Preparing IRaMuTeQ input files

Stephen Gourlay, Kingston University, Kingston-upon Thames, UK

Introduction

The text to be analysed can come from any source. Examples below cover interview transcripts, bibliographic records, articles and other printed documents, and open responses on survey questionnaires. You can analyse just one document, or a set of documents (and you can create sub-sets of the larger set if you wish, within IRaMuTeQ). A set of documents being analysed together is referred to as a "corpus".

Data files have to be in a particular format, and saved in a particular way:

- All files must be plain text files, saved with the file name extension .txt
- Files should preferably be saved using the UTF-8 encoding standard.

If you use a plain text editor like Notepad++ (downloadable free) you can easily meet these conditions. Any plain text editor will do, so long as you can save the files according to the UTF-8 encoding convention. In Notepad++ click on the `Encoding` menu at the top and select `Encode in UTF-8`. Files created in this program are normally saved as plain text files. You cannot have text effects like **bold** or *italic*.

File structure

You can have two different file input formats - one for files without thematic variables, and one for files with thematic variables.

Files without thematic variables:

- The first line must be blank
- The second line – the metadata line – indicates the beginning of a new document, and lists the associated variables. The metadata line must begin with `****` (or 4 numerals), and must be followed by at least one variable formatted like this: `*variable_attribute`
- The third and subsequent lines are the data

For example

```
**** *var1_1 *var2_1

text text text text text text text text text text text text text text text text text text
```

Files with thematic variables:

- The first line must be blank
- The second line – the metadata line – indicates the beginning of a new document, and lists the associated variables. The metadata line must begin with `****` (or 4 numerals), and must be followed by at least one variable formatted like this: `*variable_attribute`
- The third line must be the first thematic variable, formatted like this: `-*theme1`. (NB this variable is prefaced with a hyphen or dash, not an underscore character)

- The fourth and subsequent lines are the data for the first theme. Second and subsequent themes should be introduced with another `-*themeX` variable on a new line, and the text on the next line. For readability you can leave a blank line between the theme text, and the next theme metadata line[1].

For example

```
**** *var1_1 *var2_1

-*theme_1

text text text text text text text text text text text text text text text text text

-*theme_2

text text text text text text text text text text text text text text text text text
```

Naming variables

Variables:

- must only be written using a-z, A-Z, 0-9, and the underscore
- must follow the format: `*variable_attribute` - e.g. `*sex_f *interview_Jane`

Note - IRaMuTeQ is sensitive to case - `*sex_f` and `*Sex_f` are different variables

Creating the data input files

You can format interview transcriptions as you transcribe, or annotate a file of bibliographic records one at a time. But this is prone to error, and unnecessary. It's easier to create your input files with a workflow using the original source text software, a spreadsheet, and a plain text editor.

The basic process is as follows:

1. Copy the whole text to be analysed into a spreadsheet - one document per row; all the text to be analysed in one cell, and any other variables in other cells. Save this as a .csv file.
2. In a new column (e.g. to the right) use the CONCATENATE command to combine the metadata line text, text to be analysed, and some formatting commands, into one cell.
3. Copy the CONCATENATEd data to a plain text editor. Search and replace formatting commands to create the formatted input file.

To perform the search and replace operations you will need to use "regular expressions" - a powerful pattern searching tool. If you are using Notetab++, open the Search and Replace dialogue box and under Search Mode choose Regular Expressions. Other plain text editors might have a different way to ensure search and replace uses regular expressions.

Detailed examples covering interview transcripts, articles and printed documents, bibliographic abstracts, and responses to open-ended survey questions are given below.

Working with interview transcripts

Transcribe directly into a plain text editor. If you have transcribed into another word processing program, you will have to select all the text, and copy it into the plain text editor. You cannot have bold or other text formatting in plain text files.

[1] See Carmago & Justo's guide to IRaMuTeQ for an example, and the information on the IraMuTeQ web site.

1. Temporarily remove all the paragraphs to facilitate working with text in the spreadsheet. If you created the transcripts in Windows, using regular expressions, search for `\r\n` and replace with `\n` All paragraphing should disappear. If you created the source file on an Apple / Mac or a Linux computer, search for `\n`, and replace with `\n`.
2. Copy all the text into a spreadsheet - each interview transcript should be in one cell and each interview on a separate row.
3. In the spreadsheet, create columns for appropriate variables - e.g. date of interview, interviewer, interviewee etc.
4. In a new column, use CONCATENATE to add the metadata line and variables - e.g. `=CONCATENATE("\n**** *date_",A1," *int_",B1," *respondent_",C1,"\n",D1)`.
5. Copy the data into Notepad++
6. Using regular expressions, type `\\n` in the Search for box, and `\n` in the Replace box, and perform the search and replace across the whole document. Now each interview transcript should be headed with the metadata line, and the paragraphs of the original transcript will have been reinstated.
7. If you have thematic codes these will have to be inserted as you type. You will now have to clean the data before proceeding with any analysis.

Working with focus group transcripts

Focus group data poses particular challenges and choices. Daniel Pélissier has discussed preparing focus group transcripts, and has a tutorial for doing this this in IRaMuTeQ - in French.[2]

Working with articles and other published documents

Articles and published documents are usually in pdf or other formats. There may be copyright implications of using the full text - **please check your local situation before proceeding with these steps**.

1. Select all text and paste it into a plain text document (Notepad++).
2. Temporarily remove all the paragraphs to facilitate working with text in the spreadsheet. If you created the transcripts in Windows, using regular expressions, search for `\r\n` and replace with `\n` All paragraphing should disappear. If you created the source file on an Apple / Mac or a Linux computer, search for `\n` only, and replace with `\n`.
3. Copy all the text into a spreadsheet - the document text should be in one cell and each document on a separate row.
4. In the spreadsheet, create columns for appropriate variables - e.g. date of publication, source, author, citation etc.
5. In a new column, use CONCATENATE to add the metadata line and variables - e.g. `=CONCATENATE("\n**** *date_",A1," *citation_",B1," *source_",C1,"\n",D1)`.
6. Copy the data into Notepad++
7. Using regular expressions, type `\\n` in the Search for box, and `\n` in the Replace box, and perform the search and replace across the whole document. This reinstates the paragraphs. Now each document should be headed with the metadata line, and the paragraphs of the original documents will have been reinstated.
8. You will now have to clean the data before proceeding with any analysis. If you have thematic codes these will have to be inserted as you type.

Working with Bibliographic abstracts

Article titles and abstracts can readily be analysed using IRaMuTeQ.[3] Download the records as a .csv file. To create a data file of abstracts:

1. Open the .csv file in a spreadsheet program.

[2] The discussion is available here and the tutorial is linked to this page. See also Peyrat-Guillard et al 2014 (they refer to the 'ALCESTE' program, a commercial alternative to IRaMuTeQ.)

[3] For bibliometric analyses use a free program like VOSviewer.

2. Copy the abstract and any other data you need to a clean worksheet (this is not necessary, but the spreadsheet will be easier to work with).
3. You must have at least one variable - here lets assume you want to include the publication year.
4. In a new column, use CONCATENATE to add the metadata line and variables - e.g. `=CONCATENATE("\n**** *year_",A1,"\n",B1)`.
5. Copy the data into Notepad++
6. Using regular expressions, type `\\n` in the Search for box, and `\n` in the Replace box, and perform the search and replace across the whole document. This puts the abstract in a paragraph below the metadata line.
7. You will now have to clean the data before proceeding with any analysis.

When working with abstracts you will probably want to have variables such as the author, the journal, and so on, or you might want to create a variable combining author name and date to make an in-text citation. Use spreadsheet text manipulation commands to do this. For guidance, search online for "working with text strings" and the name of your favourite spreadsheet program. You must ensure the result conforms to the rules for naming variables (see above).

Working with survey open question responses

Here you most probably begin with the data in a spreadsheet - make sure this is a .csv file.

1. Select the data columns you need to include. Copy the columns you need as variables to a clean worksheet (the spreadsheet will be so easier to work with).
2. In a new column, use CONCATENATE to add the metadata line and variables - e.g. `=CONCATENATE("\n**** *boss_",C1," *job_",D2," *gender_",F2,"\n",A2)`.
3. Select this new column and copy the data into Notepad++
4. Using regular expressions, search for `\\n` and replace with `\n`. This should create a metadata line with the responses on the following line. See the IRaMuTeQ guide (p.6) for an example.
5. You will now have to clean the data before proceeding with any analysis.

Cleaning your data

Text is typically very 'messy' data. Words may be spelt wrongly, and there may be inconsistent spelling. These can produce misleading results, and need correcting. Just what counts as a problem will depend on the kind of text, what you are trying to do, and other constraints relating to your analysis strategy.

When transcribing interviews you might not put any punctuation in (people do not use punctuation when speaking) but the algorithms use punctuation, so this would not be a good idea. The algorithm also treats everything looking like a word as a word - this will include "Err" and "Um" if you have used these to indicate hesitation. And it will include the interviewer's questions and comments. You might have words like "didn't" and phrases like "did not" because that's what the speakers said. But unless you are interested in linguistic differences you should force the program to treat these as the same word for consistency.

Word hyphenation is often not applied consistently. For example, you might have "work place" and "work-place" in the same corpus. These should be treated as the same, but the program disregards hyphens, so "work-place" will be split into two. This would be consistent with "work place" - but you perhaps want to be certain that the algorithm treats "work place" and "work-place", not to mention "workplace" as the same, single term.

Published texts may use several different English spelling conventions. For example, "organization" vs. "organisation". They also often contain acronyms, and phrases that indicate a concept. "OCB" is an acronym for the concept "organizational citizenship behaviour". "OCB" will be treated as one 'word' in IRaMuTeQ but "Organizational citizenship behaviour" will be treated as three words. To ensure this concept term is treated as a single term you should edit the phrase.

Data cleaning tasks to consider:

- standardize spelling on one variant of written English
- check for abbreviations that could be mistaken for other words - the word "Fed" in documents about the US financial system refers to "Federal Reserve" but would normally be classed as the past tense of "feed" (Schonhardt-Bailey 2013)
- standardize hyphenated words to a single form. You can turn them into a single word even if that is not normal. Or you can join them with the _ to force the program to treat them as one word. E.g. you can replace "work-place", "workplace" and "work place" with 'work_place'. The program also discards other non-standard punctuation. If the text has words separated by / mark (e.g. "word/term") you should edit this to ensure the meaning is not distorted.
- decide what to do about synonyms. For example you might replace "work-place", "workplace" "work place" and "place of work" with 'work_place' because the first four terms are synonymous, and keeping the lexical variations will not add to your analysis. However, lexical variability might inhibit the algorithm from detecting simpler but interesting patterns. The extent to which you standardize synonyms depends on how far you consider it acceptable to edit the text to make it more amenable to algorithmic analysis. If you are interested in linguistic variability, then obviously you should not standardize at all.
- decide what to do about multi-word terms and acronyms - standardize on one or the other - "Organizational citizenship behaviour" can be replaced with 'organizational_citizenship_behaviour' or even OCB.[4]
- Carmago & Justo advise that numbers should be rendered as numbers, not text - i.e. 2019 (not "twenty-nineteen") and 70 (not "seventy").

To exclude a word from the analysis, but keep it in the text so that you can read it when looking at the processed text, put the word between underscores. For example if you included text to indicate hesitations in your interview transcript ("Err" or "Um", for example) put this text between underscores: `_Err_`, and `_Um_`. You can do the same with the interviewer's questions. For example, if the question was `What did you do?` change this to `_WhatDidYouDo?_` .

You can do data cleaning in the source files, in the .csv files (a useful free tool for this is OpenRefine) or in the IRaMuTeQ input files, using a plain text editor. You can also edit the dictionary of expressions to standardize words and phrases[5].

References

Formatage des corpus texte - Note on the IRaMuTeQ web site
Baril, E., & Garnier, B., 2015, Utilisation d'un outil de statistiques textuelles. IRaMuteQ 0.7 alpha 2 Interface de R pour les Analyses Multidimensionnelles de Textes et de Questionnaires
Carmago, B. V. & Justo, A. M., IRaMuTeQ Tutorial, translated from Portuguese by Terese Forte
Loubère, L. & Ratinaud, P., Documentation IRaMuTeQ 0.6 alpha 3 version 0.1
Pélissier, D. Initiation à la lexicométrie Approche pédagogique à partir de l'étude d'un corpus avec le logiciel IRaMuTeQ, MARS 2017, V6
Pélissier, D. "Pourquoi et comment utiliser la lexicométrie pour l'analyse de focus groups?," in Présence numérique des organisations, 11/07/2016
Peyrat-Guillard, D., Miltgen, C.L. and Welcomer, S., 2014 Analysing conversational data with computer-aided content analysis: The importance of data partitioning. JADT 2014 : 12esJournées internationales d'Analyse statistique des Données Textuelles
Schonhardt-Bailey, C., 2013. Deliberating American monetary policy: a textual analysis. MIT Press.

Thanks to Pierre Ratinaud who took time to correct errors in an earlier version.

Prepared and all links working: 19 March 2019

[4] See Appendix 1 to Schonhardt-Bailey's study of US monetary policy for some examples.
[5] See the IRaMuTeQ Guide p.7