WORD ASSOCIATION DATA PROCESSOR 0.6
A SHORT USER GUIDE

Andreas Buerki
buerkiA@cardiff.ac.uk

CONTENTS

1 WHAT IS NEW IN VERSION 0.6?

If you are familiar with an earlier version of the WADP, take a look at the separately supplied release notes for this version which details the full history of changes. Then turn to the sections of the manual that describe those features to find out how they work. Most of the changes since version 0.5 have been to the reporter module which is discussed on pages 9ff.

2 INTRODUCTION

The Word Association Data Processor (WADP) is an open-source software package which automates key aspects of the processing of large amounts of word association data gathered from respondents in word association tests. Word association data are typically gathered by asking respondents to write down the first word (or the first few words) that come to mind given a cue word. The word(s) that are written down in response to cues, are referred to as responses. Figure 1 shows examples of cue-response pairs.

Responses are then typically analysed by looking for patterns in word associations across or within individuals, or across or within various cues. One way to analyse associations is to categorise them into various types. At a rudimentary level, the first cue - response pair in figure 1 could be categorised as instantiating a *paradigmatic* relationship between cue and response (e.g. *boy* and *girl* are part of a set of terms that can be used in the same way in a sentence to produce different meanings). The second cue - response pair could be categorised as instantiating a *syntagmatic* relationship since a *house guest* is a complex term,

boy ⇒ girl
house ⇒ guest

Figure 1: Two cue - response pairs

the parts of which can occur in the same sentence, next to each other. A more elaborate categorisation of word associations is presented in Fitzpatrick et. al. (2015:18-9). Another way in which cue - response pairs can be analysed is by looking at typical or *primary* responses to cues and how typical individual respondents' responses are.

The WADP is designed to help researchers working with large amounts of word-association data to process cue - response pairs efficiently and consistently with regard to both categorisations as well as deriving primary responses. This brief guide introduces the reader to what exactly the WADP is and how it may be used.

The guide is structured as follows: the next section lists the key features of the WADP. This is followed by a section with installation information. A tutorial-style introduction to basic operations is presented in section 5. More advanced features and their operation are introduced in section 6 and this is followed by a section featuring a number of expected FAQs (section 7). The guide concludes with information on software licensing and copyright and an appendix describing file formats.

3 KEY FEATURES

The principal goal of the WADP is to facilitate the efficient and consistent categorisation of responses to word association tasks. The WADP comprises of three modules with the following key features:

Categoriser module:

- a tidy and efficient interface for the manual categorisation (i.e. rating) of cue - response pairs
- automatic categorisation of responses in cases where categorisations for the relevant cue - response pairs are found in a database of past category assignments
- automatic storage of all new categorisations in the database
- tracking of respondent IDs and rater IDs (if provided) in all in- and output files.
- tracking of rater IDs in categorisations stored in database files

Reporter module:

- automatic creation of individual response profiles
- automatic creation of cue profiles
- automatic creation of primary response lists
- display of inter-rater agreement ratios where several raters rated the same data

Administrator module:

- automatic comparison and reporting of differences between database files
- support for inter-rater agreement procedures through interactive resolution of differences between two database files
- automatic combination of two database files
- turning a fully-rated output file into a database file

Compatible Operating Systems

The WADP is written in the bash shell scripting language and as such can be run on wide range of operating systems, including Linux, OS X and, if Cygwin is installed, Windows. All recent versions of these systems include bash version 3.2 or later which is minimally required to run the WADP. The WADP was tested and confirmed working on OS X version 10.8, 10.9, 10.10 and 10.11, Xubuntu 14.04 and Cygwin 1.8 on Windows 7 and Windows 8.

Installation under Xubuntu

1) Open the `WADP_0.6` folder

2) Double-click on the `Xubuntu_installer` icon

3) The installer window will open, displaying processing information and finally asking if a launcher should be placed on the desktop. Answer by pressing ENTER.

4) A yellow WADP-icon should appear on the desktop (and also inside the `WADP_0.6` folder). The WADP is launched by double-clicking on this icon. The icon can be placed anywhere.

Should the installer fail, a manual installation will be necessary: the files inside the `src` folder should be placed into a folder named `bin` inside the user's home directory (create this folder if it does not already exist). Then the WADP can be launched by ctrl-clicking on the file `WADP.sh` and choosing 'open with', then choosing 'Terminal Emulator' or similar. The `WADP.sh` file can be placed anywhere for convenience (but the other files must remain in the `bin` folder).

Installation under Apple's OS X

1) Open the `WADP_0.6` folder.

2) Double-click on the `OSX_installer` icon.

3) The installer window will open, displaying processing information and finally asking if an icon should be placed on the desktop. Answer by pressing ENTER.

4) A yellow WADP-icon should appear on the desktop (and also inside your Application folder and the `WADP_0.6` folder). The WADP is launched by double-clicking on this icon. The icon can be placed anywhere.

Should the installer fail, a manual installation will be necessary: the files inside the `src` folder should be placed into a folder named `bin` inside the user's home directory (create this folder if it does not already exist). Then change the name of the file `WADP.sh` into `WADP.command`. Now the WADP can be launched by double-clicking on this file. The `WADP.command` file can be placed anywhere for convenience (but the other files must remain in the `bin` folder).

Installation under Windows

Under Windows, before the WADP is installed, the Cygwin environment needs to be installed. This is accomplished by following the step in the box below:

1) Download and then open the application `setup-x86.exe`, available free of charge from http://cygwin.com/setup-x86.exe

2) Follow the on-screen instructions. The default installation settings can be used, except in the following:

   a) On the 'Choose Installation Directory'-screen, the 'root directory' should be either `C:\cygwin64` (if 64 bit version) or `C:\cygwin` otherwise. This should be the default. Also, under 'Install For', the option '`Just Me`' should be chosen (cf. figure 2)

   b) On the 'Select Packages' screen, the following packages need to be installed in addition to the default packages: `diffutils`, `ncurses`, `cygutils-extra` (under the `utils` section), and `bc` (arbitrary precision calculator language) from the `Math` section. The easiest way to do this is to enter these names (one after another) in the search box, then click the '+' next to 'Utils' (or 'Math' for `bc`) and click on the circling arrows (cf. figure 3). For all other prompts, just click on 'next' or 'okay'. There is no need to tick to create a desktop shortcut or startmenu item on the last screen.

3) Once all configuration options are set, the installer then proceeds to install all necessary components of the Cygwin environment.

Once the Cygwin environment is installed, the WADP is installed in four steps:

1) Open the `WADP_0.6` folder.

2) Double-click on the `Cygwin_installer` or `Cygwin64_installer` icon (depending on whether or not the 64-bit version of Cygwin was installed. Try both if one doesn't work).

3) The installer window will open, displaying processing information and finally asking if a shortcut should be placed on the desktop. Answer by pressing ENTER.

4) A yellow WADP-icon should appear on the desktop (and also inside the `WADP_0.6` folder). The WADP is launched by double-clicking on this icon. The icon can be placed anywhere.

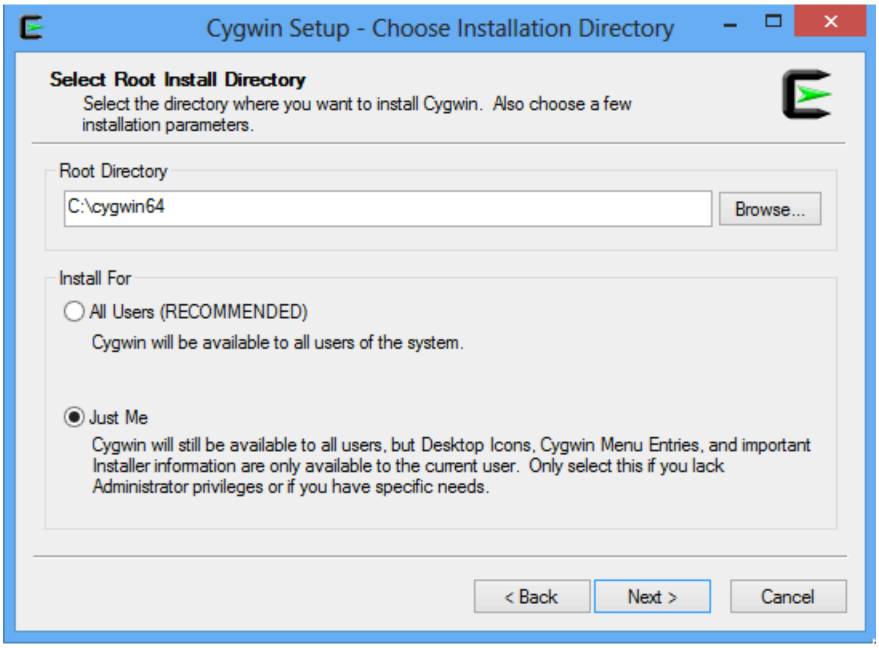
5 BASIC OPERATION

To illustrate the basic operation of the WADP, the example files found in the `test_data` directory inside the `WADP` directory will be used. To prepare for the following tutorial sections, start the WADP by double-clicking on its icon. The main menu as shown in figure 4 should appear. From here, the different modules of the WADP are accessible and this screen is automatically returned to after a task has completed. You can also exit WADP from this screen.[1]

The Categoriser module

This module helps categorise word association data. Let us assume we wish to categorise a set of responses to a word association task. Let us further assume that we have, at this moment, no database of previous category assignments available to automate category assignment. To start the process of assigning categories to cue - response pairs, type `c` and then press `ENTER`. It does not matter whether a capital 'C' or a lower case 'c' is entered.

[1] Under some systems, the window has to be closed manually after the programme has exited.



Cygwin Setup - Choose Installation Directory

Select Root Install Directory

Select the directory where you want to install Cygwin. Also choose a few installation parameters.

Root Directory

C:\cygwin64

Install For

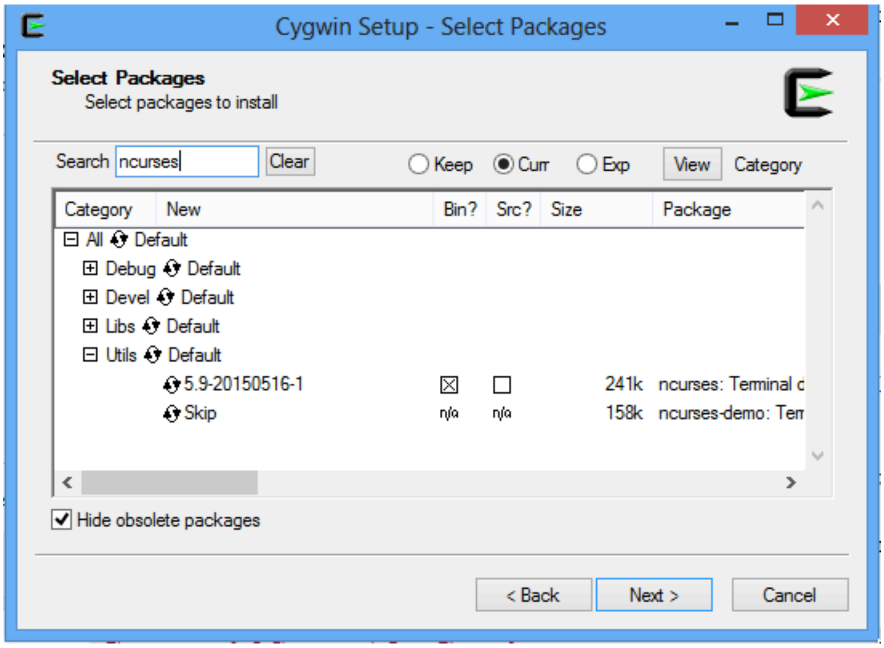
○ All Users (RECOMMENDED)

Cygwin will be available to all users of the system.

● Just Me

Cygwin will still be available to all users, but Desktop Icons, Cygwin Menu Entries, and important Installer information are only available to the current user. Only select this if you lack Administrator privileges or if you have specific needs.

Figure 2: The 'Choose Installation Directory' screen of the Cygwin installer



Cygwin Setup - Select Packages

Select Packages

Select packages to install

Search: ncurses

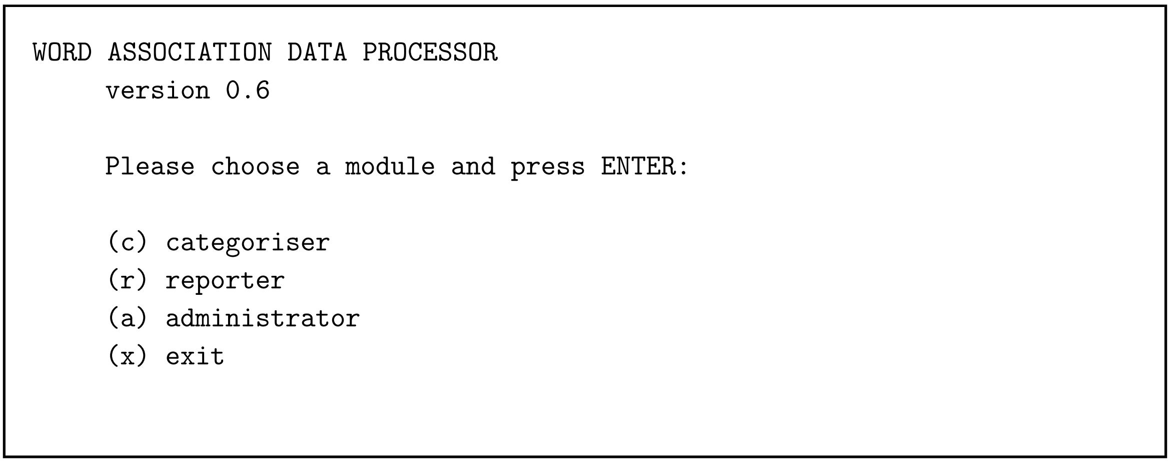
○ Keep ● Curr ○ Exp

View: Category

Category	New	Bin?	Src?	Size	Package
⊟ All Default					
⊞ Debug Default					
⊞ Devel Default					
⊞ Libs Default					
⊟ Utils Default					
	5.9-20150516-1	☒	□	241k	ncurses: Terminal d
	Skip	n/a	n/a	158k	ncurses-demo: Tem

☑ Hide obsolete packages

Figure 3: The 'Select Packages' screen of the Cygwin installer



```
WORD ASSOCIATION DATA PROCESSOR
     version 0.6

     Please choose a module and press ENTER:

     (c) categoriser
     (r) reporter
     (a) administrator
     (x) exit
```

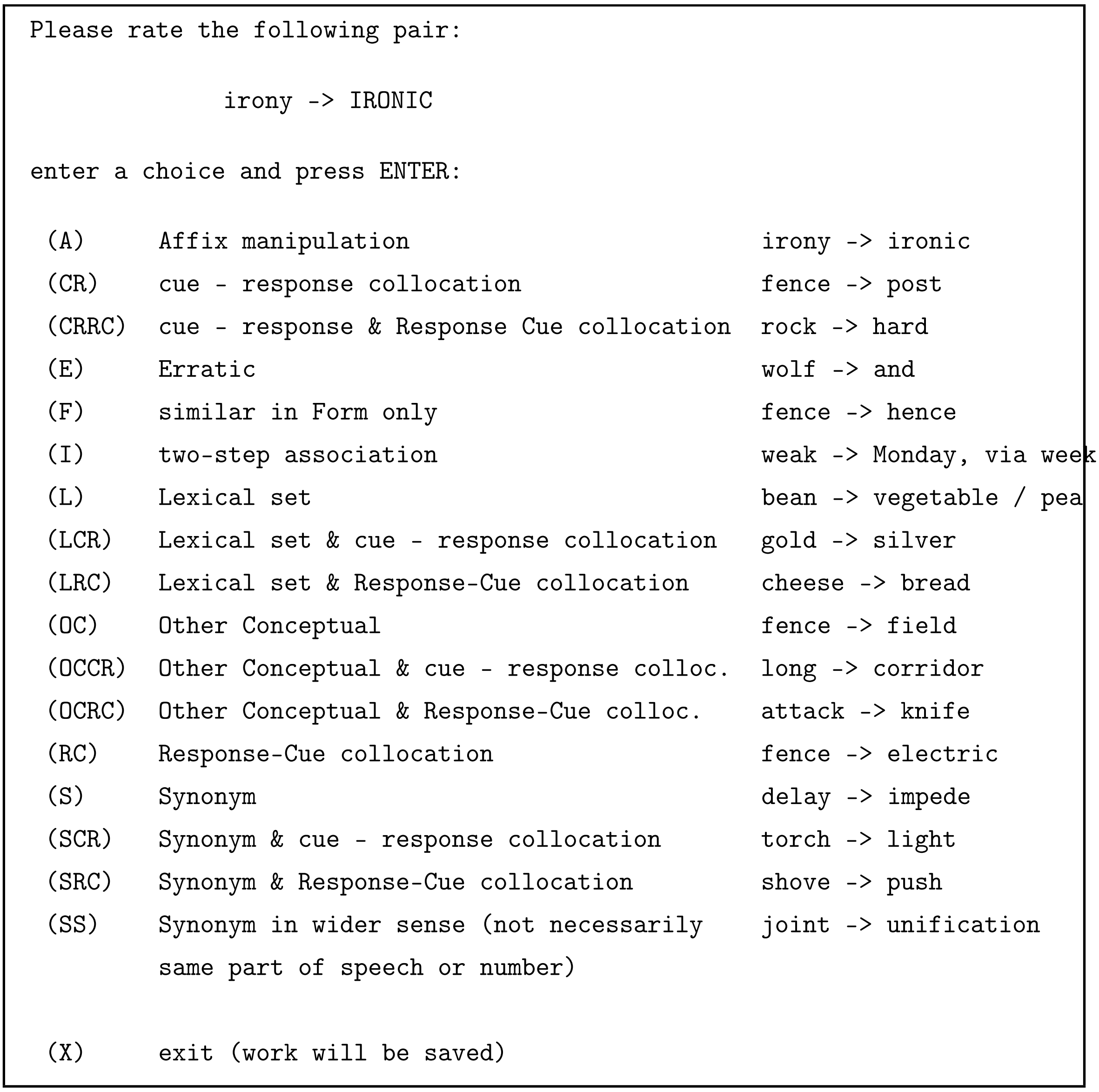
Figure 4: The WADP main menu

The WADP now asks for data to be categorised. The expected input is a .csv file containing the word association data to be categorised. This file needs to be formatted according to the formatting specification for in-files given in the appendix (page 17). Essentially, the file must be a .csv file and contain a header (first row), listing the cues, followed by any number of rows of responses to those cues, each row consisting of one respondent's responses. A file of this type might be constructed using a spreadsheet application (such as Excel or Open Office) and exported as a csv file. The `test_data` directory contains an in-file of the proper format named `five.csv`. Drag this file into the WADP-window and then press ENTER.

The WADP will now ask you to provide a database file. For now, we shall pretend that we have no existing database of categorisations and just press ENTER. Next, the WADP asks whether rater IDs should appear in the rated output file. You will notice that the sequence `(Y/n)` follows the question. This specifies the format of the expected answer, i.e. either `y` or `n`, followed by ENTER. It does not matter whether upper case or lower case letters are entered.[2] Next, WADP asks for confirmation that no database is to be used and then asks for the present rater's ID. This is to make it possible to associate categorisations with raters if this is desired. E-mail addresses generally make useful IDs because they are globally unique. Enter an ID (or none) and press ENTER.

The WADP will now present cue - response pairs for categorisation. *It is likely that the window size will need to be adjusted at this stage so that the whole window content can be displayed.* The full rating screen as shown in figure 5 should now be visible. By default, it is assumed that categories according to Fitzpatrick et. al. (2015:18-9) will be assigned. Additional categories or an entirely different set of categories can be specified if desired (see section 6). The first cue - response pair that needs to be rated is 'irony -> IRONIC'. Regardless of original case, the response will always be displayed in block capitals. To categorise this pair, choose an appropriate category and press ENTER. Note that it does not matter whether lower or upper case letters are entered. If no rating is entered (i.e. if merely ENTER is pressed), this is accepted as a valid response and will appear as an empty field in the output file. However, if that same pair is to be rated again in the future, the programme will ask for a categorisation again, treating it as though no categorisation had ever been given. Rate a few more pairs, but no more than ten at this stage. Regardless of whether all pairs in the in-file have been rated or not, it is always possible to interrupt the categorisation process by inputting `X` and pressing ENTER. Do so now. categorisations up to this point will be saved and the programme will state that the categorisation is not yet complete and ask whether

[2] The default answer, indicated by an upper-case letter, can be given by simply pressing ENTER.



```
Please rate the following pair:

             irony -> IRONIC

enter a choice and press ENTER:

 (A)     Affix manipulation                          irony -> ironic
 (CR)    cue - response collocation                  fence -> post
 (CRRC)  cue - response & Response Cue collocation   rock -> hard
 (E)     Erratic                                     wolf -> and
 (F)     similar in Form only                        fence -> hence
 (I)     two-step association                        weak -> Monday, via week
 (L)     Lexical set                                 bean -> vegetable / pea
 (LCR)   Lexical set & cue - response collocation    gold -> silver
 (LRC)   Lexical set & Response-Cue collocation      cheese -> bread
 (OC)    Other Conceptual                            fence -> field
 (OCCR)  Other Conceptual & cue - response colloc.   long -> corridor
 (OCRC)  Other Conceptual & Response-Cue colloc.     attack -> knife
 (RC)    Response-Cue collocation                    fence -> electric
 (S)     Synonym                                     delay -> impede
 (SCR)   Synonym & cue - response collocation        torch -> light
 (SRC)   Synonym & Response-Cue collocation          shove -> push
 (SS)    Synonym in wider sense (not necessarily     joint -> unification
         same part of speech or number)

 (X)     exit (work will be saved)
```

Figure 5: The rating screen

the user would like an output file for the part that is complete. Enter `Y`, or just press `ENTER`. The programme will also ask if the output directory should be opened. Simply press `ENTER` to confirm the default answer.

Take a look at the `test_data` directory which should have been opened automatically for you. You will notice that two new files have appeared:

- `categorised-five.csv`[3]
  This file contains the list of rated responses. If you rated less than five cue - response pairs, only the header will be written, otherwise all complete rows will be listed (incomplete rows do not appear in the output file, but the categorisations have been saved, see below). To see its contents, you may wish to open the file in a spreadsheet application.

- `db-DATE.dat`
  Where `DATE` is today's date. This file is a database of the category assignments you

[3] Under some operating systems, the .csv and .dat extensions might remain invisible.

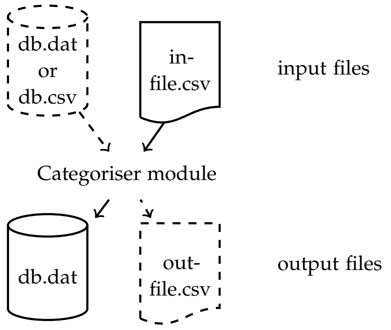


Figure 6: In- and output files of the categoriser module.
Note. Dashed files are optional

have just made. It can be used to automatically assign a category to cue - response pairs in the future. The file itself is in WADP's own database format, but since this is a plain text format, it can be examined in a plain text editor, though there usually is no need to do so.[4]

At this point, it may be worth setting up the WADP's window settings so that it is convenient to work for extended periods of time. Exact settings depend on the operating system used, but on all of them, the size of the window, the background colour as well as colour, type and size of font can be adjusted. Typically, a light background with dark font of at least size 12 or 13 pt is recommended.

Let us now assume we wish to continue categorising the set of word association data. Note that since the output file `categorised-five.csv` only contains the data that has been rated, we need to provide WADP with the original in-file (i.e. the file `five.csv`) in order to carry on with the categorisation process. Unlike the first time around, however, we now have a database of rated responses that can be used to automatically categorise responses for which we have assigned a category in the past. This means that, as more and more cue - response pairs are categorised, there will be less and less need for manual categorisation work as the correct category of any previously rated pair is automatically assigned through look-up in the database. By now the WADP-window will have reverted back to the main menu. Choose the categoriser module again, drag the `five.csv` file into the window when prompted and now also drag the newly created database file (the one with today's date) into the window when prompted to supply a database file. The categoriser will check the database provided and make any automatic category assignments it can. When it encounters a cue - response pair that has not yet been rated, it prompts the rater to assign a category as before. Wait until prompted, and then enter another few category assignments. During manual categorising, WADP features the possibility of going back to the previous rating (perhaps because a typo was made, or a rater changes their mind) by typing `b` instead of a rating and pressing ENTER. It is only possible, however, to go back to ONE previous rating. If categorisations before that need changing, the database file needs to be edited (see page 13). After categorising a few more cue - response pairs, exit (or, if continuing to the end, WADP will advise that categorisation is complete).

The WADP will ask whether you wish to update the database or create a new one. If `u` is entered, the categorisations that were just now carried out will be added to the database. This is the default response. If `n` is entered, the input database will be left untouched and a new database will be created, containing both the categorisations of the old database as

[4] See section 6 for more information on how to edit a database file.

well as the categorisations added in the current session. Press ENTER to choose the default option updating the existing database.

A new output file categorised_five-1.csv will have appeared in the test_data directory and the the database will have been updated with any category assignments made. Output files can be renamed, but the file extensions (.dat for database files and .csv for data files) must be retained and file names must not contain spaces. Figure 6 shows a schematic summary of the input and output files of the categoriser module.

The Reporter module

The reporter module creates reports from categorised word-association data. Four different kinds of report can be produced:

INDIVIDUAL RESPONSE PROFILES list instances of each category summed *by respondent*. A schematic example is shown in table 1. Respondent IDs are only listed if present in the input file.

respondent ID	CAT1	CAT2	CAT3
respondent 1	3	21	1
respondent 2	0	16	33
respondent 3	24	5	10

Table 1: Individual response profiles

CUE PROFILES list instances of each category summed *by cue*. A schematic example is shown in table 2.

cues	CAT1	CAT2	CAT3
cue 1	89	234	1
cue 2	143	259	0
cue 3	6	272	478

Table 2: Cue profiles

PRIMARY RESPONSE LISTS are csv-files that list all responses to each cue in order of frequency. Typically the most frequent response for each cue or the few most frequent ones are considered primary responses. An example list is shown in table 3, where the first line after the header would show the (most) primary responses.

blue	frequency	fence	frequency	irony	frequency
WATER	3	POST	4	IRONIC	2
GRASS	1	COW	1	TWISTED	2
CLOUD	1			SHARP	1

Table 3: Primary responses list

INTER-RATER AGREEMENT REPORT If categorised word-association data are fed in that were derived from a categorisation database that had inter-rater resolution mark-up (for an explanation of this, see the description of the administrator module, below), then a report can be displayed on-screen which gives the ratio of inter-rater agreement, that is, the number of categories where both (or all) raters agreed on categorisation, divided by the total number number of categories that appear in the data file. For databases to be marked up with agreement, they need to have been created with WADP v. 0.6 or later. An example of an inter-rater agreement report is shown in figure 7.

```
==============================================================
Inter-rater agreement report for categorised_seven.csv
==============================================================
total ratings in file:  26
ratings marked as having needed resolution:  9
inter-rater agreement ratio:  .6539
==============================================================
The accuracy of the report depends on an agreement-marked
database having been used to produce the data file.  This will be
the case if the database was created using WADP v.  0.6 or later.

Press ENTER to continue.
```

Figure 7: Example of an inter-rater agreement report

The reporter module is accessed by choosing it from the WADP's main menu. The user is prompted to provide a data file from which the report is to be produced. You may wish to drag the file `categorised_five-1.csv` from the `test_data` directory into the window and press ENTER. A primary response list can be produced by supplying an uncategorised in-file (such as the file `five.csv` in the `test_data` directory), but for all other reports a categorised data file must be supplied. The programme will ask what types of report(s) should be produced. Type `A` (or any other option) and press ENTER. The programme will inform the user that the report(s) were placed in the output directory and the inter-rater agreement report will be shown on screen (if selected). The individual response profile report output is prefixed `i-report_`, the cue profile report is prefixed `c-report_` and the primary response list is prefixed `p-report_`. These are csv files that can be viewed in a spreadsheet application. Note that where applicable, categories are listed in alphabetical order and only categories to which assignments were made appear in reports. A summary of input and output of the reporter module is shown in figure 8.

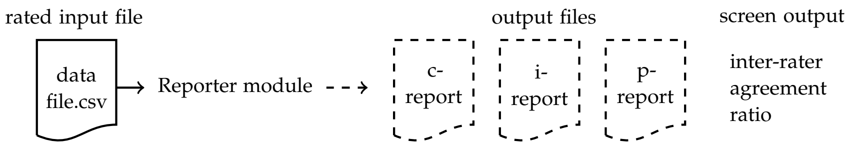
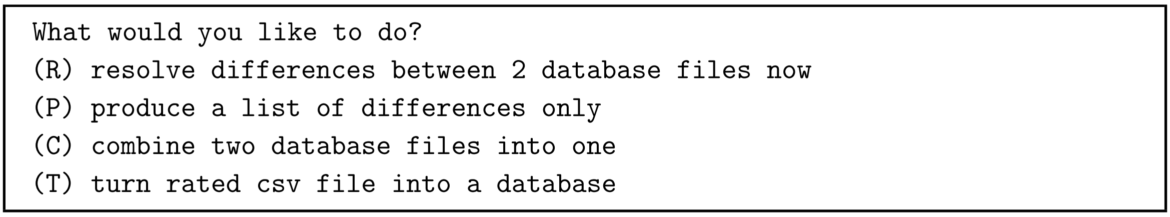


Figure 8: In- and output files of the reporter module



```
What would you like to do?
(R) resolve differences between 2 database files now
(P) produce a list of differences only
(C) combine two database files into one
(T) turn rated csv file into a database
```

Figure 9: The administrator menu

The Administrator module

This module performs a number of administrative tasks in relation to database files. It has three basic functions:

(R) Resolve differences between two database files interactively

(P) Produce a report listing all differences between two database files

(C) Combine two database files automatically

(T) Turn a rated output file (in csv-format) into a database file

The first two of these functions can be useful, for example, when wishing to establish inter-rater agreement for a data set that was rated (i.e. had categories assigned) by two different raters. One may first wish to simply produce a report listing any differences, and later, when one is certain of the correct category assignment for each instance where raters differ, one can then use the resolution function to resolve differences (the programme will ask which categorisation should be used for all instances where categorisations differ) and end up with an agreed database file that can be used to assign categories to the same or new input files. The database will also contain a mark-up for each categorisation where raters initially disagreed. This feature is used to derive inter-rater agreement in the reporter module. Refer to the FAQs (p. 14) for a suggested inter-rater agreement procedure.

The (C)-function simply combines two database files automatically. While this generally assumes that there are no conflicting categorisations in the two databases, the software carefully checks for conflicts and resolves them by giving priority to the first database file. Any conflicts will also be logged and can then be reviewed by consulting the logfile produced.

The (T)-function turns a rated output file (such as those produced by the categoriser module) into a database file of the .dat format. [b] To try out the administrator module, choose it from the WADP's main menu. The administrator menu in figure 9 appears. To try out the (R) function, type `R` and press textttENTER. The process of interactive difference-resolution is initiated. The user is now prompted to supply the first of the two database files to be resolved. Drag the database file created earlier (the one with today's date in its name) into the window and press `ENTER`. The software asks for the second database file. Drag the supplied `example_database.dat` file onto the window and press `ENTER`. Exactly two .dat-formatted database files need to be supplied. The order of the files is significant in that, where database files agree, the rater IDs present in the first file are transferred to the resolved file. The user is then prompted to enter their rater ID because this ID will be associated with the decisions made in the difference-resolution process.[5] The user is

[5] The resolved output file will contain the rater IDs of the first supplied database file where the two raters agree, and the rater ID supplied to the administrator module (plus a mark-up 'RESOLVED') where they do not.

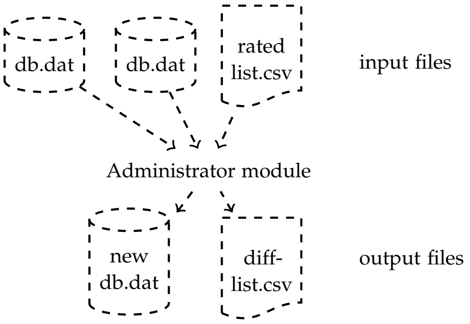


Figure 10: In- and output files of the administrator module.
Note. Dashed output files depend on options selected; diff-list is a list of differences.

alerted to differences in cues and responses and a menu is presented for differences in categorisations of responses. When all differences have been resolved, a new database file named `resolved-db-DATE.dat` (with `DATE` being today's date) is created and placed in the present output directory. The source database files are left in place. Optionally, a log file can be created that details the differences and decisions made to resolve them.

To try out the (P)-function, choose the administrator module again. This time, however, type `P` and press `ENTER` when the administrator module asks what you would like to. Supply the same database files as earlier. After processing, the differences will be displayed on screen and can be saved to a file called `difference_report.txt` in the current directory. No new database file is produced.

To try out the `(C)`-option, choose the administrator module again from the main menu and type `C` to combine two databases into one. Next, supply the two database files to be resolved. The order in which the database files are supplied is significant: the file that is supplied first overrides in case conflicting category assignments are found. The source database files are left in place. Any conflicts in categorisations will be displayed on screen and saved in a text file called `administrator-log.txt` (or `administrator-log-1.txt` in this case, because an earlier file of the same name exists in the output directory). A single new database file named `combined-db-DATE.dat` will be created.

The (T)-option works in similar ways, but rather than supplying two `.dat` database files, one fully rated output file needs to be supplied (for example the file `categorised_five-1.csv` we created earlier).

Figure 10 shows a schematic overview of input and outfiles of the administrator module. Regarding naming conventions of database files, also see the box of important notes on database files on page 12. A new database file named `converted-db-DATE.dat` will be created. If necessary, this can then be combined with an existing other database file using either the (P) or (C) options.

To quit the WADP, type `x` in the main menu and press `ENTER`. On some systems, the window will need to be closed manually, on others it disappears automatically.

Important note on file naming conventions

With time, database files will accumulate a very significant amount of information which represents many hours of work. It is therefore imperative to handle those files with great care. This includes very regular backups (copying the file(s) to a safe place, perhaps a copy to a different place on the computer and onto a flash memory stick or several). It also includes keeping on top of different versions of database files. One way of doing this is by (re-)naming database files for maximum clarity (no spaces are allowed in file names to be processed, however).

To aid users in managing database files effectively, a brief summary is here given of WADP's naming conventions regarding databases:

In the categoriser module,

- newly created database files are named `db-DATE.dat`, where `DATE` is the current date in the format DD-MM-YYYY. New database files are created when a) no database file was supplied as input, b) after categorising, the user chooses to create a new database file rather than updating the old one, and c) if a .csv-formatted database file was supplied (see section 6, below).
- updated database files retain their names.
- if any filename is identical to the name of an existing file in the output directory, the new file will have `-1` or `-2`, etc. added to its name to avoid the overwriting of existing files.

In the administrator module (see below),

- if two database files are manually resolved, the new resolved database is named `resolved-db-DATE.dat`, regardless of the names of the source database files. The source database files are left as they are.
- if two databases are automatically combined, the new combined database is named `combined-db-DATE.dat`, regardless of the names of the source database files. The source database files are left as they are.
- if any filename is identical to the name of an existing file in the output directory, the new file will have `-1` or `-2`, etc. added to its name to avoid the overwriting of existing files.

6 MORE ADVANCED OPTIONS

Use of existing database files in csv-format with the categoriser module

The categoriser module of the WADP features the ability to import existing csv files with categorised word association data for use as databases. To make use of this feature, a csv file conforming to the specification for csv-databases (see appendix, page 18) can be provided in lieu of a database in WADP's own dat format. Note that the WADP will output the usual .dat suffixed database which will include items supplied via the csv-database.

Editing a database file

If categorisations in the database need to be changed for any reason, this can be achieved by editing the respective .dat file in a text editor (by performing a 'find and replace' action, for example). This should be seen as a last resort and a backup copy of the database file to be edited should first be created. This is because of the possibility of inadvertently introducing formatting (such as extra spaces, tabs or line breaks) that can cause errors in automatic processing. Changes should therefore be kept to the necessary minimum. It is essential that after any changes, the file is saved in plain text format. For a brief explanation of the .dat file format, see the appendix (page 17).

Changing the default set of categories used by the categoriser module

If changes should be made to the categories that are displayed during the categorisation process, this can be accomplished by editing the file `categoriser.sh` in a plain text editor. It may be safer to make a copy of `categoriser.sh` before making any changes, then open it in a text editor. The section starting around line 9 of the file (entitled 'the following section can be adjusted') and ending with the comment 'end of user-adjustable section' hold the information used by the categoriser module to display and handle category assignments. Both the keys and the list of allowed categories should be changed at the same time. Only the information INSIDE of `key='...'` and `allowed_categories='...'` should be adjusted and the single quotes are essential. When editing is finished, the file must be saved in plain text format. Note, however, that the categoriser module accepts any and no category during manual categorisation (even categories not displayed as possible categories). This is to allow flexibility if needed.

7 FREQUENTLY ASKED QUESTIONS

Q1 I work with data where I have more than one response for each cue per participant. Can the WADP deal with this type of data?
Answer: Yes, it can. The key is to prepare the data in such a way that each response is on a separate line. If respondent IDs are used, each line must have a unique respondent ID, but this can be achieved by appending a response count to the end of an ID. For example if the ID is *respondent1*, their first responses to each cue might be under the ID *respondent1-1*, the second set of responses to the same cues under *respondent1-2*, etc. It is okay to leave some fields empty, for example if respondents gave different numbers of responses for different cues. An input file might then look something like this:

respondent ID	cheese	fence	rock	...
respondent1-1	moon	post	hard	...
respondent1-2	crackers	cow		...

Q2 I produced an output file without rater IDs. How can I convert it into one with rater IDs?
Answer: run the categoriser module and supply it with the *unrated* version of the in-file that needs to have categories and rater IDs added. Also supply the most up-to-date database file. Answer `y` when asked if rater IDs should be included. There should be no need for any manual category assignments since they should all be stored in the database, but if a categorisation was left blank at the first categorisation, the user will be asked to rate it now. The output list will contain the desired rater IDs.

Q3 How can I use the WADP in a scenario where inter-rater reliability needs to be checked?
Answer: One suggestion for a procedure would be following:

(a) Prepare the word-association data to be rated in a csv file according to the specifications in the appendix for in-files.

(b) Supply two raters with a) the latest version of a .dat database file and the in-file with data to be rated.

(c) Each rater independently assigns categories to the in-file data using the categoriser module. They return the updated .dat database files.

(d) The administrator module can then be used to first produce a list of inter-rater differences and then to resolve those differences.

(e) The resulting resolved-db.dat file can then be supplied to the categoriser module, together with the unrated in-file to produce a rated output file that includes the resolved categorisations. This output file can then be further processed with the reporter module.

(f) The reporter module features a menu item that allows the calculation of an inter-rater agreement ratio which shows the ratio of agreement between raters with respect to all categorisations in the input data file.

(g) The combined database file can then be used as the basis for future categorisations. If, while the two raters rated the in-file, the .dat file was changed in some way (such as being used by a third rater to rate more data), the two most recent versions can again be combined into a unified, up-to-date database file using administrator module.

Q4 I somehow ended up with several database files that all contain valid category assignments not contained in the others. What do I do?
Answer: Use the administrator module to combine the databases pairwise until a combined database with data from all the various database files results.

Q5 I am unsure which database file is the most up-to-date one. How do I find out?
Answer: It is sometimes possible to check the creation or modification date of files (or order them in a window according to any of those dates. If that cannot be done or database files have been modified after creation and do not provide reliable indications, it will be easiest to simply use the administrator module to combine the files that are thought to be the most recent into a single combined database file which will then contain the most up-to-date data.

Q6 Can I use filenames with spaces in them?
Answer: no, using filenames with spaces in them will most likely cause problems. Use underscores (_) instead.

Q7 I have a file with rated responses that are not in the database. How can I add those categorisations to the database?
Answer: This is a two-step process:

(a) Make certain the file is saved as a csv file (comma separated values) and in the correct format (see the appendix on the format of the categorised output file format). Then use the administrator module and choose the (T) option.

(b) After turning the csv-file into a dat database file, the administrator module can be used again to combine the newly created database file with an existing database file, either using the (C) or (R) options.

Q8 Where can I get support or answers to other questions about the WADP?
Answer: Please raise an issue at https://github.com/buerki/WADP/issues. Please understand that while I am grateful for feedback and suggestions, I may not be able to answer or address all issues.

8 TO DO

The automatic creation of stereotypy ratings as part of the reporter module is planned for a future release of the WADP. Requests for further features or comments can be made by raising an issue at https://github.com/buerki/WADP/issues.

9 LICENCE AND COPYRIGHT

Although the WADP has been extensively tested, results should always be checked for rough plausibility at the very least. This is why the software is offered as a beta release. The WADP is licensed under the terms of the EUPL v.1.1. The EUPL is an open-source license which allows licensees to use, reproduce, modify, distribute and sub-license the software. See the attached `EUPL.pdf` file or https://joinup.ec.europa.eu/software/page/eupl/licence-eupl for the full licence. Please read the licence, especially articles 7 and 8, prior to using the software. As of 2015, the copyright to the WADP is held by Cardiff University.

A APPENDIX: FORMAT DESCRIPTIONS

This appendix lists file format specifications used by the WADP.

in-files (files containing word association data to be categorised)

FILE FORMAT csv (comma separated values)

HEADER ROW The first row of the file must be a header row. It must be formatted as in one of the following examples: (the respondent ID field is optional)

respondent ID	cheese	fence	rock	...

or

respondent ID	cue 1: cheese	cue 2: fence	cue 3: rock	...

or

respondent ID	1. cheese	2. fence	3. rock	...

If there are no respondent IDs, the first field is left out:

1. cheese	2. fence	3. rock	4. shove	...

The field labelled 'respondent ID' can in fact contain any text at all as long as the word 'ID' is included. In the cue fields, the second pattern can be used with any text preceding the colon, so 'any text at all: cheese' would be fine.

REMAINING ROWS Depending on whether a 'respondent ID' row was declared in the header, these rows take either the format

respondent 1	moon	barbed wire	hard	...
respondent 2	goat		and roll	...

or

moon	barbed wire	hard	shovel	...
goat		and roll	push	...

There may be empty fields, such as when no response was given by a respondent. Respondent IDs can be anything, but must be unique for each respondent.

dat-database files

FILE FORMAT dat (content is in plain text)

HEADER ROW none

REMAINING LINES Each line of the document consists of a cue word, followed by a tab, followed by a response in uppercase, followed by a bar (|), followed by a category, followed by a semi-colon, followed by the ID of the rater that entered this particular categorisation. This is followed by a tab and the next response, etc. All lines end in a tab, followed by a line break. Here are the beginnings of two lines of an imaginary database file:

fence	POST | M;buerkiA@cf.ac.uk	BARBED_WIRE | M;buerkiA@cf.ac.uk
irony	IRONIC | M;buerkiA@cf.ac.uk	NOT_FUNNY | M;buerkiA@cf.ac.uk

csv-database files

FILE FORMAT csv (comma separated values)

HEADER ROW The first row of the file must be a header row. It must be formatted as in the following example:

WA001	cheese	WA002	rock	...

Note that a field with 'WA' immediately followed by one or more digits, needs to alternate with fields containing cue words.

REMAINING ROWS As in the following example:

moon	OC	barbed wire	RC	...
goat	RC	and roll	CR	...

categorised output files

FILE FORMAT csv (comma separated values)

HEADER ROW The first row of the file must be a header row. It must be formatted as in the following example. The fields with respondent IDs and the fields labelled 'rated by' are *optional* and can be left out:

respondent ID	cheese	category	rated by	...

The field labelled 'respondent ID' can in fact contain any text at all as long as the word 'ID' is included.

REMAINING ROWS As in the following example:

resp 1	MOON	OC	anonymous	...
resp 2	GOAT	RC	anonymous	...

As in the header, the fields for respondent IDs and rater IDs can be left out.

REFERENCES

Fitzpatrick, T., Playfoot, D., Wray, A., & Wright, . J. (2015). Establishing the reliability of word association data for investigating individual and group differences. Applied Linguistics. Accessible at http://applij.oxfordjournals.org/content/36/1/23.full.pdf+html