RcmdrPlugin.temis, a Graphical Integrated Text Mining Solution in R

by Milan Bouchet-Valat and Gilles Bastin

Abstract We present the package **RcmdrPlugin.temis**, a graphical user interface for user-friendly text mining in R. Built as a plug-in to the R Commander provided by the **Rcmdr** package, it brings together several existing packages and provides new features streamlining the process of importing, managing and analyzing a corpus, in addition to saving results and plots to a report file. Beyond common file formats, automated import of corpora from the Dow Jones Factiva content provider and from Twitter is supported. Featured analyses include vocabulary and dissimilarity tables, terms frequencies, terms specific of levels of a variable, term co-occurrences, time series, correspondence analysis and hierarchical clustering.

Text mining applications have become quite popular in the social sciences – in particular in sociology and political science – fostered by the availability of integrated graphical user interfaces (GUIs) that render typical analyses possible at a reasonably low learning cost. R has been technically well positioned in the area of text mining for several years, thanks to the **tm** package (Feinerer et al., 2008; Feinerer and Hornik, 2013) and to its rich ecosystem of packages dedicated to advanced text mining operations[1], but also to general purpose packages whose power can be leveraged for this particular application. Yet, the power and the flexibility of this environment present the downside that non-technical users can feel lost, especially when they are used to GUIs that do not provide them – to say the least – with so many possibilities.

In this article, we present **RcmdrPlugin.temis** ("R.TeMiS", for R Text Mining Solution, in short) version 0.6.1 (Bouchet-Valat and Bastin, 2013), a new package that intends to fill this gap by providing users with a GUI, in the form of a menu augmenting the R Commander (provided by the **Rcmdr** package, *cf.* Fox, 2005). We hope to provide an integrated and easy to use graphical solution that does not constrain the user into a specific path of analysis, as an alternative to proprietary software packages that currently dominate the market, often with a closed-source conception of what users should (and can) do, and what they should not.

Following the spirit of free software and scientific research, we intend to empower users and not lock them in a jointly technical and theoretical, rigorously defined paradigm. Though, even if we are open to different approaches of text mining, the current features that we demonstrate below are largely based on a long-established tradition of exploratory data analysis; the interested reader will find a detailed description of these techniques in *e.g.* Lebart et al. (1998).

Design choices

The fundamental choice behind the design of **RcmdrPlugin.temis** is that it is fully integrated into – and actually written in – R. Following the general principle adopted by the R Commander, every action initiated by the user via the GUI generates R code which is shown to the user while it is executed (Figure 1), and which can be saved and run again later. This makes it possible to manually edit the code, be it to customize a function call or to reuse objects to extend the analysis without starting from scratch; this also ensures the reproducibility of the results, a major issue in scientific research.

Thus, **RcmdrPlugin.temis** differs from other text mining GUIs, which do not generate source code, and provide little space for extending the set of possible analyses. In particular, two free software text mining tools offer a relatively similar set of features: Iramuteq (Ratinaud, 2013) and TXM (Heiden, 2010)[2]. These two projects use R as a back end for some analyses, but are written as standalone applications completely separated from the R session; this means that the power of R as a general purpose statistical environment is lost.

The decision of writing our package as an R Commander plug-in also presents several other advantages. First, all the features offered by **Rcmdr** are still available to the user: in particular, menus allowing to import and recode data and to export plots can be very useful for text mining work.

[1] See the "Natural Language Processing" CRAN task view at http://CRAN.R-project.org/view=NaturalLanguageProcessing for a list of packages applying to this field.

[2] See respectively http://www.iramuteq.org and http://textometrie.ens-lyon.fr/?lang=en. Interestingly, all of the three projects are developed by French researchers.

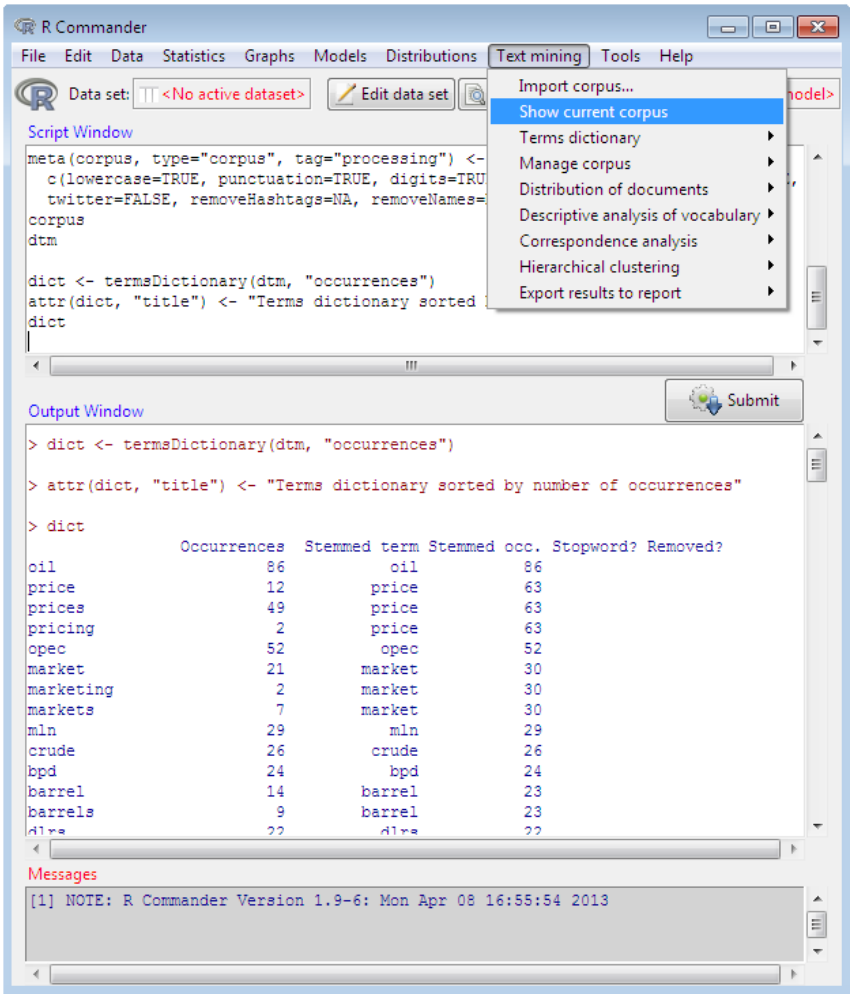

Figure 1: Main R Commander window showing text mining menu and top of the terms dictionary sorted by number of occurrences.

Second, **Rcmdr** runs without too much tweaking on all platforms, since the toolkit it relies on, Tcl/Tk, is built-in standard in R via the **tcltk** package (see the next section for instructions on how to install **RcmdrPlugin.temis**). The downside of this choice is of course that the resulting interface is not fully geared towards text mining, contrary to what a dedicated application would have made possible.

Other general purpose R GUIs such as RKWard (Rödiger et al., 2012) and Deducer (Fellows, 2012) could have been used instead of the R Commander. But at the time the project was started, no Mac OS X port of the former existed, and the latter was not even released yet (for a discussion of the different R GUIs see Valero-Mora and Ledesma, 2012, and the individual articles of that special issue). The choice of the hosting GUI for a text mining plug-in is in the end closely tied to the question of the best R GUI: our choice has been to rely on the most mature one[3].

Under the hood, **RcmdrPlugin.temis** puts together state-of-the-art R packages covering various types of applications and makes them easily accessible to perform text mining tasks. At the core of the text mining operation is the **tm** package, which provides the framework to import corpora from various sources, process texts and extract their vocabulary in the form of a *document-term matrix*. All other packages on which **RcmdrPlugin.temis** depends are plugged into this framework to provide specific features. We illustrate them below.

[3]The GUI itself represents only a share of the work required to create an integrated text mining solution: this means that plug-ins for other general purpose GUIs could be written without too much effort in the future by adapting **RcmdrPlugin.temis**'s code base.

Installing and launching the package

Like any CRAN package, **RcmdrPlugin.temis** and its mandatory dependencies can be installed from the repositories using the R command

```
install.packages("RcmdrPlugin.temis")
```

or from graphical menus provided by the R GUI you are using. Mac OS X users will need to install the Tcl/Tk libraries before running this command; bundles can be found at http://cran.r-project.org/bin/macosx/tools/. In case of problems, more details concerning that platform, as well as Windows and Linux, can be found on the R Commander web page: http://socserv.mcmaster.ca/jfox/Misc/Rcmdr/installation-notes.html.

Once installed, the package can be be loaded with the command:

```
library("RcmdrPlugin.temis")
```

(or, again, using menus).

Importing and managing corpora

The first task that **RcmdrPlugin.temis** intends to allow is the seamless importation of a corpus – an often neglected task which can prove quite time-consuming in typical research work. The program will ask to automatically install the packages needed to import corpora from the chosen source at the first attempt to do so[4]: this applies to all sources except plain text and comma/tab-separated values (CSV/TSV) files. Four types of sources are currently supported:

- A series of plain text files (typically .txt) contained in a directory.
- A spreadsheet-like file (.csv, .ods, .xls, ...) with one line per document (answers to an open-ended question, typically). The first column must contain the text to analyze; the remaining columns can provide information about the document and are automatically imported as variables. Open Document Spreadsheet files (.ods) are loaded using the **ROpenOffice** package (Temple Lang, 2011a)[5], and Microsoft Excel files (.xls, .xlsx) are imported using the **RODBC** package (Ripley and Lapsley, 2012)[6].
- One or several .xml or .html files exported from the Dow Jones Factiva content provider, using the dedicated **tm.plugin.factiva** package (Bouchet-Valat, 2013); using this portal, often available via academic subscriptions, streamlines the process of creating a corpus by allowing the mass importation of press articles from many newspapers, together with meta-data information like origin, date, author, topics and geographic coverage.
- A Twitter search (thanks to the **twitteR** package, *cf.* Gentry, 2013) on hash tags, authors or full text, or more complex criteria supported by the Twitter API.

By default, plain text and CSV/TSV files are assumed to use the native system encoding: this is usually UTF-8 (Unicode) on Mac OS X and Linux, and varies depending on the locale on Windows. An indication that this default choice is not appropriate is the presence of incorrect accentuated characters in the corpus, but it may also happen that the import fails, or does not detect documents and variables. In that case, you will need to specify the encoding of the file by choosing it in the drop-down list provided by the import dialog (Figure 2). If such problems persist, opening the relevant files in a text editor or spreadsheet application and saving them in a known encoding is a reliable solution[7].

The import dialog also offers text processing options. First, texts can be split into smaller chunks defined as a number of adjacent paragraphs, which is most useful for relatively long texts, for example press articles imported from a Factiva source; in that case, each chunk will be considered as a document in subsequent analyses, and a meta-data variable will identify the original text from which the chunk originates. Second, texts can be processed to make them more suitable for computer analysis: conversion to lower case, punctuation removal, stopwords removal, and finally stemming

[4] This operation requires a working Internet connection.

[5] This package will only be automatically installed by the GUI on Linux at the time of writing, since no binary packages are available. Users familiar with building R packages from source can install the package manually, but this requires tools which are typically not installed by most users on Windows and Mac OS X.

[6] This package will only be automatically installed by the GUI on Windows, since **RODBC** relies on an Excel ODBC driver which is installed by default on Windows but not on other platforms. Mac OS X users can install such a driver and the package manually. New solutions might be investigated in the future.

[7] For example, OpenOffice/LibreOffice Writer and Calc applications offer this option – called "charset" – when "Edit filter options" is checked in the save dialog.

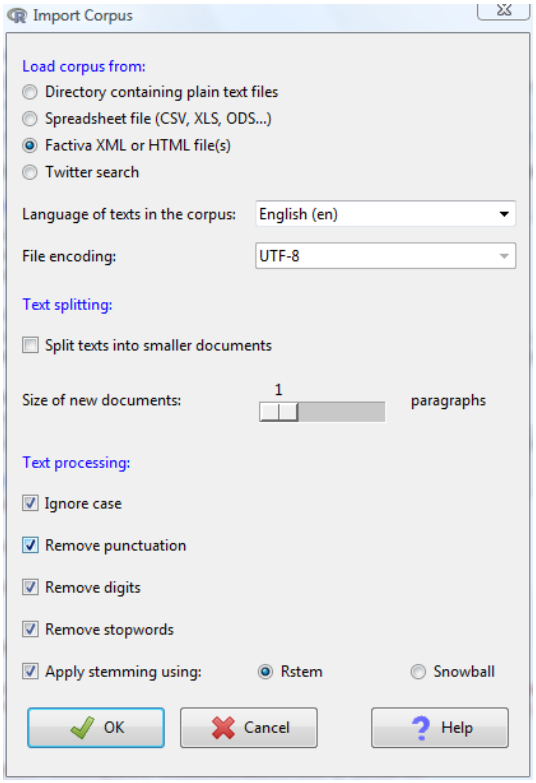

Figure 2: The import dialog with the settings used in the presented example.

can be carried out. Stopwords removal is disabled by default, since we consider that it should only be performed based on a conscious decision made by the researcher: in many cases, at least some stopwords are of interest for the analysis.

Stemming is performed using either the **Rstem** (Temple Lang, 2011b) or the **Snowball** (Hornik, 2012) packages, which implement Porter's algorithm for English, and support several other languages[8]. This processing option is most interesting for short texts because small grammatical variations can dramatically reduce the number of (exact) co-occurrences between two texts, thus hindering later analyses of between-document relationships. After this step, the document-term matrix is created, which will be the basis for most later work; a summary of the matrix is printed on the screen.

Once the corpus has been imported, additional meta-data variables can be loaded from the active data set, be it imported from an external file or filled from R itself, using the '`Manage corpus` → `Set corpus variables`' menu item. The variables will then be made available for all analyses we present below.

The dictionary of terms that are present as columns of the document-term matrix and will be the basis of analyses can be altered in two ways:

- If specific terms are known (or found out later) to disturb the analyses – as commonly happens with meta-vocabulary like "read" and "page" in press articles – they can be excluded.
- If only some terms known in advance are of interest, the matrix can be restricted to them.

Finally, documents can be managed in the same way, by only keeping those containing (or *not* containing) some terms, or those corresponding to a given level of a variable. As said earlier, documents can refer to text chunks if this option was checked during the import, in which case this feature can be used to create a thematic corpus from large and heterogeneous original texts, thanks to a simple selection of chunks according to a set of terms, or to hierarchical clustering. In any case, the original corpus can be saved and restored at will.

For demonstration purposes, we use in the rest of this presentation a small corpus of 19 documents dealing with crude oil which appeared in the Reuters newswire in 1987, and is included in the classic

[8] At the time of writing, languages supported by both packages are Danish, Dutch, English, Finnish, French, German, Norwegian, Portuguese, Russian, Spanish and Swedish. **Snowball** additionally supports Hungarian and Italian, but requires Java via the **rJava** framework (Urbanek, 2013), which more often requires manual system configuration than **Rstem**.

Reuters 21578 data set (already used in **tm** examples). This small set of articles provides a good illustration of the typical analysis of a press corpus[9]. The corpus is shipped with the **tm.plugin.factiva** package in the Factiva XML format with its meta-data; the location of this file on your system can be found using the following command provided you have installed the package:

```
system.file("texts", "reut21578-factiva.xml",
    package = "tm.plugin.factiva")
```

You can also download it directly from: http://r-temis.r-forge.r-project.org/texts/reuters21578-factiva.xml.

To import the file as a corpus, use the 'Text mining → Import corpus...' menu, select 'Factiva XML or HTML file(s)' for source, choose "English (en)" as the language of the corpus, and enable stopwords removal by checking the corresponding box; we leave other options to their default values (Figure 2). Then click 'OK': dialogs will allow you to select the file ('reut21578-factiva.xml') and the variables to retain (we will use "Date", but you may also retain other variables), and the import phase will be run. You can then check the raw contents (*i.e.* not affected by processing operations) of the corpus using the 'Show current corpus' menu item, and see the corpus' words, their stemmed form and their number of occurrences using the 'Terms dictionary → Sorted by number of occurrences' menu (Figure 1).

Elementary statistics

Because **RcmdrPlugin.temis** aims at providing users with a complete text mining solution, it provides dialog boxes to compute simple statistics not directly related to lexical analysis, but that we consider useful to explore the composition of a corpus. The 'Distribution of documents' sub-menu allows creating one- and two-way tables from meta-data variables, and optionally plots the result. This sub-menu also makes it possible to plot time series representing the number of documents over time, using a single curve, or one curve for each level of a variable; as an option, the rolling mean can be computed over a configurable time window. This feature is based on packages **zoo** (Zeileis and Grothendieck, 2005) for computations and **lattice** (Sarkar, 2008) for plotting, which means the generated code can easily be extended by users for custom representations if needed. It is especially useful for corpora imported directly from Factiva or Twitter, which contain date and time information that can prove of high value for the study of the timing of media cycles.

The 'Descriptive analysis of vocabulary' menu introduces us to tasks more oriented towards text analysis *per se*. As a first preview of our corpus, we can print the most frequent terms and their number of occurrences, for the whole corpus as well as for each level of a variable.

A slightly more sophisticated information is provided by the 'Terms specific of levels...' dialog: this feature reports terms whose observed frequency in each level is either too high or too low compared to what would be expected given the documents' lengths and the global distribution of terms in the corpus. p-values based on an hypergeometric distribution give the probability of observing such an extreme number of occurrences in the level under the independence hypothesis; the sign of t-values can be used to identify positive (printed first) and negative (printed last) associations.

	% Term/Level	% Level/Term	Global %	Level	Global	t value	Prob.
post	2.0	67	0.58	8	12	3.3	0.0004
demand	1.3	71	0.34	5	7	2.7	0.0038
analyst	1.5	60	0.48	6	10	2.6	0.0051
product	1.8	50	0.68	7	14	2.4	0.0091
contract	1.0	67	0.29	4	6	2.2	0.0145
compani	1.3	56	0.43	5	9	2.1	0.0163
crude	2.5	38	1.26	10	26	2.1	0.0172
meet	1.8	44	0.77	7	16	2.0	0.0212
saudi	0.0	0	1.06	0	22	−2.4	0.0089
kuwait	0.0	0	0.92	0	19	−2.1	0.0170

Table 1: Terms specific of the first day with more than 5 occurrences in the corpus.

[9]Even though most techniques make more sense applied to large corpora, this small set of texts is sufficient to illustrate all the steps involved in the analysis of a corpus of any size, with the advantage of easy reproducibility and low computing requirements.

The printed output (*cf.* Table 1) also provides several descriptive statistics: the share of all occurrences of all terms in the level that is accounted for by the corresponding term ("% Term/Level"), the share of the occurrences of the term that happen in the level rather than elsewhere in the corpus ("% Level/Term"), the share of the term in all occurrences of all terms in the whole corpus ("Global %"), and finally the absolute number of occurrences of the term in the level and in the corpus.

In our example, choosing the "Date" variable, we can observe (Table 1, where terms with the lowest p-value and more than 5 occurrences are retained) that articles deal with oil markets on the first day, while Middle East countries ("saudi", "kuwait") are not cited at all, contrary to other days, a difference from the corpus average which is significant at the 5% confidence level.

Other measures can be computed, both by document and by levels of a variable: vocabulary summary (number of terms, diversity and complexity); frequency and specificity of chosen terms; co-occurring terms (that is, terms that tend to appear in the same documents as chosen terms); dissimilarity table (Chi-squared distance between row profiles of the document-term matrix collapsed according to the levels of a variable).

Hierarchical clustering and correspondence analysis

The power of R and of its ecosystem is clearly visible when implementing more sophisticated techniques: from the document-term matrix created using **tm**, **RcmdrPlugin.temis** easily performs a hierarchical clustering of the documents using function '`hclust()`' from the **base** package, and a correspondence analysis (CA) using the **ca** package (Nenadic and Greenacre, 2007). These two features are often used together in text mining software, but are relatively hard to run manually for beginners; the possibility to only plot a subset of points was also missing from all CA packages available in R, a feature particularly required for text mining where terms are too many to be all shown at the same time.

For both techniques, a dialog allows eliminating from the document-term matrix *sparse terms*, *i.e.* terms that appear in too few documents. Such terms are of little use for methods that intend to group a significant number of documents together based on term co-occurrences, and they can lead to system memory issues with large corpora. Here we use the value of 94%, which means that terms present in 6% of documents and less are not taken into account: this amounts to removing terms present in only one document, reducing the number of terms from 705 to 239 without much loss of information (since terms appearing in only one document provide no information on relations among documents, and can only help singling out outliers).

Hierarchical clustering can be run from the '`Hierarchical clustering` → `Run clustering...`' menu; this command uses Ward's method based on a Chi-squared distance between documents[10].

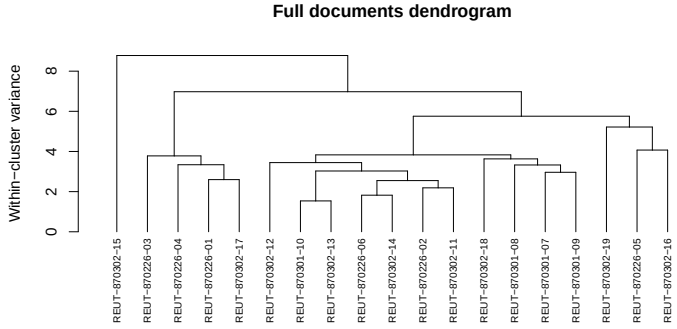

Figure 3: Dendrogram of documents resulting from hierarchical clustering.

The resulting dendrogram is automatically plotted (Figure 3), and its inspection prompts us to choose to create 5 clusters according to a scree test (the reduction in variance is relatively strong when going from 4 to 5 clusters, and slows down afterwards). If we change our mind later, the '`Create clusters...`' menu command allows choosing another number of clusters without running the full computations again. As printed in a separate window, the resulting clusters contain 1, 4, 11, 1 and 2

[10]Since this method tends to start with clusters identifying outliers, which are often of little interest, an alternative solution is also provided: instead of running the clustering procedure on the document-term matrix itself, one can choose to base the analysis on an approximation of the matrix obtained using a given number of axes of the CA (once it has been computed using the corresponding menu). This procedure reduces the noise in the data and allows for a strong relationship between clusters and CA axes. We do not retain this solution here.

documents, respectively; clusters 2 and 3 are more homogeneous (within-cluster variance of 3.8) than cluster 5, despite being the two largest, with respectively 21% and 58% of all documents versus 11%. Terms and documents specific of each cluster are also automatically printed. Specific documents are chosen according to their (as small as possible) Chi-squared distance to the centroid of the cluster, *i.e.* the mean term distribution of the cluster.

We will not detail here the interpretation of the clusters. Let us simply note that if one is interested in reading more associated terms, the '`Create clusters...`' dialog will allow raising the total number of printed terms. Once identified, clusters can be used in all other analyses offered by **RcmdrPlugin.temis**: a meta-data variable called "Cluster" is automatically created. As an illustration, the '`Distribution of documents` → `Two-way table of variables...`' can now be used to cross the "Date" variable with the "Cluster" one, and see that cluster 3 gains momentum in March compared to February.

When dealing with a larger corpus with date or time information, the '`Temporal evolution...`' item available from the same sub-menu is of great interest to help visualize the variations of clusters over time. Further, the "Cluster" variable can be introduced in a CA as a supplementary variable, which can help identifying the meaning of clusters themselves: this is what we will now present.

But first, as the only article in cluster 1 appears to be very different from the rest of the corpus, we may want to exclude it before running the CA: indeed, if we retain it, we can notice that this article alone draws the opposition on several dimensions. We thus use the '`Manage corpus` → `Subset corpus` → `By variable...`' menu, choosing the newly created "Cluster" variable, and selecting only the levels 2, 3, 4 and 5 (keep the Ctrl key pressed to select several items in the box using the mouse).

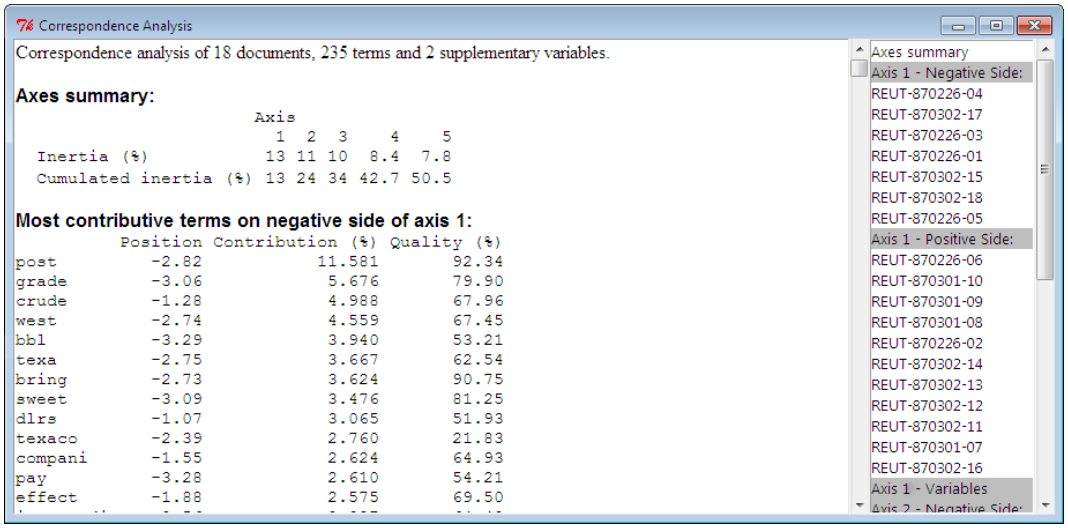

Figure 4: Information about the CA axes, and most contributive terms on the negative side of axis 1.

The CA can be run from '`Correspondence analysis` → `Run analysis...`': again, we exclude terms missing from more than 94% of documents. The number of dimensions to retain only affects the display of axes, and does not change the results: we can safely keep the default value. After running the CA, a dialog box opens automatically to show the results. It offers a few options that apply both to the text window that opens, similar as for the cluster analysis, and to the plot which is drawn at the same time. Among the most important options is the option which allows to select the number of terms and documents shown: as with clustering, only the most contributive items are shown; one can choose to show items most contributive to both axes taken together, or only to one of them. Here, we choose to print terms and documents most contributive to the first two axes.

The text window (Figure 4) presents axes one by one, first their negative, then their positive side, each with most contributive terms and documents, and finally the situation of supplementary (passive) variables on the axis. It thus constitutes a major help to interpreting the plotted plane (Figure 5).

By ticking the '`Variables`' check box in the '`Draw labels for:`' section, we can study the position of clusters on the CA plane by plotting them as supplementary levels: it is possible to select the "Cluster" variable so that other variables are not drawn (as previously, use the Ctrl key when clicking to select if you want to select several variables in the list).

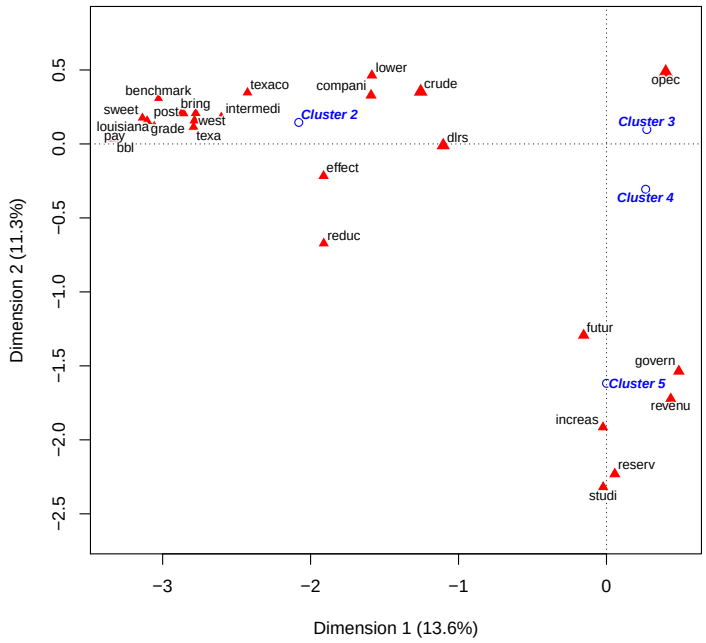

Figure 5: The first plane of the CA with most contributive terms and clusters as supplementary levels.

Exporting results

Since **RcmdrPlugin.temis** aims at providing an integrated text mining solution, we could not leave our users at this stage of the analysis without providing them with a good way of exporting results. The last sub-menu is here precisely for that. After any of the operations described above, it allows copying the last printed table or last drawn plot to an output file in the HTML format, thanks to the **R2HTML** package (Lecoutre, 2003). With this command, plots are saved to a PNG file using a reasonably high resolution in the directory containing the HTML file, and inserted into it. The '`Draw plots in black and white`' menu command converts current and future plots to a mode suitable for printing and publications, using different line types to replace colors[11].

From the HTML file that is opened automatically in the default Web browser, users can copy tables to a word processor without loss of quality. Users looking for high resolution plots will happily use the richer exporting features of the R Commander ('`Graphs` → `Save graph to file`' menu) or of their standard R GUI, in particular to export graphs in vector formats like PDF, PostScript or SVG. Given the low quality of graphics produced by most proprietary text mining software, we have no doubt that R's qualities in that area will be appreciated!

Conclusion

By putting together in a GUI the many existing pieces that already make possible advanced and flexible text mining analyses with R, we hope to fill the needs of new users, who are currently locked using proprietary solutions. We also hope to ease the life of more advanced users who can find in our package a straightforward way of running common tasks before adapting or extending the generated code.

Those looking for more detailed information than was exposed in this introduction about the exact operations run by the GUI are advised to consult **RcmdrPlugin.temis**'s documentation available under the '`Help`' button in all dialog boxes, and to already cited introductions to the **tm** package, as well as online functions documentation.

RcmdrPlugin.temis does not impose on users any canonical model of text mining, but intends to provide users with all available techniques: we are open to suggestions and additions of new features

[11] CA plots, which are not drawn using **lattice**, do not benefit from this feature.

that will help leverage the power of R's ecosystem.

Acknowledgments

The authors would like to thank the maintainers of the packages used by this GUI, in particular Ingo Feinerer, John Fox, Kurt Hornik and Duncan Temple Lang, who have been most helpful by their responses and the fixes they applied to their code to make it suit our needs when necessary, and Bénédicte Garnier and Élisabeth Morand (INED) for their useful remarks and harsh testing.

Bibliography

M. Bouchet-Valat. *tm.plugin.factiva: Import Articles from Factiva Using the tm Text Mining Framework*, 2013. URL http://CRAN.R-project.org/package=tm.plugin.factiva. R package version 1.2. [p3]

M. Bouchet-Valat and G. Bastin. *RcmdrPlugin.temis: Graphical Integrated Text Mining Solution*, 2013. URL http://CRAN.R-project.org/package=RcmdrPlugin.temis. R package version 0.6.1. [p1]

I. Feinerer and K. Hornik. *tm: Text Mining Package*, 2013. URL http://CRAN.R-project.org/package=tm. R package version 0.5-8.3. [p1]

I. Feinerer, K. Hornik, and D. Meyer. Text mining infrastructure in R. *Journal of Statistical Software*, 25 (5):1–54, 2008. URL http://www.jstatsoft.org/v25/i05/. [p1]

I. Fellows. Deducer: A data analysis GUI for R. *Journal of Statistical Software*, 49(8):1–15, 2012. URL http://www.jstatsoft.org/v49/i08/. [p2]

J. Fox. The R Commander: A basic statistics graphical user interface to R. *Journal of Statistical Software*, 14(9):1–42, 2005. URL http://www.jstatsoft.org/v14/i09/. [p1]

J. Gentry. *twitteR: R Based Twitter Client*, 2013. URL http://CRAN.R-project.org/package=twitteR. R package version 1.1.0. [p3]

S. Heiden. The TXM platform: Building open-source textual analysis software compatible with the TEI encoding scheme. In R. Otoguro, K. Ishikawa, H. Umemoto, K. Yoshimoto, and Y. Harada, editors, *Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation*, pages 389–398, Sendai, Japan, Nov. 2010. Institute for Digital Enhancement of Cognitive Development, Waseda University. URL http://halshs.archives-ouvertes.fr/halshs-00549764. [p1]

K. Hornik. *Snowball: Snowball Stemmers*, 2012. URL http://CRAN.R-project.org/package=Snowball. R package version 0.0-8. [p4]

L. Lebart, A. Salem, and L. Berry. *Exploring Textual Data*. Kluwer Academic Press, Dordrecht/Boston, 1998. ISBN 0-7923-4840-0. [p1]

É. Lecoutre. The R2HTML package. *R News*, 3(3):33–36, 2003. URL http://CRAN.R-project.org/doc/Rnews/Rnews_2003-3.pdf. [p8]

O. Nenadic and M. Greenacre. Correspondence analysis in R, with two- and three-dimensional graphics: The ca package. *Journal of Statistical Software*, 20(3):1–13, 2007. URL http://www.jstatsoft.org/v20/i03/. [p6]

P. Ratinaud. *Iramuteq: Interface de R pour les Analyses Multidimensionnelles de Textes et de Questionnaires*, 2013. URL http://sourceforge.net/projects/iramuteq/files/iramuteq-0.6-alpha3/. Version 0.6-alpha3. [p1]

B. Ripley and M. Lapsley. *RODBC: ODBC Database Access*, 2012. URL http://CRAN.R-project.org/package=RODBC. R package version 1.3-6. [p3]

S. Rödiger, T. Friedrichsmeier, P. Kapat, and M. Michalke. RKWard: A comprehensive graphical user interface and integrated development environment for statistical analysis with R. *Journal of Statistical Software*, 49(9):1–34, 2012. URL http://www.jstatsoft.org/v49/i09/. [p2]

D. Sarkar. *Lattice: Multivariate Data Visualization with R*. Springer, New York, 2008. [p5]

D. Temple Lang. *ROpenOffice: Basic Reading of Open Office Spreadsheets and Workbooks*, 2011a. URL http://www.omegahat.org/ROpenOffice/. R package version 0.4-0. [p3]

D. Temple Lang. *Rstem: Interface to Snowball Implementation of Porter's Word Stemming Algorithm*, 2011b. URL http://www.omegahat.org/Rstem/. R package version 0.4-1. [p4]

S. Urbanek. *rJava: Low-level R to Java Interface*, 2013. URL http://CRAN.R-project.org/package=rJava. R package version 0.9-4. [p4]

P. M. Valero-Mora and R. Ledesma. Graphical user interfaces for R. *Journal of Statistical Software*, 49(1): 1–8, 2012. URL http://www.jstatsoft.org/v49/i01/. [p2]

A. Zeileis and G. Grothendieck. zoo: S3 infrastructure for regular and irregular time series. *Journal of Statistical Software*, 14(6):1–27, 2005. URL http://www.jstatsoft.org/v14/i06/. [p5]

Milan Bouchet-Valat
Quantitative Sociology Laboratory (LSQ-CREST),
Centre for Studies in Social Change (OSC-Sciences Po & CNRS),
National Institute for Demographic Studies (INED)
France
nalimilan@club.fr

Gilles Bastin
Pacte Laboratory
Sciences Po Grenoble
France
gilles.bastin@sciencespo-grenoble.fr