Behavior Research Methods

U-Learn : Finding optimal coding units from unsegmented sequential databases

Journal:	*Behavior Research Methods*
Manuscript ID:	Draft
Manuscript Type:	Original Manuscript
Date Submitted by the Author:	n/a
Complete List of Authors:	Perruchet, Pierre; Universite de Bourgogne, LEAD Robinet, Vivien; INRIA Lemaire, Benoit; Universite Pierre Mendes France

U-Learn : Finding optimal coding units from unsegmented sequential databases.

Pierre Perruchet[1], Vivien Robinet[2], and Benoît Lemaire[3]

[1] CNRS and University of Bourgogne, France

[2] INRIA, France

[3] CNRS and University of Grenoble, France

Running Head: U-Learn

* Corresponding author:

Pierre Perruchet
Université de Bourgogne
LEAD/CNRS UMR5022
Pole AAFE, Esplanade Erasme
21000 Dijon, France

email: pierre.perruchet@u-bourgogne.fr

The programs and the User Manual can be downloaded at the following URL: http://leadserv.u-bourgogne.fr/~perruchet/.

Abstract

Finding optimal coding units from unsegmented sequential databases (e.g., finding the words from a continuous speech stream) can be solved by looking for the unit boundaries, often defined as the point where the predictability of the next element of the sequence is the lowest. However, other models rely on a very different strategy, in which chunks are built progressively by the concatenation of the initial primitives, then selected through some kind of competition process between different segmentation modes. This paper introduces to U-Learn, a Windows-based, user friendly software, which implements two representative chunk-based models: The MDLChunker (Robinet, Lemaire, & Gordon, 2011), which relies on a Minimum Description Length method such as used in standard compression algorithms, and PARSER (Perruchet & Vinter, 1998), which relies on basic principles of associative learning and memory. U-Learn allows to generate corpora from a list of items and the desired frequency for each item, with a large number of options, or alternatively, to start from an existing database (such as a child-directed language). There are two running modes. The *step by step* mode is set up for maximum transparency in terms of access to all the operations performed by the models on a single run, while the '*normal*' mode allows efficiently performing and analyzing simulations over several runs. Due to its modular design, U-Learn may be easily complemented with other models or some variants of the initial models.

U-Learn : Finding optimal coding units from unsegmented sequential databases.

Background

Finding optimal coding units is a pervasive issue in artificial intelligence. For instance, research on data compression has lead to the creation of a number of algorithms aimed at replacing a set of consistent events by a single coding unit. However, the very same issue arises in cognitive science, because adaptive purposes require that the human mind codes the overwhelming complexity and diversity of the events occurring in the world by a manageable number of internal representations. There is considerable evidence that the mind is especially efficient in this task, and a number of psychological phenomena may be conceived of as a compression process. For instance, categorizing consists in searching for a unique label to uncover a set of different objects or events. Likewise, performance in memory tasks improves as the information the learner has to remember can be encoded in a more compressed fashion (e.g., Brady, Konkle, & Alvarez, 2009). More generally, several models of the mind are based on the notion that the formation of relevant representational units is essential for adaptive purposes (Pothos & Wolff, 2006, Wolff, 1991). Perruchet and Vinter (2002) have proposed that understanding how conscious representations become increasingly isomorphic with the world should be the primary objective of the psychological research.

Making the issue computationally tractable

Such as framed above, however, the issue is exceedingly general, and any empirical or computational approach needs to focus on a more manageable problem. The main conditions that stand as a necessary prerequisite for the computational models implemented in U-Learn are twofold: (1) the original data set must be composed from a *sequence* of units, and (2) the units must be composed themselves from a sequence of one or several *contiguous* primitives. Even with those restrictions, the issue remains wide-ranging, because the primitives, that is, the elements that are considered as undividable processing entities for a given learner at a given moment, may belong to various sensory modalities or domains: They may be phonemes, graphemes, syllables, notes of music, spatial locations, response signals, and so on. All these potential domains of application have not been considered with equal attention in the literature, however. Since the seminal studies of Saffran et al. (1996), artificial languages composed of unsegmented sequences of oral syllables have become the paradigmatic situations to investigate this kind of issues. Due to the prevalence of this paradigm, the terminology used below is borrowed from the language domain. Accordingly, the primitives are called the *syllables*, the relevant units are called the *words*, and the whole corpus may be composed of one or several *sentences*, each sentence comprising a variable number of words. Of course, this terminology is only used for the sake of convenience, and in no way implies that the domain of application of U-Learn is restricted to lexicon formation.

In addition to the two conditions above, another limiting characteristic is that only the distributional information is considered in looking for the relevant units. Considering the acquisition of the lexicon in natural settings, for instance, it has been shown for long that the discovery of words by infants also depends on phonological, prosodic, and contextual cues (e.g., Creel, Tanenhaus, & Aslin, 2006; Curtin, Mintz, & Christiansen, 2005; Dahan & Brent, 1999; Johnson & Jusczyk, 2001; Onnis, Monaghan, Chater, & Richmond, 2005; Perruchet & Tillmann, 2010; Thiessen & Saffran, 2003). However, the point of interest is that experimental studies have demonstrated that infants, children, and adults were able to extract word-like units from continuous artificial languages without any phonological or prosodic markers (e.g., Giroux & Rey, 2009; Graf Estes et al., 2007; Perruchet & Poulin-Charronnat, submitted; Saffran, 2001; Saffran & Wilson, 2003). These results strongly suggest that learners exploit the statistical information embedded in the speech stream, and notably the fact that the relationships between word-internal syllables are more consistent than the relationships between syllables straddling word boundaries. The models implemented in U-Learn are designed to account for this specific ability.

Bracketing vs. Clustering approaches

How learners exploit statistical information is the topic of a growing literature. In the context of artificial languages, word segmentation is generally attributed to the ability of participants to compute transitional probabilities between successive elements (Aslin, Saffran & Newport, 1998).

Chunks would be inferred from their boundaries, which are themselves defined as the points where the predictability of the next element is the lowest. This interpretation is largely prevalent in statistical learning research, both for oral stimuli (e.g. Saffran, 2001) and for visual scenes (e.g. Fiser & Aslin, 2005). This prevalent approach is sometimes coined as the *bracketing* approach (Swingley, 2005), because chunks are inferred from the knowledge of their boundaries. By contrast, the *clustering* approach (Swingley, 2005) posits that chunks are created incrementally, with the sensitivity to transitional probabilities being a by-product of this process (e.g., Frank, Goldwater, Griffiths & Tenenbaum, 2010; Perruchet & Vinter, 1998; Robinet, Lemaire, & Gordon, 2011; Servan-Schreiber & Anderson, 1990).

At the computational level, the bracketing approach is generally implemented by connectionist networks, most often Simple Recurrent Networks (SRN, e.g. Christiansen et al., 1998), because the distribution of activations in the output layer of SRNs provide a very good approximation of transitional probabilities, which are thought of as essential to set word boundaries. For the neural network connectionist models, including the SRNs, Ruh and Westermann (2009) have recently presented in this Journal a software that allows a quick and easy start to this form of modeling (OXlearn). Up to now, there is no equivalent software for the chunk-based models. We suspect that this may be at least one of the reasons why only a handful of studies (e.g., Frank et al., 2010; Giroux & Rey, 2009; Perruchet & Tillmann, 2010, Robinet, Lemaire & Gordon, 2011) have undertaken a comparison between the predictions and the level of achievement of the different models. Indeed, models are not always described in exhaustive ways in the literature, and even if they are, implementing a new model is a risky and time consuming endeavor. The primary objective of U-Learn (U stands for Units) is to provide a user-friendly software to run representative chunk-based models, hence filling the same objective as OXlearn for the connectionist models.

The clustering models

Instead of looking for units' boundaries, the general strategy shared by all the chunk-based models is that chunks are built progressively by the concatenation of the initial primitives, then selected through some kind of competition process between different segmentation modes. How chunks are generated and selected substantially differ between models. To put it shortly, two general approaches, which differ by their origin and the deep structure of model's algorithms, may be distinguished. To anticipate, the present version of U-Learn includes models that are representative of each approach: the MDLChunker (Robinet et al., 2011), and PARSER (Perruchet & Vinter, 1998). As a consequence, the presentation of these models will be privileged in the brief outline that follows.

The first approach originates from artificial intelligence research. It is based on the idea that a chunk is created whenever the overall representation of the data when this chunk is used as a coding unit becomes simpler than before chunk creation. But what does "simpler" mean in this context? Creating a chunk changes the complexity of a system along two dimensions: There is one more chunk to store, but, at the same time, an opportunity for representing the data in another way, which could be shorter. For instance, grouping some syllables into a word may result in a shorter way of representing new sentences, although a new word has to be managed. Therefore, there is a trade-off between the cost of representing a new chunk and the cost of representing the data. A common approach to solve this problem is to use the Minimum Description Length principle (MDL, Rissanen, 1978). This method consists in computing the length of the codes for representing the description of the chunks (the lexicon) and the length of the codes for representing the input data rewritten using the lexicon, and minimize their sum. Codelengths are estimated by means of Shannon's formula, according to which a symbol s, occurring with probability p, can be ideally compressed with a binary code whose length is $-\log_2(p)$. In our case, p is estimated by the frequency of s. Frequent items have therefore short codelengths.

Considering models of the form "data + set of chunks", the MDL principle can be viewed as a selection criterion able to select the most compressed model among competing ones, which is therefore the most plausible one if we assume that cognition is compression. In fact, this search for simplicity has been proposed as a general mechanism of cognition (Chater & Vitanyi, 2003). Obviously, this problem is not computationally easy. Many chunks may be candidates for creation at each step and the rewriting of stimuli in terms of chunks is not unique (given chunks abc and cd, abcde can be rewritten as abc+d+e or a+b+cd+e).

Brent & Cartwright (1996) used the MDL principle to account for the ability of children to segment a stream of phonemes into words. Their model generates all possible segmentations of the input stream and selects the one with the shortest codelength. This model was applied to transcripts of child-directed speech and showed a good ability to recover the words.

As opposed to this off-line approach, MDLChunker (Robinet et al., 2011) works online. It predicts the time course of the creation of chunks. It works in the following way. First, the beginning of the current input stream is rewritten using the existing chunks, in a way that minimizes its codelength. Actually, not the whole input stream is rewritten, but only a sub-part of fixed size called "focus". This focus may be viewed as a buffer containing the information currently processed, whose size is the sum of chunks' codelengths in the buffer. The first two units are then candidates for forming a chunk. If the creation of this chunk leads to a smaller codelength of the system, composed of the data plus the set of chunks, then the chunk is created.

Because it would be cognitively implausible to consider all the data processed so far in the rewriting step, another limited buffer is considered. Its size (also in terms of codelength) is a parameter of the model. MDLChunker has been shown to reproduce the time course of the creation of chunks in a spatial environment (Robinet et al., 2011) as well as a specific sub-word effect in a segmentation task (Robinet & Lemaire, 2009).

In the second general approach, the researchers' primary motivation is to account for human behavior in terms of psychologically plausible processes. Two main models have been proposed: the Competitive Chunking model (Servan-Schreiber and Anderson, 1990) and PARSER (Perruchet & Vinter, 1998), which are both an application of a general view of the mind to a specific issue involving chunking. The Competitive Chunking model is an application to artificial grammar learning of the ACT* model of Anderson (e.g., 1983). Likewise, PARSER (Perruchet & Vinter, 1998) is a computationally implemented version of the Self-Organizing Consciousness concept proposed by Perruchet and Vinter (2002), which is aimed at accounting how conscious representations become increasingly isomorphic with the world structure. Perruchet and Vinter (2002) have characterized the organization of the cognitive system as the interplay of two interrelated principles. The first principle stipulates that perception shapes internal representations. This means that the primitives that are perceived within one attentional focus as a consequence of their experienced spatial or temporal proximity (i.e., they are perceived as a chunk) become the constituents of one new representational unit. The future of this provisional unit, they argued, depends on ubiquitous laws of associative learning and memory. If the association between the primitives that form a provisional unit is not strong enough in the language, this representation rapidly vanishes, as a consequence of both natural decay and interference with the processing of similar material. However, if the degree of cohesiveness between the primitives is sufficient, the internal representation is progressively strengthened.

The second principle is that internal representations guide perception. Perception involves an active coding of the incoming information constrained by the perceiver's knowledge. Internal representations serve as perceptual primitives. Because the representational landscape changes with increased experience in a domain, perception, and notably the composition and the size of the perceived chunks, also evolves. The resulting picture is that perception builds the internal representations which, in turn, guide further perception, hence leading to the self-organization of the mind.

These principles have been exploited in PARSER to discover words from a nonsegmented speech flow (Perruchet & Vinter, 1998). How does PARSER work? Based on the observation that, in humans, attentional coding of the ingoing information naturally segments the material into disjunctive parts, the model is provided online with a succession of candidate units, some of them relevant to the structure of the language and others irrelevant. According to the first principle described above, an internal representation that matches a percept is reinforced in the model if its components are cohesive and occur repeatedly in the input. This means that a word or a part of a word are more likely to create a long-lasting internal representation than between-word segments. The relevant units emerge through a selection process based on forgetting. Forgetting due to both decay and interference leads to the selection of the most cohesive parts among all parts generated by the initial, presumably mostly irrelevant, chunking of the material. The second principle described above ensures the convergence of this process toward an optimal parsing solution. The fact that perception is guided by internal representations allows the system to build representations of words whose components could hardly be perceived in one attentional focus if perception were driven only by the initial primitives in the language. Also, once internal representations providing an appropriate coding of the input have been built, an endless generation of new candidate units is avoided.

Several studies report simulation studies using either a MDL-based approach (e.g., Brent, 1995) or PARSER (e.g., Giroux & Rey, 2009; Perruchet & Tillmann, 2010), but to our best knowledge, only two studies (Frank et al., 2010; Robinet et al., 2011) provide a direct comparisons between the two models. One of the more surprising conclusions given the striking structural differences between the models is the relative similarity of outcomes. But needless to say, additional studies are needed to explore further the relationships between the two chunk-based models, as well as the relationships of these models with the clustering approach.

U-learn: main functionalities

As mentioned above, two models are implemented in the present version of U-Learn: the MDLChunker (Robinet al., 2011), and PARSER (Perruchet & Vinter, 1998). A simple click on the appropriate button allows to shift from one model to another, hence ensuring a very easy comparison of the models' results starting from the very same database. However, the program has been designed to be easily complemented by other models or some variants of the initial models. Indeed, the program can play the role of an interface to launch any executable files, with simple text files (the content of which is exhaustively depicted in the User manual of U-Learn) ensuring the bidirectional transfer of data. New models could be integrated in the current version upon modeler's agreement.

The opening window of U-learn is shown in Figure 1. Skimming through the options allows to outline the main possibilities of the software.

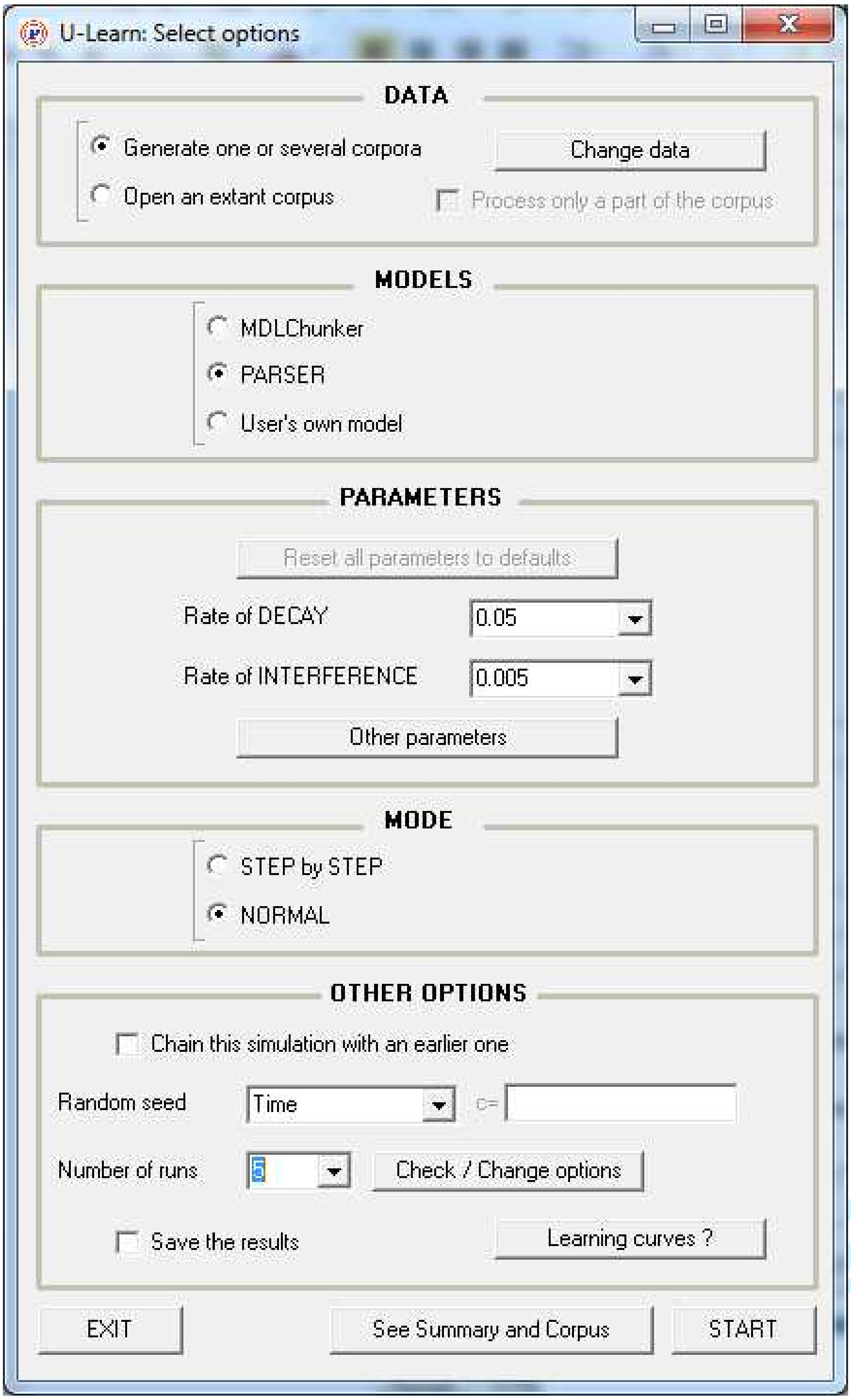

Figure 1: The opening window

- U-Learn can generate a corpus from a list of items and the desired frequency for each item. A very large number of options are available, such as the presence/ absence of immediate repetition and the introduction of hard boundaries (e.g., to simulate the exposure to physically separate sentences). In addition, the corpus may be divided into sections, with each section having its own parameters. For instance, the corpus may begin with certain words as isolated utterances, followed by a continuous language composed of the same or other words. More generally, this allows to present models with sequential information that is *not* uniformly distributed across the entire corpus (e.g., Gebhart et al., 2009). The algorithms of randomization avoid the flaws that have been described by French and Perruchet (2009) in the usual algorithms. The random seed may be a pseudo-random value (the current time), but may also be user-defined to generate the same sequence upon request.

- Although the generation option allows to build a large diversity of corpora, there are cases where one may wish to enter a specific dataset, such as an excerpt of child-directed language. If each character (e.g., a letter or a phoneme) stands for a primitive, any text file can be loaded as such. If primitives, or at least some of them, comprise two or more characters, separators (/) must be inserted throughout the corpus. Practically: if the user considers the syllables as the primitives of the language, *a baby* must be written "/a/ba/by/. In addition, a hard boundary (e.g., a perceptually salient pause preventing learners from linking the two surrounding primitives) must be marked by "//". Note the option "Chain this simulation with an earlier one", whose the recursive usage makes it possible to process a virtually infinite database.

- There are two main running modes: *step by step* and *normal*. '*Step-by-step*' provides a detailed analysis on a single run, while '*normal*' only provides the final results of a multi-run session. The two modes have very different objectives. The '*normal*' mode should be used the most often, notably because it allows to efficiently perform and analyze simulations over an unlimited number of runs. The results for each run are shown and can be saved upon request, but the details of the computations are not available. By contrast, the '*Step-by-step*' mode is set up for maximum transparency in terms of access to all the operations performed by the models on a single run. At any given moment during training, the user is shown the state of the system memory at Step *n-1* and Step *n*, and the currently processed part of the corpus that triggered the changes between the two states. This option may be useful for the aim of understanding how the models work, but it may also be useful to proficient users who may want to examine, for instance, how a surprising pattern of results may have emerged on a particular run under the *normal* mode.

- The results are displayed in text format, and individual and/or averaged learning curves are also plotted (but cannot be edited). The scores of completeness (the proportion of units that are extracted) and precision (the proportion of actual units among the extracted units) are also provided. All the data can be saved as text files, hence allowing to run further analyzes, and/or to plot more sophisticated learning curves with other, specially designed softwares.

Starting with U-Learn

U-Learn may be freely downloaded at the following URL: http://leadserv.u-bourgogne.fr/~perruchet/. It is currently composed of three executable files:

-1 U-Learn.exe. This is the main interface, and the only program that the user has to launch. This is a Windows-based program, which has been used extensively under WindowsXP and Windows7.

2- Parser.exe

3- MDLCh.exe.

The executable files #2 and #3 are called by the main interface, as a function of the model in use. Locate all three files in the same folder make things easier. However, if the program doesn't find the appropriate .exe file, then the user is required to indicate the path of this file through a standard Windows dialog box.

A user manual may be downloaded at the same URL (UserManual.pdf). Its use should be limited, given that the standard operations are quite intuitive. The artificial languages used in a number of papers are directly available in the software (*'Generate one or several corpora*-> *Ready-to-use configurations*), so doing simulations on most sets of previously published data only requires a few clicks.

References

Aslin, R. N., Saffran, J. R., & Newport, E. L. (1998). Computation of conditional probability statistics by 8-month-old infants. *Psychological Science*, *9*, 321-324.

Brady, T. F. , Konkle, T., & Alvarez, G. A. (2009). Compression in Visual Working Memory: Using Statistical Regularities to Form More Efficient Memory Representations. *Journal of experimental psychology. General*, *138*, 487-502.

Brent, M. (1999). Speech segmentation and word discovery: A computational perspective. *Trends in Cognitive Sciences*, *3*, 294-301**.**

Brent, M., & Cartwright, T. (1996). Distributional regularity and phonotactic constraints are useful for segmentation. *Cognition*, 61(1-2), 93-125.

Chater, N., & Vitanyi, P. (2003). Simplicity : a unifying principle in cognitive science? *Trends in Cognitive Sciences*, 7(1), 19-22.

Christiansen, M.H. et al. (1998) Learning to segment speech using multiple cues: A connectionist model. *Language and Cognitive Processes*, *13*, 221–268.

Creel, S. C., Tanenhaus, M. K., & Aslin, R. N. (2006). Consequences of lexical stress on learning an artificial lexicon. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *32*, 15-32.

Curtin, S., Mintz, T. H., & Christiansen, M. H. (2005). Stress changes the representational landscape: Evidence from word segmentation. *Cognition*, *96*, 233-262.

Dahan, D., & Brent, M. R. (1999). On the discovery of novel word-like units from utterances: An artificial-language study with implications for native-language acquisition. *Journal of Experimental Psychology: General*, *128*, 165-185.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*, 179-211.

Fiser, J. and Aslin, R.N. (2005) Encoding multielement scenes: Statistical learning of visual features hierarchies. *Journal of Experimental Psychology: General*, *134*, 521–537.

Frank, M. C., Goldwater, S., Griffiths, T., & Tenenbaum, J. B. (2010). Modeling human performance in statistical word segmentation. *Cognition, 117*, 107-125.

French, R. M., & Perruchet, P. (2009). Generating constrained randomized sequences: Item frequency matters. *Behavior Research Methods*, *41*, 1233-1241.

Gebhart, A.L., Aslin, R.N., Newport, E.L. (2009) Changing structures in midstream: Learning along the statistical garden path. *Cognitive Science*, *33*, 1087-1116.

Giroux, I., & Rey, A. (2009). Lexical and sub-lexical units in speech perception. *Cognitive Science*, *33*, 260-272.

Graf Estes, K. M., Evans, J., Alibali, M. W., & Saffran, J. R. (2007). Can infants map meaning to newly segmented words? Statistical segmentation and word learning. *Psychological Science*, *18*, 254-260.

Johnson, E., & Jusczyk, P. (2001). Word segmentation by 8-month-olds: When speech cues count more than statistics. *Journal of Memory and Language*, *44*, 548-567.

Onnis, L., Monaghan, P., Chater, N., & Richmond, K. (2005). Phonology impacts segmentation in online speech processing. *Journal of memory and language*, *53*, 225-237.

Perruchet, P., & Pacton, S. (2006). Implicit learning and statistical learning: Two approaches, one phenomenon. *Trends in Cognitive Sciences*, *10*, 233-238.

Perruchet, P., & Tillmann, B. (2010). Exploiting multiple sources of information in learning an artificial language: Human data and modeling. *Cognitive Science*, *34*, 255-285.

Perruchet, P., & Vinter, A. (1998). PARSER: A model for word segmentation. *Journal of Memory and Language*, *39*, 246-263.

Perruchet, P., & Vinter, A. (2002). The Self-Organizing Consciousness. *Behavioral and Brain Sciences*, *25*, 297-388.

Pothos, E.M., & Wolff, J.G. (2006). The Simplicity and Power model for inductive inference. *Artificial Intelligence Review*, *26*, 211-225.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, 14(5), 465-471.

Robinet, V., & Lemaire, B. (2009). MDLChunker: a MDL-based model of word segmentation? In N.A. Taatgen & H. van Rijn (Eds.), *Proceedings of the 31th Annual Conference of the Cognitive Science Society* (pp. 2866-2871). Austin, TX: Cognitive Science Society.

Robinet, V., Lemaire, B., Gordon, M.B. (2011). MDLChunker: a MDL-based Cognitive Model of Inductive Learning. *Cognitive Science*, 35, 1352-1389.

Ruh, N. & Westermann, G. (2009) OXlearn: A new MATLAB-based simulation tool for connectionist models. *Behavior Research Methods*, *41*, 1138–1143

Saffran, J. R. (2001) Words in a sea of sounds: The output of statistical learning. *Cognition*, *81*, 149-169.

Saffran, J. R., Aslin, R. N., & Newport, E. L. (1996). Statistical learning by 8-month-old infants. *Science, 274*(5294), 1926-1928.

Saffran, J. R., & Graf Estes, K. M. (2006). Mapping sound to meaning: Connections between learning about sounds and learning about words. In R. Kail (Ed.), Advances in Child Development and Behavior. New York: Elsevier (p. 1-38).

Saffran, J. R., Newport, E. L., & Aslin, R. N. (1996). Word segmentation: The role of distributional cues. *Journal of Memory and Language*, *35*, 606-621.

Saffran, J. R., Newport, E. L., Aslin, R. N., Tunick, R. A., & Barrueco, S. (1997). Incidental language learning: Listening (and learning) out of the corner of your ear. *Psychological Science, 8*(2), 101-105.

Saffran, J. R., & Wilson, D.P. (2003). From syllables to syntax: Multilevel statistical learning by 12-month-old infants. *Infancy*, 4, 273-284.

Swingley, D. (2005). Statistical clustering and the contents of the infant vocabulary. *Cognitive Psychology, 50,* 86-132.

Thiessen, E. D., & Saffran, J. R. (2003). When cues collide: Use of stress and statistical cues to word boundaries by 7- to 9-month-old infants. *Developmental Psychology, 39*, 706-16.

Wolff, J.G. (1991). *Towards a theory of cognition and computing*. Ellis Horwood Upper Saddle River, NJ, USA.