Modeling spatial–temporal operations with context-dependent associative memories

Eduardo Mizraji¹ · Juan Lin²

Received: 2 December 2014 / Revised: 31 March 2015 / Accepted: 7 May 2015 / Published online: 17 May 2015
© Springer Science+Business Media Dordrecht 2015

Abstract We organize our behavior and store structured information with many procedures that require the coding of spatial and temporal order in specific neural modules. In the simplest cases, spatial and temporal relations are condensed in prepositions like “below” and “above”, “behind” and “in front of”, or “before” and “after”, etc. Neural operators lie beneath these words, sharing some similarities with logical gates that compute spatial and temporal asymmetric relations. We show how these operators can be modeled by means of neural matrix memories acting on Kronecker tensor products of vectors. The complexity of these memories is further enhanced by their ability to store episodes unfolding in space and time. How does the brain scale up from the raw plasticity of contingent episodic memories to the apparent stable connectivity of large neural networks? We clarify this transition by analyzing a model that flexibly codes episodic spatial and temporal structures into contextual markers capable of linking different memory modules.

Keywords Neural computation · Cognitive order relations · Hierarchical models · Context-dependent associative memories

Introduction

From the very beginning of natural science, physical theories have continuously provided powerful and refined concepts about the nature of space and time. However, preceding this elaborate and highly technical knowledge, human cognition has developed since the birth of language, remarkable abilities to identify and communicate the sense of time flow and the relative positions of objects and events in space. We spontaneously know and transmit to others that yesterday events happened before those of today, and that the events of tomorrow will happen after today. We do know and can communicate factual information when we say that a tree grows behind a house or that a book sits on a table. These cognitive capacities would seem trivial because of their remarkable familiarity, but involve nevertheless complex neural computations. It is a remarkable fact that with the exception of some illusory perceptions, cognitive processing of space and time relations is empirically consistent with external reality. This processing happens in the spontaneous, pre-scientific experiential level and on time and space scales of perceptually and pragmatically observed phenomena.

We shall assume that complex brain dynamics support these cognitive activities by structuring layers of neural modules capable of performing specific tasks, including syntactic parsing, conceptual combinations, semantic evaluations and many logical arguments and linguistic performances (Eliasmith et al. 2012; Szelag et al. 2009; Wang et al. 2014). In this work, we are concerned with

Partial results of this study were presented in the Tandem Workshop on Optimality in Language and Geometric Approaches to Cognition, Berlin, December 11–13, 2010.

✉ Eduardo Mizraji
mizraj@fcien.edu.uy; emizraji@gmail.com

Juan Lin
jlin2@washcoll.edu

¹ Group of Cognitive Systems Modeling, Biophysics Section, Facultad de Ciencias, Universidad de la República, Igua 4225, 11400 Montevideo, Uruguay

² Department of Physics, Washington College, Chestertown, MD 21620, USA

neural computations that occur on time scales much longer than the time scales of sensory processing. In fact, the conceptual processing mechanisms we intend to model spring from the raw data provided by the complex processing of sensory organs.

The search for the neural processing of order relations has been pre-dated by a series of investigations into the neural codifications of number systems and their operations.

In the decade 1990 to 2000, numerous theoretical and empirical investigations into the concept of number and the detection of “numerosity” were based on the existence of parallel neural codes. Some of these codes were correlated to lexicons or visual imageries characterizing a number, while other codes were correlated to qualitative notions about the magnitude of a number (e.g. small, medium and large) that seem closely connected to relations of order. Theoretically, these approaches had a lasting influence on neural network modeling (Anderson 1995, Chapter 17; Dehaene and Changeux 1993; Dehaene et al. 1998). An important earlier work on the analog processing by the brain of certain mathematical problems is Davis and Anderson (1979). As we shall see, our proposed model is partially based on the qualitative codes previously developed to explain the ordering of numbers.

One of the principal aims of this communication is to present a set of neural modules capable of structuring a hierarchical system that computes order relations. These order relations are frequently coded in strategic words (prepositions), which provide linguistic procedures—including both the production as well as the decoding of language—with a blueprint for the organization of recorded events in space and time. We shall present a heuristic approach based on the theory of distributed neural memories to compute some of the activities displayed by the brain while it processes databases already preinstalled on memory modules.

The problem

We state our problem as follows: How can we build a neural model capable of representing the coding of spatial and temporal relations in the cognitive space created by the human mind? We want to emphasize from the very beginning some important epistemological points. Presently, the mathematical modeling of neural networks does not provide sufficient information to identify unique solutions from experimental data. This single fact suggests that a family of acceptable and coexistent neural models can explain provisionally, certain areas of neurobiological and cognitive behaviors. In this article, we shall explore only one member of this family of neural

models under the conviction that all hierarchical modular structures can be translated into different mathematical representations. However, each representation would retain similar computing abilities and exhibit similar dynamic behavior.

Another important associated issue, in fact, a corollary of the main problem previously stated is concerned with the factual bases of heuristic theories based on neural networks. We shall postulate that in the context of vector symbolic architectures, many cognitive activities linked to language processing—e.g. the formal reasoning activated in proving a theorem or the establishment of spatial and temporal order of events during a verbal description of a story—can indirectly be considered as quasi-operational observables. Obviously, these quasi-observables are far echoes of measurable bioelectrical and neurochemical activities as most cognitive computations are represented by operators that are not directly observable (beim Graben and Potthast 2009). Yet, the cognitive operations herein presented provide an organizational structure in models that connect cognitive computations with their electrochemical and neuronal substrates. Modules based on matrix memories are uniquely adaptable to operate on several levels of information processing.

A brief summary of our results

In our search for solutions to the issues posed in the previous section, we shall develop a hierarchical neural model capable of detecting order relations on three levels of abstraction. The neural modules first start with concrete information and then move toward abstract conceptual computations.

At the most basic echelon, the High-Level Processing (Level 1) is where the ordering of events associated with our immediate experience—Is a dog smaller than an elephant?—is processed by modular associative memories that map asymmetric events into qualitative markers of order by size. In our example, “elephant” is associated with ‘large’ and “dog” with ‘medium’. These associations can also be applied to temporal order—Which event came first, the American or French revolution?—by mapping outputs as time labels ‘anterior’ or ‘posterior’. On this level the attributes of order are experiential, not logical. The connection between complex processes of temporal perception and cognitive representations of simple propositions leading to neural computations of time positions is an on-going research theme (Szelag et al. 2009).

The next level of abstraction, the Medium-Level Processing (Level 2) is where we find neural modules operating on the outputs of Level 1. These modules evaluate a question and establish a question-driven order of events.

The outcome of this neural processing is an abstract conceptual label indicating the ordering sequence involved in the query and diagnosed by experiential Level 1. The final stage, the Low-Level Processing (Level 3), is where the ordered outputs from Level 2 are processed by abstract neural modules that compute a final answer in binary form, Yes or No.

In the hierarchical approach, an output of a given level is an input to the next level. The output at the highest level produces an answer to a query or a diagnostic to an ordering sequence that is compatible with experience. Thus, the basic unit of information is a pair of connected patterns to which we add a query operating across three levels. On Level 1 a query is processed as

$$\text{Level 1 : } [query, (f, f')] = [query_{<}, (u, v)] \quad (1)$$

The $query_{<}$ is a question of the type, “Is f smaller than f' ?” and the pair (u, v) symbolizes the positions of order assigned to (f, f') . When f is “dog” and f' is “elephant”, the information stored in these memories suggests u as ‘medium’ (or ‘anterior’) and v as ‘large’ (or ‘posterior’). Another possibility is the processing of $query_{>}$, which asks the opposite question, “Is f larger than f' ?” On Level 2, the triplet on the right hand side of Eq. (1) is an input that yields a pair of labels (a, b) , equally valid for spatial or temporal order relations,

$$\text{Level 2 : } [query_{<}, (u, v)] = [query_{<}, (a, b)] \quad (2)$$

On Level 3, neural modules process the now abstract order variables and decide on the correct diagnosis, which is triggered by a context-dependent query,

$$\text{Level 3 : } [query_{<}, (a, b)] = \text{Final answer (Yes or No)} \quad (3)$$

In the example of “dog versus elephant” the answer will be Yes. We shall assume that this final low-level diagnosis triggers further neural actions that travel along the hierarchical scale to make contact with high-level neural motor

interfaces and consequently produce behavioural responses. Figure 1 depicts the levels involved in our model.

Context-modulated associative memories

In the decade following 1980, neural network models were developed to do computations in situations involving semantic contexts (Arbib 1995). Multilayer memories trainable via back propagation (BP) algorithms advanced neural computing dramatically and enlarged the scope of theoretical and applied neural models (Rumelhart et al. 1986a, b). The multilayer scheme was particularly important because it overcame the inability of one-layer perceptrons and matrix memories to operate on semantic contexts, a limitation associated with the failure to compute the logical function exclusive-or (XOR) (Anderson 1995). The extraordinary neurocomputational advances promoted by BP algorithm did not generate parallel advances on formal mathematical theories of neurodynamics that would capture the complexity of interacting neural modules. One reason can be traced to the mathematical complexities of BP algorithm, a gradient-descent procedure combining chain-rule error minimization with the necessary non-linear input–output neuron responses in some layers.

However, a different way to deal with problems involving semantic contexts or equivalently, to construct neural networks capable of solving the XOR problem, was implicit in many independent proposals that postulated some kind of multiplicative capacity during synaptic processing (Koch and Poggio 1992; Mel 1992). Kohonen (1977) explored the possibility that associative matrix memories could enhance their recognition capacities by undergoing some non-linear processing of input vectors. Later, Pike (1984) proposed a matrix scalar product that included context modulation of data (Humphreys et al. 1989). From an engineering framework, Pao (1989) explored the effectiveness of non-linear processing of inputs in associative memories; he showed that “functional links” were able to solve the XOR-problem more efficiently than the BP algorithm—at least in the small-sized examples he analyzed. About the same time, Paul Smolensky (1990) described his very influential tensor approach and established a relationship between this formalism and Pikes’ proposal. Smolensky also introduced filler/role bindings for the representation of complex data structures, such as syntactic phrase structure trees. In order to counter the combinatorial explosion and hence the explosion of spatial dimensionality he uses some compression techniques, such as circular convolution. However, these non-linear mappings result in tensor product representations that may not be faithful anymore, in contrast to linear operators based on dyadic products of orthogonal bases (Smolensky 2006; Smolensky and Legendre 2006; beim Graben and

Fig. 1 Drawing showing the three processing levels involved in the model. The information flows from a large variety of inputs processed at Level 1 towards a more abstract but reduced variety of inputs processed at Level 3

Gerth 2012). Interestingly, these contributions have become recently recognized in Eliasmith et al. (2012) “neural engineering” approach.

One approach to represent cognitive processes is using dynamic neural fields to take advantage of their infinite-dimensional representation spaces—as function spaces over some manifolds—thereby preventing the compression problems of tensor product based neural networks. All that is needed in this formulation would be to select one or more orthonormal bases for representing symbolic expressions and expressing cognitive operations as operators on a function space. Moreover, dynamic neural fields have a straightforward interpretation as neural population models. The underlying manifold can be interpreted as a “feature space” and regions in that space represent neural populations or activity modes (Erlhagen and Schöner 2012; beim Graben and Potthast 2014).

Mizraji (1989) proposed another approach involving multiplicative processes. It was rooted in Ross Ashby’s theory of adaptive control systems (Ashby 1956, 1960) and allowed for a neural implementation of Ashby’s machine with input. In Ashby’s theoretical approach, machine actions modulated by potentially arbitrary parameters were essential precursors of complex adaptive behaviors. These parameters allow for evolutionary adaptation to changing and unpredictable environments. Neural versions of these adaptive machines are memories that modulate associations among arbitrary semantic contexts—usually symbolized by high dimensional vectors. This approach is based on the properties of the Kronecker product, which represents tensors of rank two as matrices and also tensors of higher rank via recursions (Mizraji 2008a; Mizraji et al. 2009). Context-dependent associative memories act as matrix operators that can perform the calculus of symbolic logic. Their study has helped to clarify some aspects of formal human reasoning (Mizraji 2008a; Mizraji and Lin 2011).

In the following sections we shall assume that cognitive processes composed of large clusters of electrochemical signals are most naturally coded in high-dimensional vectors and processed by natural modules sustaining associative memories. The specificity of these memories depends on synaptic connectivity—for classic panoramic introductions to the biological and mathematical study of these memories refer to Anderson and Rosenfeld (1988) and Anderson (1995). Associative memories are structured around input–output vectors, and stored vectors become statistical averages of conceptual representations within the neural system (Cooper 1974).

The simplest structure of an associative matrix memory is of the form

$$A = \sum_{i=1}^K \mu_i g_i f_i^T \quad (4)$$

The vector f_i represents an input pattern associated with the output g_i . Each pair (input, output) = (f_i, g_i) belongs to a real (n, m) -dimensional Euclidean vector space. The upper index T is the transpose operation. The real numbers μ_i represent frequencies of associated pairs (f_i, g_i) in the experience of consolidating a memory. In the distributed, associative memory (4), K associated pairs of patterns have been stored in a rectangular matrix where each coefficient in the matrix represents a synaptic strength during signal transduction. An analysis of Eq. (4) shows that associated pairs become scattered and superimposed among all matrix coefficients. The mixing of patterns clarifies some cardinal properties of distributed memories: high tolerance to partial physical deterioration in the material substrate of the memory, reconstruction of partial or fuzzy patterns, and conceptualization via statistical averaging (Cooper 1974; Kohonen 1977). Matrix A in (4) processes an input f_h in the following way,

$$A f_h = \sum_{i=1}^K \mu_i g_i \langle f_i, f_h \rangle \quad (5)$$

The scalar product $\langle f_i, f_h \rangle$ measures the similarity between input and stored patterns—similar means quasi-parallel, $\langle f_i, f_h \rangle \approx 1$ and very different means quasi-orthogonal, $\langle f_i, f_h \rangle \approx 0$. To simplify the mathematical formalism, we shall assume that different concepts map onto orthonormal vectors. This assumption is discussed in Anderson (1972). Kohonen (1977) showed that the theory could also be formulated with non-orthogonal vectors by replacing the transpose by an adjoint operation, e.g. by computing Moore–Penrose pseudoinverses. However, orthonormality tends to be approximately true for most large and sparse random vectors (Amari 1977). We can generalize this formalism to represent context-dependent matrix memories as

$$M = \sum_{i,j} \mu_{ij} g_{ij} (p_j \otimes f_i)^T \quad (6)$$

Contexts p_j and key inputs f_i vectors are stored by means of a Kronecker (or tensor) product $p_j \otimes f_i$ (Mizraji 1989, 2008b). Contextualized inputs $p_k \otimes f_h$ are processed by memory M as follows (Graham 1981):

$$M(p_k \otimes f_h) = \sum_{i,j} \mu_{ij} g_{ij} \langle p_j, p_k \rangle \langle f_i, f_h \rangle \quad (7)$$

The computation in (7) implies a double filtering via two scalar products, which enhances the computational ability of M , including the quasi-trivial solution of the XOR problem and the implementation of matrix logical gates (Mizraji 1992). When dealing with high-dimensional random vectors, the Kronecker product can be imperfectly evaluated with minimal deterioration of computational

performance (Mizraji et al. 1994). In the latter case we encounter a kind of statistical Kronecker product where the key input is weighted by the random components of the context vector.

An operational procedure to produce contextualized outputs can be achieved by memory modules built on the following structure,

$$M = \sum_{ij} \mu_{ij} (p_{ij}^* \otimes g_{ij}) (p_{ij} \otimes f_{ij})^T \quad (8)$$

The Kronecker product in (8) can be manipulated to yield an equivalent memory

$$M = \sum_{ij} (p_{ij}^* p_{ij}^T \otimes \mu_{ij} g_{ij} f_{ij}^T) \quad (9)$$

It will be discussed later that this form of M will be essential to maintain a query as a constant label through the three levels of cognitive processing. In fact, a query is the context that will identify the final answer.

Computing with words

Language is a natural target for cognitive and neurocomputational theories. Lipinski et al. (2006) have correlated data about spatial working memory and spatial language using a neural model based on dynamic field theory. This approach has been expanded (Lipinski et al. 2009; Ursino et al. 2011) with dynamic neural fields that integrate spatial language coding and spatial sensory-motor systems, or generate category formation and semantic memories to produce context-dependent behavioral responses upon cue presentation.

The tensor representation of matrix memories was an important component of the Optimality Theory developed by Prince and Smolensky (1994), a theory widely applied to the investigation of relationships between language and cognition (see for instance, Besnard et al. 2003). Recently, beim Graben and collaborators have used these tensor representations to explore linkages among syntactic and semantic language structures and the core neural dynamics that sustain these structures (beim Graben et al. 2008a, b; beim Graben and Gerth 2012); in their approach, a correspondence is established between neural observables and the syntactic structures that support semantic constructions.

The inherent structure of language requires some complex computing capacities in the human brain (see Blutner 2004). For instance, we compute with words; when using “and” we produce the conjunction of conceptually coded objects in a way that is similar to what happens with the conjunction operator in formal logic. In more complex situations our brain is capable of computing logical modalities when confronted with phrases such as: “to live, it is necessary to

breathe” or “to live, it is not possible not to breathe”. The words “necessary” and “possible” are logical modalities, and the word “not” is associated to logical negation. It was already known to Aristotle (350 BC) that the mind computes the last two phrases as being equivalent. Context-dependent matrix memories can also compute modalities by means of iterated Kronecker products of conjunction or disjunction matrices (Mizraji and Lin 1997). Another possibility to compute multivalued logical schemes is by using partially overlapping neural spike trains to address and process information in the brain (Bezrukov and Kish 2009).

Some basic prepositions of common languages imply another kind of computation with words, one that involves spatial or temporal order relations. In phrases such as “The notebook is on the table” or “He is behind you”, the prepositions “on” and “behind” create in our brain an emergent, well-defined imaginary scenario. We can naturally postulate that these words articulate a variety of logical and asymmetrical order operations—such as before or after, on or under, from or towards—that are computed by complex neural programs. A preposition, as well as some logical words, acts as a kind of password that accesses specific neural computations. How are neural processes structured to give meaning to these words and how do we model neural operators behind these neural computations? In the next subsection, we shall invoke the mathematical formalism of matrix associative memories to help us build operators representing asymmetrical prepositions.

Neural modules for computing order relations

Inspired in part by a neural theory of logical operations (Mizraji and Lin 2002, 2011), we will construct elementary neural models capable of representing the abstract computations triggered by asymmetrical prepositions such as “from” or “towards”. The proposed matrix memories have certain formal similarities to the operators used in temporal and/or tense logics (Prior 1967, 1968; Rescher and Urquhart 1971; Øhrstrøm and Hasle 1995). These logics combine the truth-values of Boolean logic with new logical operators representing positions in time or space. We proceed to define a list of matrix modules that compute, given some particular empirical information, the truth or falsity of an assumed order relation, e.g. “the book is on the table”, or “a heat wave passed before the storm”. We base our theory on the assumption that asymmetric prepositions are installed as neural versions of anti-commutative functions,

$$f(x, y) = -f(y, x) \quad (10)$$

The translation of this property to the language of neural modules requires a re-coding of information stored in the variables x and y , to analogous column vectors u and v . Let

us use a hybrid representation based on the logical column vectors s (true) and n (false) by assigning specific coding vectors to the positional pairs.

$$\Psi(u, v) = N\Psi(v, u), \quad u, v \in \mathbb{R}^n \quad (11)$$

Here, N is a projection operator corresponding to logical negation—the neural version of the minus sign in Eq. (10). In our view, Eq. (11) can be directly associated to prepositions that involve asymmetric operations. It simplifies the execution of cognitive shortcuts within the multilevel scheme we describe. These shortcuts appear in situations such as “bring me the book on the table”. The objects “book” and “table” may be construed as neural vectors u and v , and “on” as an antisymmetric operator Ψ installed previously as a neural module. The whole sentence may correspond to a well-known syntactic structure (beim Graben and Gerth 2012).

A small repertoire of modular processors

In order to construct matrix modules, the following definitions are needed,

$$\text{Logic truth-values : } \Lambda = \{s, n\}, s, n \in \mathbb{R}^m \quad (12)$$

The vector s codes for an affirmative diagnosis (“true” in propositional logic) and n codes for a negative diagnosis (“false”).

$$\text{Positional values : } \Omega = \{b, i, a\}, b, i, a \in \mathbb{R}^n \quad (13)$$

These three vectors code for positional information, b corresponding to “before”, a to “after” and i to an intermediate position between b and a . In order to simplify these representations, we are going to assume that all these vectors form an orthonormal set. The events involved in the description of an episode can be classified by patterns parameterized as functions of their corresponding position vectors. As an example, the associations between vectors and patterns translate to,

$(b, \text{pattern}) \rightarrow$ Pattern positioned “before” (or “under”)

$(i, \text{pattern}) \rightarrow$ Pattern positioned “between”

$(a, \text{pattern}) \rightarrow$ Pattern positioned “after” (or “on”)

A small sample of basic positional operators is given below.

Monadic operators for temporal order

These operators are generated through the following mappings:

$$Op1: \Omega \rightarrow \Lambda \quad (14)$$

These are matrices that compute (similar to the classic operators F and P of temporal logic) an answer to the following questions:

Matrix F : Will this event happen in the future?

Matrix P : Did this event happen in the past?

$$F = nb^T + sa^T, P = sb^T + na^T \quad (15)$$

The two operators are connected through the negation operator $N = sn^T + ns^T$ (Mizraji and Lin 2011), $F = NP$

Dyadic operators for asymmetric prepositions

Here we have functions of the type

$$Op2: \Omega \times \Omega \rightarrow \Lambda \quad (16)$$

Let A be a matrix that codes abstract order relations and processes questions such as: “is u in front of v ?” or “is u on v ?” A possible form for this matrix would be

$$A = s(a \otimes b)^T + n(b \otimes a)^T \quad (\text{Matrix “after”}) \quad (17)$$

Another operator for the questions “is u behind v ?” or “is u under v ?” would be,

$$B = n(a \otimes b)^T + s(b \otimes a)^T \quad (\text{Matrix “before”}) \quad (18)$$

The following properties hold, $A(a \otimes b) = NA(b \otimes a)$, and $B = NA$

Dynamic order

The words “towards” and “from” describe dynamic order relations that code transitions. We can order these high-level processors with the following matrices:

Matrix “Towards”

$$Tow = n(b \otimes a)^T + n(b \otimes i)^T + n(i \otimes a)^T + s(a \otimes b)^T + s(a \otimes i)^T + s(i \otimes b)^T \quad (19)$$

Matrix “From”

$$Fro = NTow \quad (20)$$

In the figure below, we show a diagram of these dynamic connections (Fig. 2).

If the intermediate value i does not exist, these matrices reduce to the previous matrices A and B . The coding of all intermediate positions between b and a by a single vector representing an undetermined “fuzzy” position is similar to the strategy developed by J. Lukasiewicz to define modal operators using a 3-valued logic.

Fig. 2 Graph showing the markers b , i and a , together with the direction (arrows) of the operations “From” and “Towards”

A multi-level model

We are frequently confronted with the task of establishing an ordering sequence, be it spatial (Is the trachea of humans in front or behind the esophagus?), or temporal (Was Julius Caesar emperor of Rome before Nero?). To resolve this issue we construct visual representations of these events in timelines that are projected onto some sort of “interior screens”. Looking into one of these screens is analogous to a translation of linguistic codes into neural maps that activate certain areas of the visual cortex (Elia-smith et al. 2012). Visualization means translating one coding type into another, but does it solve the ordering sequence? To obtain an unequivocal answer the brain requires additional information stored in some memories. Even if we memorize and are able to visualize the images of two people, we still would not be able to ascertain who is taller. We may know the Greek alphabet and visualize it and still fail to discern which letter, gamma or delta, comes first.

To someone familiar with the sizes of ordinary objects the sentence, “a bacteria is larger than an elephant”, is deemed to be false and the sentence, “an elephant is larger than a bacteria”, is recognized as true. These familiar responses underline a subtle notion: the computations of relations of order are in many cases related to operations involving logical variables (“true” or “false”). Relations of order can be seen as connected to abstract propositions P and Q , usually involving the logical implication $P \Rightarrow Q$. They obey a transitive rule ($A < B$ and $B < C$ imply $A < C$), as understood by an informed person when A , B and C are the sizes of bacteria, dog and elephant, respectively. Analogously, logical implication is also transitive; $P \Rightarrow Q$ and $Q \Rightarrow R$ imply $P \Rightarrow R$. Nevertheless, the classical implication has properties (P and Q both true imply $P \Rightarrow Q$ is true, and P and Q both false imply $P \Rightarrow Q$ is true) that are essentially irrelevant to the type of order relations we are looking for. It is an open question how we comprehend the cognitive function of logical implication during formal reasoning (Mizraji and Lin 2011). Perhaps, understanding this function may help clarify how the brain computes order rankings from concepts directly connected to sensory experiences.

The mapping following a series of visual cues or acoustic signals usually involves abstract operations that are carried out by neural modules distributed over several layers of processing. In this modeling approach we shall assume the simple approximation that spatial size or temporal order rely on the same reference scales, spatial or temporal, of human sensory experience (e.g. we conceptualize objects as small, medium or large).

The brevity of words—prepositions—that are used to express spatial or temporal order suggests the existence of underlying neural modules that disentangle these words and then compute the relations of order in abstract form. It is especially interesting to note that certain prepositions are used indistinctively to characterize spatial or temporal order. “Behind” can be used as the back or rear of something (spatial) whereas in other situations it is used to indicate something happening in the past or being late for certain tasks (temporal). We shall describe the three levels of neural processing that takes a query, requests an ordering sequence of sensory data, and then proceeds to convert this pattern into an abstract entity that is finally resolved in a logical outcome.

Level one

Let us take the example, “Is a dog larger than an elephant?” and represent this question in the compact form, [larger, dog, elephant]. We shall assume that the sensory inputs that codify this query, after several layers of processing, will be converted into a neural triplet $[q, f, f']$, where q is the query, f represents “dog”, and f' represents “elephant”. Without loss of generality we may take the f 's as vectors of equal dimension and orthogonal to each other. The vector q represents the query and has a dimension that reflects the connectivity space of the neural module that codifies the question. All these vectors represent coherent bioelectric activities of extensive groups of neurons. It is appropriate to assume their components to be real and to measure deviations from basal activity (Anderson 1972; Cooper 1974; Kohonen 1972, 1977). Vector components can be positive for excited neurons, negative for inhibited neurons, and zero for neurons insensitive to the stimuli.

On this first level of information processing the column vectors f and f' map on the categories, small (sm), medium (md), or large (lg) in the triplet $[q, f, f']$. We shall take the neural module that processes the query as a trained associative memory endowed with context-dependent inputs–outputs as detailed in section “Context-modulated associative memories”. To simplify our computations we shall assume that all vectors on Level 1 and the outcomes of direct sums are normalized and that interferences among recall memories are negligible. We proceed to outline the components of this module.

(a) Upward order associations

$$H_{sm} = (sm \oplus md[+]) \sum_i (f_i \oplus f'_i[+])^T \quad (21a)$$

$$H_{ml} = (md[-] \oplus lg) \sum_j (f'_j[-] \oplus f''_j)^T \quad (21b)$$

(b) Downward order associations

$$K_{lm} = (lg \oplus md[-]) \sum_k (f_k \oplus f'_k[-])^T \quad (22a)$$

$$K_{ms} = (md[+] \oplus sm) \sum_m (f'_m[+] \oplus f''_m)^T \quad (22b)$$

The summations in (21)–(22) refer to all pairs of objects that can be associated to the categories (sm , md), (md , lg), or their reverse directions. The global structure of the associative memory would then be

$$M = H_{sm} + H_{ml} + K_{lm} + K_{ms} \quad (23)$$

The direct sum $u \oplus v$ in Eqs. (21)–(22) has the following meaning. Let u and v be vectors belonging to orthogonal subspaces,

$$u = [u_1 \dots u_r]^T \quad v = [v_1 \dots v_r]^T \quad (24)$$

The direct sum $u \oplus v$ is a vector in the 2r-dimensional space

$$u \oplus v = [u_1 \dots u_r, v_1 \dots v_r]^T \quad (25)$$

Consequently, when u , u' , v and v' have the same dimension, the scalar product satisfies

$$\langle u \oplus v, u' \oplus v' \rangle = \langle u, u' \rangle + \langle v, v' \rangle \quad (26)$$

The qualifiers $[+]$ and $[-]$ attached to vectors that codify intermediate patterns offer a subtle bridge to help us establish transitive relations. In neural computations of transitive relations $A < B$ and $B < C$, therefore $A < C$, each pattern is represented by a vector, but the pattern B does not appear in the final outcome. How does B disappear? In the theory of matrix memories, there is a natural answer: by generating orthogonal patterns with scalar products equal to zero. When a specific object is categorized as a concept, the concept occupies a neural subspace that includes many independent vectors—slightly different shades of red books, distinguishable small-sized books, etc. It is then possible to find a base where the vectors in this subspace are orthogonal, e.g. sm , $md[-]$, $md[+]$ and lg are all orthogonal to each other as well as the vectors f , f' and f'' . In processes comparing pairs of patterns as in Eqs. (21)–(22), we shall assume that intermediate vectors are orthogonal even though they represent the same pattern. We assume that a memory that has experienced several patterns, including bridge patterns, in the end will deform the bridge pattern augmenting its size when it is associated to a smaller pattern, and reducing its size when it is associated to a larger pattern. This is the meaning of the pairs $(sm \oplus md[+])$ and $(md[-] \oplus lg)$ in Eqs. (21)–(22). In Fig. 3 we display this pairing as a one-dimensional diagram.

Is there any empirical basis for the introduction of deformations in the bridge patterns? Indirect evidence can be

Fig. 3 Pictorial representation of bridge patterns and their deformations

found in perceptual deformations. The Titchener effect, a famous optical illusion, is shown in Fig. 4.

The query “Is f larger than f' ?” is mapped onto a vector q_a . Similarly, the query “Is f smaller than f' ?” is mapped onto a vector q_b . As is usual in the construction of categories (Anderson 1995; Kohonen 1977, 1988) we shall assume that clearly distinct queries are mapped onto orthogonal vectors. If the query acts as a context in both input and output memories (21)–(22), the module L_1 that processes data on Level 1 takes the form,

$$L_1 = q_a q_a^T \otimes M + q_b q_b^T \otimes M \quad (27)$$

In the earlier question, “Is a dog larger than an elephant?” we end up with the mappings $\text{dog} \rightarrow f[-]$, $\text{elephant} \rightarrow f'$, and $\text{query} \rightarrow q_a$. Module L_1 processes this query as

$$(q_a q_a^T \otimes M + q_b q_b^T \otimes M) [q_a \otimes (f[-] \oplus f')] = q_a \otimes (md[-] \oplus lg) \quad (28)$$

The successful execution of this operation relies on adequate experiential information stored in the memory, in this case, the comparison of $(\text{dog} \oplus \text{elephant})$ with $(\text{medium} \oplus \text{large})$. At this level, the module is still unable to give a positive or negative answer to the query.

Level two

The outputs of Level 1 are mapped onto pairs of variables on Level 2, vectors like (b, a) or (before, after) that “conceptualize” abstract order relations. Both column vectors are of the same dimension, orthogonal to each other, which define patterns associated to sm , md and lg , in accordance to their ordering position on Level 1. For pairs increasing in size we have the matrix memory

$$G = (b \otimes a) [(sm \oplus md[+])^T + (md[-] \oplus lg)^T] \quad (29)$$

Perceptual Titchener Effect

Fig. 4 The Titchener illusion, brought about by neighboring objects surrounding the *central circle* even though the *central circle* has the same size in both figures

and for pairs decreasing in size,

$$R = (a \otimes b)[(lg \oplus md[-])^T + (md[+] \oplus sm)^T] \quad (30)$$

The sum of the two previous matrices gives rise to a Global Linkage (*GL*) memory that sorts ordered relations between abstract patterns *b* and *a*,

$$GL = G + R \quad (31)$$

We represent the relations between *a* and *b* by means of Kronecker products in order to make possible the integration of these memories into a neural theory of logical operations (Mizraji and Lin 2011). As an example, *GL* acting on outputs of Level 1 computes the following results,

$$\begin{aligned} GL(sm \oplus md[+]) &= 2(b \otimes a) \\ GL(md[+] \oplus sm) &= 2(a \otimes b) \end{aligned} \quad (32)$$

Equation (32) shows that *GL* can discriminate between two different situations even though we have the sum of matrices *G* and *R* in Eq. (31). The module *GL* can also “infer” correctly a transitive relation as shown below

$$GL(sm \oplus lg) = 2(b \otimes a) \quad (33)$$

Equation (33) processes an input correctly as can be seen by expanding matrices *G* and *R*,

$$\begin{aligned} GL(sm \oplus lg) &= [\langle sm, sm \rangle + \langle md[+], lg \rangle + \langle md[-], sm \rangle \\ &\quad + \langle lg, lg \rangle](b \otimes a) \\ &\quad + [\langle lg, sm \rangle + \langle md[-], lg \rangle + \langle md[+], sm \rangle \\ &\quad + \langle sm, lg \rangle](a \otimes b) \\ &= (1 + 0 + 0 + 1)(b \otimes a) \\ &\quad + (0 + 0 + 0 + 0)(a \otimes b) \\ &= 2(b \otimes a) \end{aligned} \quad (34)$$

It is interesting to note that memory *GL* is able to compute an output correctly even though the pair (*sm* $\oplus$ *lg*) is not in the data stored in memory *GL*.

The information that enters *GL* from the previous level is processed and transported to the next level after been contextualized by the queries. In the situation we have described, Level 2 is globally represented by a matrix of the form,

$$L_2 = q_b q_b^T \otimes GL + q_a q_a^T \otimes GL \quad (35)$$

Level three

On this level, the deepest level of information processing, there is a limited repertoire of multiusable memories to which the previous lower level memories can converge. Empirical motivation for the existence of this level, are the many uses and polymorphic variations of prepositions that establish a sense of order (e.g. behind, before, after, on, or

under, etc.). On Level 3 we use two modules. They connect the inputs from Level 2 to a final binary answer in response to the initial Level 1 query. Let us use the matrices

$$A = s(a \otimes b)^T + n(b \otimes a)^T \quad (36a)$$

$$B = n(a \otimes b)^T + s(b \otimes a)^T \quad (36b)$$

Vectors *s* and *n* form an orthonormal set, with *s* associated to the logical answer “Yes” and with *n* to the logical answer “No”. Matrix *A* produces a final answer “Yes” to the query “Is *X* after *Y*?” and matrix *B* returns an answer “Yes” to the reverse question, “Is *X* before *Y*?” In the example we are analysing, a Level 3 matrix memory can be constructed from *A* and *B*,

$$L_3 = q_a^T \otimes A + q_b^T \otimes B \quad (37)$$

Bird’s-eye view of the multilevel model

Let us see how the three modules, *L*₁ to *L*₃, process a query such as, “Is a dog larger than an elephant?” On Level 1, the module processes the triplet of vectors, (*q_a*, *f*, *f'*) or (*q_b*, *f*, *f'*), and generates the following output

$$\begin{aligned} L_1[q_a \otimes (f[-] \oplus f')] &= [(q_b q_b^T + q_a q_a^T) \otimes M][q_a \otimes (f[-] \oplus f')] \\ &= q_a \otimes (md[-] \oplus lg) \end{aligned} \quad (38)$$

On Level 2, the processing of the previous outcome yields

$$\begin{aligned} L_2[q_a \otimes (md[-] \oplus lg)] &= [(q_b q_b^T + q_a q_a^T) \otimes GL][q_a \otimes (md[-] \oplus lg)] \\ &= q_a \otimes 2(b \otimes a) \end{aligned} \quad (39)$$

Finally, on Level 3 we end up with an answer “No” to the initial query,

$$\begin{aligned} L_3[q_a \otimes (b \otimes a)] &= (q_a^T \otimes A + q_b^T \otimes B)[q_a \otimes 2(b \otimes a)] \\ &= 2A(b \otimes a) = 2n \approx n \end{aligned} \quad (40)$$

In the last step of Eq. (40) the final vector has been renormalized. It must be emphasized that these modules have been built to address the questions posed in our examples. However, it is possible to extend the number of modules on each level by contextualizing queries into different classes. The small number of final categorized queries would be projections of myriad linguistic varieties of the natural lexicon onto distinct orthogonal subspaces. This is an important topic in data mining (Berry et al. 1995; Berry and Browne 2005). In the final analysis, the anatomical connectivity and the storage capacity of the brain place upper bounds on the total number of simultaneously operating memory modules as well as on the dimensionality of input vectors (Kohonen 1988, p. 228).

We summarize in Fig. 5 the transfer of information and the decisions adopted in each level of our model.

In order to simplify the description of this hierarchical model we have avoided using the neural modules (19) and (20), representing the dynamic propositions “towards” and “from”. Yet, in neural regions that process questions concerning spatial or temporal order these modules would probably operate as naturally as the ones we have used in our example and with similar processing designs. Our model can easily incorporate the matrices (19) and (20) that compute these dynamical prepositions.

To establish order relations among several events (What is the temporal order among the execution of Marie Antoinette, the storming of Bastille, and the declaration of the rights of man and of the citizen?), pairs of events must be compared and then stored in a working memory to be later dynamically evaluated by a recursive conjunction AND (Mizraji and Lin 2011). As an example, let A, B and C, be three events to be classified according to their temporal order of appearance. To the question, “Does X occur before Y?” (X and Y being any pair of events), our model would produce answers like: $(A, B) \mapsto \text{Yes}$, $(A, C) \mapsto \text{Yes}$, $(B, A) \mapsto \text{No}$, $(B, C) \mapsto \text{Yes}$, $(C, A) \mapsto \text{No}$ and $(C, B) \mapsto \text{No}$. A recursive AND can only provide an answer Yes in three cases (in Boolean symbols, $(A, B) \wedge (A, C) \wedge (B, C) \mapsto \text{Yes}$). Dynamic recursion is necessary in this case because the logical operator AND is a binary function. The computation of conjunction among three logical variables u, v, w can be realized on neural modules that compute AND recursively as in AND $[\text{AND}(u, v), w]$. Consequently, given the empirical

data from Level 1, the correct order is detected when the AND memory produces a positive answer Yes.

Discussion

One of the fundamental aims of neuroscience is to comprehend how the brain manages to process signals from the outside world into a coherent informational universe that allows us to survive and prosper. In human cognition, the existence of language adds a new and novel codification strategy, one in which conceptual constructions are mapped onto words (Gayler 2006).

In this work we have tried to model the neural processing of order relations codified in certain prepositions. We have assumed a hierarchical processing of outside information where perceptual imagery is converted into concepts by representing them as high dimensional neural vectors. Therefore, a decision problem that involves certain relations of order (Ex. Is a mouse heavier than an elephant?) must be channeled to specialized content-addressable neural memories (Kohonen 1988) that self-organize by processing the corresponding information to produce an answer. The functional scope of the networks employed in the present model largely exceeds the formal problem of classifying objects according to an ordering sequence. In Valle-Lisboa et al. (2014) there are several examples where the performances of these types of neural modules have been applied to pattern recognitions and language processing.

Fig. 5 An illustration of the partial decisions adopted in each one of the levels of the model when confronted with the question “Is a dog larger than an elephant?”

On the first processing level sensory objects are associated to a small number of categories (in our example, “large”, “medium” and “small”). On the second processing level these objects are translated into multi-usable neural vectors that are connected through abstract relations of order, e.g. vectors b , a and eventually i in Level 2 of Fig. 1.

The model assumes these abstract vectors as sharing common roots to diverse contexts that involve inequalities of some kind—comparisons of unequal masses, of events in time, of positions in space, etc. When the original problem requires a resolution that involve an answer of the type Yes or No, neural processing moves to its most abstract level, Level 3, where neural modules associate the original query with a decision as shown in section “[Level three](#)”.

We previously pointed out that in certain uses of prepositions as in “Bring me the book on the table”; immediate answers can be provided by shortcuts that only involve the intermediate Level 2. On Level 1 the directive and the two objects are mapped onto vectors, book $\rightarrow f$, table $\rightarrow f'$, and directive $\rightarrow q_a$, giving an input $q_a \otimes (f \oplus f')$. The module that processes this input would be $L_1 = q_a q_a^T \otimes M$, with the associative memory $M = a \sum_i (f_i \oplus f'_i)^T$, where a refers to the vector that conceptualizes “above”. The output of $L_1[q_a \otimes (f \oplus f')] = q_a \otimes a$ enters the next processing level as an input to the module $L_2 = q_a^T \otimes I$, which selects the object above the table, $L_2[q_a \otimes a] = a$. The innocuous identity operator I substitutes for the more complex GL linkage operator (31). Here the memory only needs to select the object on the table and does not have to discriminate between objects below and above the table.

A possible observable outcome of this model is the convergence of situations that activate distinct sensory modes toward common brain regions that process all these situations in similar manner. The existence of Level 3 may indicate regions in the brain dedicated to abstract processing and potentially reachable through electroencephalography, functional magnetic resonance imaging or magnetoencephalography. The constant refinement of experimental techniques offers a promising avenue for investigational evaluations and modifications of the scalable model presented here. Christoff (2009) suggests that some particularly important classes of human thought, e.g. reasoning and introspective cognition, are associated with activity in the lateral prefrontal cortex (LPFC). In her review, LPFC is organized in sub-regions arranged according to a complexity gradient, where higher levels of abstraction in thought correspond to more anterior LPFC regions.

Acknowledgments This work was partially supported by the Programa de Desarrollo de las Ciencias Básicas (PEDECIBA) and the

Comisión Sectorial de Investigación Científica (CSIC), Universidad de la República, Uruguay, and a grant from Washington College to J. L. The authors thank Peter beim Graben, Andrés Pomi and Juan C. Valle-Lisboa for stimulating discussions during the preparation of this work.

References

- Amari SI (1977) Neural theory of association and concept formation. *Biol Cybern* 26:175–185
- Anderson JA (1972) A simple neural network generating an interactive memory. *Math Biosci* 14:197–220
- Anderson JA (1995) An introduction to neural networks. MIT Press, Cambridge
- Anderson JA, Rosenfeld E (eds) (1988) Neurocomputing. MIT Press, Cambridge
- Arbib MA (ed) (1995) The handbook of brain theory and neural networks. MIT Press, Cambridge
- Aristotle (350 BC) On interpretation (trans: Edghill EM). Provided by the internet classics archive. <http://classics.mit.edu/Aristotle/interpretation.html>. 28 Oct 2012
- Ashby WR (1956) An introduction to cybernetics. Wiley, New York
- Ashby WR (1960) Design for a brain, 2nd edn. Wiley, New York
- Beim Graben P, Gerth S (2012) Geometric representations for minimalist grammars. *J Logic Lang Inf* 21:393–432
- Beim Graben P, Potthast R (2009) Inverse problems in dynamic cognitive modeling. *Chaos* 19:015103
- Beim Graben P, Potthast R (2014) Universal neural field computation. In: Coombes S, Beim Graben P, Potthast R, Wright JJ (eds) Neural fields: theory and applications. Springer, Berlin, pp 299–318
- Beim Graben P, Pinotsis D, Saddy D, Potthast R (2008a) Language processing with dynamic fields. *Cogn Neurodyn* 2:79–88
- Beim Graben P, Gerth S, Vasishth S (2008b) Towards dynamical system models of language-related brain potentials. *Cogn Neurodyn* 2:229–255
- Berry MW, Browne M (2005) Understanding search engines: mathematical modeling and text retrieval, 2nd edn. SIAM, Philadelphia
- Berry MW, Dumais ST, O'Brien GW (1995) Using linear algebra for intelligent information retrieval. *SIAM Rev* 37:573–595
- Besnard P, Fanselow G, Schaub T (2003) Optimality theory as a family of cumulative logics. *J Logic Lang Inf* 12:153–182
- Bezrukov SM, Kish LB (2009) Deterministic multivalued logic scheme for information processing and routing in the brain. *Phys Lett A* 373:2338–2342
- Blutner R (2004) Nonmonotonic inferences and neural networks. *Synthese* 142:143–174
- Christoff K (2009) Human thought and the lateral prefrontal cortex. In: Kraft E et al (eds) Neural correlates of thinking. Springer, Berlin, pp 219–252
- Cooper LN (1974) A possible organization of animal memory and learning. In: Proceedings of the Nobel symposium on collective properties of physical systems, Aspen, Sweden
- Davis PJ, Anderson JA (1979) Nonanalytic aspects of mathematics and their implication for research and education. *SIAM Rev* 21:112–127
- Dehaene S, Changeux JP (1993) Development of elementary numerical abilities: a neuronal model. *J Cogn Neurosci* 5:390–407
- Dehaene S, Cohen L, Changeux JP (1998) Neuronal network models of acalculia and prefrontal deficits. In: Parks RW, Levine DS, Long DL (eds) Fundamentals of neural network modeling. The MIT Press, Cambridge, pp 233–255

- Eliasmith C, Stewart TC, Choo X, Bekolay T, DeWolf T, Tang Y, Rasmussen D (2012) A large-scale model of the functioning brain. *Science* 338:1202–1205
- Erlhagen W, Schöner G (2012) Dynamic field theory of movement preparation. *Psychol Rev* 109:545–572
- Gayler RW (2006) Vector symbolic architectures are a viable alternative for Jackendoff's challenges. *Behav Brain Sci* 29:78–79
- Graham A (1981) Kronecker products and matrix calculus with applications. Ellis Horwood, Chichester
- Humphreys MS, Bain JD, Pike R (1989) Different ways to cue a coherent memory system: a theory for episodic, semantic, and procedural tasks. *Psychol Rev* 96:208–233
- Koch C, Poggio T (1992) Multiplying with synapses and neurons. In: McKenna T, Davis J, Zornetzer SF (eds) *Single neuron computation*. Academic Press, San Diego, pp 315–345
- Kohonen T (1972) Correlation matrix memories. *IEEE Trans Comput* C-21:353–359
- Kohonen T (1977) *Associative memory: a system-theoretical approach*. Springer, New York
- Kohonen T (1988) *Self-Organization and associative memory*, 2nd edn. Springer, Berlin
- Lipinski J, Spencer JP, Samuelson LK, Schöner G (2006) SPAM-ling: a dynamical model of spatial working memory and spatial language. In: *Proceedings of the twenty-eighth annual conference of the cognitive science society* pp 489–494
- Lipinski J, Sandamirskaya Y, Schöner G (2009) Swing it to the left, swing it to the right: enacting flexible spatial language using a neurodynamic framework. *Cogn Neurodyn* 3:373–400
- Mel BW (1992) NMDA-based pattern discrimination in a modeled cortical neuron. *Neural Comput* 4:502–517
- Mizraji E (1989) Context-dependent associations in linear distributed memories. *Bull Math Biol* 51:195–205
- Mizraji E (1992) Vector logics: the matrix-vector representation of logical calculus. *Fuzzy Sets Syst* 50:179–185
- Mizraji E (2008a) Vector logic: a natural algebraic representation of the fundamental logical gates. *J Logic Comput* 18:97–121
- Mizraji E (2008b) Neural memories and search engines. *Int J Gen Syst* 37:715–732
- Mizraji E, Lin J (1997) A dynamical approach to logical decisions. *Complexity* 2:56–63
- Mizraji E, Lin J (2002) The dynamics of logical decisions. *Phys D* 168–169:386–396
- Mizraji E, Lin J (2011) Logic in a dynamic brain. *Bull Math Biol* 73:373–397
- Mizraji E, Pomi A, Alvarez F (1994) Multiplicative contexts in associative memories. *Biosystems* 32:145–161
- Mizraji E, Pomi A, Valle-Lisboa JC (2009) Dynamic searching in the brain. *Cogn Neurodyn* 3:401–414
- Øhrstrøm P, Hasle PFV (1995) *Temporal logic: from ancient ideas to artificial intelligence*. Springer, Dordrecht
- Pao YH (1989) *Adaptive pattern recognition and neural networks*. Addison-Wesley, Reading, MA
- Pike R (1984) Comparison of convolution and matrix distributed memory systems for associative recall and recognition. *Psychol Rev* 91:281–294
- Prince A, Smolensky P (1994) Optimality: from Neural Networks to Universal Grammar. *Science* 275:1604–1610
- Prior AN (1967) *Past, present and future*. Oxford University Press, London
- Prior AN (1968) *Papers on past and tense*. Oxford University Press, London
- Rescher N, Urquhart A (1971) *Temporal logic*. Springer, New York
- Rumelhart DE, Hinton GE, McClelland JL (1986a) A general framework for parallel distributing processing. In: Rumelhart DE, McClelland JL (eds) *parallel distributing processing*. MIT Press, Cambridge
- Rumelhart DE, Hinton GE, Williams RJ (1986b) Learning representations by back-propagating errors. *Nature* 323:533–536
- Smolensky P (1990) Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artif Intell* 46:159–216
- Smolensky P (2006) Harmony in linguistic cognition. *Cogn Sci* 30:779–801
- Smolensky P, Legendre G (2006) *The harmonic mind. From neural computation to optimality-theoretic grammar*. MIT Press, Cambridge
- Szelag E, Dreszer J, Lewandowska M, Szymaszek A (2009) Neural representation of time and timing processes. In: Kraft E et al (eds) *Neural correlates of thinking*. Springer, Berlin, pp 187–199
- Ursino M, Cuppini C, Magosso E (2011) An integrated neural model of semantic memory, lexical and category formation, based on a distributed representation. *Cogn Neurodyn* 5:183–207
- Valle-Lisboa JC, Pomi A, Cabana A, Elvevåg B, Mizraji E (2014) A modular approach to language production: models and facts. *Cortex* 55:61–76
- Wang L, Li X, Yang Y (2014) A review on the cognitive function of information during language comprehension. *Cogn Neurodyn* 8:353–361