

Update History of OTSoft

Bruce Hayes
UCLA

January 2013, version 2.3.2

- Sorting of tableaux by harmony is now optional.
- Input files may be saved in Praat format.
- GLA now can use the Magri update rule, proposed to fix the problem with GLA pointed out by Joe Pater.
- Output file formats now include HTML.

January 2011, version 2.3.1

- Ranking argumentation is now done with an algorithm devised by Prince and Brasoveanu; it yields more complete results (and is apparently fully trustable: Prince and Brasoveanu will soon publish a proof of completeness for their algorithm.)
- Factorial typology now includes the “t-orders” proposed by Arto Anttila.
- There is a simple (no-Gaussian-prior) implementation of maxent grammars.
- The bug kindly spotted by Alan Prince (Rutgers Optimality Archive 882) is fixed.

1/12/04

I’m working on ways to keep the program running in increasingly hostile operating system environments.

- The program no longer writes to its own application folder, which would cause it to crash under many Windows setups. Instead, it seeks out some safe haven: first "c:\windows\temp", then "c:\winnt\temp", then "d:". If more safe havens are needed I will add them. The safe havens are where files remembering the user’s choices (otsoft.ini, SoftwareLocations.ini, RecentlyOpenedFiles.ini) are kept.
- In addition, some versions of Windows seem to be set up so that OTSoft cannot output to its own output screen. This error is now detected, and a backup plan is in place. (Specifically, the ShellExecute method is used to find whatever program is the default for displaying plain text.)
- Lastly, an increasing number of users avoid Excel and use OpenOffice instead. For now, they are stuck re. spreadsheets, but at least OTSoft now warns them to use tab-delimited text. I hope someday to make OTSoft compatible with OpenOffice, but I’m not optimistic—the very programming language in which OTSoft has been written has been discontinued by Microsoft.

- The program now always checks that file exists before trying to open it (duh...)
- The program detects when it is the first time it has been run, and if so, it copies TinyIllustrativeFile.txt and TinyIllustrativeFile.xls to the safe place on the hard disk, letting them be the first files to be run.

6/12/03

Bug fix: GLA now remembers the upper and lower plasticity values from run to run.

GLA secret feature: if the last six letters of the file name (not suffix) are “tabbed”, a very plain output file, with just ranking values and output frequencies, will be created: FilenamePlain.txt. This can be more easily used in Excel for further processing and interpretation.

3/25/03: Version 2.1

OTSoft can now do **Hasse diagrams**. They are made by using the ATT GraphViz software (freely downloadable) to graph the results of the ranking argument generator (else, the results of the GLA). Hasse diagrams can be viewed automatically in OTSoft, and are automatically imported into the high-quality printed output.

A priori rankings (externally imposed prior to ranking, by the user) are now supported for most of OTSoft’s algorithms (Classical CD, LFCD, Gradual Learning Algorithm, factorial typology calculator, ranking argument calculator). Click the **A priori rankings** menu on either the main or GLA screens to access this capability.

The **GLA** component of OTSoft now generate **stochastic tableaux**, and is also now capable of **reporting exactly what it did** to rank the constraints (this capacity was installed earlier for the discrete algorithms). To view, go to the Options menu on the GLA screen, and select “Print file with history of all actions”.

The GLA also allows full control over its learning parameters: initial ranking values, number of trials, plasticity (separately for Faithfulness and Markedness) and noise (ditto separately for Faithfulness and Markedness). See the **Initial rankings** and **Learning schedule** menus on the GLA screen.

You can now enter (some) markedness violations using **structural descriptions**. List strings that violate the constraint in the column for that constraint in the Excel spreadsheet, and OTSoft will search each candidate for each of these strings, and assign violations accordingly.

OTSoft now **relates more comfortably to local Windows programs**. Open **SoftwareLocations.ini** to specify where your programs are. The Edit command causes your spreadsheet to open your input file; more generally, all files created by OTSoft are accessible from the interface and need not be opened outside OTSoft.

The new **Help** menu auto-opens the OTSoft manual (with Word or the PDF reader); and information can be accessed by the internal links of the manual.

OTSoft now outputs files to a separate folder, set up as a daughter folder to the folder containing the input file. This should help avoid chaotic file situations.

Various effort to avert crashes: no file is opened without ascertaining that both the opening program and the file exist; etc.

Bug fix: Word macro is again able to delete the pesky diagonal line that was showing up in tableaux in Word 2000.

OTSoft now remembers from run to run (in OTSoft.ini) the four most crucial parameters for the GLA (number of learning trials, initial plasticity, final plasticity, number of trials to test).

For the GLA, “Number of times to go through forms” has been replaced by “Number of learning trials.” The program selects from all available forms according to their overall frequency in the input file.

Output files for PCWrite are no longer produced. PCWrite, while in many ways a superb word processor, can't handle file names longer than eight characters. Since most people use longer file names now, I felt that the PCWrite files were reduced to being just clutter in one's folders.

8/7/02

Bug fix: the code for computing how accurately the GLA mimicked input frequencies now works properly.

Bug fix: If you select Factorial Typology twice on the same input file, the second run now works properly.

Consecutive identical inputs in an input file (Excel or tab-delimited text) are now treated as two inputs, not one.

Two functions of the code for the Gradual Learning Algorithm have been made optional. If you want either (a) to have the program produce an Excel file showing the history of ranking values over the course of learning; or (b) include in its output file the probability of pairwise ranking for each individual constraint, you must ask the program to do this by opening the Options menu of the Gradual Learning Algorithm screen, and click to make your choice.

2/3/02

The GLA code now outputs the probability that any two constraints will be ranked in the order suggested by their ranking values (e.g., small difference = probability somewhat over .5; large difference = probability approaching 1).

The macros for Word file conversion had a bug, such that tables with 10-19 columns were treated as 1 column; 20-29 columns as 2 etc. This is now fixed.

12/28/01

Bug fix: the GLA algorithm now decrements plasticity correctly, for both markedness and faithfulness constraints. Earlier version decremented only Markedness, and involved a computational error due to use of base-10 logarithms rather than natural logarithms.

Bug fix: in computing factorial typology, the program no longer traces the course of the algorithm in the HowIDidIt.txt output file—this took many meg of disk space, much time (so that it appeared that the program has crashed), and was not helpful.

4/24/2001

Algorithms

There is now a revised version of Biased Constraint Demotion that favors specific over general Faithfulness constraints. This seems to result in improved performance, as discussed in the second version of my “Phonological Acquisition in OT: The Early Stages” (<http://www.humnet.ucla.edu/humnet/linguistics/people/hayes/Acquisition>). To access this revised algorithm, select Biased Constraint Demotion from the “Choose Ranking Algorithm” menu, then pick “BCD favors specific Faithfulness constraints” from the Options menu.

A minor problem in the Biased Constraint Demotion code is also fixed: the algorithm now informs the user (via a message box) when it is making a decision at “random”. (Actually, it never truly behaves randomly; in practice, a “random” decision means that it based its ranking on the arbitrary order of the constraints as they were listed in the input file.)

Self-reportage

Constraint Demotion, Low-Faithfulness Constraint Demotion, and both versions of Biased Constraint Demotion now report in detail how they arrived at their conclusions. To see this report, run an algorithm, then click on **View, Show how ranking was done**.

To save the file that includes the detailed report, uncheck **Delete temporary files on exit** from the **Options** menu before you exit OTSoft. The file will appear in the same folder as the original input file, with the name **HowIDidIt.txt**.

Macro for Output File Conversion

A bug is fixed that was producing incorrectly formatted minitableaux for ranking arguments. This also occasionally was crashing the Word macro.

Lines in tableaux separating non-crucially ranked constraints are now the industry standard dashes-with-small-gaps, rather than little dots. This makes them more noticeable.

The conversion macros for Word 97 and Word 2000 now convert the whole file to Normal format (instead of Plain Text) before doing anything else. This averts frustrations that arose in trying to edit the file in Word.

The macro now automatically saves the file in Word format rather than text format. This saves work and also averts the accidental loss of formatting that occurred earlier if the user neglected to do this task by hand.

12/27/2000

Program usability issues:

- There is now a Print menu:
 - Open the diacritic-laden file in Word for further processing (this just repeats an item already available in the View menu)
 - Send a rough copy of the output (specifically: FileName.tmp) directly to the printer. Useful for any users who are having Word problems.
 - Options for rough printing (change default printers, portrait/landscape, reduction)
- You can now start OTSoft by right-clicking on your Excel file. Instructions on how to set up this capacity are in the ReadMe.doc file.
- A file entitled **FilenameBackup.txt** is made for every file that is run. You can access it by selecting “.txt” files on the Open File screen.
- When you close the program by clicking on the little “X” in the upper right hand corner, it now saves the current filename and settings.
- When there is a blank line in the input file (specifically, no candidate in the second column, which should always be full), OTSoft now warns the user.
- The **View here** option for looking at output now swells up the screen to full size, with scroll bars visible.
- The interface now pops up properly centered on the user’s screen.
- Little tableaux for ranking argumentation now leave out some irrelevant constraints; specifically, constraints for which the two candidates are tied in having a non-zero number of violations.

Bug fix:

- Diagnostic tableaux for failed grammars: now it prints just the unrankable constraints and the relevant candidates, rather than the whole constraint/candidate set. Fancy output still needs work; which is low priority, since one doesn't usually make a fancy output for a failed grammar.

12/1/2000

Improvement in the code that reports "useless" constraints. Now, if a constraint is technically useless, but is a Faithfulness constraint violated in a winning candidate, the software will label it as such. Reason: explicit analyses should always include Faithfulness constraints violated by winners, so such constraints shouldn't really count as useless.

10/15/2000

Separate, labeled versions of the file conversion macro for Word 6, Word 97, and Word 2000. The latter gets rid of the maddening diagonal line that overwrites the input cell.

Also, a minor bug fix in the conversion macro: it now specifies "first page header different from other headers," and therefore won't hang up when the user's word processor isn't already set up this way.

9/12/2000. Two improvements from other programmers.

- The Biased Constraint Demotion algorithm of Alan Prince and Bruce Tesar is now included. The code for the algorithm was created by Prof. Bruce Tesar of Rutgers University.
- Improvements allowing more control over the learning regime of the Gradual Learning Algorithm. These were programmed by Prof. Kie Zuraw of the University of Southern California.

I welcome further contributions from other programmers; please contact [me](#) for details of how to proceed.

12/7/1999:

There is now a new version, called **OTSoft** [named 2.0, posthoc BH 2/25/03]. This version can read input files from Microsoft Excel, facilitating the editing of these files. It also has three new algorithms, including a much faster factorial typology checker. Lastly, it is Windows 95/98 software, eliminating the earlier need to worry about DOS.