BALI: A software tool to build experimental material in psycholinguistics

Christophe Coupé
Laboratoire Dynamique du Langage (CNRS – Université Lyon 2), Institut Rhône-Alpin des Systèmes complexes (IXXI)
contact: ccoupe@ish-lyon.cnrs.fr

Experimenters usually need to solve a set of constraints when building their materials. On the one hand, investigating the cognitive impact of a given factor requires contrasted modalities. On the other hand, other factors out of the focus of the experiment may be controlled and 'averaged out' to avoid loss of statistical significance or biases. Settings in experimental psychology therefore often involve items carefully sorted into a number of lists according to their features; in psycholinguistic studies, these features are usually those of words. Differences in frequency, number of letters, syllabic structure etc. are commonly either enhanced or neutralized, depending on the target of the study. Additional constraints are sometimes required: preventing similar items to repeat too closely in the lists, having all lists start with items from a specific category etc.

Building lists is a difficult and time-consuming problem. Picking and switching items while keeping an eye on averages or standard deviations require many trials and errors. Pen and paper may suffice when one restricts themselves to 2 or 3 features, but more factors usually need to be taken care of, and the problem quickly becomes intractable. Within the framework of theoretical computer science, the problem can be proven to be NP-complete; this means that it cannot be solved exactly in a number of operations proportional to a polynomial function of its "size" – in our case, typically the number of items to be sorted into lists. Solving NP problems exactly require too much computational power, and they are therefore usually better dealt with heuristic approaches yielding near-optimal results.

Software already exists which assists psycholinguists in creating lists (Van Casteren & Davis, 2006; 2007). We here propose a new tool, BALI (***Ba***lancing ***Li***sts), which goes further to offer more flexibility and ease in the construction of lists. Its heuristic search algorithm relies on genetic algorithms, an optimization paradigm which mimics natural selection in biology (Holland, 1975). Following the evolutionary idea of the 'survival of the fittest', it gradually evolves – by evaluating, selecting, mutating and reproducing - competing sets of lists to answer the constraints imposed by the user. This leads to an optimal or near-optimal result, depending on the constraints and on their potentially conflicting interactions.
At the heart of the algorithm lies a 'fitness function', which mathematically translates the user's constraints. Sets can be ranked according to it. 'Mutating' and 'reproducing' the sets are achieved by randomly switching pairs of items in their respective lists. The process is simple yet powerful when iterated many times.

Like Match (Van Casteren & Davis, 2007), our algorithm accepts any number of items and features, and produces a desired number of lists. However, relying on a carefully fitness function opens a wide range of possibility. Firstly, it can simultaneously contrast some features *and* balance others, and these features may take continuous values (e.g. frequencies) *or* belong to finite sets (e.g. syntactical categories or gender). Secondly, rather than being only divided into different unsorted lists, items are actually organized in a *2-dimensional array*. There are therefore interlaced horizontal lists – lines – and vertical lists – columns. This allows for more freedom when preparing experimental material, e.g. defining two

balanced lists of primes and targets where each couple (prime-target) is additionally balanced with respect to the same or other features. Thirdly, constraints *between* and *within* rows – lines or columns – of the 2-dimensional array may be combined, when they were previously addressed by distinct tools. It is hence possible for example to build ordered lists where words are always separated by at least one pseudo-word, and which are moreover balanced in terms of number of letters and contrasted in terms of morphological structures.
The constraints readily available in BALI allow addressing a wide range of experimental settings. However, for specific needs, a programmer may easily modify the code to devise new constraints.

BALI offers an intuitive and graphic interface without command lines. Input and output files are easily managed with Excel, and file reading and writing procedures are Unicode compliant. Although the software's primary function is not to prepare the dataset from which items will be picked to build lists, it offers a few filtering functions that allow reading large inventories like Lexique (New, 2006) and then reduce the number of entries, e.g. to keep only nouns or verbs. On the one hand, this leaves the door open for diversified items to be matched and selected. On the other hand, working with large datasets may lead to resulting lists containing undesirable items, e.g. words unsuitable for children or simply unnatural in the context of the experiment. It is however possible to solve this issue by adding new constraints impeding these specific items and rerunning the optimization process to eventually reach a new near-optimal result that will be free of unwanted elements.

We will introduce our software and illustrate its performances with examples from concrete cases.

References:

Holland, J. (1975). Adaptation in Natural and Artificial Systems. The MIT Press.

New, B. (2006). Lexique 3 : Une nouvelle base de données lexicales. Actes de la Conférence Traitement Automatique des Langues Naturelles, avril 2006, Louvain, Belgique.

Van Casteren, M. & Davis, M. H. (2006). Mix, a program for pseudorandomization. *Behavior Research Methods*, 38(4), pp. 584-589.

Van Casteren, M. & Davis, M. H. (2007). Match: A program to assist in matching the conditions of factorial experiments. *Behavior Research Methods*, 39(4), pp. 973-978.