import module interprets certain XML tags to format the HTML editions:
• Text tags
  ◦ produce a h3 title with the contents of the @id attribute at the beginning of the first page of the edition
  ◦ followed by a table of metadata present in the text: name, value
imposes a line break after the table

• **Head tags**
  ◦ produce a h2 title with the contents of the element

• **Note tags**
  ◦ produce a footnote with a span containing the text '[*]' in red that displays a hoverbox containing the contents of the w/form sub-elements and of text outside <w> elements
  ◦ by default, the contents of the notes (the hoverbox contents) is tokenized and indexed by the search engine (but it is impossible to return to the text from the concordances). To avoid indexing the contents of the notes, it is necessary to add the <note> tag to the ignored elements importation parameter (see section 16.5.1, Input).

• **Graphic tags**
  ◦ produce a div element containing an img element that has the attribute @src with the value of the @url attribute for the <graphic> tag. If the @url attribute is not present, then this information is ignored.

• **lg, p and q tags**
  ◦ produce a p paragraph that has the @class CSS attribute with the value of the @rend attribute from the original element. If the original element does not have an @rend attribute present, then the paragraph will have no particular style.

• **lb and br tags**
  ◦ force a br line break

• **pb tag** (or the pagination tag indicated by the editionpage import parameter)
  ◦ closes the current page and opens a new edition page. If this element appears within the embedded elements, then these elements will be closed before closing the current page, then reopened with the new page.
  ◦ numbers the new page with a p paragraph centred at the top of the page. The '-' n '-' format for the page number is shown in red. 'N' equals the value of the @n attribute of the pb element. If the pb element has no @n attribute, then an automatic page number is posted, staring at 1.

### 16.5.3.2 Styling the editions with CSS

Each HTML page of the edition declares a CSS style sheet named 'mycorpus.css' ('mycorpus' signifies the name of the imported corpus in lowercase).

You can therefore create a personalised style sheet 'mycorpus.css' to be placed in the folder $HOME/corpora/MYCORPUS/html/default, so that all the HTML pages of the edition can benefit from it.

This CSS style sheet determines the overall style of each HTML element or the classes that will be associated with the different types of lg, p and q elements (see the section above).
16.6 XML-PPS

16.6.1 Input

This module uses files produced by exporting XML from the Factiva portal as input\(^{50}\). The files are processed, then imported using the XML/w+CSV module.

The processing procedure used for PPS files has been proposed by Daniel Marin and Florent Bédécarrats.

16.7 Transcriber+CSV module

16.7.1 Input

Text body

This module uses a folder of transcripts in XML-TRS format (with the extension '.trs') as input. To be validated, they must be accompanied by the file 'trans-14.dtd'. Each transcript is considered to be a documentary unit or text.

The transcripts must follow the specifications defined here (only French version available):
http://sourceforge.net/projects/txm/files/documentation/Guide_de_Transcription_d_entretiens_Transcriber-TXM_0.2_FR.pdf/download

Text metadata

Metadata from texts is encoded in a CSV formatted file named '<f0>metadata.csv</f0>', located in the same folder as the source files.

The field delimiter is ',', The text delimiter\(^{51}\) is '"'.

The first column must be named 'id'. The others can be named at the user's discretion, though no special or accented characters may be used.

The first column must contain the name of the source file (excluding its extension) that corresponds to the metadata in the row.

Metadata will be inserted into each transcription, if present.

Additional parameters

This module uses a parameters file called “import.properties” located in the same folder as the transcriptions.

It enables three parameters to be configured:

\(^{50}\) See http://www.factiva.com.

\(^{51}\) The text delimiter allows you to frame complex values containing, notably, spaces or characters that usually delimit fields.
removeInterviewer: has a value of either 'true' or 'false', which indicates whether an interviewer's words should be ignored. The interviewers in each text are defined in the 'enqN' columns (N represents a number) in the metadata.csv file;

- metadataList: the list of metadata. Each metadata is separated from the next by the '|' character, eg. title|date|place

- csvHeaderNumber the number of header rows in the CSV file (if it exists):
  - 1 = there are only the metadata identifiers;
  - 2 = there is one row of identifiers and one row of extended identifiers;
  - 3 = there is a row of identifiers, a row of extended identifiers and the type of metadata.

16.7.2 Output

The XML file structure for Transcriber is reproduced:

- a Transcriber section corresponds to a div structure;
- a speech turn corresponds to a 'sp' structure;
- a speech segment corresponds to a 'u' structure (signifying 'utterance' in TEI).

Two forms of Transcriber events are managed:
- occasional: comments, murmurs;
- across the word span: pronunciation, incertitudes, etc.

Descriptions associated with occasional events are encoded in the subsequent word. For events that span, the descriptions are concatenated as part of the 'event' lexical property, for words between 'start' and 'end' events.

Certain metadata is duplicated at the word level (spk) and structural level (u@spkattrs, textAttr@<metadata>, div@topic@endtime@starttime@type, sp@speaker@endtime@starttime@overlap, event@type@desc) to facilitate sub-corpora construction.

16.7.3 Annotation

Morphosyntactic annotations and the lemma are added using TreeTagger.

16.7.4 Edition

The edition reproduces the same edition as Transcriber. The list of corresponding metadata can be found at the beginning of each text (or transcript).

---

52 This latter information is not used in this version of the software.

53 For transcripts in French, users are advised to use the TreeTagger language model, developed for both written and oral texts under the framework of the PERCEO project <http://www.cnrtl.fr/corpus/perceo>.
16.8 XML-TEI BFM module

16.8.1 Input

The input format is defined by the encoding documentation for the Base de Français Médiéval (BFM) (Old French Base). Each text is represented as a XML TEI P5 formatted file that encodes both the body of the texts and their metadata.

In addition to the text files, a parameters file called 'import.properties' contains the XPath\textsuperscript{54} expression that serves to extract textual metadata contained in their TEI header. Here is an example of the contents of this file:

\begin{quote}
\end{quote}

Gloss:
- the value of the 'title' metadata corresponds to the contents of a \texttt{<title>} element located in a specific place in the TEI header, with its 'type' attribute being equal to 'reference';
- the value of the 'author' metadata corresponds to the contents of an \texttt{<author>} element located in a specific place in the TEI header.

During importation, each word is equipped with a 'ref' property that is used, when calculating concordances, to display a default reference. This property is built using various information:
- the text's identifier, extracted from the attribute \texttt{text@sigle}
- the paragraph number, extracted from the attribute \texttt{p@n}
- and, if the text contains verses, then the verse number extracted from the attribute \texttt{lb@n}

The \texttt{text@sigle} attribute will be built during the 'import' stage of importation by taking information, by order of preference, from:
- the 'idbfm' property declared in the 'import.properties' file
- the 'sigle' property declared in the 'import.properties' file
- the XML file name

\textsuperscript{54} XML Path Language (Xpath) 2.0: \url{http://www.w3.org/TR/xpath20}. 

- 166 -
Pay attention, as the import module does not manage the TEI tags div1, div2, etc. You must replace them with div tags, either directly in the XML source files, or by using a XSL input sheet.

For more information about encoding BFM texts:
- XML-TEI encoding manual for BFM texts: http://bfm.ens-lyon.fr/article.php3?id_article=158
- Text Encoding Initiative Consortium: http://www.tei-c.org

16.8.2 Annotation
Morphosyntactic annotations are added by TreeTagger, using the 'fro.par' linguistic model. The tagset used by this model is CATTEX2009 (see http://bfm.ens-lyon.fr/article.php3?id_article=176).

16.8.3 Edition
The edition for the texts is quite similar to that created for the 'Queste del Saint Graal' project (see http://portal.textometrie.org/txm). However, this part of the module will eventually be replaced by XSLT+CSS style sheets, created by Alexis Lavrentiev, in order to produce an equivalent, yet permanent edition.

16.9 XML-TEI Frantext Module
This module is similar to the XML-TEI BFM module. Before launching the XML-TEI BFM importation, an XSL sheet pre-processes the source files:
- the <br/> element is recoded as <lb/>
- starred words are re-encoded as <w type="caps">…</w>
- the TEI header is corrected, particularly as the <auteur> element should be <author>
- <seg> elements are re-encoded as <w>…</w>

16.10 XML-TMX

16.10.1 Input
This module uses a TMX formatted XML file as input.

16.10.2 Output
The module builds a corpus for each language, and each corpus is aligned with the others.

16.10.3 Edition

The result is a simple edition (there is no editorial information in the TMX format).

16.11 XML-TXM module

16.11.1 Input

This module imports texts directly in XML-TXM pivot format\(^{56}\). This module does not produce tokenizations as the XML-TXM format already encodes the words with \(<w>\) elements.

16.11.2 Output

Metadata for texts is generated by properties encoded in the headers of the texts. Intra-textual structures and their properties are generated by all the XML elements not included in the XML-TXM schema. Words and their properties are generated by \(<w>\) elements.

16.11.3 Annotation

There are no annotations added to this module.

16.11.4 Edition

There is one edition per text, paginated by blocks of n words.

16.12 CNR+CSV module

16.12.1 Input

Text body

The texts are Cordial CNR formatted files, in other words, a TSV file with a tab character as the field delimiter and without a text delimiter. The columns of the CNR files, in order, are:

- para: paragraph number
- sent: sentence number
- form: written form of a lexical unit
- lem: lemma
- pos: morphosyntactic property
- func: syntactic function


- 168 -
Text metadata
Metadata for the text is encoded as a CSV formatted file named 'metadata.csv', which is located in the same folder as the source files. The field delimiter is ',', The text delimiter57 is '"'. The top header row labels each metadata column. The first column must be named 'id'. The others can be named at the user's discretion, though no special or accented characters may be used. The first column must contain the name of the source file (excluding its extension) that corresponds to the metadata in the row.

16.12.2 Output
Structures for the paragraphs (p), sentences (s) and texts (text) are obtained as output. Words are equipped with all the properties that correspond to the CNR columns.

16.12.3 Annotation
There are no annotations added to this importation, it does not need TreeTagger.

16.12.4 Edition
The edition pages are paginated by blocks of n words. The first page of the edition for each text documents a list of metadata read in the CSV file.

16.13 Alceste module

16.13.1 Input
This module uses a file in Alcest format as input (the same format as IRaMuTeQ - http://www.iramuteq.org). This is a plain text format that uses several simple encoding conventions.
There are two ways you can choose to encode the start of the text (which corresponds with the initial context unit 'icu' in the d+Alceste terminology) and its metadata:

1. one line for the form: 0001 *Meta1_Val1 *Meta2_Val2... *MetaN_ValN
2. one line for the form: **** *Meta1_Val1 *Meta2_Val2... *MetaN_ValN

*Meta_val, declares the metadata of a text with 'Meta' being the name and 'val' the value. If the metadata of the text does not have a value, you can use the following notation: *Meta.
For TXM, attribute names are composed solely of letters without any special cases (all letters in lowercase) and with no diacritics (no accents).

57 The text delimiter allows you to frame complex values containing, notably, spaces or characters that usually delimit fields.
To pre-encode a compound word, you can replace the spaces between its constituents with a '‐' character. For example, 'human being' can be encoded/tokenised as one word: 'human_being'.

The Alceste format also offers a way of encoding sections from within the uci, sections characterised by an asterisked variable (notation: -*Meta_Val as a single line), but this TXM import module does not yet handle this.

16.13.2 Output
Text structures (text) and words segmented by separator characters are obtained as output.

16.13.3 Annotation
Morphosyntactic annotations and the lemma are added using TreeTagger.

16.13.4 Edition
There is one edition per text, paginated by blocks of n words.

16.14 Hyperbase module

16.14.1 Input
This module uses a file formatted for the older version of Hyperbase as input. This means the separator lines for the text appear in the following form:

```
&&& Long text name, TextName, ShortName &&&
```

Page break lines (encoded by '\\$') are interpreted. They are then re-encoded as p structures.

16.14.2 Annotation
Morphosyntactic annotations and the lemma are added using TreeTagger.

16.14.3 Edition
There is one edition per text, paginated by blocks of n words.
16.15 Factiva TXT module

This module converts source files from the Mail by Factiva exportation format to the Alceste format, then applies the Alceste import module.

16.15.1 Input

This module uses the Mail by Factiva exportation format as input, while following the exportation recommendations for Factiva, conceived by Pierre Ratinaud and Lucie Loubere:

- conduct a classic search;
- once you have selected your articles, in 'View Options' located above them, select 'Full article and indexing';
- request to see your articles;
- copy all contents to a .txt document.

16.16 Factiva XML module

This module uses the Factiva XML format. This format is no longer available for Factiva's educational and research licences.

16.16.1 Input

The module begins by restructuring the information using the header so that it can be exploited by CQP.

The module finishes by applying an XML/w+CSV import module to the XML files generated.
17 Keyboard Shortcuts

17.1 Results tables

<table>
<thead>
<tr>
<th>Commands</th>
<th>Shortcuts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search a character string</td>
<td>Ctrl+F</td>
</tr>
<tr>
<td>Copy selected rows to the clipboard</td>
<td>Ctrl+C</td>
</tr>
</tbody>
</table>

17.1 Text Editor

<table>
<thead>
<tr>
<th>Commands</th>
<th>Shortcuts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Help</td>
<td></td>
</tr>
<tr>
<td>Display available Shortcut Keys</td>
<td>Ctrl+Shift+L</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Select</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Select All</td>
<td>Ctrl+A</td>
</tr>
<tr>
<td>Select to start of line</td>
<td>Shift+Home</td>
</tr>
<tr>
<td>Select to end of line</td>
<td>Shift+End</td>
</tr>
<tr>
<td>Select word to the right</td>
<td>Ctrl+Shift+Right</td>
</tr>
<tr>
<td>Select word to the left</td>
<td>Ctrl+Shift+Left</td>
</tr>
</tbody>
</table>
**Edit**

Copy
Ctrl+C, Ctrl+Insert

Paste
Ctrl+V, Shift+Insert

Cut
Ctrl+X, Shift+Suppr

Delete
Delete

Undo
Ctrl+Z

Redo
Ctrl+Y

Set to Uppercase
Ctrl+Shift+X

Set to Lowercase
Ctrl+Shift+Y

**Find**

Find and Replace
Ctrl+F

Find Next
Ctrl+K

Find Previous
Ctrl+Shift+K

Open Incremental Search
Ctrl+J

Quit Incremental Search
Ctrl+Shift+J

**Move**

Start of text
Ctrl+Home

End of text
Ctrl+End

Start of line
Home

End of line
End
Next Word Ctrl+Right
Previous Word Ctrl+Left
Go to line Ctrl+L
Last Edition Location Ctrl+Q

**Delete**
Delete line Ctrl+D
Delete to end of line Ctrl+Shift+Delete
Delete to the right Ctrl+Delete
Delete to the left Ctrl+Backspace

**Move Lines**
Move Current Line Up Alt+Up
Move Current Line Down Alt+Down

**Insert Line**
Insert Line above Current Line Ctrl+Shift+Enter
Insert Line Below Current Line Shift+Enter

**Other**
Merge Lines Ctrl+Alt+J
Scroll Lines Up Ctrl+Up
Scroll Lines Down Ctrl+Down
Duplicate Lines Ctrl+Alt+Up
Copy Lines Ctrl+Alt+Down

Toggle folding Ctrl+Numpad_Divide

Mode
Toggle insert mode Ctrl+Shift+Insert
Toggle overwrite mode Insert

Toggle selection mode Alt+Shift+A
Toggle quick comparison Ctrl+Shift+Right

Show ruler context menu Ctrl+F10

File
New Ctrl+N
Save Ctrl+S
Close Ctrl+W, Ctrl+F4
Close All Ctrl+Shift+W
Print Ctrl+P
Properties
Alt+Enter

Refresh
F5

Miscellaneous
Word Completion
Alt+/ 

Scripts
Run Selected Text
F11
Run Script
Ctrl+F11
Re-Run Last Executed Script or Macro
F12

17.2 Graphic Shortcuts
Move View
Right-click and move mouse or keyboard arrows
Change Scale
Use either the mouse wheel or Ctrl + and Ctrl - on the numeric keypad

17.3 Windows
Manage Windows
Next window
Ctrl+F6
Previous window
Ctrl+Shift+F6
Window's drop-down menu
Ctrl+E
Return to Window Menu
Ctrl+Shift+E
Displays the Window's menu
Alt+-
Views

Maximise current window  
Next View  
Previous View  
Display View Menu  
Display available Shortcut Keys  
Display Console View

<table>
<thead>
<tr>
<th>Shortcut</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctrl+M</td>
<td>Maximise current window</td>
</tr>
<tr>
<td>Ctrl+F7</td>
<td>Next View</td>
</tr>
<tr>
<td>Ctrl+Shift+F7</td>
<td>Previous View</td>
</tr>
<tr>
<td>Ctrl+F10</td>
<td>Display View Menu</td>
</tr>
<tr>
<td>Ctrl+Shift+L</td>
<td>Display available Shortcut Keys</td>
</tr>
<tr>
<td>Alt+Shift+Q, C</td>
<td>Display Console View</td>
</tr>
</tbody>
</table>

17.4 Miscellaneous Shortcuts

- Run last executed script  
- Quit TXM

<table>
<thead>
<tr>
<th>Shortcut</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>F12</td>
<td>Run last executed script</td>
</tr>
<tr>
<td>Ctrl+Q</td>
<td>Quit TXM</td>
</tr>
</tbody>
</table>

18 TXM Settings

18.1 TXM

- Export settings as a file: saves TXM settings as a file;  
- Import from a file: imports settings previously exported back into TXM.

18.2 Advanced Settings

The following settings configure the launching of TXM and should be manipulated with caution.

- TXM Installation folder: folder containing files essential for launching TXM;  
- Users folder: TXM work folder that notably contains a user's settings and corpora;
Degree of detail for log: enables more or fewer messages to be displayed in the Console;

Display run time: displays the run time for each command (Concordances, Index, etc.);

Add technical comments: displays internal TXM messages in the Console;

Copy log to a file: saves the contents of the Console as a file in the user's folder under the name 'TXM (date)'.

18.2.1 Corpus Engine

Use network protocol: enforces socket communication between TXM and CQP;

Path to 'cqpjni' library: do not modify (expert configuration only);

Remote CWB server: to connect TXM to a remote CQP server (see the following settings);

Machine name: remote server network address;

Port number for CWB server: communication port to be used with remote server (set to 4877 by default);

Login for CWB server: username on remote server (set to 'anonymous' by default);

Password for CWB server: password for remote server;

Path to executable file 'cqpserver': the executable cqpserver file can be located in the TXM installation folders (for a local network connection);

Path to Registry folder: 'registry' reference folder in the TXM user's folder;

Path to cqpserver set up file: cqpserver configuration file, used at the very beginning when launching the cqpserver. Useful for changing the environment variables of the cqpserver;

Additional Options: options to add to the cqpserver command line. For example, the option '-b 1000000' allows you to define the maximum size of an unbound CQL expression.

18.2.2 Graphics Engine

Current graphics engine: defines the engine to be used to render graphical images (R/SVG or Java)
18.2.3 Statistics engine

- Remote connection: connects TXM to a remote R server;
- Path to R executable file: path to R software (for a local connection);
- R data transmission by file: to be used in the event of R communication problems (for a local connection);
- Server address: for a remote connection;
- User: username for the remote R server;
- Password: user password.

18.2.4 NLP/TreeTagger

- TreeTagger installation folder path: unzipped TreeTagger folder;
- Folder pathway for TreeTagger's language models: folder containing TreeTagger's language models, unzipped and renamed following the ISO 639 standard ('en.par', 'fr.par', etc.);
- Options [separated by 2 blank spaces]: options to be added to the TreeTagger command line. These options are separated by 2 blank spaces to allow TXM to create a list. For example: "-arg value -arg2 value2".

18.3 User

General settings for the user interface:

- No confirmation before deleting an object: suspends the request for confirmation before deleting an object from the Corpus view.

18.3.1 Clustering

- Number of clusters: number of clusters to be calculated;
- Method: algorithm for constructing clusters;
- Metric: distance to be used by the algorithm;
- Display the graphs in 2D or 3D: display or do not display clusters in the same graph as the AFC factorial planes.
18.3.2 Concordances

- Line per page: number of lines displayed per page of concordances;
- Left Context (in words): number of words in the left context;
- Right Context (in words): number of words in the right context.

18.3.3 Co-occurrences

- Score format: configures the viewing format for the specificities score;
- Minimum frequency threshold for the co-occurent: minimum frequency for a word to be included in the calculation of co-occurrences;
- Minimum frequency threshold for co-occurrences: minimum frequency of encounters of a word with a pivot for it to be included in the calculation of the co-occurrences;
- Minimum score threshold for the co-occurrences: threshold under which the co-occurent is not included in the results list;
- Minimum to the left: closest distance for the left-side co-occurrences;
- Maximum to the left: furthest distance for the left-side co-occurrences;
- Minimum to the right: closest distance for the right-side co-occurrences;
- Maximum to the right: furthest distance for the right-side co-occurrences.

18.3.4 Description

- Number of property values displayed: maximum number of different values to be displayed per property;
- Order the parts by size: organises parts in descending order according to the number of occurrences in the partition description;
- Display the number of parts in the bar plot title: displays the number of parts in the title of the partition dimension bar plot.

18.3.5 File browser

Display settings for the file browser:

- Regular expression for hidden files: filter enabling files with a name that corresponds to a regular expression not to be listed (especially useful for masking files beginning with '.' in Linux);
18.3.6 Exporting

Settings for all TXM exports:

- **Display hidden files**: displays those files the system considers 'hidden'.

**Exporting**

**18.3.6 Exporting**

Settings for all TXM exports:

- **Exported encoding files**: character encoding table to be used for exportation. The 'UTF-8' value is recommended as it is the most universal;

- **Field delimiter**: a character used as a field delimiter for exports in CSV format (set to ',' by default). This character varies according to the spreadsheet software and the operating system. Even though 'CSV' may be the acronym for 'Comma Separated Values', in this role the comma (',') is often instead replaced by a semi-colon (';'), or sometimes the tab character ('→').

- **Text delimiter**: a character used to delimit the text values for exports in CSV format (set to '' by default).

- **Display export results in a text editor**: if the export is carried out correctly, the results are displayed in a new TXM Text Editor window.

- **Default export file format for R graphs**: format used for exporting graphs.

**18.3.7 Importation**

Settings for reading the 'metadata.csv' file by the import modules applying it (TXT+CSV, XML/w+CSV, etc.):

- **Encoding**: character encoding table to be used. Set to 'UTF-8' by default;

- **Field delimiter**: a character used as a field separator. Set to ',' by default;

- **Text delimiter**: a character used as a text field delimiter for values. Set to '' by default.

**18.3.8 Progression**

Default values for the Progression window settings:

- **Progression graph**: type of graph to be generated (set to 'cumulative' by default);

- **Greyscale**: produces a graph in shades of grey instead of colour (for certain publications);

- **Different line styles**: enables a different style of line to be used for each curve (useful for greyscale graphs);
– Repeat the property values of structures: structural limits sharing the same property value are not displayed, by default;

– Scale of structural limits: number that configures the edge of the structural limits in the density graph.

18.3.9 References

– Order references by frequency: orders references by frequency rather than alphabetical order.

18.3.10 Scripts

– Next session number: R scripts are saved as a file that uses the session number as a suffix in its name. This number is incremented each time the script is run;

– Root folder: folder where TXM is authorised to run scripts.

18.3.11 Specificities

– Banality: banality threshold for graphs;

– Regroupe bars by lines in the table: transfers the lexical table before producing the graph;

– Greyscale: produces a graph in shades of grey instead of colour (for certain publications);

– Show lines: produces a graph using lines;

– Show bars: produces a bar plot.

18.3.12 Lexical Table

– Minimum frequency: minimum frequency of entries composing a lexical table, created during an intermediary calculation;

18.3.13 Correspondence Analysis

– Quality column format: configures the display format for the values of representation quality on the plane. See section 18.3.13.1;

– Contribution column format: configures the display format for the values of contribution to the axes;

– Mass column format: configures the display format for the mass values;
Distance column format: configures the display format for the distance values;
Cos² column format: configures the display format for the cos² values;
Coord column format: configures the display format for the coordinates values;
Show columns: displays the column points (or variables);
Show rows: displays the row points (or individuals).

18.3.13.1 Defining the display formatting for real or whole numbers

In the results table, the numbers can be formatted according to a pattern defined by the following special characters:

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>represents a compulsory digit that must be present, even if it is a seemingly unimportant zero</td>
</tr>
<tr>
<td>#</td>
<td>represents a number, however can ignore unimportant zeros</td>
</tr>
<tr>
<td></td>
<td>represents the separator for the decimal part</td>
</tr>
<tr>
<td>,</td>
<td>represents the group separator (thousands, millions, etc.)</td>
</tr>
</tbody>
</table>

Table 7: Formatting characters for numbers

The number '0' or '#' in the pattern defines the size of the whole and decimal parts of the numerical value, given that 0 will represent a compulsory digit that must be present (and eventually replaced by an unimportant zero) and # an optional figure (that ignores unimportant zeros). Formatting examples:

<table>
<thead>
<tr>
<th>Format</th>
<th>0</th>
<th>0.02</th>
<th>0.8</th>
<th>12.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>13</td>
</tr>
<tr>
<td>#,#</td>
<td>0</td>
<td>0.02</td>
<td>0.8</td>
<td>12.9</td>
</tr>
<tr>
<td>0,#</td>
<td>0</td>
<td>0.02</td>
<td>0.8</td>
<td>12.9</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.02</td>
<td>0.80</td>
<td>12.90</td>
</tr>
<tr>
<td>#.00</td>
<td>.00</td>
<td>.02</td>
<td>.80</td>
<td>12.90</td>
</tr>
<tr>
<td>#,.00</td>
<td>0.00</td>
<td>0.02</td>
<td>0.80</td>
<td>12.90</td>
</tr>
</tbody>
</table>

Table 8: Examples of number formats

---

58 According to [http://java.developpez.com/faq/java/?page=langage_chaine#LANGAGE_STRING_nombre_en_chaine_formattee](http://java.developpez.com/faq/java/?page=langage_chaine#LANGAGE_STRING_nombre_en_chaine_formattee)
19 Morphosyntactic tagset used for French.

'frpos' property values for the contemporary French TreeTagger model\(^{59}\).

\(^{59}\) Source: French TreeTagger Part-of-Speech Tags Achim Stein, April 2003 [http://www.ims.uni-stuttgart.de/~schmid/french-tagset.html]
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABR</td>
<td>abbreviation</td>
</tr>
<tr>
<td>ADJ</td>
<td>adjective</td>
</tr>
<tr>
<td>ADV</td>
<td>adverb</td>
</tr>
<tr>
<td>DET:ART</td>
<td>article</td>
</tr>
<tr>
<td>DET:POS</td>
<td>possessive pronoun (ma, ta...)</td>
</tr>
<tr>
<td>INT</td>
<td>interjection</td>
</tr>
<tr>
<td>KON</td>
<td>conjunction</td>
</tr>
<tr>
<td>NAM</td>
<td>proper noun</td>
</tr>
<tr>
<td>NOM</td>
<td>noun</td>
</tr>
<tr>
<td>NUM</td>
<td>numeral</td>
</tr>
<tr>
<td>PRO</td>
<td>pronoun</td>
</tr>
<tr>
<td>PRO:DEM</td>
<td>demonstrative pronoun</td>
</tr>
<tr>
<td>PRO:IND</td>
<td>indefinite pronoun</td>
</tr>
<tr>
<td>PRO:PER</td>
<td>personal pronoun</td>
</tr>
<tr>
<td>PRO:POS</td>
<td>possessive pronoun (mien, tien...)</td>
</tr>
<tr>
<td>PRO:REL</td>
<td>relative pronoun</td>
</tr>
<tr>
<td>PRP</td>
<td>preposition</td>
</tr>
<tr>
<td>PRP:det</td>
<td>preposition plus article (au, du, aux, des)</td>
</tr>
<tr>
<td>PUN</td>
<td>punctuation</td>
</tr>
<tr>
<td>PUN:cit</td>
<td>punctuation quotes</td>
</tr>
<tr>
<td>SENT</td>
<td>sentence tag</td>
</tr>
<tr>
<td>SYM</td>
<td>symbol</td>
</tr>
<tr>
<td>VER:cond</td>
<td>conditional verb</td>
</tr>
<tr>
<td>VER:futu</td>
<td>future verb</td>
</tr>
<tr>
<td>VER:impe</td>
<td>imperative verb</td>
</tr>
<tr>
<td>VER:impf</td>
<td>imperfect verb</td>
</tr>
<tr>
<td>VER:infi</td>
<td>infinitive verb</td>
</tr>
<tr>
<td>VER:pper</td>
<td>past participle verb</td>
</tr>
<tr>
<td>VER:ppre</td>
<td>present participle verb</td>
</tr>
<tr>
<td>VER:pres</td>
<td>simple present verb</td>
</tr>
<tr>
<td>VER:simp</td>
<td>simple past verb</td>
</tr>
<tr>
<td>VER:subi</td>
<td>imperfect subjunctive verb</td>
</tr>
<tr>
<td>VER:subp</td>
<td>present subjunctive verb</td>
</tr>
</tbody>
</table>
20 TXM Glossary

Categories:
- com: Command
- mod: data model
- fmt: file format
- int: user interface
- nlp: Natural Language Processing (NLP)
- exp: search query expression
- sof: software component
- met: Textometry Methodology

<table>
<thead>
<tr>
<th>Input</th>
<th>Cat</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFA</td>
<td>com</td>
<td>the act of reducing the number of dimensions of a matrix (demonstrating the form “parts X words”) using the correspondence factor analysis algorithm (CFA). The new dimensions are represented by eigenvectors called factors. The parts and words from the original matrix can be displayed simultaneously across the resulting factorial planes.</td>
</tr>
<tr>
<td>FRO</td>
<td>nlp</td>
<td>a standard code signifying Old French</td>
</tr>
<tr>
<td>Alceste</td>
<td>soft</td>
<td>a commercial textometry program</td>
</tr>
<tr>
<td>annotation</td>
<td>mod</td>
<td>the property of a unit (either lexical or structural) within a digital text.</td>
</tr>
<tr>
<td>tag</td>
<td>mod</td>
<td>delimited representation of an element that contains its properties in XML language</td>
</tr>
<tr>
<td>character</td>
<td>mod</td>
<td>a basic unit from which the form of a word is constituted.</td>
</tr>
<tr>
<td>CATTEX2009</td>
<td>nlp</td>
<td>a morphosyntactic tagging system used for Old French.</td>
</tr>
<tr>
<td>import module</td>
<td>com</td>
<td>a software component that imports texts from an external source into the TXM platform.</td>
</tr>
<tr>
<td>ClipboardN</td>
<td>int</td>
<td>corpora created using the clipboard are named 'Clip'+&lt;a number&gt;.</td>
</tr>
<tr>
<td>Word</td>
<td>Action</td>
<td>Definition</td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CNR</td>
<td>fmt</td>
<td>data format for Cordial software (file extension .cnr)</td>
</tr>
<tr>
<td>command</td>
<td>com</td>
<td>an action made available in TXM</td>
</tr>
<tr>
<td>concordance</td>
<td>com</td>
<td>a way of presenting search results, whereby each occurrence appears in one central column, with its surrounding context displayed to the right and left.</td>
</tr>
<tr>
<td>console</td>
<td>int</td>
<td>while running TXM displays various messages in a window entitled “Console”.</td>
</tr>
<tr>
<td>Cordial</td>
<td>nlp</td>
<td>a commercial lemmatizer and morphosyntactic tagger</td>
</tr>
<tr>
<td>corpus</td>
<td>mod</td>
<td>set of words. These collections/sets come from whole or partial texts (à REFAIRE§§§). The 'root' corpora are constructed using these bases/foundations.</td>
</tr>
<tr>
<td>CQL</td>
<td>exp</td>
<td>stands for &lt;Corpus Query Language&gt;, a query language managed by CQP and applied to the corpora.</td>
</tr>
<tr>
<td>CQP</td>
<td>soft</td>
<td>stands for &lt;Corpus Query Processor&gt;, a software component that manages queries in order to build indexes, concordances, etc.</td>
</tr>
<tr>
<td>CSV</td>
<td>fmt</td>
<td>stands for &lt;Comma Separated Values&gt;. A text file where each result line is separated by a line break and the values are separated by a delimiter character (such as a comma).</td>
</tr>
<tr>
<td>Ctrl</td>
<td>int</td>
<td>the 'Ctrl' or 'Control' key on the keyboard.</td>
</tr>
<tr>
<td>document</td>
<td>mod</td>
<td>a digital text</td>
</tr>
<tr>
<td>editor</td>
<td>com</td>
<td>a window in which a text (such as a source file or script) can be modified.</td>
</tr>
<tr>
<td>encoding</td>
<td>mod</td>
<td>how information is represented in a digital text.</td>
</tr>
<tr>
<td>workspace</td>
<td>int</td>
<td>all the objects (corpora, sub-corpora...) available in TXM.</td>
</tr>
<tr>
<td>tagger</td>
<td>soft</td>
<td>independent software capable of segmenting words within textual sources, and associating a morphosyntactic tag or lemme to them.</td>
</tr>
<tr>
<td>tag</td>
<td>nlp</td>
<td>the morphosyntactic property of a word</td>
</tr>
</tbody>
</table>
### TXM User Manual 0.7 - June 2015

<table>
<thead>
<tr>
<th>Term</th>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>to export</td>
<td>com</td>
<td>the act of saving the results of a TXM command in an external file.</td>
</tr>
<tr>
<td>file</td>
<td>mod</td>
<td>a component of the operating system that contains information on the user's hard drive, for example a source text. A file may be identified by an access path.</td>
</tr>
<tr>
<td>focus</td>
<td>int</td>
<td>to focus a command on a specific lexical event, for example through a query.</td>
</tr>
<tr>
<td>graphic form</td>
<td>mod</td>
<td>the graphic representation of a word, generally calculated by tokenizers.</td>
</tr>
<tr>
<td>frequency</td>
<td>met</td>
<td>total number of occurrences of an event (an occurrence of word or a sequence of words, etc.)</td>
</tr>
<tr>
<td>Groovy</td>
<td>soft</td>
<td>computing language used to write TXM scripts.</td>
</tr>
<tr>
<td>HTML</td>
<td>fmt</td>
<td>presentation format of web page data</td>
</tr>
<tr>
<td>Hyperbase</td>
<td>soft</td>
<td>non-commercial textometry software.</td>
</tr>
<tr>
<td>to import</td>
<td>mod</td>
<td>act of integrating a corpus into the platform, from a source file.</td>
</tr>
<tr>
<td>index</td>
<td>com</td>
<td>lists all the combinations of word properties, along with their frequency, for all the occurrences of a query.</td>
</tr>
<tr>
<td>index</td>
<td>soft</td>
<td>a file created by TXM in order to accelerate the processing of query results.</td>
</tr>
<tr>
<td>score</td>
<td>nlp</td>
<td>a numerical value provided by a statistical model</td>
</tr>
<tr>
<td>info-bubble</td>
<td>int</td>
<td>temporary window that appears when the mouse cursor hovers over an object, for example a word in a text edition.</td>
</tr>
<tr>
<td>Java</td>
<td>soft</td>
<td>the computing language in which TXM is coded.</td>
</tr>
<tr>
<td>tagset</td>
<td>mod</td>
<td>the set of possible morphosyntactic values belonging to words</td>
</tr>
<tr>
<td>language</td>
<td>mod</td>
<td>the language in which a text or corpus is written.</td>
</tr>
<tr>
<td>lem</td>
<td>mod</td>
<td>see lemma.</td>
</tr>
<tr>
<td>lemma</td>
<td>mod</td>
<td>the standard dictionary entry of a word</td>
</tr>
<tr>
<td>Term</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>lemmatizer</td>
<td>soft</td>
<td>software component that matches each word in the text to a dictionary entry</td>
</tr>
<tr>
<td>lexicon</td>
<td>com</td>
<td>a list of all the possible word forms in a corpus or a frequency list of word properties</td>
</tr>
<tr>
<td>status line</td>
<td>soft</td>
<td>TXM temporarily displays comments about operations it conducts in a space located at the bottom left-hand side of the interface.</td>
</tr>
<tr>
<td>literal</td>
<td>exp</td>
<td>a character that matches its graphic form (is not considered to be a special character), often in a query</td>
</tr>
<tr>
<td>localisation</td>
<td>int</td>
<td>the TXM interface can be displayed in different languages, which can be set up through the File menu, 'Change language'.</td>
</tr>
<tr>
<td>to match</td>
<td>nlp</td>
<td>to establish structural correspondence within the algebra of property characters or occurrences.</td>
</tr>
<tr>
<td>metadata</td>
<td>mod</td>
<td>the properties of a whole text or a document. Each metadata has a name, type and value.</td>
</tr>
<tr>
<td>modifier</td>
<td>exp</td>
<td>a special character used to express certain variants in a query.</td>
</tr>
<tr>
<td>word</td>
<td>mod</td>
<td>a lexical unit identified by its graphical form and its position in a sequence of words. It is generally constructed by tokenizers.</td>
</tr>
<tr>
<td>Multext</td>
<td>nlp</td>
<td>standard European tagset</td>
</tr>
<tr>
<td>occurrence</td>
<td>met</td>
<td>the appearance of a textual event in a corpus, such as the occurrence of a word.</td>
</tr>
<tr>
<td>operator</td>
<td>exp</td>
<td>special character or keyword that plays a particular role in a query.取自</td>
</tr>
<tr>
<td>page</td>
<td>mod</td>
<td>a segment of text displayed on a support that generally corresponds to the page of a paper edition.</td>
</tr>
<tr>
<td>part</td>
<td>mod</td>
<td>a part is one element of a partition.</td>
</tr>
<tr>
<td>partition</td>
<td>mod</td>
<td>the division of a corpus into different parts. The sum of all these parts is equal to the corpus as a whole. The partitions are used to analyse contrasting parts (for example between authors, sections of the same text, discourse dates, etc.)</td>
</tr>
<tr>
<td>Term</td>
<td>Context</td>
<td>Definition</td>
</tr>
<tr>
<td>------------</td>
<td>---------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>sentence</td>
<td>nlp</td>
<td>a syntactically homogeneous sequence of words constructed using tokenizers</td>
</tr>
<tr>
<td>pivot</td>
<td>com</td>
<td>the concordance tool's central column that displays all the occurrences of a given query.</td>
</tr>
<tr>
<td>pos</td>
<td>mod</td>
<td>stands for 'part of speech', a word's morphosyntactic information</td>
</tr>
<tr>
<td>setting</td>
<td>int</td>
<td>each command in TXM has its own parameters. Some of these can be modified in the 'Settings' window.</td>
</tr>
<tr>
<td>clipboard</td>
<td>mod</td>
<td>a function of the operating system that allows a text selection to be stored, by using the 'copy' command.</td>
</tr>
<tr>
<td>Property</td>
<td>mod</td>
<td>information pertaining to a lexical or structural unit</td>
</tr>
<tr>
<td>reference</td>
<td>int</td>
<td>information displayed at the beginning of a concordance line which stems from the properties of lexical and structural units.</td>
</tr>
<tr>
<td>directory</td>
<td>mod</td>
<td>a folder containing files or other folders located on the user's hard-drive. A directory may be identified through an access path.</td>
</tr>
<tr>
<td>query</td>
<td>com</td>
<td>character string that expresses a combination of words and word properties to be searched.</td>
</tr>
<tr>
<td>script</td>
<td>soft</td>
<td>file containing a description of specific actions that can be executed by TXM</td>
</tr>
<tr>
<td>selection</td>
<td>met</td>
<td>a list of sequences of words. A selection is the outcome that results from a search query.</td>
</tr>
<tr>
<td>source</td>
<td>mod</td>
<td>the initial representation of a corpus in a set format, contained in several files or folders. For example, the format may be TXT (plain text file), XML or TEI.</td>
</tr>
<tr>
<td>specificity</td>
<td>com</td>
<td>act of listing specific forms of words, or word properties, in each partition part, in accordance with the quantitative model of specificities</td>
</tr>
<tr>
<td>T</td>
<td>met</td>
<td>the total number of occurrences in a corpus</td>
</tr>
<tr>
<td>NLP</td>
<td>soft</td>
<td>stands for 'Natural Language Processing'</td>
</tr>
<tr>
<td>TEI</td>
<td>fmt</td>
<td>stands for 'Text Encoding Initiative', the standard method of</td>
</tr>
<tr>
<td>term</td>
<td>type</td>
<td>description</td>
</tr>
<tr>
<td>--------------</td>
<td>------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>text</td>
<td>mod</td>
<td>a homogeneous sequence of words, described using properties called metadata.</td>
</tr>
<tr>
<td>textometry</td>
<td>met</td>
<td>the methodology implemented by TXM. Textometry assists by using quantitative and qualitative tools to analyse textual corpora. See <a href="http://textometrie.ens-lyon.fr">http://textometrie.ens-lyon.fr</a>.</td>
</tr>
<tr>
<td>tokenizer</td>
<td>soft</td>
<td>a software component capable of separating words and characterising them by their properties into source files</td>
</tr>
<tr>
<td>TreeTagger</td>
<td>soft</td>
<td>independent non-commercial tagging software</td>
</tr>
<tr>
<td>TXT</td>
<td>fmt</td>
<td>data format of a plain text file (without any annotation).</td>
</tr>
<tr>
<td>unit</td>
<td>mod</td>
<td>structural or lexical unit of a text</td>
</tr>
<tr>
<td>structural unit</td>
<td>mod</td>
<td>an element marking the logical structure of a text. In TXM all structural units are organised in a hierarchical order: each unit nests within another unit – right up to the 'text' unit. The smallest structural unit is located just above the lexical unit.</td>
</tr>
<tr>
<td>V</td>
<td>met</td>
<td>the total number of diverse graphic forms in a corpus</td>
</tr>
<tr>
<td>vocabulary</td>
<td>com</td>
<td>générer un lexique ou un index.</td>
</tr>
<tr>
<td>Weblex</td>
<td>soft</td>
<td>non-commercial textometry software</td>
</tr>
<tr>
<td>window manager</td>
<td>int</td>
<td>component that helps organise the user's work interface</td>
</tr>
<tr>
<td>XML</td>
<td>fmt</td>
<td>the principal data format (file extension) for corpora source texts.</td>
</tr>
</tbody>
</table>
21 Bibliography (not complete)


22 Index

Image index
Image 2.1: Security window .............................................................................................................. 15
Image 2.2: Installation folder ........................................................................................................... 16
Image 2.3: Error opening file 'R.dll' .............................................................................................. 16
Image 2.4: Finalising the installation ............................................................................................. 17
Image 2.5: Homepage ....................................................................................................................... 18
Image 2.6: Installation drive ............................................................................................................ 18
Image 2.7: Installation ..................................................................................................................... 18
Image 2.8: Authorisation .................................................................................................................. 18
Image 2.9: Finalising the installation ............................................................................................. 19
Image 2.10: Opening with Gdebi .................................................................................................... 21
Image 2.11: Accepting the TXM licence ......................................................................................... 22
Image 2.12: TXM will install the R statistical libraries ................................................................. 23
Image 2.13: Installation progress window ..................................................................................... 23
Image 2.14: Finalising the installation ........................................................................................... 24
Image 3.1: Updated versions ........................................................................................................... 25
Image 3.2: Chosen update level ...................................................................................................... 26
Image 3.3: Step 1: Available updates ............................................................................................. 27
Image 3.4: Step 2: Update details .................................................................................................... 27
Image 3.5: Step 3: Agreeing to the distribution licenses ............................................................... 28
Image 3.6: Step 4: Downloading updates ....................................................................................... 28
Image 3.7: Step 5: Final security confirmation before installation ................................................ 29
Image 3.8: Step 6: Restart TXM to apply updates .......................................................................... 29
Image 4.1: List of available extensions ............................................................................................ 30
Image 5.1: Installing third-party extensions ................................................................................... 31
Image 10.1: The standard TXM interface ....................................................................................... 39
Image 10.2: The Browser ................................................................................................................ 40
Image 10.3: The Corpus view ......................................................................................................... 41
Image 10.4: The File view .............................................................................................................. 42
Image 10.5: The Toolbars .............................................................................................................. 44
Image 10.6: The File menu .............................................................................................................. 45
Image 10.7: The Corpus menu shows commands for the corpus (left) and commands for the partition (right) ........................................................................................................... 45
Image 10.8: The Tools menu for corpora (left) and for partitions (right) ........................................ 46
Image 10.9: Corpus context menu .................................................................................................. 47
Image 10.10: Example of results windows .................................................................................... 54
Image 10.11: The messages ............................................................................................................ 55
Image 10.12 The Text Editor Toolbar ............................................................................................ 57
Image 10.13: Import parameters form ............................................................................................ 61
Image 10.14: TreeTagger Settings .................................................................................................. 68
Image 11.1: DISCOURS corpus description ............................................................................................................. 72
Image 11.2: Edition of the DISCOURS corpus ............................................................................................................ 74
Image 11.3: Window for browsing within the editions of various parts of the partition ........................................... 75
Image 11.4: 'Simple' mode: constructing a sub-corpus using all De Gaulle's speeches ........................................... 75
Image 11.5: 'Assisted' mode: creating a sub-corpus using Pompidou's televised radio interviews .......................... 78
Image 11.6: 'Advanced' mode: constructing a sub-corpus based on Pompidou's speeches dating back to 1970 ............. 79
Image 11.7: Using simple mode to build a partition based on dates ........................................................................ 80
Image 11.8: Assisted mode: building a partition based on the dates of the DISCOURS corpus ................................. 82
Image 11.9: Building a partition for each president for the year 1970 .................................................................... 83
Image 11.10: The Concordance window .................................................................................................................. 84
Image 11.11: Constructing a query based on the word 'je' followed by a verb ......................................................... 86
Image 11.12: Concordance of the word 'je' followed by a verb in the DISCOURS corpus ........................................... 88
Image 11.13: 'References pattern' dialogue box ...................................................................................................... 90
Image 11.14: Co-occurrences of words beginning with 'j' ......................................................................................... 91
Image 11.15: Hierarchy list of the written forms of words from the DISCOURS corpus ........................................... 93
Image 11.16: Index command window ................................................................................................................... 94
Image 11.17: Selection window for word properties .................................................................................................. 94
Image 11.18: Index formed using the 'word' and 'pos' properties for the lemma 'pouvoir', in the DISCOURS corpus ........................................................................................................ 96
Equation 11.19: Maximum likelihood/probability of occurrence in a part ............................................................. 98
Equation 11.20: Probability for occurrence in a part ............................................................................................... 99
Equation 11.21: Specificity score .......................................................................................................................... 99
Image 11.22: Macro parameters for the ExecR example ............................................................................................. 100
Image 11.23: Probability distribution of the specificity for parameters no. 296, 1084 and 61449 .............................. 100
Image 11.24: Specificity window in a partition .......................................................................................................... 101
Image 11.25: Specificities for words in a partition based on the text property (or variable) named 'type' in the DISCOURS corpus ..................................................................................................... 104
Image 11.26: Specificity graph of lemmas 'je', 'nous' and 'vous' based on three types of speeches found in the DISCOURS corpus ........................................................................................................ 105
Image 11.27: Specificities of written forms from the 'Allocution radiotélévisée' part of the DISCOURS corpus ......... 106
Image 11.28: Progression graph calculation for the words 'France' and 'Algérie' in the speeches of De Gaulle and Pompidou ........................................................................................................... 108
Image 11.29: Cumulative progression graph depicting the words France and Algérie in the speeches of De Gaulle and Pompidou ........................................................................................................... 109
Image 11.30: CFA obtained through a lexical table based on 'Dates' from the DISCOURS corpus .......................... 111
Image 11.31: Example of highlighting points using the multiple selection tool in a CFA, created using a partition based on the presidents in the VOEUX corpus ......................................................... 113
Image 11.32: Property of the lexical table ................................................................................................................ 114
Image 11.33: Lexical table for the partition based on Dates in the DISCOURS corpus ........................................... 115
Index of tables
Table 1: Metacharacters (or Operators).................................................................135
Table 2: Quantifiers..........................................................................................135
Table 3: Character codes..................................................................................136
Table 4: Character classifications.................................................................136
Table 5: Common Unicode Classes...............................................................137
Table 6: Common POSIX classes (an older and more basic system than the Unicode classes).................................................................137
Table 7: Formatting characters for numbers.................................................187
Table 8: Examples of number formats.........................................................187

Updates Table (in date order)

<table>
<thead>
<tr>
<th>Date</th>
<th>Author</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>13/03/10</td>
<td>Serge Heiden (SH)</td>
<td>Creation</td>
</tr>
<tr>
<td>02/07/10</td>
<td>Matthieu Decorde (MD)</td>
<td>Update for version 0.4.7</td>
</tr>
<tr>
<td>15-29/07/10</td>
<td>SH</td>
<td>Rewrite for version 0.4.7</td>
</tr>
<tr>
<td>27/08/10</td>
<td>SH</td>
<td>Numbering of headings, reorganisation of layout</td>
</tr>
<tr>
<td>08/10/10</td>
<td>Lauranne Bertrand (LB)</td>
<td>French version of the manual for 0.5</td>
</tr>
<tr>
<td>05/01/11</td>
<td>MD</td>
<td>Update for the released version 0.5</td>
</tr>
<tr>
<td>14/01/11</td>
<td>SH</td>
<td>Corrections</td>
</tr>
<tr>
<td>18/01/11</td>
<td>SH</td>
<td>Corrections, additions to the section on import modules</td>
</tr>
<tr>
<td>06/06/11</td>
<td>MD</td>
<td>Copy and update for version 0.6</td>
</tr>
<tr>
<td>09/04/12</td>
<td>SH</td>
<td>Update 0.6: garde, tdm</td>
</tr>
<tr>
<td>17/06/12</td>
<td>SH</td>
<td>Update 0.6 preface, tdm</td>
</tr>
<tr>
<td>17/07/12</td>
<td>MD</td>
<td>Copy and update for version 0.7</td>
</tr>
<tr>
<td>Date</td>
<td>Author(s)</td>
<td>Notes</td>
</tr>
<tr>
<td>------------</td>
<td>-----------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>13/03/10</td>
<td>SH</td>
<td>Creation</td>
</tr>
<tr>
<td>06/08/12</td>
<td>SH</td>
<td>Update, save, preface, Windows, Mac and Linux installation instructions</td>
</tr>
<tr>
<td>29-30/09/12</td>
<td>SH</td>
<td>Addition of 'French tagset' and 'regular expressions' sections. Section entitled 'In the event of a problem' moved. Corrections to 'Settings' section.</td>
</tr>
<tr>
<td>05/03/2013</td>
<td>SH</td>
<td>Addition of 'temporary' stamp Replaced section 6 with &quot;Mémo CQL&quot; by B. Pincemin</td>
</tr>
<tr>
<td>07/03/2013</td>
<td>SH</td>
<td>Light revision carried out on the import modules section</td>
</tr>
<tr>
<td>17/05/2013</td>
<td>SH</td>
<td>Update to regexp section and cqpman.pdf link</td>
</tr>
<tr>
<td>08/11/2013</td>
<td>MD</td>
<td>Added 'Macros', 'Automatic updates' and 'Installing extensions' sections</td>
</tr>
<tr>
<td>08/11/2013</td>
<td>BP</td>
<td>Update to 'Import Alceste' section</td>
</tr>
<tr>
<td>19/11/2013</td>
<td>MD</td>
<td>Addition to 'Importing Factiva XML' section</td>
</tr>
<tr>
<td>28/11/2013</td>
<td>SH</td>
<td>Finalisation of sections 'Automatic updates', 'Adding extensions', 'Macros', settings and 'Import Alceste'</td>
</tr>
<tr>
<td>10/12/2013</td>
<td>SH</td>
<td>Minor adjustments made to sections: 'Automatic updates', 'Installing an extension', 'In the event of a problem'</td>
</tr>
<tr>
<td>16/12/2013</td>
<td>SH</td>
<td>Rewrite of the section 'Driving the TXM software with macros and Groovy scripts or R code', added to the section 'Run R code within TXM'.</td>
</tr>
<tr>
<td>06/01/2014</td>
<td>SH</td>
<td>Update and reformatting section 1 and partial rewrite section 6.3</td>
</tr>
<tr>
<td>Date</td>
<td>Author</td>
<td>Changes/Notes</td>
</tr>
<tr>
<td>----------</td>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>13/03/10</td>
<td>Serge Heiden (SH)</td>
<td>Creation</td>
</tr>
<tr>
<td>06/02/2014</td>
<td>SH</td>
<td>Adjustment to sections on the updates and extensions, TOC and indexes updates.</td>
</tr>
<tr>
<td>14/04/2014</td>
<td>SH</td>
<td>Update of running title</td>
</tr>
<tr>
<td>06/05/2014</td>
<td>SH</td>
<td>Updates to Mac OS X installation and Recommended settings</td>
</tr>
<tr>
<td>19/06/2014</td>
<td>SH</td>
<td>Update to section 11.12 CFA</td>
</tr>
<tr>
<td>04/07/2014</td>
<td>Sébastien Jacquot (SJ)</td>
<td>Updates to TXM 0.7.6: graphics</td>
</tr>
<tr>
<td>22/07/2014</td>
<td>MD</td>
<td>Updates to TXM 0.7.6: see tickets #740, #747, #762, #777, #779, #782, #799, #914, #920</td>
</tr>
<tr>
<td>17/09/2014</td>
<td>MD</td>
<td>Update for ticket #1013</td>
</tr>
<tr>
<td>20/10/2014</td>
<td>SH</td>
<td>Additions to section 16.5.3.2 (CSS of the editions)</td>
</tr>
<tr>
<td>24/10/2014</td>
<td>SH</td>
<td>Accepted all SJ and MD modifications, began to rewrite import section</td>
</tr>
<tr>
<td>30/10/2014</td>
<td>SH</td>
<td>Transferred ‘Specificity’ section elements from Weblex manual</td>
</tr>
<tr>
<td>06/11/2014</td>
<td>SH</td>
<td>Updated installation sections</td>
</tr>
</tbody>
</table>