HAL
open science

Merging Spatial Coding and Open Bigrams theories of the orthographic coding

Pierre Courrieu, Sylvain Madec, Arnaud Rey

To cite this version:

Pierre Courrieu, Sylvain Madec, Arnaud Rey. Merging Spatial Coding and Open Bigrams theories of the orthographic coding. 2018. hal-01687304v1

HAL Id: hal-01687304

https://hal.science/hal-01687304v1

Preprint submitted on 18 Jan 2018 (v1), last revised 23 Apr 2019 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Merging Spatial Coding and Open Bigrams theories of the orthographic coding

Pierre Courrieu, Sylvain Madec, and Arnaud Rey

Laboratoire de Psychologie Cognitive, CNRS & Aix-Marseille University, Marseille, France

Running Head: Orthographic coding

Corresponding author:

Pierre Courrieu

Laboratoire de Psychologie Cognitive

CNRS & Aix-Marseille Université

3, place Victor Hugo - Case D

13331 Marseille Cedex 03 – France

Authors E-mail:

P. Courrieu: courrieu@free.fr, pierre.courrieu@univ-amu.fr

S. Madec: symadec@gmail.com

A. Rey: arnaud.rey@univ-amu.fr

Abstract. Simple numerical versions of the Spatial Coding and of the Open Bigrams coding of character strings are presented, together with a natural merging of these two approaches. Comparing the predictive performance of these three orthographic coding schemes on orthographic masked priming data, as well as on lexical decision and word naming data, we observe that the merged coding scheme always provides the best performance, and that both the spatial coding component and the open bigrams component provide specific and significant contributions. While the open bigrams component provides the largest contribution to the fits, the spatial coding component allows the code to be decodable in all cases.

Key words. Orthographic Code; Spatial Coding; Open Bigrams; Orthographic Regressors

1. Introduction

In recent years, there has been a rapid development in the use of large-scale databases as a tool to study single word reading (Balota, Yap, Cortese, Hutchison, Kessler, Loftis, Neely, Nelson, Simpson, & Treiman, 2007; Ferrand, New, Brysbaert, Keuleers, Bonin, Méot, Augustinova, & Pallier, 2010; Keuleers, Diependaele, & Brysbaert, 2010; Keuleers, Lacey, Rastle, & Brysbaert, 2012). Such large-scale studies have been used to test item-level predictions of word naming or lexical decision performance derived from computational models (Perry, Ziegler, & Zorzi, 2007, 2010), as well as from regression models (Yap & Balota, 2009).

Previous studies using regression modeling approaches have addressed the issue of looking at the influence of various variables, including orthographic and phonological ones, in accounting for item level variance in lexical decision and word naming times (e.g., Yap & Balota, 2009). Commonly used variables are item properties and statistics (number of letters, word log-frequency, orthographic and phonological neighborhood counts, ...), and in the case of word naming response times, variables such as the articulatory features of the first and second phonemes of the word, which are known to influence the measured naming time, mainly due to the differential sensitivity of naming response recording devices to the various phonemes (Rastle, Croot, Harrington, & Coltheart, 2005; Rastle & Davis, 2002 ; Rey, Courrieu, Madec, & Grainger, 2013).

Another very promising application field of item level regressors is the analysis of electrophysiological data such as cerebral event related potentials (ERP) in various psycholinguistic tasks. For instance, Hauk, Davis, Ford, Pulvermüller, and Marslen-Wilson (2006) computed the correlations of various word-descriptors with ERPs in a lexical decision task, which allowed these authors to identify the time windows and scalp locations associated

with various sub-processes in printed word recognition. A quite similar approach was successfully used by Rey, Madec, Grainger, and Courrieu (2013), with word-descriptors similar to those described hereafter, in a speeded word-naming task.

However, except in the last cited study, the tested regression models never included variables completely describing the stimulus, such as orthographic coding and phonological coding, despite the fact that such variables probably relate to a major part of the process of word recognition, and thus to a significant part of the behavioral and electrophysiological data. Full simulation computational models always need to encode the stimulus in order to work, however, regression models can omit this encoding just because they do not need to be complete. One of the possible reasons for not testing complete stimulus description variables is the lack of suitable theoretical modeling of such descriptions, which is also a challenge for full simulation models.

One of the main families of orthographic coding models available to date is based on the concept of "open bigrams" (Grainger, Granier, Farioli, Van Assche, & van Heuven, 2006; Hannagan & Grainger, 2012; Whitney, 2001). An open bigram is an ordered pair of non-necessarily adjacent characters, and the open bigrams code of a character string is basically the list of all the open bigrams of the string (e.g. {SO, SN, ON} for the word SON). Open bigrams are commonly associated with numerical values depending on the gap between the two symbols in the string and the number of occurrences of the open bigram. For instance, following Hannagan and Grainger (2012), the open bigram ME appearing in the string MEMES is associated with the numerical value $2\lambda^2+\lambda^4$ (for some real λ such that $0>\lambda>1$) because the open bigram ME appears two times in subsequences of two characters, and one time in a subsequence of four characters. Unfortunately, several empirical arguments against the open bigrams coding theory have recently been stated (Davis & Bowers, 2006; Kinoshita

& Norris, 2013; Lupker, Zhang, Perry, & Davis, 2015), so that there is now a serious doubt about the capability of this family of models to make relevant predictions.

Another important family of orthographic coding models is based on the concept of "spatial coding" (Davis, 1999, 2010). The spatial coding principle originates from Grossberg's theory of the encoding of event sequences (Grossberg, 1978; Grossberg & Pearson, 2008). The spatial coding model developed by Davis is a complete simulation model of visual word identification, including a number of possibly realistic but complex features. Empirical arguments supporting the spatial coding principle in word recognition can be found in the paper of Davis and Bowers (2006).

In short, in spatial coding, one associates a dedicated detector to each possible symbol of an alphabet, this detector being activated when an input string includes the corresponding symbol. In the simplest approaches, the activation value of each detector depends on the serial position of the corresponding symbol in the current input symbol string. For instance, Davis (1999) suggested that the activation at time T for a character appearing at the ith position in a string is of the form $\mu\omega^{(T-i)}$, for some real μ and $\omega>1$. There was a difficulty whenever several occurrences of the same symbol appeared in the same string. Davis (1999, 2010) solved this problem assuming that there are several detectors for each alphabetical character, that is, one detector for each possible occurrence of this character in a string. This requires that one a priori fixes the maximum number of occurrences of a given character in a string (e.g. four in common English words), and that the total number of nodes (code length) is equal to the alphabet length time the maximum number of occurrences. A simplified approach of the spatial coding of character strings was proposed by Courrieu (2012) to encode the output of handwritten words recognition systems. It allows one to compactly encode every symbol string in the form of a fixed length numerical vector. An important property of spatial coding models is that every code vector can be exactly decoded back into the corresponding symbol

string, which guaranties that the code completely and unequivocally represents the string, and allows one to use it as a decodable numerical output of various systems. The code format also allows one to use such orthographic codes as multidimensional predictors in regression analyses.

However, this leads to some methodological difficulties. High dimension independent variables tend to mechanically account for a large part of the data variance in multiple regression analyses, even if they are purely random, and it is known that the usual R^2 statistic is positively biased. This problem can be partially solved using the so-called "adjusted R^2", denoted $\underline{R}^2$ hereafter, which is a well-known unbiased estimator of the corresponding population parameter, and is designed to be independent of the regressor dimension (Cohen, Cohen, West, & Aiken, 2003, pp. 83-84; Theil, 1961, p. 212). In its usual formulation, the adjusted R^2 is given by: $\underline{R}^2 = 1 - (m-1)(1- R^2)/(m-k-1)$, where m is the number of items, and k is the dimension of the regressor. One can easily see in this formula that R^2 and $\underline{R}^2$ tend to equality when m is much greater than k. However, it remains that high dimension regressors cannot be used with moderate size data sets, and it is usually recommended to have at least 10-20 times more items than we have regressor dimensions, in order to obtain suitable statistics. So, if we want to use a high dimension regressor, then we must necessarily have a very large data set, while if we have only a moderate size data set, then we must necessarily lower the regressor dimension. A possible approach to this problem is to build and validate low dimension regressors from high dimension ones on large item databases, which allows one to use the obtained low dimension regressors for moderate size data sets. This type of approach will be studied hereafter.

The cross-database generalization of regressors leads to another difficulty. Depending on the data collection process and on the number of measures per item, the data accuracy and consistency are not necessarily the same in distinct databases, and this substantially affects the

regression statistics (Courrieu and Rey, 2011). Under quite general conditions, the proportion of systematic variance in a vector of item means (as those provided by large scale behavioral databases) can be estimated using an intraclass correlation coefficient of type ICC(2, k), the remaining variance being random (Courrieu, Brand-D'Abrescia, Peereman, Spieler, & Rey, 2011; Courrieu & Rey, 2011, 2015; McGraw & Wong, 1996; Rey & Courrieu, 2010; Rey, Courrieu, Schmidt-Weigand, & Jacobs, 2009; Shrout & Fleis, 1979). The expected value of the ICC(2, k) depends on the signal to noise ratio in the data and on the number of observations per item, while its variance also depends on the number of items. Large-scale behavioral databases usually include many missing data, in addition to the fact that most of these databases were collected using an incomplete design. As a result, the item variance is contaminated by the participant effect, and the computation of the ICC on the raw data is biased. Fortunately, if one uses the z-scores instead of the raw data (Faust, Balota, Spieler, & Ferraro, 1999), then this difficulty vanishes because the participant effect reduces to zero and one can suitably estimate the ICC (Courrieu & Rey, 2011, pp. 314-315). One can show that if r (or R) is the correlation (or multiple correlation) coefficient between the item means of a data set and a regressor, and if ICC denotes the ICC(2, k) of this data set, then the ratio r^2/ICC (or R^2/ICC) is stable and does not depend on the data precision (Courrieu & Rey, 2011), which solves our problem.

In the following, we will first define numerical orthographic codes belonging to the spatial coding and to the open bigrams coding families of models. Then we will see that there is a natural way of merging these codes, which solves some critical problems. We provide some preliminary test of the models on available masked orthographic priming data. After this, we define a method for building, validating, and reusing generalizable one dimension orthographic regressors from the high dimension orthographic codes. Then we apply this method to three large scale behavioral databases and we test the generalization power of the

obtained regressors in various ways, before concluding. Useful computer programs in Matlab/Octave code are provided in Appendix, for practical use of the proposed tools.

2. Spatial Coding (SC)

2.1 Code definition

Consider an alphabet of n symbols $\{s_1, s_2, ..., s_n\}$, for instance the 26 lower-case letters of the Roman alphabet. The spatial coding associates to each symbol of the alphabet one component of a real vector $(c_1, c_2, ..., c_n)$. Let X be a symbol string of m characters, one first determines the "symbol position bits" as $b_{k,i} = 1$ if the symbol s_i appears at rank k in X, else one has $b_{k,i} = 0$. Then the components of the orthographic code are given by:

$$c_i(X) = (\Sigma_{k=1..m}\, b_{k,i}\, 2^{-k})^p \ , \ i=1..n,\ 0 < p \leq 1, \qquad (1)$$

where p is a free parameter. For instance, the code for the word "parabola", in the 26 letters Roman alphabet, is the following 26 components vector:

C(parabola) =

$$[(2^{-2}+2^{-4}+2^{-8})^p, (2^{-5})^p, 0,0,0,0,0,0,0,0,0, (2^{-7})^p, 0,0, (2^{-6})^p, (2^{-1})^p, 0, (2^{-3})^p, 0,0,0,0,0,0,0,0].$$

The encoding of strings can be performed using the function "str2scob" listed in Appendix, and examples of string spatial codes are visualized in Figure 1.

2.2 Decoding

Such a code can be completely and unequivocally decoded back into the corresponding character string in all cases. If a component is zero, then the corresponding character does not appear in the string. For each non-zero component, the corresponding character appears one or several times in the string. To know where it appears, it suffices to raise the component to the power $1/p$, and to compute the binary form of the result. The non-zero bits of this form correspond to the symbol position bits. For instance, in the above

example of the word "parabola", consider the code vector component corresponding to the letter "a", its value is $(2^{-2}+2^{-4}+2^{-8})^{p}$. Raising this value to the power *1/p*, we obtain $(2^{-2}+2^{-4}+2^{-8})=0.31640625$, whose binary form is (.01010001), which indicates that the letter "a" appears at ranks 2, 4, and 8 in the character string. Note that the actual decoding procedures must also manage the possible presence of noise and approximation errors in realistic models using spatial coding. This is what does the Matlab/Octave function named "scob2str" whose code is listed in Appendix.

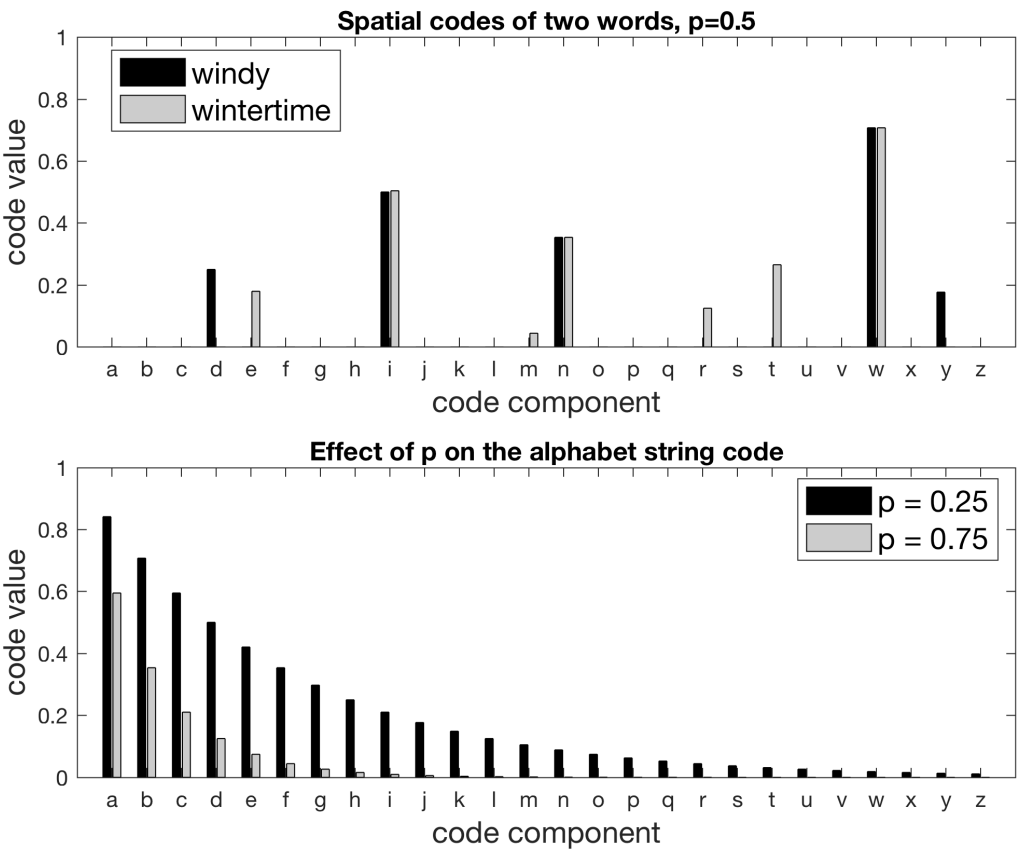

Figure 1. Visualization of the spatial codes of two words (upper panel), and of the alphabet string spatial code with two different values of the p parameter (lower panel).

The use of base 2 exponential functions for the coding is motivated by the fact that 2 is the minimum base that allows complete and unequivocal decoding. On the other hand, the

exponential function of base 2 decreases very fast as the rank of letters increases, which tends to crush the code values for most letters in the string, except the initial ones. The use of the parameter *p* allows us to correct for this drawback, and to obtain a function that is possibly more suitable to cognitive modeling. In particular, the use of an appropriate *p* value allows minimizing the influence of noise and of approximation errors occurring in natural or artificial systems, because *p* determines the minimum difference (spacing) between two distinct exact values in the code (including 0). The effect of *p* on spatial codes is visualized in the lower panel of Figure 1.

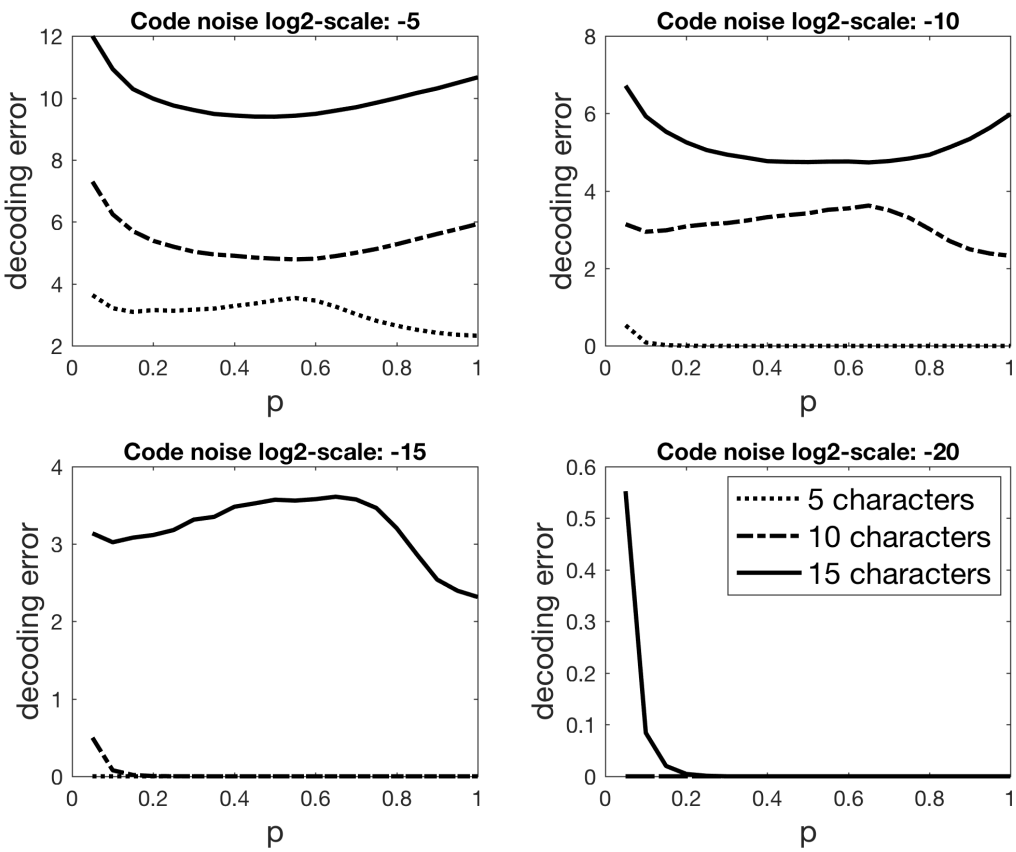

Figure 2. Summary of a computational experiment measuring the decoding error of noisy spatial codes as a function of the scale of the Gaussian noise in the codes (2^{-5}, 2^{-10}, 2^{-15}, 2^{-20}), the length of the original character string (5, 10, or 15 letters), and the p parameter value (from 0.05 to 1 by steps of 0.05).

Figure 2 summarizes a computational experiment illustrating the behavior of the decoding process as a function of *p*, the amount of noise in the spatial codes, and the length of the character strings. A total of 480000 computational tests has been performed, each of them using a randomly generated character string of a given length (5, 10, or 15 characters), a *p* parameter value (varying from 0.05 to 1 by steps of 0.05), and a given amount of Gaussian noise (with mean 0 and standard deviation 2^{-5}, 2^{-10}, 2^{-15}, or 2^{-20}) added to the spatial code components of the string, resulting in a noisy spatial code which was decoded back using the "scob2str" routine. The resulting string was then compared to the original one using the Damerau-Levenshtein string distance (Damerau, 1964) as a "decoding error" measure. This was repeated 2000 times for each combination of the experimental variables modalities, and the average decoding error was used as the dependent variable in the plots of Figure 2. Note that for zero noise, the decoding error is always zero if the length of strings does not exceed the precision of the used real numbers (the maximum is 52 characters with the usual standard IEEE 754 double-precision binary floating-point format). With non-zero noise, one can observe in Figure 2 that the decoding error increases with the amount of noise and with the length of strings, which is not surprising. However, the decoding error is not a monotonic function of *p*, and its shape depends on the relation between the noise scale and the length of the string. In short, let L be the number of characters of the string, then there is almost no effect of *p* on the decoding error (zero) if the noise scale is lower than 2^{-L}, except a small increase for very low *p* values. However, the decoding error function has a maximum on the middle zone of the *p* values if the noise scale is close to 2^{-L}, while it has a minimum on the middle zone of the *p* values if the noise scale is greater than 2^{-L}. Thus, in weakly noisy systems, one can use any value of p, even 1, which is equivalent to remove the *p* parameter. In highly noisy systems, the critical string length is low, and most words are longer than this critical length, thus it is preferable for the decoding accuracy to choose an intermediate value

for p (in a neighborhood of 0.5). Now, in a system where (unfortunately) the noise scale corresponds to the modal string length, the plots in Figure 2 show that the best choice is p=1.

2.3 Example of application in electrophysiological data analysis

As mentioned in the introduction, an application field of special interest of orthographic or phonological regressors is the analysis of ERP data in various psycholinguistic tasks (lexical decision, word naming, …). The problem of the dimension of regressors is particularly critical in this area because the number of distinct stimuli used in ERP experiments is usually limited. As an example, we rapidly summarize the work of Rey, Madec, Grainger, and Courrieu (2013). These authors collected ERPs associated to 200 printed French test words, 4-8 letters long, in a speeded word naming task, using averaged ERPs on 4 repetitions per word for 48 French participants. The EEG activity was recorded continuously using 64 electrodes, positioned on the scalp according to the 10-10 International System, in a time window of -100 ms to +500 ms with respect to the stimulus onset. The between-participant consistency of ERPs, as measured by the ICC, allowed to detect latencies and scalp locations where systematic electrophysiological responses occurred. At these spatiotemporal points, various regressors were applied (with test inflation control) to attempt to identify the nature of the involved processes. In particular, one used an orthographic Spatial Code (1) of 26 lowercase letters to detect orthographic processing, a phonological Spatial Code (1) of 35 French phonemes to detect phonological encoding, and the usual word log-frequency to detect a lexical level processing. Spatial codes were computed using a p parameter value of about 1/3 in order to obtain non-negligible code values at all serial positions. However, the size of the orthographic code matrix was 200-by-26, while the size of the phonological code matrix was 200-by-35, which in both cases lead to substantial regressor overfittings for 200 words. So, it was necessary to lower the dimension of the regressors,

which was done by replacing each of the two string code matrices by its first three (left) singular vectors (Golub & Reinsch, 1970). This provided acceptable three-dimensional regressors (i.e. 200-by-3 matrices), while preserving a maximum part of each original regressor variation for the considered 200 test words.

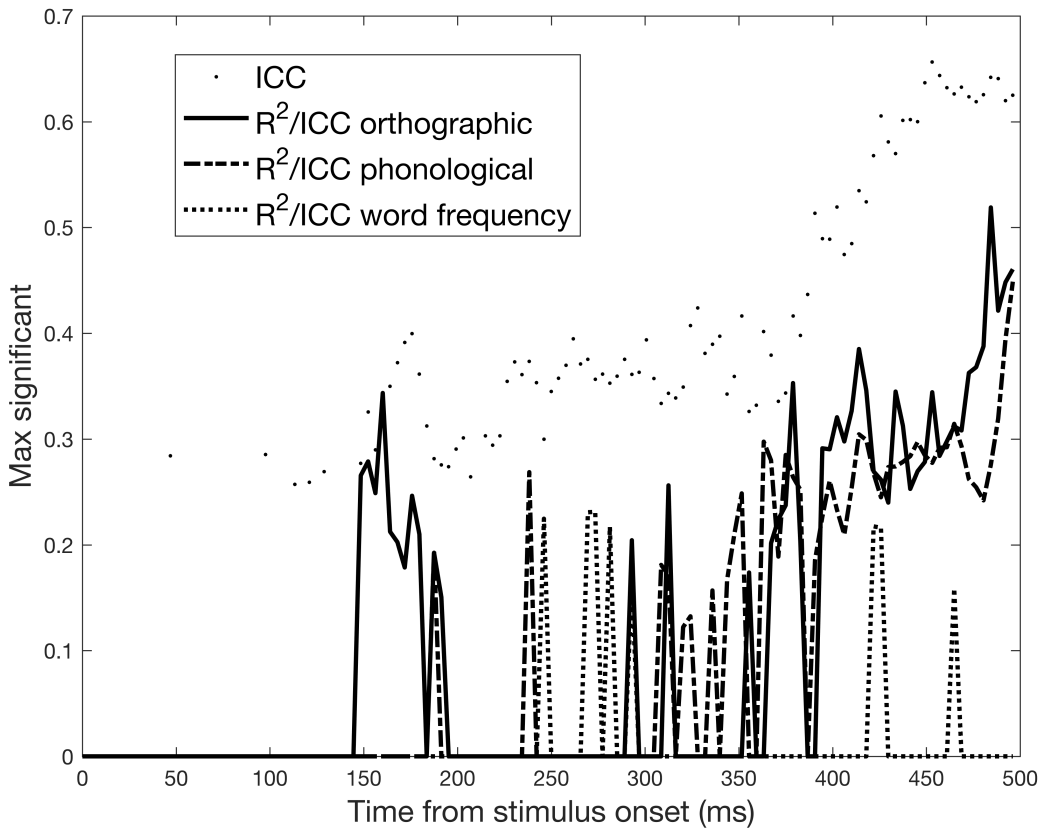

Figure 3. Time course of the between-participant consistency (ICC) and regressor fits (R^2/ICC), for orthographic, phonological, and word frequency regressors applied to cerebral Event Related Potentials in a word-naming task. Non-significant statistics are set to zero for readability, and only the maximal significant values among 64 electrodes are displayed for each latency (after Rey, Madec, Grainger, & Courrieu, 2013).

Figure 3 shows the obtained time course of the detected processes. The first systematic (significant ICCs) but unidentified processes appeared before 100 ms, while the beginning of an orthographic processing was detected at a latency of about 148 ms on a right occipital area, migrating to an occipital area at 188 ms, where and when also appeared the beginning of a

phonological encoding. The phonological processing then migrated to a left occipital area at about 238 ms, and was followed by a word-frequency effect beginning at about 246 ms, also on a left occipital area. The sequence of detected processes seems logical in a word naming task, and the scalp locations of corresponding ERPs are consistent with those observed with other methods in other tasks involving visual character processing and phonological transcoding (Madec et al., 2016).

3. Open Bigrams Coding (OB)

The coding model described hereafter is a variant of the one described in Hannagan and Grainger (2012), and in Lodhi, Saunders, Shawe-Taylor, Cristianini, and Watkins (2002). Contrarily to the original model, this variant does not detect one character strings since it encodes only open bigrams, thus at least two character strings. For instance, in the word 'hat', the open bigrams are 'ha', 'ht', and 'at', while the word 'at' is itself a bigram, but in the one-letter word 'a', there is no bigram in the usual sense. In fact, the following model was designed to be compatible with the above spatial coding scheme (1), in the perspective of merging the two approaches, as described in the next section.

In an alphabet of n characters, the open bigrams code of a string X of m characters is defined as a real matrix of $n \times n$ components c_{ij}, each one corresponding to a possible open bigram whose first character has the index i in the alphabet, and whose second character has the index j. The symbol position bits $b_{k,i}$ are defined as previously, and one has:

$$c_{ij}(X) = (\Sigma_{k=1..m-1} \Sigma_{l=k+1..m} b_{k,i}\, b_{l,j}\, 2^{-(l-k)})^p, \quad i, j = 1..n, \quad 0 < p \leq 1, \qquad (2)$$

For instance, using the 26 letters Roman alphabet, the open bigram 'aa' in the word 'parabola', has the code value $c_{1,1}(\text{parabola})=(2^{-(4-2)}+2^{-(8-2)}+2^{-(8-4)})^p$, while the open bigram 'oa' has the code value $c_{15,1}(\text{parabola})=(2^{-(8-6)})^p$. Since it is more convenient to store the codes in the form of row vectors than in the form of matrices, one vectorizes the code matrix by

concatenating its rows one after the other, which results in a row vector of n^2 components. The Matlab function "str2scob" listed in Appendix compute spatial codes if the input argument p (two components vector) has its first component greater than zero and the second one has a zero value. It computes open bigrams codes if the first component of p is zero and the second one is greater than zero.

The size of an open bigrams code is the square of the size of a spatial code, which is somewhat cumbersome, but also much more redundant, and thus potentially more robust in case of approximation errors and noisy code. An open bigrams code is easy to decode if the target character string does not include more than one repeated character (as the A in PARABOLA). However, in the general case, decoding an open bigrams code is a hard-to-solve problem and there is no known practical solution for large scale applications that require fast decoding.

4. Merging Spatial and Open Bigrams Codes (SCOB)

There is in fact a very simple and quite natural solution to the main drawbacks of the open bigrams coding. Assume that the beginning of any character string is not its first symbol (letter), but a "start character". For instance, one can consider the left whitespace as a tag of the start character for printed words. One can append a start character at the beginning of the alphabet, and append a start character at the beginning of each character string. If the size of the original alphabet was n characters, then it is now $n+1$, but the size of the corresponding open bigrams code is $(n+1)n$, since the start character is not a stop character and it never appears at the second position in a bigram. The code is computed in the same way as the basic OB code, and one obtains an open bigrams code where one character strings are represented as (start character + symbol). Assigning to the start character the index 0 in the alphabet, and

the serial position 0 in the character strings, one sets $b_{0,0}=1$, and the code components are defined as:

$$c_{ij}(X) = (\Sigma_{k=0..m-1} \Sigma_{l=k+1..m} b_{k,i}\, b_{l,j}\, 2^{-(l-k)})^p\ ,\ \ i = 0..n, j = 1..n,\ \ 0 < p \leq 1. \qquad (3)$$

The Matlab function "str2scob" listed in Appendix computes merged spatial and bigrams codes if the input argument p has its two components greater than zero.

This code is easily decodable in all cases since its first n components (c_{0j} components) are exactly equal to those of the spatial code (1) of the same string, which is decodable, for instance using the "scob2str" Matlab function listed in Appendix. For this reason, we will abbreviate this code as SCOB, for "Spatial Code + Open Bigrams", but also remembering "Start Character + Open Bigrams". In this context, one can consider that the spatial coding is just the initial part of an open bigrams coding.

5. Preliminary test on masked orthographic priming data

Orthographic coding models are commonly tested using masked orthographic priming techniques (Davis & Bowers, 2006; Grainger, Granier, Farioli, Van Assche, & van Heuven, 2006; Van Assche & Grainger, 2006; Welvaert, Farioli, & Grainger, 2008), where one assumes that the more the prime and the target are orthographically similar, the more the priming effect is large. The orthographic similarity of two character strings depends on the considered coding model, together with an associated similarity function. In the case where the orthographic codes are fixed length numerical vectors, say x and y, one can for instance use a similarity function of the form $S(x, y)= \langle x, y\rangle/(||x||.||y||)$, where $\langle ., .\rangle$ denotes the inner (dot) product, and $||.||$ is the Euclidean norm (Hannagan & Grainger, 2012). If the considered orthographic coding model is correct, then one can expect a strong positive correlation between S(x, y) and the perceptual priming effect of the string whose code is x on the string whose code is y. We used 27 masked priming effects obtained with different prime structures,

reported by Adelman et al. (2014), in order to test the predictive capability of the orthographic coding schemes SC, OB, and SCOB described above. The parameter p was optimized for each model in order to obtain the best possible correlation between S(x, y) and the corresponding empirical priming effects. For SC, one obtained p=0.54, r=0.59; for OB, one obtained p=0.55, r=0.90; finally, for SCOB, one obtained p=0.81, and r=0.92. Using Williams T2 test (Steiger, 1980; Williams, 1959), we observed that the correlation was significantly better for OB than for SC (T2(24)=3.73, p<.001), and significantly better for SCOB than for SC (T2(24)=4.45, p<.001). However, the correlations were not significantly different for OB and SCOB, although the fit was a bit better for SCOB (T2=1.01, n.s.). Thus, contrarily to what was expected, it seems that the open bigrams coding, even alone, is more suitable than the spatial coding for predicting orthographic priming effects.

6. Building and validating orthographic regressors

6.1. Building one-dimensional orthographic regressors

One can use the above described orthographic codes as multidimensional regressors on large scale item level behavioral databases, where the number of items is sufficiently larger than the number of regressor dimensions. However, this is clearly not possible when one has only a moderate size data set, which is the most common case. The idea presented hereafter is to use a large scale database to build one-dimensional orthographic regressors that can be used on other databases, even small ones, and even if they do not share any item with the large scale database. The principle is very simple: using the large-scale database, one computes the regression (least squares) coefficients of all the components of the multidimensional orthographic code in order to fit the behavioral data (RTs or other) associated to the items of the database. The obtained coefficients allow to capture the part of the data variation that is accountable for by the orthographic code. Then one can reuse these

coefficients to compute a linear combination of the components of the orthographic codes of the same as well as of other items to predict the part of new data variation that is accountable for by the orthographic codes. The obtained regressor is now one-dimensional because the used multiple regression coefficients have been fixed on an independent data set, thus it can be used even on small data sets. This is similar to a supervised learning process on the large-scale database, followed by generalization processes on other data sets. What is learned is just a set of regression coefficients that allows partially predicting a behavioral variable as a function of an orthographic code.

In large OB or SCOB codes, there are frequently components corresponding to open bigrams that never occur in a given database. The regression coefficients of these components are *a priori* zero, and the corresponding components must be temporarily removed to compute the other coefficients. As a result, the number of degrees of freedom of the multidimensional regressors in regression analyses can be lower than the number of code components. The Matlab function "str2scob" listed in Appendix optionally computes the set of (all the) regression coefficients as an output ("lscoef" output argument) if a numerical data set ("data") is provided as an input argument, together with the list of character strings to be encoded (a large-scale database in this case). The same Matlab function optionally applies given coefficients ("lscoef" input argument) to the orthographic codes of the current input character strings, in order to compute a one-dimensional orthographic regressor ("lsaprx" output argument) for these strings. Note that "lscoef" includes the bias regression coefficient as the first one, and one must not provide "lscoef" and "data" simultaneously as input arguments (use the empty matrix [] to skip an argument). Useful information and examples of use are given as comments (at right of "%") in the Matlab code of functions.

6.2. Cross-validation of the regressors

In such a context, one can use well-known validation methods such as the Monte-Carlo cross-validation procedure to estimate the generalization power of the built regressor. Cross-validation, as a generalization process, avoids the overfitting problems that systematically occur in least squares multiple regression models (Picard & Cook, 1984). Using a large-scale database, one repeatedly randomly sample a subset of items of a given size as the cross-validation (generalization) set, and one uses the remaining subset of items as the learning set. One computes the regression coefficients of the orthographic code components on the learning set data, then one applies these coefficients to the orthographic codes of the items of the cross-validation set, and one computes the correlation coefficient of the obtained one-dimensional orthographic regressor with the target data of the cross-validation set. One obtains a sample of cross-validation correlation coefficients, which can be summarized by its mean and a confidence interval (for instance the 99% one). To compute the useful statistics, it is convenient to use the r-to-z Fisher transformation, then to compute the mean and 99% confidence interval on the z values, and finally to transform the results in r values using the inverse Fisher transformation. The Matlab function "crossvalSCOB" listed in Appendix do all this work, given a large-scale database, a size for the cross-validation sets (we always use Nitcv=2000 items in the present study), and the number of random repetitions of the test (we always repeat Ntest=120 times in this study).

6.3. Data independent regressors

One can also build one-dimensional regressors that are independent of data. For instance, in the present study, we will use a regressor that relates to the inconsistency of the grapheme-to-phoneme correspondence, in the analysis of word naming times. The orthographic coding schemes described above can in fact be applied to any type of symbol

strings with any type of alphabet, including phonological ones. Assume that we have a long list of words with their spelling and their phonological form, then we can compute the orthographic and the phonological numerical codes of these words, which results in two rectangular matrices having the same number of rows. One can easily compute the least squares linear transcoding of the orthographic codes into the phonological codes, and the obtained approximation leaves a residual that is the part of the phonological codes that cannot be linearly accounted for by the orthographic codes. Taking the Euclidean norm of this residual for each word of the list, one obtains a one-dimensional regressor estimating the "spelling to pronunciation linear transcoding inconsistency", abbreviated SPLTI in this paper. Note that the transcoding matrix, which is the solution of the least squares system, can be reused for other, possibly shorter lists of words, which allows one to estimate the SPLTI even for words that do not belong to the initial large-scale dictionary. The Matlab function "TranscobResMat" listed in Appendix allows one to easily compute transcoding matrices and SPLTI's.

7. Application to the French Lexicon Project (Ferrand et al., 2010)

The French Lexicon Project (Ferrand, New, Brysbaert, Keuleers, Bonin, Méot, Augustinova, & Pallier, 2010) provides valid lexical decision data for 38335 French words. The alphabet of 39 characters includes characters with diacritic marks necessary to write French words: abcdefghijklmnopqrstuvwxyzàâçèéêëîïôùûü. The average number of valid observations per item is 22.1, and the proportion of systematic item variance in the RT z-scores is given by the ICC= 0.8450, 99% CI= [0.8420, 0.8479], which is close to the split-half Spearman-Brown reliability estimate r_{corr}=0.84 provided by Ferrand et al. (2010) for these data.

7.1. Determination of the optimal code and p parameter value

In order to determine a suitable orthographic code for these data, we tried to fit (by the least squares method) the RT z-scores and the frequency of errors, on the basis of spatial codes (SC), open bigrams codes (OB), and merged spatial and open bigrams codes (SCOB), as a function of the p parameter value, which was varied from 0.05 to 1.00 by steps of 0.05. One can observe in Figure 4 that open bigrams codes always provided better fits (Pearson's r) than the spatial code alone, while the SCOB codes seem to have a small but regular advantage on OB codes. The smallest p parameter values are optimal for the spatial code alone, while p=1 is optimal for the OB and SCOB codes. Thus, globally, p=1 is the optimal choice, which is equivalent to remove the p parameter.

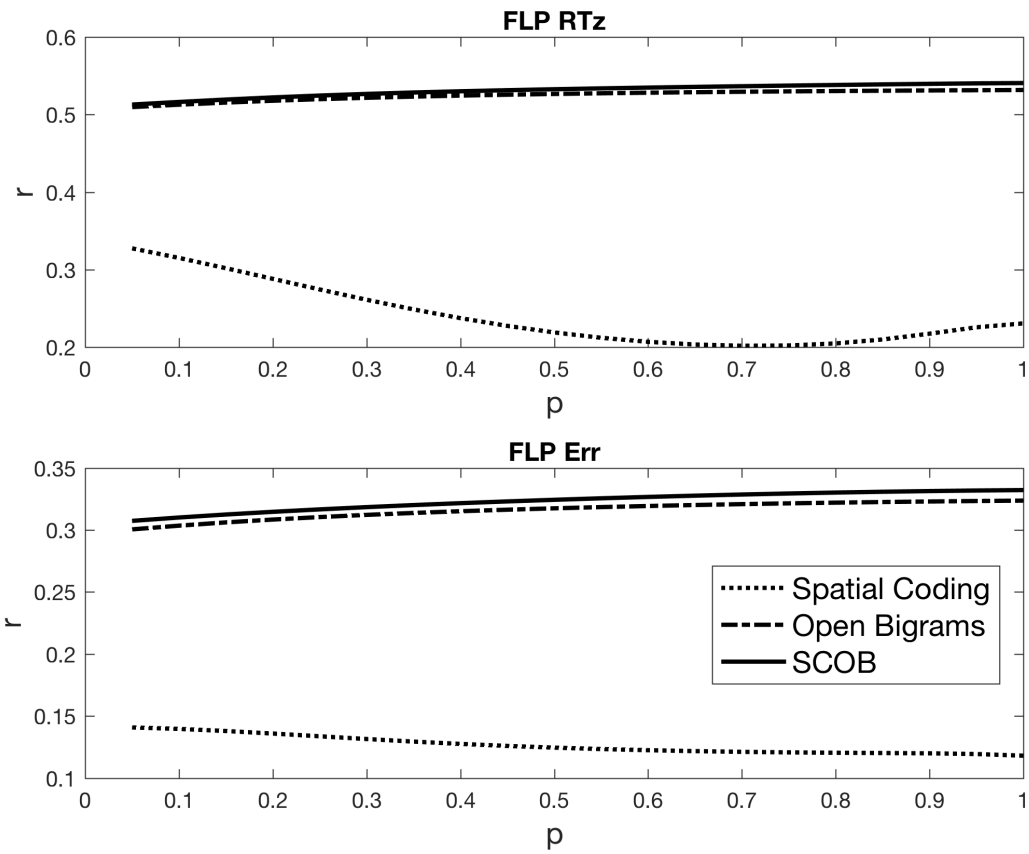

Figure 4. Correlation of FLP lexical decision z-times (upper panel), and error frequencies (lower panel) with their least square approximations based on spatial codes (SC), on open bigrams codes (OB), and on merged SC and OB codes (SCOB), as functions of p parameter.

In order to see whether or not both the SC and the OB components provided specific significant contributions to the performance of the SCOB code regressor, we performed a series of hierarchical multiple regressions, with the RT z-scores as dependent variable, while the orthographic codes SC and OB were entered as multidimensional regressors in this order, and in the reverse order, with various values of the p parameter. These analyses are presented in Table 1, where one can see that in all cases, both the SC and the OB components provide specific significant contributions to the data fit, while p=1 for the two components is globally the best parameter choice. We conclude that the SCOB code without p parameter is the most appropriate orthographic code for these data, while the contribution of the SC component is small but always relevant.

Table 1. Hierarchical multiple regression analyses of the lexical decision z-times of the French Lexicon Project (Ferrand et al., 2010). The multidimensional orthographic regressors are the spatial code (SC) and the open bigrams code (OB) of the words, with various values of the code parameter p.

Regressor 1	R^2	Regressor 2	R^2	ΔR^2	ΔR^2 significance	Adj. R^2
SC (p=1)	0.0553	OB (p=1)	0.2924	0.2391	F(1076, 37219)=11.69	0.2712
OB (p=1)	0.2828	SC (p=1)	0.2924	0.0096	F(39, 37219)=12.97	0.2712
SC (p=0.72)	0.0408	OB (p=0.72)	0.2884	0.2476	F(1076, 37219)=12.03	0.2671
OB (p=0.72)	0.2806	SC (p=0.72)	0.2884	0.0078	F(39, 37219)=10.42	0.2671
SC (p=0.05)	0.1073	OB (p=0.05)	0.2632	0.1559	F(1076, 37219)=7.32	0.2411
OB (p=0.05)	0.2598	SC (p=0.05)	0.2632	0.0034	F(39, 37219)=4.36	0.2411
SC (p=0.05)	0.1073	OB (p=1)	0.2880	0.1806	F(1076, 37219)=8.78	0.2666
OB (p=1)	0.2828	SC (p=0.05)	0.2880	0.0052	F(39, 37219)=6.95	0.2666

Underscored F-values are highly significant ($p<0.0001$)

7.2. Relation with usual regressors

In Table 2, one can see the inter-correlations of the SCOB based one-dimensional orthographic regressors (A.RTz and A.Err, where we use the prefix "A." for "Approximate"), the corresponding targeted data variables (RTz and Err), and three usual regressors, namely the word length, the word log-frequency, and the OLD20 measure of orthographic neighborhood (Yarkoni, Balota, & Yap, 2008), whose values were found in the Lexique 3 database (New, Pallier, Brysbaert, & Ferrand, 2004). For the word frequency, we used the logarithm of the sum of the frequency in books and the frequency in movies plus one (to avoid log(0) for very rare words). We note, in particular, the high correlations of A.RTz with the word length and the OLD20.

Table 2. Inter-correlations of the FLP one-dimensional SCOB based regressors, A.RTz and A.Err, targeting the lexical decision z-times (RTz) and the error frequencies (Err), respectively. Correlations with 3 usual regressors: word length, word log-frequency, and old20 are also provided.

r_{38335}	A.RTz	A.Err	RTz	Err	length	log-Fr	old20
A.RTz	-	0.4124	0.5407	0.1370	0.7270	-0.3657	0.6636
A.Err	0.4124	-	0.2230	0.3322	-0.1726	-0.0246	0.0241
RTz	0.5407	0.2230	-	0.6057	0.3931	-0.5676	0.4423
Err	0.1370	0.3322	0.6057	-	-0.0573	-0.3232	0.0928
length	0.7270	-0.1726	0.3931	-0.0573	-	-0.3799	0.7756
log-Fr	-0.3657	-0.0246	-0.5676	-0.3232	-0.3799	-	-0.3134
old20	0.6636	0.0241	0.4423	0.0928	0.7756	-0.3134	-

In Table 3, one analyzed the relations of the multidimensional SCOB regressor with the three usual regressors in fitting the RT z-scores. This was done using a series of hierarchical multiple regression analyses from which we can observe that all tested regressors provided specific significant contributions to the fit, except the word length whose

contribution was completely explained by the SCOB (but not the reciprocal). This is not surprising given that the sum of all components of an OB code is almost a linear function of the string length.

Table 3. Hierarchical multiple regression analyses of the FLP lexical decision z-times. The regressors are the multidimensional SCOB code of the words and the usual regressors: word length, word log-frequency, and old20. All regressors provide significant specific contributions, except the word length whose effect is completely explained by the SCOB.

Regressor 1	R^2	Regressor 2	R^2	ΔR^2	ΔR^2 significance	Adj. R^2
old20 + log-Fr + Length	0.4003	SCOB	0.5201	0.1199	F(1115, 37216)=8.34	0.5057
SCOB	0.2924	old20	0.3281	0.0357	F(1, 37218)=1978.2	0.3080
SCOB	0.2924	log-Fr	0.4922	0.1998	F(1, 37218)=14643	0.4770
SCOB	0.2924	length	0.2924	0	F(1, 37218)= 0	0.2712

Underscored F-values are highly significant ($p<0.0001$)

7.3. Orthographic regressors cross-validation

The learning and cross-validation of SCOB orthographic regressors were tested on FLP lexical decision z-times and on error frequencies. The two distributions of 120 cross-validation r values are shown in Figure 5. Each r value was computed using 2000 randomly selected generalization test words, while 36335 other words were used to compute regression coefficients of the SCOB code components in order to approximate the data (learning). For the RT z-scores, one obtained the learning average r = 0.5418, 99% CI = [0.5416, 0.5420], and the cross-validation average r = 0.4563, 99% CI = [0.4417, 0.4707]. For the error frequency, one obtained the learning average r = 0.3342, 99% CI = [0.3339, 0.3346], and the cross-validation average r = 0.1997, 99% CI = [0.1881, 0.2112]. The averages and confidence

intervals were computed using Fisher r-to-z transforms. The cross-validation mean r value obtained for the RT z-scores corresponds to 20.82% item variance accounted for, and to 24.64% systematic item variance accounted for, given the ICC=0.8450. The difference between the cross-validation r and the learning r gives an idea of the overfitting provided by the least squares fit of the regressors to the data. Note, however, that an overfitting in learning generally results in an under-fitting in generalizations, so that the difference between the learning and the cross-validation fits is the sum of these two misfits.

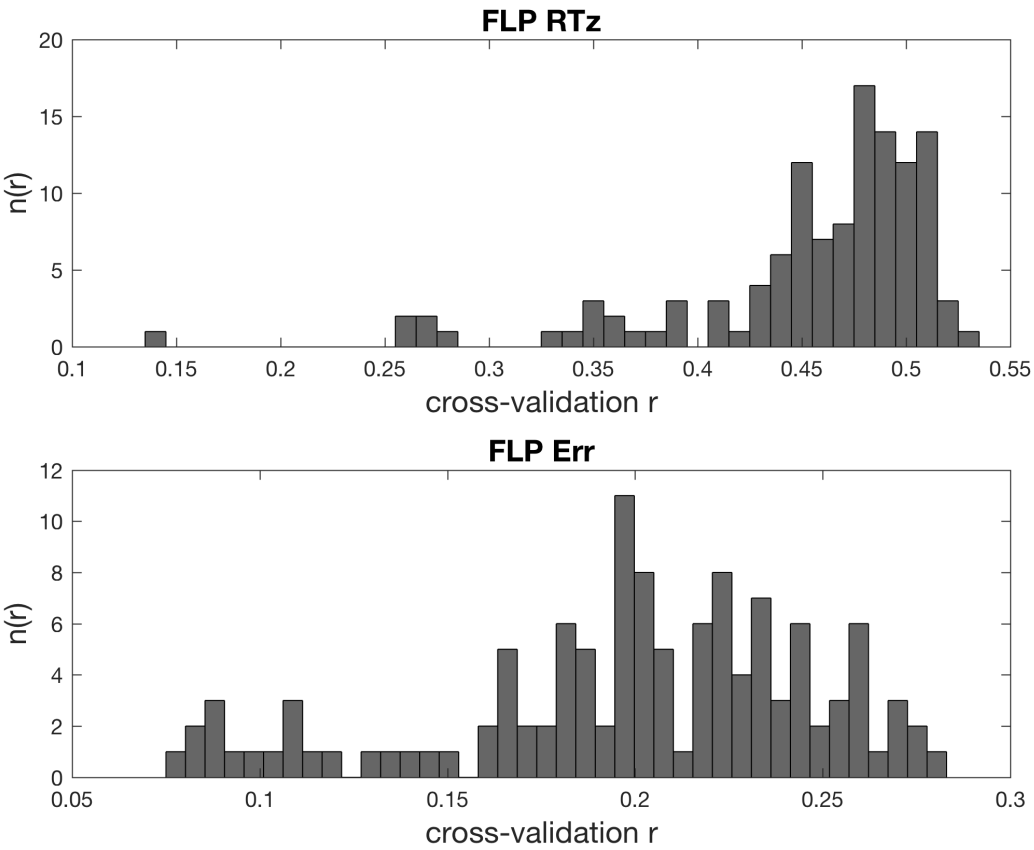

Figure 5. Distribution of 120 cross-validation r values for FLP RTz, and FLP error frequencies. Each r value was computed using 2000 randomly selected generalization test words, while 36335 other words were used to compute regression coefficients of the SCOB code components in order to approximate the data (learning). RTz : learning average r = 0.5418, 99% CI = [0.5416, 0.5420] , cross-validation average r = 0.4563, 99% CI = [0.4417, 0.4707]. Error frequency : learning average r = 0.3342, 99% CI = [0.3339, 0.3346], cross-validation average r = 0.1997, 99% CI = [0.1881, 0.2112]. The averages and confidence intervals were computed using Fisher r-to-z transforms.

8. Application to the English Lexicon Project (Balota et al., 2007)

The English Lexicon Project (Balota, Yap, Cortese, Hutchison, Kessler, Loftis, Neely, Nelson, Simpson, & Treiman, 2007) includes two behavioral databases. The first one provides lexical decision data, and the second one provides speeded word naming data for 40481 English words. For the present study, the items selection was conditional to the availability of valid data concerning: the word spelling, its pronunciation, its frequency, its OLD20, OLD20 frequency, PLD20, PLD20 frequency, the mean lexical decision z-time and response accuracy, the mean speeded naming z-time and response accuracy. The number of selected words for the two tasks was finally 39302. Both uppercase and lowercase letters were used, resulting in an alphabet of 52 characters:

ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz.

In addition, for the word naming ELP database, we must also consider the phonological alphabet of 46 characters:

"%.34=@ADEHINORSTUVXZ_`abdefghijklmnoprstuvwxz, which was used to describe the pronunciation of words.

8.1. Lexical decision data

In what concerns the lexical decision database, the average number of valid observations per item was 27.8, and the proportion of systematic item variance in the RT z-scores is given by the ICC= 0.8954, 99% CI= [0.8934, 0.8973].

8.1.1. Determination of the optimal code and p parameter value

As for the FLP database, we tried to determine a suitable orthographic code for the ELP lexical decision data by fitting (least squares method) the RT z-scores and the accuracy,

on the basis of spatial codes (SC), open bigrams codes (OB), and merged spatial and open bigrams codes (SCOB), as a function of the p parameter value, which was varied from 0.05 to 1.00 by steps of 0.05. One can observe in Figure 6 that open bigrams codes always provided better fits (Pearson's r) than the spatial code alone, while the SCOB codes seem to have a small but regular advantage on OB codes. The smallest p parameter values are optimal for the spatial code alone, while p=1 is optimal for the OB and SCOB codes. Thus, globally, p=1 this is the optimal choice, which is equivalent to remove the p parameter.

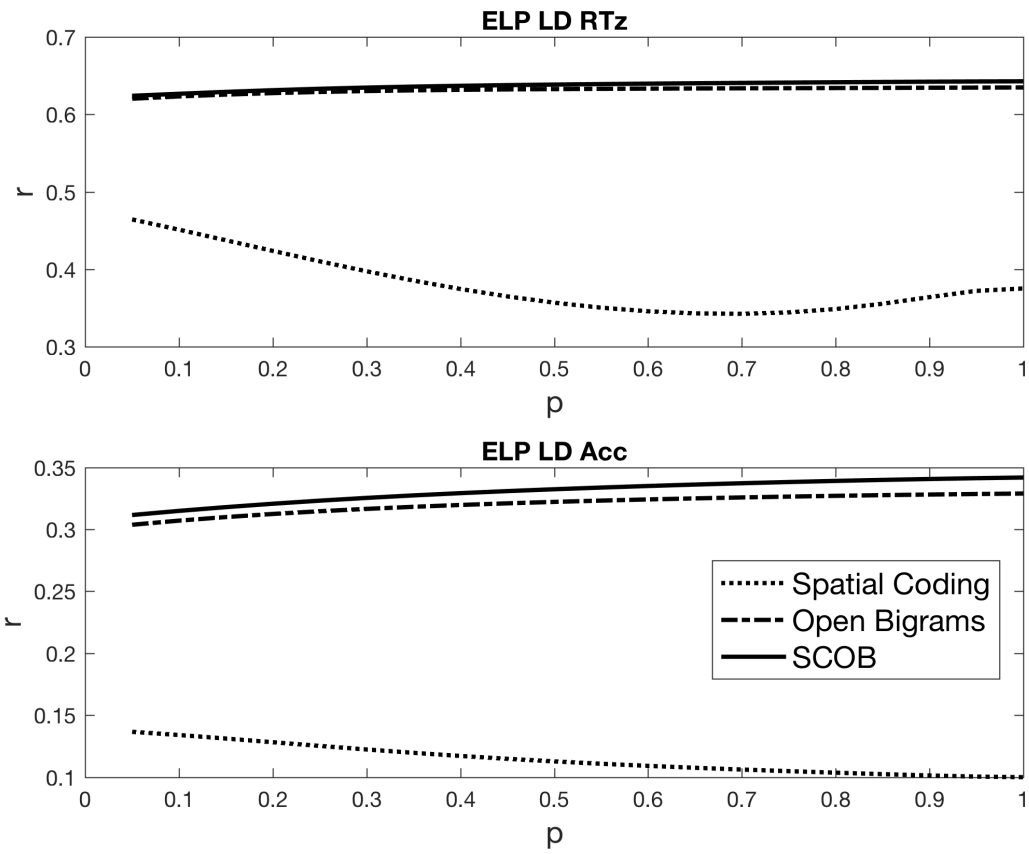

Figure 6. Correlation of ELP lexical decision z-times (upper panel), and response accuracy (lower panel) with their least squares approximations based on spatial codes (SC), on open bigrams codes (OB), and on merged SC and OB codes (SCOB), as functions of the code parameter p.

In order to see whether or not both the SC and the OB components provided specific significant contributions to the performance of the SCOB code regressor, we performed a series of hierarchical multiple regressions, with the RT z-scores as dependent variable, while the orthographic codes SC and OB were entered as multidimensional regressors in this order, and in the reverse order, with various values of the p parameter. These analyses are presented in Table 4, where one can see that in all cases, both the SC and the OB components provide specific significant contributions to the data fit, while p=1 for the two components is globally the best parameter choice. We conclude that the SCOB code without p parameter is the most appropriate orthographic code for these data, while the contribution of the SC component is small but always relevant. In summary, the results and conclusions are very similar to those obtained with the FLP database.

Table 4. Hierarchical multiple regression analyses of the lexical decision z-times of the English Lexicon Project (Balota et al., 2007). The multidimensional orthographic regressors are the spatial code (SC) and the open bigrams code (OB) of the words, with various values of the code parameter p.

Regressor 1	R^2	Regressor 2	R^2	ΔR^2	ΔR^2 significance	Adj. R^2
SC (p=1)	0.1410	OB (p=1)	0.4132	0.2723	F(1243, 38006)=<u>14.19</u>	0.3932
OB (p=1)	0.4031	SC (p=1)	0.4132	0.0102	F(52, 38006)=<u>12.65</u>	0.3932
SC (p=0.70)	0.1175	OB (p=0.70)	0.4106	0.2931	F(1243, 38006)=<u>15.20</u>	0.3905
OB (p=0.70)	0.4017	SC (p=0.70)	0.4106	0.0089	F(52, 38006)=<u>11.00</u>	0.3905
SC (p=0.05)	0.2158	OB (p=0.05)	0.3895	0.1737	F(1243, 38006)=<u>8.70</u>	0.3687
OB (p=0.05)	0.3849	SC (p=0.05)	0.3895	0.0047	F(52, 38006)=<u>5.57</u>	0.3687
SC (p=0.05)	0.2158	OB (p=1)	0.4089	0.1931	F(1243, 38006)=<u>9.99</u>	0.3887
OB (p=1)	0.4031	SC (p=0.05)	0.4089	0.0058	F(52, 38006)=<u>7.17</u>	0.3887

Underscored F-values are highly significant ($p<0.0001$)

8.1.2. Relation with usual regressors

In Table 5, one can see the inter-correlations of the SCOB based one-dimensional orthographic regressors (A.RTz and A.Acc), the corresponding targeted data variables (RTz and Acc), and four usual regressors, namely the word length, the word log-frequency, the OLD20, and the OLD20 frequency (Yarkoni, Balota, & Yap, 2008). For the word frequency, we used the logarithm of the HAL frequency plus one (to avoid log(0) for very rare words). As for the FLP database, we note the high correlations of A.RTz with the word length and the OLD20.

Table 5. Inter-correlations of the ELP one-dimensional SCOB based regressors, A.RTz and A.Acc, targeting the lexical decision z-times (RTz) and the accuracy (Acc), respectively. Correlations with 4 usual regressors: word length, word log-frequency, old20 and old20 frequency are also provided.

r_{39302}	A.RTz	A.Acc	RTz	Acc	length	log-Fr	old20	old20F
A.RTz	-	-0.3627	0.6428	-0.1240	0.8637	-0.3559	0.8568	-0.6412
A.Acc	-0.3627	-	-0.2332	0.3418	0.0474	0.1054	-0.1438	0.0305
RTz	0.6428	-0.2332	-	-0.5974	0.5552	-0.6594	0.6114	-0.4313
Acc	-0.1240	0.3418	-0.5974	-	0.0162	0.4915	-0.1154	0.0191
length	0.8637	0.0474	0.5552	0.0162	-	-0.3514	0.8683	-0.7217
log-Fr	-0.3559	0.1054	-0.6594	0.4915	-0.3514	-	-0.4016	0.4583
old20	0.8568	-0.1438	0.6114	-0.1154	0.8683	-0.4016	-	-0.6622
old20F	-0.6412	0.0305	-0.4313	0.0191	-0.7217	0.4583	-0.6622	-

In Table 6, one analyzed the relations of the multidimensional SCOB regressor with the four usual regressors in fitting the RT z-scores. This was done using a series of hierarchical multiple regression analyses from which we can observe that all tested regressors provided specific significant contributions to the fit, except the word length whose

contribution was completely explained by the SCOB (but not the reciprocal). This last result is similar to the one obtained with the FLP database, for the same reason.

Table 6. Hierarchical multiple regression analyses of the ELP lexical decision z-times. The regressors are the multidimensional SCOB code of the words and the usual regressors: word length, word log-frequency, and old20, and old20 frequency. All regressors provide significant specific contributions, except the word length whose effect is completely explained by the SCOB.

Regressor 1	R^2	Regressor 2	R^2	ΔR^2	ΔR^2 significance	Adj. R^2
old20 + old20F + log-Fr + Length	0.5959	SCOB	0.6744	0.0785	F(1295, 38002)=7.08	0.6632
SCOB	0.4132	old20	0.4505	0.0372	F(1, 38005)=2575.4	0.4317
SCOB	0.4132	old20F	0.4148	0.0015	F(1, 38005)=98.45	0.3948
SCOB	0.4132	log-Fr	0.6604	0.2472	F(1, 38005)=27660	0.6488
SCOB	0.4132	length	0.4132	0	F(1, 38005)= 0	0.3932

Underscored F-values are highly significant ($p<0.0001$)

8.1.3. Orthographic regressors cross-validation

The learning and cross-validation of SCOB orthographic regressors were tested on ELP lexical decision z-times and on the response accuracy. The distributions of 120 cross-validation r values for ELP lexical decision z-times, and ELP lexical decision accuracy are shown in Figure 7. Each r value was computed using 2000 randomly selected generalization test words, while 37302 other words were used to compute regression coefficients of the SCOB code components in order to approximate the data (learning). For the RT z-scores, one obtained the learning average r = 0.6437, 99% CI = [0.6435, 0.6438], and the cross-validation average r = 0.4313, 99% CI = [0.3888, 0.4720]. For the response accuracy, one obtained the

learning average r = 0.3440, 99% CI = [0.3438, 0.3443], and the cross-validation average r = 0.1430, 99% CI = [0.1267, 0.1592]. The averages and confidence intervals were computed using Fisher r-to-z transforms. The cross-validation mean r value obtained for the RT z-scores corresponds to 18.6% item variance accounted for, and to 20.78% systematic item variance accounted for, given the ICC=0.8954. The difference between the cross-validation r and the learning r reveals a substantial overfitting resulting from the least squares fit of the regressors to the data.

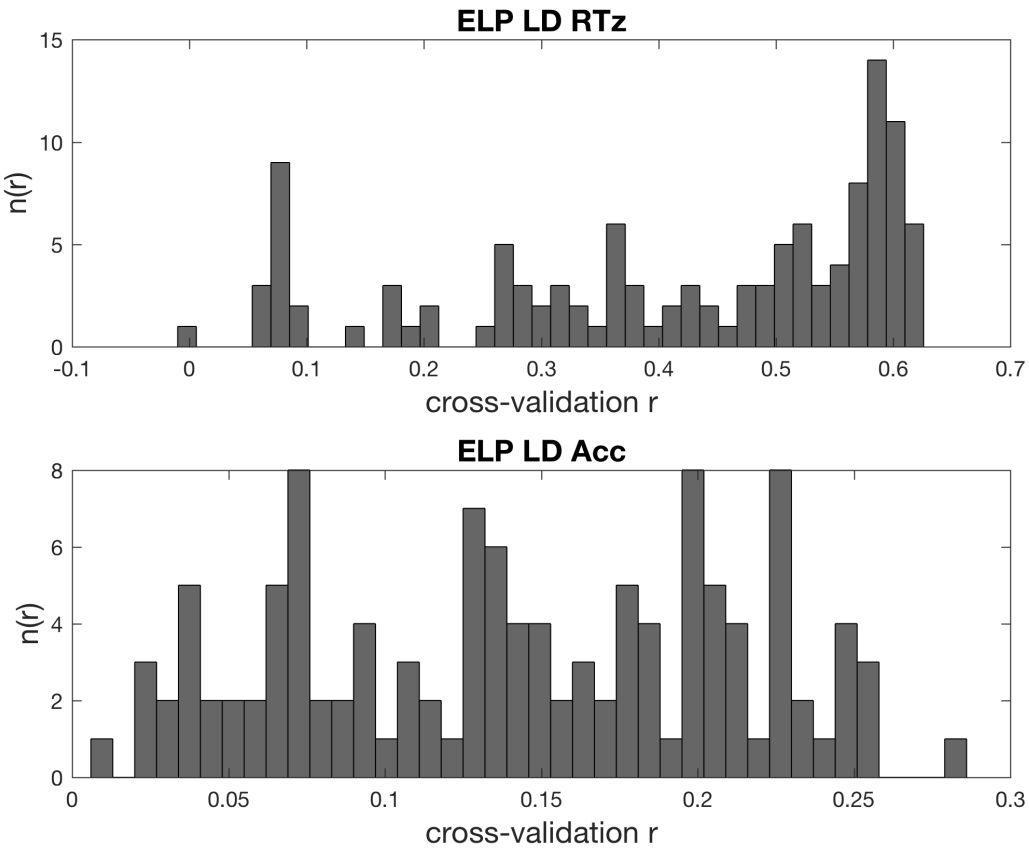

Figure 7. Distribution of 120 cross-validation r values for ELP lexical decision RTz, and ELP lexical decision accuracy. Each r value was computed using 2000 randomly selected generalization test words, while 37302 other words were used to compute regression coefficients of the SCOB code components in order to approximate the data (learning). RTz : learning average r = 0.6437, 99% CI = [0.6435, 0.6438] , cross-validation average r = 0.4313, 99% CI = [0.3888, 0.4720]. Accuracy: learning average r = 0.3440, 99% CI = [0.3438, 0.3443], cross-validation average r = 0.1430, 99% CI = [0.1267, 0.1592]. The averages and confidence intervals were computed using Fisher r-to-z transforms.

8.2 Speeded word naming data

For the speeded word naming database, the average number of valid observations per item was 19.2, and the proportion of systematic item variance in the RT z-scores is given by the ICC= 0.8921, 99% CI= [0.8901, 0.8941].

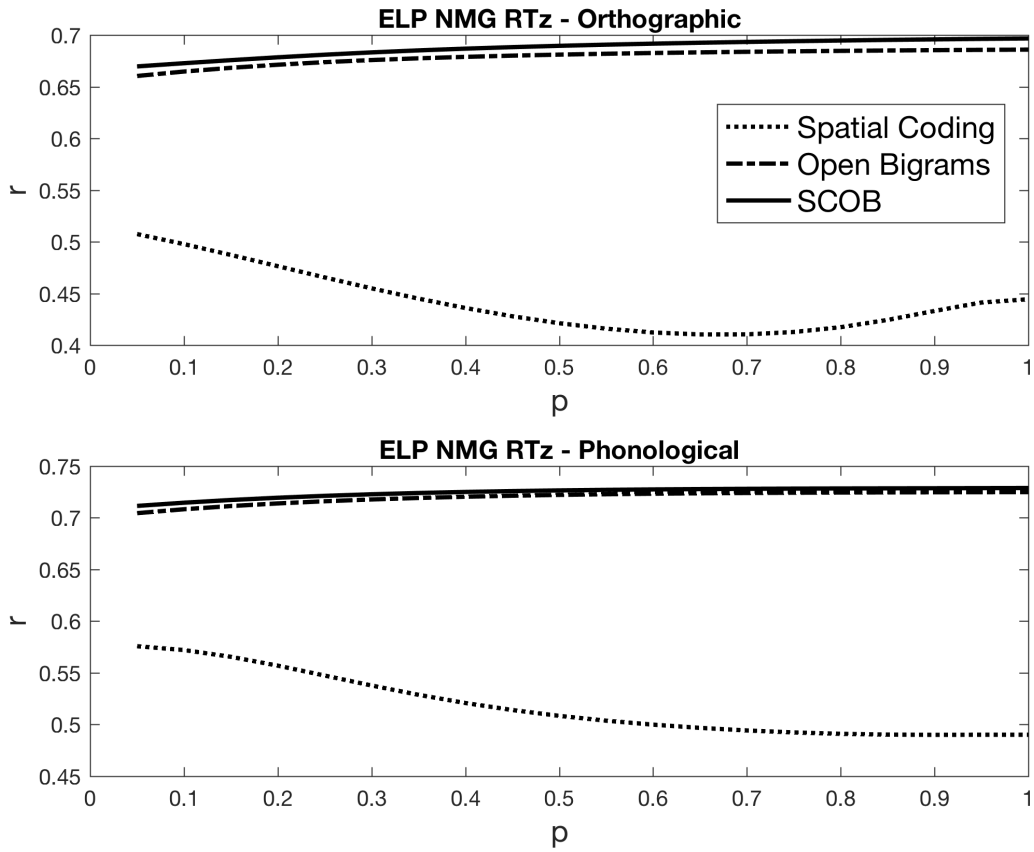

Figure 8. Correlation of ELP word naming RT z-scores with their least square approximations based on spatial orthographic codes, on open bigrams orthographic codes, on SCOB orthographic codes (upper panel), and on spatial phonological codes, on open "biphones" codes, and on SCOB phonological codes (lower panel), as functions of the code parameter p.

8.2.1. Determination of the optimal orthographic and phonological codes

As for the other databases, we must determine a suitable orthographic code for the ELP word naming data, but it is also relevant to determine a suitable phonological code in

order to take into account the phonological nature of the response in the word naming task. This was done by using the phonological alphabet in a least-squares fitting of the RT z-scores on the basis of spatial codes (SC), open "biphones" codes (OB), and merged spatial and open "biphones" codes (SCOB), as functions of the p parameter value, which was varied from 0.05 to 1.00 by steps of 0.05. One can observe in Figure 8 (upper panel) that orthographic open bigrams codes always provided better fits than the spatial code alone, while the orthographic SCOB codes have a small but regular advantage on OB codes. One can also observe, in the lower panel of Figure 8, that the pattern of results is the same for the phonological codes as for the orthographic codes. Moreover, for both the orthographic and the phonological codes, the smallest p parameter values are optimal for the spatial code alone, while p=1 is optimal for the OB and SCOB codes. Thus, globally, SCOB codes with p=1 are always the optimal choices.

In order to see whether or not both the SC and the OB components provided specific significant contributions to the performance of the SCOB code regressors, we performed a series of hierarchical multiple regressions, with the RT z-scores as dependent variable, while the orthographic codes SC and OB were entered as multidimensional regressors in this order, and in the reverse order, for the orthographic code and for the phonological code. In addition, the respective contributions of the orthographic and of the phonological code to the data variations were also tested. These analyses are presented in Table 7, where one can see that in all cases, both the SC and the OB components provided specific significant contributions to the data fit, while both the orthographic and the phonological codes also provided specific significant contributions. We conclude that the SCOB code without p parameter provides the most appropriate orthographic code as well as the most appropriate phonological code for these data, while the contribution of the SC component is small but always relevant. In

summary, the results and conclusions are very similar to those obtained with the other databases, and in addition, they generalize to the phonological dimension.

Table 7. Hierarchical multiple regression analyses of the speeded word naming z-times of the English Lexicon Project (Balota et al., 2007). The multidimensional regressors are the orthographic spatial code (SC ortho), the open bigrams orthographic code (OB ortho), the phonological spatial code (SC phono), the phonological open "biphones" code (OB phono), the merged SCOB orthographic code (SCOB ortho), and the merged SCOB phonological code (SCOB phono) of the words, with p parameters always equal to 1.

Regressor 1	R^2	Regressor 2	R^2	ΔR^2	ΔR^2 significance	Adj. R^2
SC ortho	0.1977	OB ortho	0.4854	0.2878	F(1243, 38006)=<u>17.10</u>	0.4679
OB ortho	0.4705	SC ortho	0.4854	0.0149	F(52, 38006)=<u>21.18</u>	0.4679
SC phono	0.2402	OB phono	0.5309	0.2907	F(1733, 37522)=<u>13.42</u>	0.5087
OB phono	0.5252	SC phono	0.5309	0.0057	F(46, 37522)=<u>10.00</u>	0.5087
SCOB ortho	0.4854	SCOB phono	0.5862	0.1008	F(1779, 36227)=<u>4.96</u>	0.5511
SCOB phono	0.5309	SCOB ortho	0.5862	0.0553	F(1295, 36227)=<u>3.74</u>	0.5511

Underscored F-values are highly significant ($p<0.0001$)

8.2.2 Relation with usual regressors

In Table 8, one can see the inter-correlations of the SCOB based one-dimensional orthographic and phonological regressors (Ao.RTz and Ap.RTz, respectively), the spelling-to-pronunciation linear transcoding inconsistency (SPLTI), the word naming RT z-score (RTz), and six usual regressors, namely the word length, the word log-frequency, the OLD20, the OLD20 frequency, the PLD20 (i.e. the same as the OLD20, but on the phonological forms), and the PLD20 frequency. We observe, in particular, a substantial correlation of the SPLTI with the RTz, but also with the OLD20 and the PLD20.

Table 8. Inter-correlations of the ELP one-dimensional orthographic SCOB based regressor Ao.RTz, the one-dimensional phonological SCOB based regressor Ap.RTz, targeting the speeded word naming z-times (RTz), and the "spelling to pronunciation linear transcoding inconsistency" (SPLTI). Correlations with 6 usual regressors: word length, word log-frequency, old20, old20 frequency, pld20, and pld20 frequency are also provided.

r_{39302}	Ao.RTz	Ap.RTz	SPLTI	RTz	log-Fr	old20	old20F	pld20	pld20F	length
Ao.RTz	-	0.869	0.642	0.697	-0.295	0.768	-0.559	0.752	-0.505	0.792
Ap.RTz	0.869	-	0.687	0.729	-0.329	0.734	-0.517	0.773	-0.505	0.735
SPLTI	0.642	0.687	-	0.558	-0.324	0.689	-0.464	0.739	-0.482	0.647
RTz	0.697	0.729	0.558	-	-0.545	0.591	-0.400	0.592	-0.388	0.552
Log-Fr	-0.295	-0.329	-0.324	-0.545	-	-0.402	0.458	-0.388	0.481	-0.351
old20	0.768	0.734	0.689	0.591	-0.402	-	-0.662	0.910	-0.643	0.868
old20F	-0.559	-0.517	-0.464	-0.400	0.458	-0.662	-	-0.611	0.792	-0.722
pld20	0.752	0.773	0.739	0.592	-0.388	0.910	-0.611	-	-0.649	0.835
pld20F	-0.505	-0.505	-0.482	-0.388	0.481	-0.643	0.792	-0.649	-	-0.644
length	0.792	0.735	0.647	0.552	-0.351	0.868	-0.722	0.835	-0.644	-

In Table 9, one analyzed the relations of the multidimensional SCOB orthographic and phonological regressors, as well as the SPLTI, with the six usual regressors in fitting the RT z-scores. This was done using a series of hierarchical multiple regression analyses from which we can observe that all tested regressors, including the phonological SCOB and the SPLTI, provided specific significant contributions to the fit, except the word length whose contribution was completely explained by the SCOB, as previously.

Table 9. Hierarchical multiple regression analyses of the ELP speeded word naming z-times. The regressors are the multidimensional orthographic and phonological SCOB codes of the words, the Spelling to Pronunciation Linear Transcoding Inconsistency (SPLTI), and 6 usual regressors: word length, word log-frequency, old20, old20 frequency, pld20, and pld20 frequency. All regressors provide significant specific contributions, except the word length whose effect is, in fact, completely explained by the orthographic SCOB code.

Regressor 1	R^2	Regressor 2	R^2	ΔR^2	ΔR^2 significance	Adj. R^2
(6 usual)* + SCOB ortho+ SCOB phono	0.7208	SPLTI	0.7250	0.0042	F(1, 36220)=547.85	0.7016
(6 usual)* + SCOB ortho+ SPLTI	0.6646	SCOB phono	0.7250	0.0604	F(1779, 36220)=4.47	0.7016
(6 usual)* + SPLTI + SCOB phono	0.6857	SCOB ortho	0.7250	0.0393	F(1295, 36220)=4.00	0.7016
SPLTI + SCOB ortho+ SCOB phono	0.6007	Log-Fr	0.7196	0.1189	F(1, 36225)=15353	0.6957
SPLTI + SCOB ortho+ SCOB phono	0.6007	old20	0.6118	0.0111	F(1, 36225)=1039.6	0.5789
SPLTI + SCOB ortho+ SCOB phono	0.6007	old20F	0.6011	0.0004	F(1, 36225)=36.00	0.5672
SPLTI + SCOB ortho+ SCOB phono	0.6007	pld20	0.6072	0.0065	F(1, 36225)=597.42	0.5738
SPLTI + SCOB ortho+ SCOB phono	0.6007	pld20F	0.6022	0.0015	F(1, 36225)=140.48	0.5685
SPLTI + SCOB ortho+ SCOB phono	0.6007	length	0.6007	0	F(1, 36225)= 0	0.5668

Underscored F-values are highly significant ($p<0.0001$)
(*) 6 usual: word log-Frequency, old20, old20 frequency, pld20, pld20 frequency, word length

8.2.3. Orthographic and phonological regressors cross-validation

The learning and cross-validation of SCOB orthographic and phonological regressors were tested on ELP word naming RT z-scores. Figure 9 shows two distributions of 120 cross-validation r values between ELP word naming RTz and approximations based on orthographic (upper panel), or phonological (lower panel) SCOB codes. Each r value was computed using 2000 randomly selected generalization test words, while 37302 other words were used to compute regression coefficients of the SCOB code components to approximate the data (learning). For the orthographic code, one obtained the learning average r = 0.6975, 99% CI = [0.6973, 0.6976], and the cross-validation average r = 0.4613, 99% CI = [0.4075, 0.5120], corresponding to 21.28% item variance accounted for, and 24.85% systematic item variance accounted for, given the ICC=0.8921. For the phonological code, one obtained the learning average r = 0.7296, 99% CI = [0.7294, 0.7297], and the cross-validation average r = 0.6678, 99% CI = [0.6572, 0.0.6782], corresponding to 44.60% item variance accounted for, and 50% systematic item variance accounted for. As one can see, the phonological SCOB regressor is a very powerful predictor for the word naming RT z-scores. However, one must remember that naming RTs include a substantial part of variance resulting from the particular effects of the first two phonemes of each word on the vocal response recording devices (Rastle, Croot, Harrington, & Coltheart, 2005; Rastle & Davis, 2002; Rey, Courrieu, Madec, & Grainger, 2013). These initial phonemes are encoded in the SCOB phonological code, which probably partially explains the large amount of item variance accounted for.

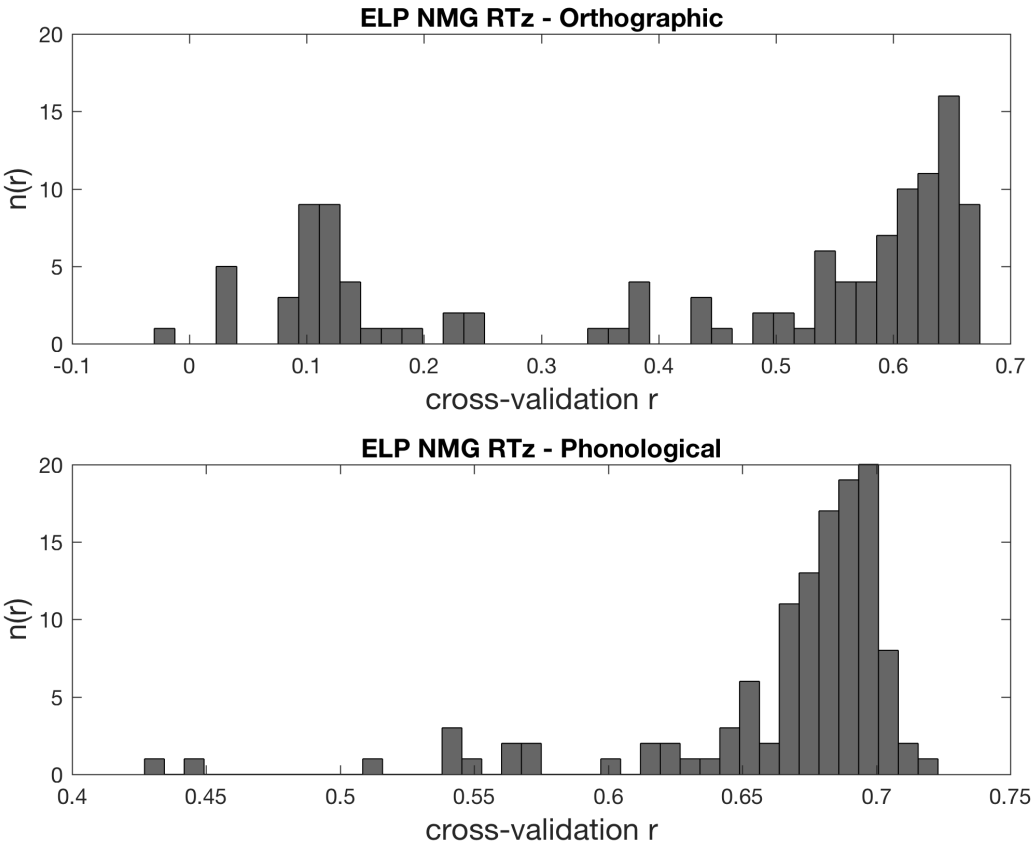

Figure 9. Distribution of 120 cross-validation r values between ELP word naming RT z-scores, and approximations based on orthographic (upper panel), or phonological (lower panel) SCOB codes. Each r value was computed using 2000 randomly selected generalization test words, while 37302 other words were used to compute regression coefficients of the SCOB code components in order to approximate the data (learning). Orthographic : learning average r = 0.6975, 99% CI = [0.6973, 0.6976] , cross-validation average r = 0.4613, 99% CI = [0.4075, 0.5120]. Phonological: learning average r = 0.7296, 99% CI = [0.7294, 0.7297], cross-validation average r = 0.6678, 99% CI = [0.6572, 0.0.6782]. The averages and confidence intervals were computed using Fisher r-to-z transforms.

8.3. Cross-task correlations

Given that the items are the same in the ELP lexical decision database and in the ELP word naming database, one can directly observe and compare the inter-correlations between the regressors and variables of these two databases, which is presented in Table 10. We note that all correlations are substantial and highly significant, while significant correlation

differences (according to Williams T2 test) show that the regressors are logically predominant in one task or in the other. For instance, the SPLTI regressor is strongly correlated with the lexical decision RT z-scores, suggesting a role of the grapheme to phoneme transcoding in the lexical decision process, however, the SPLTI correlation is significantly stronger with the word naming RT z-scores, where phonological processes are clearly involved. One can note that spelling-to-sound regularity effects are usually observed in word naming tasks, but these effects can vanish in lexical decision tasks when the stimuli are presented in capital letters (Hino & Lupker, 2000). However, Parkin and Underwood (1983) showed that spelling-to-sound irregularity effects persist in lexical decision tasks if the stimuli are presented in lower-case letters, which is the case of most stimuli in the ELP database. So, the SPLTI effects observed in the present study are consistent with previous findings.

Table 10. Cross-task correlations of one-dimensional regressors and response z-times from the ELP database. Ao.LD: orthographic SCOB based regressor targeting lexical decision z-times; Ao.NMG: orthographic SCOB based regressor targeting speeded word naming z-times; Ap.NMG: phonological SCOB based regressor targeting speeded word naming z-times; SPLTI: spelling to pronunciation linear transcoding inconsistency; RTz.LD: lexical decision z-times; RTz.NMG: speeded word naming z-times. All correlations are highly significant (p<0.0001). At the right of the table, the correlations of RTz.LD and RTz.NMG with each regressor are compared using Williams T2 test (with 39299 degrees of freedom).

r_{39302}	Ao.LD	Ao.NMG	Ap.NMG	SPLTI	RTz.LD	RTz.NMG	T2: $r_{LD} - r_{NMG}$
Ao.LD	-	0.8997	0.7897	0.6671	0.6428	0.6269	6.59, p<.0001
Ao.NMG	0.8997	-	0.8685	0.6417	0.5784	0.6967	-50.78, p<.0001
Ap.NMG	0.7897	0.8685	-	0.6869	0.6062	0.7287	-55.06, p<.0001
SPLTI	0.6671	0.6417	0.6869	-	0.5317	0.5575	-9.73, p<.0001
RTz.LD	0.6428	0.5784	0.6062	0.5317	-	0.7940	
RTz.NMG	0.6269	0.6967	0.7287	0.5575	0.7940	-	

9. Effect of the size of the learning set

Figure 10 shows the evolution of the learning r (upper panel), and of the cross-validation r (lower panel) of SCOB based regressors targeting ELP lexical decision RT z-scores, as functions of the learning set size, which was varied from 2000 to 36000 items by steps of 2000. The size of the cross-validation set was always 2000 items, and 120 independent tests (with random sampling of the item subsets) were performed for each learning set size.

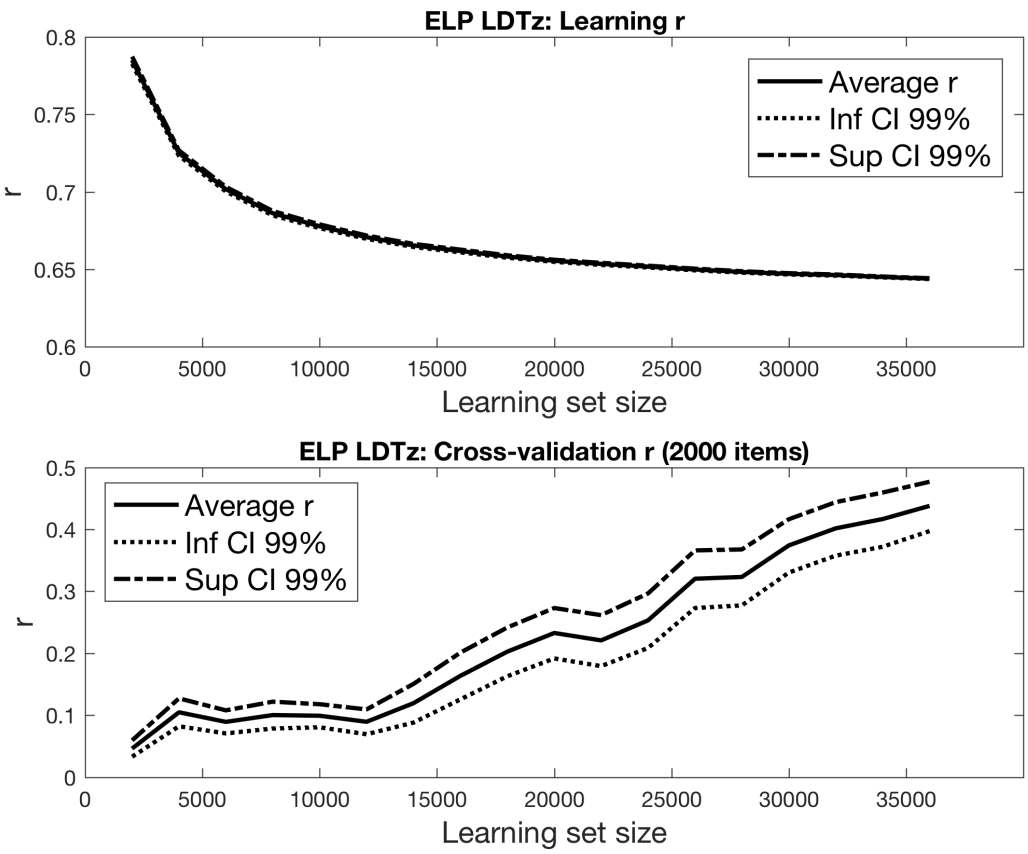

Figure 10. Evolution of the learning r (upper panel) and of the cross-validation r (lower panel) of a SCOB based regressor targeting ELP lexical decision RT z-scores, as functions of the learning set size, varying from 2000 to 36000 items by steps of 2000. The size of the cross-validation set is always 2000 items, and 120 independent tests were performed for each learning set size.

As one can see, the learning set overfitting regularly decreases, while the generalization fit increases as the learning set size increases. The explanation of this is that when the learning set size is moderate, the least squares fitting can exploit item idiosyncratic effects to minimize the residual error. As a result, the fit is high but the generalization power of the resulting coefficients is low. However, as the learning set size increases, the item idiosyncratic effects tend to equilibrate toward a zero-mean idiosyncratic effect, while the least squares fitting exploits only general regularities to minimize the residual error. As a result, the learning fit decreases, but the generalization power of the obtained coefficients increases. This clearly points out the necessity of using databases as large as possible, not only by the number of participants, but also by the number of items, in order to obtain reliable and general results.

10. Conclusion

We have first defined a simple, decodable, numerical spatial code for character strings. The property of being decodable guarantees that the code completely and unequivocally represents the encoded character string, and this also allows one to use this code in models that imply decoding processes. Then we have defined a pure open bigrams coding, which is not simply decodable in the general case. Finally, we have shown that if one makes the simple and natural hypothesis that all character strings begin with a "start character" (tagged by the left whitespace of printed words, for instance), then the initial part of an open bigrams code is equal to the spatial code of the same string, thus the open bigrams code becomes decodable. We tested these coding models on orthographic masked priming data, which clearly showed that coding schemes including open bigrams are significantly much better predictors of priming effects than the simple spatial coding scheme. Then we used the coding schemes to build numerical regressors that were applied to response times (and accuracy data) belonging

to well-known large-scale behavioral databases, namely, the French Lexicon Project (Ferrand et al., 2010), and the two databases of the English Lexicon Project (Balota et al., 2007). It was observed, using hierarchical multiple regression procedures, that the best data fits are always obtained with orthographic (or possibly phonological) open bigrams codes whose initial part is a spatial code, and that both the open bigrams part and the spatial code part of the code provide specific significant contributions to the fit. We also verified that the new orthographic regressors are not redundant with representative samples of usual regressors. An exception to this is the word length (number of letters), whose effect is completely explained by the orthographic code, but the reciprocal is not true. Given that least squares fits, as used in multiple regression analyses, are known to provide overfitting, we defined generalizable one-dimensional regressors computed from the multidimensional orthographic codes, and we validated them using procedures of the type learning and cross-validation (Picard & Cook, 1984). This allowed us to estimate that, in generalization processes, the new orthographic regressors are able to account for about 18.6- 21.3% of the item variance of RT z-scores, when the items learning sets are large enough.

These results not only show that merging the two main current theories of orthographic coding is relevant, but they also provide efficient orthographic or phonological regressors for word recognition modelling and for the analysis of behavioral and electrophysiological data such as ERPs in various word recognition tasks.

Appendix

Matlab/Octave code of useful functions (for academic use only).

```
function [v,alphabet,lsaprx,lscoef] = str2scob(s,p,alphabet,lscoef,data,RL)
% Spatial Coding and/or Open Bigrams coding of character strings.
% Optionally compute regression coefficients and data approximation
% ---------------------------------------------------------------------
```

```
%                 Input arguments:
% s: cell/char array of m strings (m >= 1).
% p: 1x2 vector; SC included if p(1)>1, OB included if p(2)>1.
%  default ([]): p=[1,1], i.e. both SC and OB with power equal to 1.
% data: optional data vector or matrix to be approximated (m-by-dw).
% RL: if provided then the strings are encoded from right to left.
%
%                 Input or output arguments:
% alphabet: optional string of lenght N (set to '' if unknown).
% lscoef: optional least square approximation coefficients such that
%    lsaprx=[ones(m,1),v]*lscoef; (set lscoef to [] if unknown)
%
%                 Output arguments:
% v: table of numerical codes of all strings. The size of v is:
%    m-by-N for SC, m-by-N*N for OB, or m-by-N(N+1) for SC + OB.
% lsaprx: optional least square approximation of data on the v basis.
%
%                 Usage:
% Exemple 1. Simple SCOB encoding
% v=str2scob('word',[1/3 1],'a':'z');
%     result:
% size(v) = [1 702]
%
% Exemple 2. SC encoding, LS coefficients & LS approximation of data
% s{1}='caba'; s{2}='bab'; s{3}='bacaba'; s{4}='ababa'; data=[4;3;6;5];
% p=[1/(6*log(2)),0];      % Note: this is a simple SC since p(2)=0
% [v,alphabet,lsaprx,lscoef]=str2scob(s,p,'',[],data);
%     result:
% v = 0.7560    0.6065    0.8465
%     0.7165    0.8931         0
%     0.7649    0.8589    0.6065
%     0.9037    0.7560         0
% alphabet = 'abc'
% lsaprx = [4.0000; 3.0000; 6.0000; 5.0000]
% lscoef = [-20.4385; 18.8399; 11.1284; 4.0703]
%
% Exemple 3. Reuse of coefficients for generalization on new strings
%     new input:
% t{1}='baa'; t{2}='cabb';
% [v2,alphabet,aprx2]=str2scob(t,p,alphabet,lscoef);
%     result:
% v2 = 0.7899    0.8465         0
%      0.7165    0.6686    0.8465
% aprx2 = [3.8632; 3.9471]
% ----------------------------------------------------------------------
if ischar(s), s=cellstr(s); end
m=length(s);
if (nargin>5) && ~isempty(RL), RL=true; else RL=false; end
if nargin<2 || length(p)<2, p=[1,1]; end
scflag=false; obflag=false;
if p(1)>0, scflag=true; end
if p(2)>0, obflag=true; end
if (nargin<3) || isempty(alphabet)            % Compute the alphabet
    alphabet='';
    for i=1:m
        alphabet=unique(strcat(alphabet,s{i}));
    end
end
N=length(alphabet);
if scflag && obflag
    v=zeros(m,(N+1)*N);
else if scflag
        v=zeros(m,N);
```

```
    else if obflag
            v=zeros(m,N*N);
        else
            error('No coding method selected')
        end
    end
end
sc=[]; ob=[];
for i=1:m                        % Compute codes of the m strings
    si=s{i}; L=length(si);
    if RL, si=fliplr(si); end
    if scflag                    % Spatial Code or start-OB
        sc=zeros(1,N);
     for j=1:L
        c=strfind(alphabet,si(j));
        sc(1,c)=sc(1,c)+2^(-j);
     end
     sc=sc.^p(1);
    end
    if obflag                    % Open Bigrams coding
        ob=zeros(N,N);
     for j1=1:(L-1)
        for j2=(j1+1):L
            c1=strfind(alphabet,si(j1));
            c2=strfind(alphabet,si(j2));
            gap=j2-j1;
            ob(c1,c2)=ob(c1,c2)+2^(-gap);
        end
     end
     ob=ob.^p(2); ob=ob'; ob=ob(:)';
    end
    v(i,:)=[sc,ob];
end
if nargin<4, lscoef=[]; lsaprx=[]; end
% Reuse given lscoef on the codes of new input strings
if (nargin>=4) && ~isempty(lscoef)
    lsaprx=[ones(m,1),v]*lscoef;
end
% Compute lscoef and lsaprx from given data to be approximated
if (nargin>=5) && ~isempty(data) && isempty(lscoef)
    [vh,vw]=size(v); [dh,dw]=size(data);
    if vh~=dh, error('data size error'); end
    lscoef=zeros(vw+1,dw); nzv=find(sum(v)>0);
    x=pinv([ones(m,1),v(:,nzv)])*data; lsaprx=[ones(m,1),v(:,nzv)]*x;
    lscoef([1;nzv(:)+1],:)=x;
end
end


function st = scob2str(v,p,alphabet,RL)
% Decoding of a SC or a SCOB numerical code of a character string
% ---------------------------------------------------------------
%              Input arguments:
% v: SC or SCOB numerical code of a character string
% p: power parameter of the code, or only p(1).
% alphabet: character string including all reference characters
% RL: if provided then the output string is reversed.
%
%              Output argument:
% st: character string resulting from the decoding of v
%
%              Usage:
%    Preliminary encoding:
```

```
% v=str2scob('word',[1/3 1],'a':'z');
%    result:
% size(v) = [1 702]
%    Decoding:
% st=scob2str(v,1/3,'a':'z')
%    result:
% st = word
% ----------------------------------------------------------------
if (nargin>3) && ~isempty(RL), RL=true; else RL=false; end
N=length(alphabet); v=v(1:N); maxlen=-log2(eps);
v(v>1)=1-eps; v(v<0)=0; v= v.^(1/p(1));
  st=''; nextk=1;
  vmax=max(v,[],2);
    while (vmax>=eps) && (nextk<=maxlen)
      j=find(v==vmax,1,'first');
      ch=alphabet(j);
      k=ceil(-log2(v(j)));
      if abs(k-nextk)>1, break, end
      st=strcat(st,ch);
      k=min(k,nextk);
      v(j)=v(j)-2^(-k);
      vmax=max(v,[],2);
      nextk=nextk+1;
    end
if RL, st=fliplr(st); end
end


function [CVrCI99,rcrossval,alphabet,lscoef0,lsaprx0,r0,v0]=...
    crossvalSCOB(s,p,data,Nitcv,Ntest,t)
% Computation and cross-validation of SCOB regressors
% ---------------------------------------------------------------
%                Input arguments:
% s: cell/char array of m strings (m > 1).
% p: 1x2 vector; SC included if p(1)>1, OB included if p(2)>1.
%   default ([]): p=[1,1], i.e. both SC and OB with power equal to 1.
% data: data vector or matrix to be approximated (m-by-dw).
% Nitcv: number of randomly selected items for each cross-validation test
% Ntest: number of cross-validation tests.
% t: optional cell array of dw strings (titles of the output histograms)
%
%                Output arguments:
% CVrCI99: 2(learning, cross-validation) x 3(average r, infCI99%, supCI99%)
%          x dw(independent variables to be predicted) table of r values.
% rcrossval: Ntest x 2(learning, cross-validation) x dw table of r values.
% alphabet: character string.
% lscoef0: global regressions coefficients such that:
%   lsaprx0=[ones(m,1),v0]*lscoef0, and ||lsaprx0 - data|| = min.
% r0: lsaprx0 and data correlation coefficient(s).
% v0: matrix off all SCOB codes of the m strings in s.
%
%                Usage:
% s=FLPword; p=[1,1]; data=[FLPrtz,FLPerr];
% t{1}='FLP RTz'; t{2}='FLP Err'; Nitcv=2000; Ntest=120;
% [CVrCI99,rcrossval,alphabet,lscoef0,lsaprx0,r0,v0]=...
%    crossvalSCOB(s,p,data,Nitcv,Ntest,t);
%    result:
%  size(CVrCI99) = [2 3 2] ...
%      ... and a figure is generated.
% ---------------------------------------------------------------

% Preliminary computation on the whole data set
[v0,alphabet,lsaprx0,lscoef0]=str2scob(s,p,'',[],data);
```

```
r0=correl(lsaprx0,data);
% Cross-validation tests
[m,dw]=size(data); rcrossval=zeros(Ntest,2,dw);
for j=1:Ntest
    remaining=strcat(num2str(100*(1-(j-1)/Ntest)),'% tests')
    % Select 2 independent data subsets
I1=randperm(length(data))';I2=I1(end-Nitcv+1:end);I1=I1(1:end-Nitcv);
    % Estimate the regression coefficients on subset 1
    v=v0(I1,:); d=data(I1,:); [vh,vw]=size(v);
    lscoef=zeros(vw+1,dw); nzv=find(sum(v)>0);
    x=pinv([ones(vh,1),v(:,nzv)])*d; lsaprx=[ones(vh,1),v(:,nzv)]*x;
    lscoef([1;nzv(:)+1],:)=x;
    rcrossval(j,1,:)=correl(lsaprx,d);
    % Perform a cross-validation test on subset 2
    v2=v0(I2,:); d2=data(I2,:);
    aprx2=[ones(Nitcv,1),v2]*lscoef;
    rcrossval(j,2,:)=correl(aprx2,d2);
end
% Mean r values and 99% CIs, computed using Fisher r-to-z transforms
zcrossval=r2z(rcrossval);
mcrossval=mean(zcrossval); stdcrossval=std(zcrossval);
inf99crossval= mcrossval-2.58.*stdcrossval./sqrt(Ntest);
sup99crossval= mcrossval+2.58.*stdcrossval./sqrt(Ntest);
mcrossval=z2r(mcrossval);
inf99crossval=z2r(inf99crossval);
sup99crossval=z2r(sup99crossval);
CVrCI99=[permute(mcrossval,[2 1 3]),permute(inf99crossval,[2 1 3]),...
    permute(sup99crossval,[2 1 3])];
CVrCI99=squeeze(CVrCI99);
rcrossval=squeeze(rcrossval);
for h=1:dw
    % Cross-validation r histogram(s)
    subplot(dw,1,h)
    histogram(rcrossval(:,2,h),40,'FaceColor','black')
    xlabel('cross-validation r','FontSize',14);
    ylabel('n(r)','FontSize',14)
    if (nargin>5) && ~isempty(t)
        title(t{h},'FontSize',12)
    end
end
end

function r=correl(X,Y)
% Correlations between columns of 2 data tables of the same size
sz=size(X); m=sz(1);
sxy=sum(X.*Y)-sum(X).*sum(Y)/m;
scx=sum(X.^2)-sum(X).^2/m;
scy=sum(Y.^2)-sum(Y).^2/m;
r=sxy./sqrt(scx.*scy);
end

function z = r2z(r,n)
% r to z Fisher transformation
if nargin<2, n=2; end
z = sqrt(0.5*(n-1)./n).*log((1+(n-1).*r)./(1-r));
end

function r = z2r(z,n)
% Inverse Fisher transformation (z to r)
if nargin<2, n=2; end
x=exp(z./sqrt(0.5*(n-1)./n));
r = 1-n./(x+n-1);
end
```

```
function [TresAB,alphA,alphB,TmatAB]=...
    TranscobResMat(sA,pA,alph1,sB,pB,alph2,Tmat12)
% Computation of transcoding matrices and transcoding residuals
% ---------------------------------------------------------------------
%         Input arguments:
% sA, sB: char/cell arrays of strings
% pA, pB: 2-vectors of p parameters (default []: p=[1,1])
% alph1, alph2: character strings = alphabets (default '')
% Tmat12: optional transcoding matrix
%         Output arguments:
% TresAB: transcoding residual norms of sA strings to sB strings
% alphA, alphB: alphabets (useful if alph1 and alph2 are unknown)
% TmatAB: transcoding matrix
%         Usage:
%  Compute the SPLTI and the transcoding matrix of the FLP database:
% [FLP_SPLTI,FLPalO,FLPalP,FLPmatOP]=...
%     TranscobResMat(FLPword,[],'',FLPpron,[],'');
%  Compute the SPLTI of a list of French words 'FrWd':
% FrWd_SPLTI=TranscobResMat(FrWd,[],FLPalO,FrWdPron,[],FLPalP,FLPmatOP);
% ---------------------------------------------------------------------
[vA,alphA]=str2scob(sA,pA,alph1); [vB,alphB]=str2scob(sB,pB,alph2);
if nargin<7
    nzA=(sum(vA)>0); nzB=(sum(vB)>0);
    TmatAB=zeros(length(nzA),length(nzB));
    x=pinv(vA(:,nzA))*vB(:,nzB);
    TmatAB(nzA,nzB)=x;
else
    TmatAB=Tmat12;
end
TresAB=sqrt(sum((vA*TmatAB - vB).^2,2));
end
```

Funding: This work, carried out within the Labex BLRI (ANR-11-LABX-0036) and the Institut Convergence ILCB (ANR-16-CONV-0002), has benefited from support from the French government, managed by the French National Agency for Research (ANR) and the Excellence Initiative of Aix-Marseille University (A*MIDEX)

References

Adelman, J. S., Johnson, R. L., McCormick, S. F., McKague, M., Kinoshita, S., Bowers, J. S., Perry, J. R., Lupker, S. J., Forster, K. I., Cortese, M. J., Scaltritti, M., Aschenbrenner, A. J., Coane, J. H., White, L., Yap, M. J., Davis, C., Kim, J., Davis, C. J. (2014). A behavioral database for masked form priming. *Behavior research methods*, *46*(4), 1052-1067.

Balota, D. A., Yap, M. J., Cortese, M. J., Hutchison, K. A., Kessler, B., Loftis, B., Neely, J. H., Nelson, D. L., Simpson, G. B., & Treiman, R. (2007). The English Lexicon Project. *Behavior Research Methods,* 39, 445–459.

Cohen, J., Cohen, P., West, S.G., & Aiken, L.S. (2003). *Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences (3rd Ed.)*. London, Lawrence Erlbaum Associates, Publishers.

Conrad, M., Grainger, J., & Jacobs, A. M. (2007). Phonology as the source of syllable frequency effects in visual word recognition: Evidence from French. *Memory & Cognition*, *35*(5), 974-983.

Courrieu, P. (2012). Density Codes, Shape Spaces, and Reading. *ERMITES 2012 : Representations and Decisions in Cognitive Vision*. La Seyne-sur-Mer, August 30-31 and September 1. Proceedings : http://glotin.univ-tln.fr/ERMITES12/

Courrieu, P., Brand-D'Abrescia, M., Peereman, R., Spieler, D., & Rey, A. (2011). Validated intraclass correlation statistics to test item performance models. *Behavior Research Methods, 43*, 37-55. doi: 10.3758/s13428-010-0020-5

Courrieu, P., & Rey, A. (2011). Missing data imputation and corrected statistics for large-scale behavioral databases. *Behavior Research Methods, 43*, 310-330. doi: 10.3758/s13428-011-0071-2

Courrieu, P., & Rey, A. (2015). General or idiosyncratic item effects : what is the good target for models ? *Journal of Experimental Psychology : Learning, Memory, and Cognition, 41(5),* 1597-1601. DOI : 10.1037/xlm0000062

Damerau, F. J. (1964). A technique for computer detection and correction of spelling errors. *Communications of the ACM, 7(3)*, 171-176.

Davis, C. J. (1999). *The Self-Organising Lexical Acquisition and Recognition (SOLAR) model of visual word recognition*. Unpublished doctoral dissertation, University of New SouthWales, Australia.

Davis, C.J. (2010). The spatial coding model of visual word identification. *Psychological Review, 117(3)*, 713-758.

Davis, C. J., & Bowers, J. S. (2006). Contrasting five different theories of letter position coding: Evidence from orthographic similarity effects. *Journal of Experimental Psychology: Human Perception and Performance*, *32*(3), 535-557.

Ferrand, L., New, B., Brysbaert, M., Keuleers, E., Bonin, P., Méot, A., Augustinova, M., & Pallier, C. (2010). The French Lexicon Project: Lexical decision data for 38,840 French words and 38,840 pseudowords. *Behavior Research Methods, 42*, 488–496.

Faust, M. E., Balota, D. A., Spieler, D. H., & Ferraro, F. R. (1999). Individual differences in information-processing rate and amount: Implications for group differences in response latency. *Psychological Bulletin, 125*, 777–799.

Golub, G. H., & Reinsch, C. (1970). Singular value decomposition and least squares solutions. *Numerische Mathematik, 14(5)*, 403-420.

Grainger, J., Granier, J.P., Farioli, F., Van Assche, E., & van Heuven, W. (2006). Letter position information and printed word perception: The relative-position priming constraint. *Journal of Experimental Psychology: Human Perception and Performance, 32*, 865–884.

Grossberg, S. (1978). A theory of human memory: Self-organization and performance of sensory-motor codes, maps, and plans. *In* R. Rosen & F. Snell (Eds.), *Progress in theoretical biology* (pp. 233–374). New York, NY: Academic Press.

Grossberg, S., & Pearson, L. R. (2008). Laminar cortical dynamics of cognitive and motor working memory, sequence learning and performance: toward a unified theory of how the cerebral cortex works. *Psychological Review*, *115*(3), 677.

Hannagan, T., & Grainger, J. (2012). Protein analysis meets visual word recognition: A case for string kernels in the brain. *Cognitive Science, 36*, 575–606.

Hauk, O., Davis, M.H., Ford, M., Pulvermüller, F., and Marslen-Wilson, W.D. (2006). The time course of visual word recognition as revealed by linear regression analysis of ERP data. *NeuroImage, 30*, 1383–1400.

Hino, Y., & Lupker, S.J. (2000). Effects of word frequency and spelling-to sound regularity in naming with and without preceding lexical decision. *Journal of Experimental Psychology: Human Perception and Performance, 26(1)*, 166-183.

Keuleers, E., Diependaele, K., & Brysbaert, M. (2010). Practice effects in large-scale visual word recognition studies: A lexical decision study on 14,000 Dutch mono- and disyllabic words and nonwords. *Frontiers in Psychology, 1, 174*. doi:10.3389/fpsyg. 2010.00174

Keuleers, E., Lacey, P., Rastle, K., & Brysbaert, M. (2012). The British Lexicon Project: Lexical decision data for 28,730 monosyllabic and disyllabic English words. *Behavior Research Methods, 44(1)*, 287-304.

Kinoshita, S., & Norris, D. (2013). Letter order is not coded by open bigrams. *Journal of memory and language*, *69*(2), 135-150.

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., & Watkins, C. (2002). Text classification using string kernels. *Journal of Machine Learning Research, 2*, 419–444.

Lupker, S. J., Zhang, Y. J., Perry, J. R., & Davis, C. J. (2015). Superset versus substitution-letter priming: An evaluation of open-bigram models. *Journal of Experimental Psychology: Human Perception and Performance*, *41*(1), 138-151.

Madec, S., Le Goff, K., Anton, J-L., Longcamp, M., Velay, J-L., Nazarian, B., Roth, M., Courrieu, P., Grainger, J., & Rey, A. (2016). Brain correlates of phonological recoding of visual symbols. *NeuroImage, 132*, 359-372. doi: 10.1016/j.neuroimage.2016.02.010

McGraw, K.O., & Wong, S.P. (1996). Forming inferences about some intraclass correlation coefficients. *Psychological Methods, 1(1)*, 30-46.

New, B., Pallier, C., Brysbaert, M., & Ferrand, L. (2004) Lexique 2 : A New French Lexical Database. *Behavior Research Methods, Instruments, & Computers*, 36 (3), 516-524. (Current version: Lexique 3)

Parkin, A.J., & Underwood, G. (1983). Orthographic vs. phonological irregularity in lexical decision. *Memory & Cognition, 11(4)*, 351-355.

Perry, C., Ziegler, J. C., & Zorzi, M. (2007). Nested incremental modeling in the development of computational theories: the CDP+ model of reading aloud. *Psychological Review, 114*, 273-315.

Perry, C., Ziegler, J. C., & Zorzi, M. (2010). Beyond single syllables: Large-scale modeling of reading aloud with the Connectionist Dual Process (CDP++) model. *Cognitive Psychology, 61(2)*, 106-151.

Picard, R., & Cook, D. (1984). Cross-validation of regression models. *Journal of the American Statistical Association, 79 (387)*, 575–583.

Rastle, K., Croot, K. P., Harrington, J. M., & Coltheart, M. (2005). Characterizing the motor execution stage of speech production: Consonantal effects on delayed naming latency and onset duration. *Journal of Experimental Psychology. Human Perception and Performance, 31*, 1083–1095.

Rastle, K., & Davis, M. H. (2002). On the complexities of measuring naming. *Journal of Experimental Psychology. Human Perception and Performance, 28*, 307–314.

Rey, A., & Courrieu, P. (2010). Accounting for item variance in large-scale databases. *Frontiers in Psychology 1:200*. doi:10.3389/fpsyg.2010.00200

Rey, A., Courrieu, P., Madec, S., & Grainger, J. (2013). The unbearable articulatory nature of naming: on the reliability of word naming responses at the item level. *Psychonomic Bulletin & Review, 20(1),* 87-94. DOI:10.3758/s13423-012-0336-5

Rey, A., Courrieu, P., Schmidt-Weigand, F., & Jacobs, A.M. (2009). Item performance in visual word recognition. *Psychonomic Bulletin & Review, 16(3)*, 600-608

Rey, A., Madec, S., Grainger, J., Courrieu, P. (2013). Accounting for variance in single-word ERPs. Oral communication presented at the *54th Annual Meeting of the Psychonomic Society*, Toronto, Canada, November 14-17.

Shrout, P. E. & Fleis, J. L. (1979). Intraclass Correlations: Uses in Assessing Rater Reliability. *Psychological Bulletin, 86*, 420-428.

Steiger, J. H. (1980). Tests for comparing elements of a correlation matrix. *Psychological Bulletin, 87(2)*, 245–251.

Theil, H. (1961). *Economic Forecasts and Policy* (2nd ed., 3rd printing, 1970). Amsterdam, North-Holland Publishing Company.

Van Assche, E., & Grainger, J. (2006). Astudy of relative-position priming with superset primes. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 32*, 399–415.

Welvaert, M., Farioli, F., & Grainger, J. (2008). Graded effects of number of inserted letters in superset priming. *Experimental Psychology, 55(1)*, 54–63.

Whitney, C. (2001). How the brain codes the order of letters in a printed word: The SERIOL model and selective literature review. *Psychonomic Bulletin & Review, 8*, 221–243.

Williams, E. J. (1959). The comparison of regression variables. *Journal of the Royal Statistical Society, Series B, 21*, 396–399.

Yap, M. J., & Balota, D. A. (2009). Visual word recognition of multisyllabic words. *Journal of Memory & Language, 60*, 502-529.

Yarkoni, T., Balota, D. A., & Yap, M. J. (2008). Moving beyond Coltheart's N: A new measure of orthographic similarity. *Psychonomic Bulletin & Review, 15*, 971–979.