

Template Construction Grammar: a Schema-Theoretic Computational Construction Grammar

Victor Barrès

University of Southern California, barres@usc.edu

Abstract

Template Construction Grammar (TCG) was developed as part of an effort to build a system-level neuro-computational model simulating the dynamics at play during language-vision interactions. This paper presents TCG as a computational framework designed to model the human brain's capacity to dynamically coordinate two concurrent incremental processes, one generating a message and the other organizing its mapping onto a linguistic form. It highlights how Schema Theory provides guidelines to implement cognitive-level hybrid computational models that operate in the style of the brain.

Introduction

Template Construction Grammar (TCG) is a novel implemented computational construction grammar framework. It is part of a more general effort to develop a neurolinguistic model of vision-language interactions and follows the tenets of Schema Theory as a cognitive-level brain modeling philosophy (Arbib 1989).

The nature of the cognitive processes at play during the generation of visual scene descriptions has been investigated by psycholinguistic experiments based on the Visual World Paradigm (VWP): the subject is asked to verbally describe a visual scene while her eye-movements and utterances are recorded. The relations between those two temporal sequences (saccades and words) reveal complex dynamic interactions between three cognitive systems: visual, semantic, and grammatical, each having its own internal dynamic behavior (Knoeferle and Crocker 2008). As visual information is actively gathered through attentional exploration of the scene, this information is readily used to update the semantic representation (message) to be linguistically conveyed in a description. The state of the grammatical processes, mapping meaning onto verbal form, are constantly updated to adapt to changes in the semantic state.

Schema Theory offers a top-down counterpart to the bottom-up neural network modeling approach. It focuses on the adaptive and dynamic nature of the interactions between distributed computational units, respecting the computational style of the brain.

Template Construction Grammar (TCG) provides a new chapter in the application of schema theory to language (Arbib, Conklin, and Hill 1987). As a construction grammar (CxG), it assumes that constructions, defined as a meaning-form mappings, are the only units of grammatical knowledge. Building on the VISIONS-Schema System model of visual scene interpretation (Draper et al. 1988), TCG plays a central role in the Schema-Architecture Language-Vision InterAction model (SALVIA) where it operationalizes the grammatical processes at work in the dynamic and adaptive translation of visually extracted, incrementally built semantic representations into online utterance production (Barrès, Lee, and Arbib in preparation).

TCG investigates the modeling space defined by the challenges posed both by the VWP results and by brain theory. Its goal is not offer a CxG analysis of a particular language. It insists on the need to develop models that accommodate the requirements of distributed computation. Anchoring the model in vision-language interactions offers a fruitful terrain to explore those computational issues. TCG however is not a priori limited to generating visual descriptions.

System-level view

A schema-theoretic model of language production TCG supports the online grammatical processes ensuring the flexible coordination of an incrementally built message and the ongoing production of utterances that reflect whole or part of the current semantic content to be conveyed. Fig. 1 presents the integration of this process within the Language Production sub-system of the Schema Architecture Language-Vision InterAction model (SALVIA). In what follows, I will describe the TCG processes as part of the language production system of SALVIA. The suffix '(P)' in Fig. 1 indicates that the systems are linked to language production. The TCG framework does not assume a priori a symmetry of processes between production and comprehension (see (Barrès and Lee 2013) for a discussion of the comprehension processes).

The model takes as INPUT the specification of a temporally unfolding message content which incrementally updates the semantic representation hosted by the semantic working memory system (SemanticWM). The grammatical working memory system (GrammaticalWM) builds on top of the semantic representation by applying the appropri-

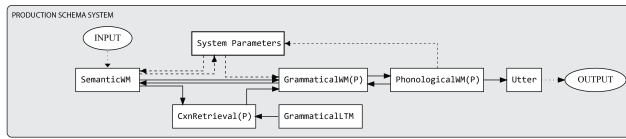


Figure 1: Language production sub-system of the Schema Architecture Language-Vision InterAction model (SALVIA). Each box corresponds to a system with arrows indicating message passing. The core of the system lies in the articulation and temporal coordination of the two main working memory systems: SemanticWM (message) and GrammaticalWM (grammatical processing). (See main text for details)

ate constructions to build a mapping from meaning to form (Those constructions are retrieved by the CxnRetrieval system from a grammatical knowledge stored in the grammatical long term memory (LTM) system (GrammaticalLTM)). Those two working memory systems host time dependent states and processes. The main challenge for TCG is to dynamically and adaptively handle their interactions. The phonological working memory system (PhonologicalWM) simply hosts the current state of word sequences that have already been chosen as the basis for an utterance. Those are posted as OUTPUT. The remainder of the paper details those systems and processes as they relate to TCG.

Schema Theory and Cooperative Computation At a cognitive level, schemas represent portions of knowledge (declarative or procedural). They are organized into schema networks that form the state of long term memory systems (LTMs), each defining a type of knowledge over a given domain. A LTM is always linked to a Working Memory (WM) in which the knowledge it stores is put to use. Once a schema is deemed relevant to the current state of the computation, it is invoked in WM in the form of a schema instance. Each instance represents a hypothesis offering a partial solution to the problem the WM attempts to solve. It carries an activation value that indicates the degree of confidence associated with its hypothesis.

Cooperative computation (C2) fuels WM processes. Instances compete and cooperate, respectively forming inhibitory competition links (comp_link) and excitatory cooperation links (coop_link). At each time the whole set of instances and C2 links (coop_links and comp_links) form a C2 network. The dynamic system it defines governs the temporal trajectories of the instances' activation values. Cooperating instances form assemblages, each corresponding to a potential way to compose instances and generate a solution. Schema Theory prescribes that instances corresponding to hypotheses that support each-other engage in cooperation while those that correspond to contradictory hypotheses compete. The precise process through which instances organize into a C2 network however is specific to each WM sub-system.

Fig. 2 provides an informal example of cooperative computation between schema instances as defined by TCG. It

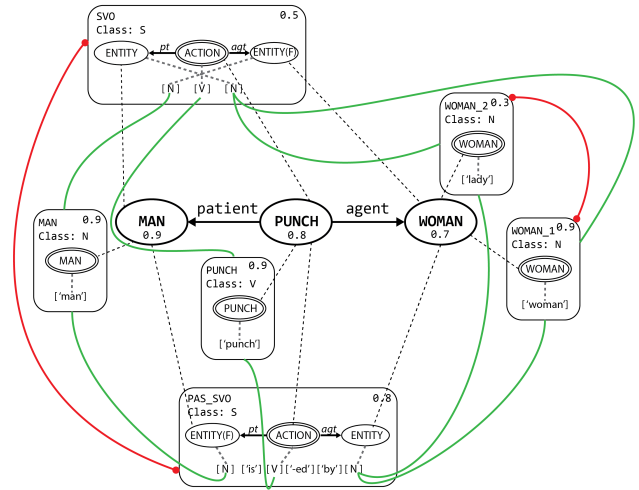


Figure 2: Dynamic coordination between Semantic and Grammatical WM. Semantic WM's state: At the center, concept schema instances form a semantic representation graph (SemRep). Grammatical WM's state: Construction schema instances (boxes) form a C2 network (green cooperation, red competition). The dashed lines linking constructions to SemRep represent the portion of the SemRep for which each construction provides a partial meaning-to-form mapping hypothesis. (Here, activation values are not shown for the SemRep edges (semantic relations).)

illustrates the dynamic coordination taking place between Semantic and Grammatical WM sub-systems (Fig. 1). Here instances are of two types: (1) concept instances forming a semantic representation (shown at the center) that defines the state of the SemanticWM, and (2) construction instances that attempt to dynamically map the semantic content of the message onto a linguistic form by forming a C2 network.

In what follows, I will present in order: the semantic representation format, the TCG constructions, the process by which the construction instances are invoked, and what governs the creation of competition and cooperation links governing the C2 dynamics.

Incremental and dynamic semantic representation (SemRep)

The expressiveness of the semantic representation is limited in order to focus on its time dependent nature as an incremental and dynamic semantic structure: we assume that the message is incrementally built and that this process occurs concurrently with the process of formulating it into utterances.

Conceptual schema instances are invoked in Semantic WM to form a Semantic Representation (SemRep) (Conceptual LTM not shown in Fig. 1). Since all the conceptual relations are binary, the SemRep is conveniently expressed as a labeled (not necessarily connected) directed graph: edges correspond to RELATION, while nodes correspond to EVENT, ENTITY, ACTION, or PROPERTY con-

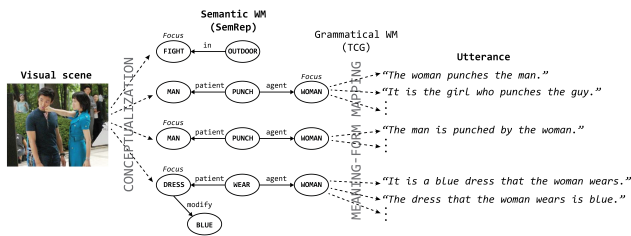


Figure 3: From visual processing to utterances. From left to right. Perceptual representations resulting from visual processing can be conceptualized in many ways into Semantic Representations (SemReps). Conceptualizations can vary in the semantic content they encode (for example in its scope: an outdoor fight vs. a woman wearing a blue dress), but also in terms of what semantic information is highlighted (Focus). As a result of grammatical processing generating meaning-to-form mappings, a given SemRep can yield different linguistic forms: a focus on the agent (WOMAN) can be expressed in the use of an active voice (mild focus) or in a cleft subject (strong focus), MAN can be lexicalized as ‘man’ or ‘guy’. (Note: The utterances shown here are examples of human-generated descriptions. TCG only handles morphology in a very limited way and therefore would not produce the proper morphological markers.)

cept schema instances. No cooperative computation is implemented within Semantic WM (i.e. the semantic message does not contain any conflict).

At each time step, the SemRep can be updated, modifying the content of the message that has to be expressed (Fig. 7). Incrementality takes place both through updating the semantic graph structure and through the activation value dynamics of the conceptual schema instances that compose the SemRep graph.

The goal of grammatical processing using TCG is to generate a flexible grammatical structure articulating the incrementally built SemRep and the production of utterances.

In the context of vision-language interactions and in line with the theories of situated cognition (Pylyshyn 2001), if a SemRep abstracts away much of the perceptual details used by the visual system in the process of attentionally parsing a scene, any part of the SemRep can serve as a ‘deictic pointer’ (Ballard et al. 1997) capable of re-orienting the attentional focus back to the visual region the concept refers to. This enables the system to flexibly use the ‘external world as memory’ (O’Regan 1992): The SemRep can be incrementally updated by requesting more details from the perceptual system through the reorientation of attention towards the region of the external world where this information is most likely to be found. Despite this application to visually extracted semantics, SemRep is not inherently limited to only express the content of visual scenes.

Grammatical Processing

Template Construction Grammar We propose Template Construction Grammar (TCG) as the basis for a Schema

Theory model of grammatical processing. TCG, as a computational construction grammar, builds on the insights of more complex symbolic models (Embodied Construction Grammar and Fluid Construction Grammar) (Steels 2011; Feldman 2010; Bergen and Chang 2005). TCG however significantly reduces the complexity of the semantic and grammatical representations tackled in order to better focus on the use of the constructions as language schemas engaging in C2.

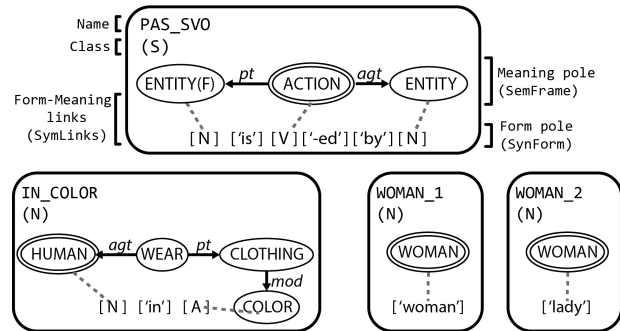


Figure 4: Constructions in Template Construction Grammar: A few examples. (Top) Template features. Constructions range from lexical constructions (WOMAN_1, WOMAN_2) that are fully lexicalized, to partially lexicalized constructions (IN_COLOR), all the way to constructions with little or no form content specification (PAS_SVO). Double circle SemFrame nodes mark head nodes.

Template Construction Grammar defines a *construction* as a tuple

$$(Class, SemFrame, SynForm, SymLink)$$

where:

- *Class* represents the general grammatical category the construction belongs to.
- *SemFrame* (Semantic Frame) represents the meaning pole.
- *SynForm* (Syntactic Form) represents the form pole.
- *SymLink* (Symbolic Links) represents the symbolic linkages between form and meaning elements.

Figure 4 presents a few construction examples illustrating those features. Each construction is assigned a class. If for simplicity the classes used here are similar to the classic syntactic classes, there is no a priori constraint on the number or nature of those classes. Following the main tenets of cognitive linguistics focusing of language in use, linguistic knowledge is not divided into components (phonology, syntax, semantics, and pragmatics), rather any construction can potentially cut across all those strata. For this reason, constructions ran the gamut from lexical constructions (e.g. WOMAN.1, WOMAN.2) all the way to argument structure constructions (e.g. PAS.SVO).

The meaning pole of each construction (SemFrame) is represented using the SemRep format with additional features. A head node indicates the semantic head of a construction. A focus feature F can be associated with a node to encode the information structure features carried by the construction’s meaning pole (cf. PAS_SVO).

The form pole of constructions (SynForm) is limited to representing sequences of form contents and slots. Slots play a key role as variables that need to be filled by the form of another cooperating construction. Slots also express constraints on the constructions that can be used as filler (set of admissible construction classes).

The mapping between meaning and form is defined through symbolic links (SymLinks, dashed lines) linking semantic to form elements, denoting that a specific form element symbolizes a given part of the meaning. In the current format, symbolic links only appear between a node and a form element, any semantic element that is not associated with a symbolic link is assumed to be de facto symbolized by the construction although the nature of this symbolization is not stipulated. Semantic relations (SemFrame edges) are always symbolically represented in the form (e.g. as sequential relations). Similarly some semantic nodes can appear in the meaning pole that are not symbolically linked to any form element (c.f. IN_COLOR construction).

Preference and Group *Preference* and *Group* features can be added to the constructions. *Preference* captures usage preferences (e.g. defined from usage frequency) and during processing modulates the initial activation value of construction schema instances. *Group* defines construction subsets (e.g. lexical and grammatical constructions) that can then be processed differently.

Grammar A grammar \mathcal{G} is a set of constructions $\{Cxn_i\}$. As a construction always includes a SemFrame which is defined in terms of concepts, a construction, and by extension the whole grammar, is necessarily defined in relation to a conceptual knowledge. The model does not impose a particular content for the grammar and offers the option to write and test new grammars using simple json format.

Language schemas A language schema or *construction schema* defines a functional unit of grammatical knowledge. The construction schema is defined as a tuple

$$(Cxn, act^0)$$

where Cxn is a construction as defined above, and $act^0 \in [0, 1]$ is a scalar value used to define the initial activation value when an instance of the schema is invoked.

Grammatical knowledge Although schema theory hypothesizes that long term memory (LTM) should be represented as a schema network, TCG in its current version simply models Language LTM as the set of all construction schemas defined based on the grammar: $GrammaticalLTM = \{(Cxn_i, act_i^0); Cxn_i \in$

$Grammar\}$. Future work will need to follow in the footsteps of Fluid Construction Grammar that has made use of a dynamic priming network to simulate the temporal evolution of the state of grammatical knowledge (Wellens and Steels 2011).

Construction schema instantiation At each time step, the state of the Grammatical WM is defined by the construction schema instances that are currently active as well as by the cooperation and competition links that they have established and that governs the cooperative computation (Fig. 2).

Instantiation is incremental. When new SemRep nodes or edges (i.e. conceptual schema instances) are invoked in Semantic WM, constructions whose SemFrame semantically matches (SemMatch) a SemRep subgraph containing new elements are invoked as instances in Grammatical WM (Fig. 7). A semantic match between a SemRep subgraph and a construction schema SemFrame indicates that the construction expresses in its form, at least in part, the semantic content of this subgraph and is therefore a candidate hypothesis for participating in the mapping of the SemRep onto a linguistic form in Grammatical WM.

In the example shown in Figure 5, the SVO and PAS_SVO construction schema semantically match the SemRep graph. This results in the invocation of an instance of each of those construction schemas in Grammatical WM. As they are invoked, the construction instances also build linkages across WMs (dashed lines).

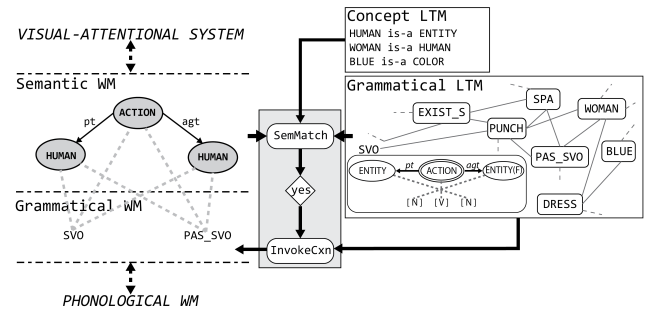


Figure 5: Construction instantiation: SemMatch process. High level view of the construction instance retrieval and invocation process. In this simplified view, the SVO and PAS_SVO constructions’ SemFrames are isomorphic to the SemRep graph (ENTITY is a hypernym of HUMAN). Those isomorphisms are determined by SemMatch which results then in the invocation in GrammaticalWM of an instance of each of those construction schemas, each covering the SemRep graph it is isomorphic to (in this case, the whole graph, dashed grey lines, coverage of edges is omitted for clarity).

The semantic match (SemMatch) requirement functions as a filter. The Conceptual LTM is represented as an input to SemMatch since the conceptual knowledge is necessary for one of the two SemMatch steps. A match between a SemRep subgraph (S) and a SemFrame requires first graph isomorphy, but also matching between the corresponding conceptual content of the nodes and edges of S and SemFrame. This

latter step requires access to conceptual knowledge since, for example, it allows for semantic matching between a concept and its hypernyms, etc.

Using *Group* features, the system can require that certain constructions be invoked first (e.g. lexical construction before argument structure constructions).

In Fig. 1 the construction invocation process is handled by the construction retrieval sub-system (CxnRetrieval(P)).

Construction cooperative computation The goal of the Grammatical WM consists in incrementally building mappings to express the semantic content of the SemRep (itself built incrementally) in a linguistic form. Construction schemas that correspond to relevant meaning-form mapping hypotheses are invoked in Grammatical WM (see above) where they enter in cooperative computation (C2).

Each construction instance carries an activation value, whose initial value is modulated by the preference value stored in the schema, representing the idiosyncratic usage preferences of the speaker (to which can be added a factor reflecting the quality of the semantic match). They organize into a C2 network, whose dynamics defines at each time step the values of the instances activation values. If a construction instances activation value falls below a given threshold, the instance is pruned out of the Grammatical WM. The C2 network is therefore intermittently reshaped following either the invocation of new constructions instances or the pruning of construction instances that ‘lost’ the competitions in which they were involved.

C2 links are built based on the Match operation. Two instances that do not overlap in their coverage of the SemRep do not form any C2 link. Informally, if two instances overlap in their SemRep coverage, the core constraint is that one of the constructions (child) needs to provide a SynForm that can (partially) fill in the missing form information of the other construction (parent).

This process is exemplified in Fig. 6. The bottom example shows two constructions that overlap on the WOMAN SemRep node. However, in PAS_SVO, the SemFrame node that covers WOMAN (ENTITY) is linked to a slot and therefore linguistic information to express the semantic content of WOMAN is missing. WOMAN_1 also covers the WOMAN SemRep node. In addition, it can serve to fill in the slot in PAS_SVO since: WOMAN_1 has a class that matches the class requirement of the slot (N) (SynForm requirement), WOMAN node in the SemFrame of WOMAN_1 is semantically compatible with the ENTITY node in PAS_SVO (SemFrame requirement) to which the slot is symbolically linked, and finally WOMAN in the SemFrame of WOMAN_1 is a HEAD node. Match therefore results in the creation of a cooperation link between the two constructions (green) that link WOMAN_1 to PAS_SVO through the relevant slot for which WOMAN_1 provides the missing form content.

The top-left figure presents a situation largely similar to the previous one with the exception that in this case, both constructions associate the SemFrame node that covers WOMAN with a form content (and not a slot). Here, this represents the case of two synonymous lexical item com-

peting to express a concept¹. In this case Match creates a competition link between the two constructions instances.

The last example, on the top-right, presents the case of two argument structure constructions that overlap on a sub-graph and not only on a single node. This necessarily results in competition since, as we have mentioned above, it is implicit in the formalism of TCG that edges of the SemFrame are symbolically represented in the SynForm (i.e. there is not equivalent of slot’ variables for edges).

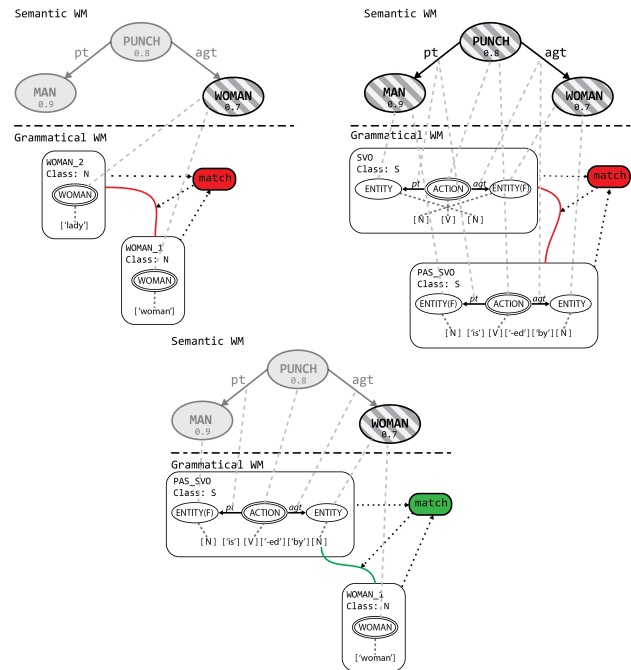


Figure 6: Examples of matching outcomes between construction instances. Highlighted part of the SemRep are covered by both constructions. Dashed lines across WMs indicates the relations between the constructions’ SemFrames and the SemRep. The Match process takes two constructions as input and generates as output either a cooperation link (green) or a competition link (red). The case in which no link is created is not shown (case in which constructions express subgraphs). (See main text for details)

Each construction instance active in Grammatical WM carries a mapping hypothesis of a portion of the current semantic representation onto a linguistic form. Cooperation emerges between two constructions whose mapping can be composed to generate a new mapping covering a larger portion of the semantic content, or refining the mapping. Competition, on the other hand, is triggered when two constructions represent incompatible mapping hypotheses.

C2 links are created incrementally: each time a new construction instance is invoked it is matched against the ones

¹Although it can be claimed that no two constructions are synonymous (principle of no synonymy (Goldberg 1995)), this only holds for an idealized speaker, not at the level of performance of individual speakers that we consider.

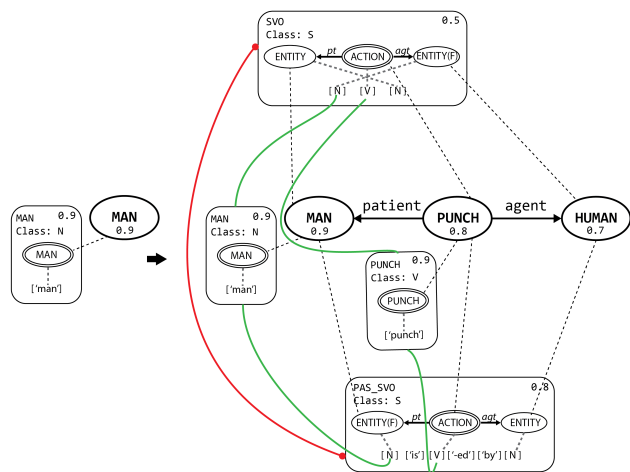


Figure 7: Generating a flexible grammatical structure mapping on top of an incrementally built semantic representation (SemRep). (Left) The SemRep only consists of a MAN node covered by a single MAN construction instance. (Right) Semantic content has been added to the previous SemRep. The MAN construction remains while other construction instances have been invoked, adapting the grammatical structure to the new SemRep state. Fig. 2 presents yet a following state in which the agent HUMAN has been specified as WOMAN.

that are already active in the Grammatical WM (Fig. 7).

Construction Instances Assemblage Through the process of competition and cooperation, construction instances generate construction assemblages, each representing a potential (possibly partial) self-organized program to translate the message (SemRep) into a form content.

At each time step, a constructions assemblage A can be defined as a set of cooperating construction instances $A = (Insts, Coop_Links, act)$. The hypothetical meaning-to-form mapping it represents is associated with its own activation value act derived from that of the assemblage component instances and that reflects its relevance as a meaning-form mapping solution.

Looking back at Fig. 2 it appears that lexical constructions WOMAN_1 and WOMAN_2 compete as synonymous lexical constructions, with WOMAN_1 winning. At the more abstract level of argument structure/voice: PAS_SVO and SVO compete as they both build on top of the same portion of the SemRep but express the agent-patient semantic roles in different ways in their SynForm. PAS_SVO is winning due its patient focus that is a better semantic match for the high activation patient MAN. Assuming that WOMAN_2 loses the competition and is pruned out, we are left with two construction instance assemblages, corresponding respectively to the use of active and passive voice. If forced to choose, the system employs a winner-take-all strategy and the passive would win since PAS_SVO has a higher activation value than SVO, yielding an assemblage with a higher activation.

Generating form When the system is required to generate an utterance, the winner assemblage is selected, the construction instances are unified, and the form of the resulting meaning-form mapping is sent to the Phonological WM as the basis for generating the utterance. The Phonological WM plays an important role as the system might be required, in order to continue the incrementally production of utterances, to take into account the form content of previous utterances.

Good enough production of utterances: Speaker and Task relevant parameters

Much work on language comprehension has by now outlined the necessity to understand the comprehension process as solving a satisficing problem: finding an interpretation for an utterance that is good-enough for communication to succeed while satisfying the constraints defined by the current task as well as by the system itself. To this ‘good-enough comprehension’ principle (Ferreira and Patson 2007) the TCG framework proposes that should be added a ‘good-enough production’ principle: the output of the language production system corresponds to a good-enough solution to a given task. Whether or not fluency and well-formedness are the overarching constraints depends on the task at hand. TCG within a language production model (Fig. 1) accounts for the fact that the processes can function at various regimes and can be impacted by task-related requirements.

Assemblage score Alongside its activation value, an assemblage is assigned a score $score$. The score of an assemblage is introduced to account for modulations of the qualities of the meaning-form mapping desired that can translate in difference in utterance production style. For each assemblage four criteria are taken in consideration when computing the score: the assemblage activation value (v_{act}), amount of semantic information covered (v_{sem}), length of the associated form (v_{form}), utterance continuity value (v_{cont}). v_{sem} is function of the size of the SemRep graph covered by the assemblage. v_{form} reflects the length of the utterance generated by the assemblage. v_{cont} accounts for how much the form associated with the assemblage smoothly overlaps and continue an already produced utterance.

System preferences parameters The main parameters of the system are those that define the dynamics of each WM (in particular their relative characteristic times). In addition to those, preference parameters are defined.

To simulate the impact of time pressure on utterance production, $t_{time_pressure}$ constrains the model to attempt the production of an utterance at each $\Delta T = t_{time_pressure}$ intervals. Crucially, the system has to do so whether or not all the required semantic information has been gathered, and also whether or not the state of the GrammaticalWM has converged to a unique solution (no more competition).

Four style parameters define the weights associated with each of the assemblage scoring criteria

$\vec{w}_{style} = (w_{act}, w_{sem}, w_{form}, w_{cont})$, with the constraint that $|\vec{w}_{style}| = 1$.²

The score of an assemblage is defined as:

$$score = \vec{w}_{style} \cdot (v_{act}, v_{sem}, 1 - v_{form}, v_{cont})^T \in [0, 1] \quad (1)$$

Varying the value \vec{w}_{style} associated the grammatical working memory results in changes in the style of utterance produce. For example, v_{form} appears as $1 - v_{form}$ in the scoring equation so that a higher w_{form} style parameter value pushes the system towards generating shorter, more semantically compact, utterances.

Output Utterances generated by the model are defined as time stamped sequences of words (and occasionally bound morphemes). The interaction of the system parameters and time pressure (and task parameters in general) impacts the dynamics of the language processes yielding qualitatively different types of utterances ranging from well-formed sentences efficiently packaging the semantic information to short disfluent utterances with little grammatical complexity.

The focus of TCG on online incremental processing enables the exploration of the impact of the dynamics of constructional processes on the quality of utterance production.

Simulation example

The model received as input the same simple succession of semantic states as the one used in the conceptual examples above. First, no time pressure is applied.

The model outputs: *[START](602)man is punch -ed by girl[END]* (time of utterance is indicated in parenthesis).

The temporal profiles of the construction instances' activation values are shown in Figure 8.

The dashed red line indicates the initial activation values of instantiated constructions (here there is no modulation of initial activation so all construction instances start with the same activation value).

The MAN lexical construction is the first to be invoked in GrammaticalWM. Its activity builds up, driven by the activation it receives from the SemRep subgraph it maps onto.

As the information about the action event is received, the PUNCH lexical construction gets activated while competition starts between the SVO and PAS_SVO construction instance.

Just before $t=200.0$, PAS_SVO emerges as a winner.

When the semantic information about the woman agent is received, the two synonymous WOMAN lexical constructions are invoked and enter in competition. Meanwhile, as cooperation builds up between the lexical constructions and the PAS_SVO construction, the latter gets an extra boost of activation and emerges as the structure that organizes the grammatical mapping.

²In the current model this parameter is taken to be time independent, but further developments should investigate the possibility to define adaptive scoring policies based on the varying requirements of the communicative task.

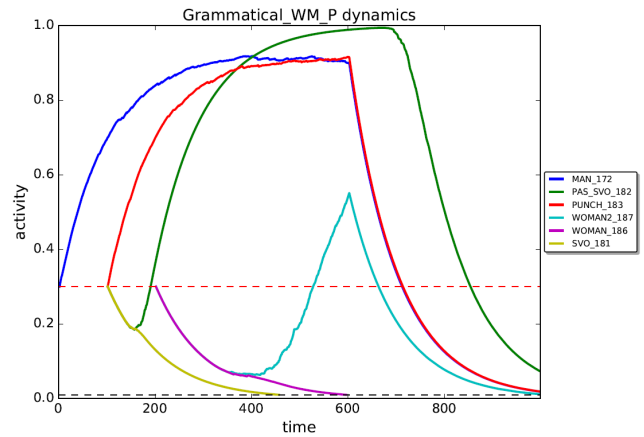


Figure 8: Construction instances' activations (semantic state update rate = 100)

At around $t=400$, the symmetry between WOMAN instances breaks.

The bottom dashed line indicates the value under which instances are pruned out of GrammaticalWM. At around $t=600$, both SVO and the loser WOMAN construction have been pruned. There is no more competition in the network.

A single assemblage remains. It is used to map the full SemRep onto the output utterance mentioned above.

Following this step, the construction instances stop receiving activation from the SemRep instances that have been expressed (all of them in the present case) and therefore their activities start to decay; they will all eventually be pruned.

However, if new semantic information were provided before the pruning occurs, i.e. during a time window proportional to the time characteristics of the GrammaticalWM, the old grammatical structures would still be available to cooperate with the new structures, influencing the continuity between utterances.

To illustrate this point, the model was then run with *time.pressure* set at 200 (forcing the system to attempt to produce an utterance every 200 steps). It outputs: *[START](208)man is punch -ed by (402) woman[END]*.

Here the system first produces a partial utterance at $t=208$. PAS_SVO construction wins as it enables the language system to start expressing the semantic content, even though the information about the agent is not yet available. The system then pauses. When the nature of the agent becomes available, the grammatical processes, piloted by the PAS_SVO instance, can smoothly incorporate the newly invoked lexical constructions and generate a single word utterance at $t=400$ that finishes the passive structure.

Conclusion and Future directions

The TCG computational approach to construction grammar places at its heart the challenge to model the human brain's capacity to dynamically coordinate two concurrent incremental processes, one generating a message and the other organizing its mapping onto a linguistic form.

Supporting the grammatical processes of the Schema Architecture Language-Vision InterAction model (SALVIA) (Barrès, Lee, and Arbib in preparationb), TCG is used to simulate key Visual World Paradigm results (Kuchinsky 2009) regarding the nature of the interactions between visual scene attentional parsing and utterance characteristics (Barrès, Lee, and Arbib in preparationa; Lee 2012).

While this paper focuses on language production, TCG was the basis of a conceptual neurolinguistics model of agrammatic language comprehension (Barrès and Lee 2013). A joint effort is underway between TCG and the other computational construction grammar frameworks (Fluid Construction Grammar (Steels 2011), Embodied Construction Grammar (Feldman 2010) and Dynamic Construction Grammar (Hinaut et al. 2015)) to derive a set principles as well as core challenges that will form a common starting point in designing a computational construction grammar for neurolinguistics.

Comparison with other computational frameworks highlights the two main challenges that TCG faces. Scale: How the C2 dynamics scales with the size of both the grammar remains to be studied. In particular, the amount of redundancy in the grammar and therefore the ratio of competition to cooperation in the network can potentially have profound impacts on the system's behavior. Semantic expressiveness: the SemRep format was designed not for its expressiveness but to enable the study of how incrementally built semantic representations can be processed online. It will be necessary to enrich it (it is already being extended to support a form of frame semantics). The challenge is to always do so while preserving the dynamic nature of the operations that build and process the SemRep.

Finally SALVIA and TCG are being expanded into an implemented model of language comprehension that will be made available at <https://victorbarres.github.io/TCG/> alongside all the TCG related work.

Acknowledgements

This material is based in part on work supported by the National Science Foundation under Grant No. 0924674 and Grant No. BCS-1343544 'INSPIRE Track 1: Action, Vision and Language, and their Brain Mechanisms in Evolutionary Relationship' (Michael A. Arbib Principal Investigator). Many thanks to Dr. Michael A. Arbib for his guidance and all his help.

References

- Arbib, M. A.; Conklin, E. J.; and Hill, J. A. C. 1987. *From schema theory to language*. Oxford University Press.
- Arbib, M. A. 1989. *The Metaphorical Brain 2: Neural Networks and Beyond*. New York, NY, USA: John Wiley & Sons, Inc., 2nd edition.
- Ballard, D. H.; Hayhoe, M. M.; Pook, P. K.; and Rao, R. P. 1997. Deictic codes for the embodiment of cognition. *The Behavioral and Brain Sciences* 20(4):723–742; discussion 743–767.
- Barrès, V., and Lee, J. 2013. Template Construction Grammar: from visual scene description to language comprehension and agrammatism. *Neuroinformatics* 1–28.
- Barrès, V.; Lee, J.; and Arbib, M. A. in preparationa. From gaze patterns to utterances: modeling the dynamics of visual scene description.
- Barrès, V.; Lee, J.; and Arbib, M. A. in preparationb. SALVIA: An Implemented Schema-Theoretic Framework for Investigating the Linkage of Vision and Language.
- Bergen, B., and Chang, N. 2005. Embodied construction grammar in simulation-based language understanding. *Construction grammars: Cognitive grounding and theoretical extensions* 147–190.
- Draper, B.; Collins, R.; Brolio, J.; Hanson, A.; and Riseman, E. 1988. *The Schema System*. Technical report, University of Massachusetts, Amherst, MA, USA.
- Feldman, J. 2010. Embodied language, best-fit analysis, and formal compositionality. *Physics of Life Reviews* 7(4):385–410.
- Ferreira, F., and Patson, N. D. 2007. The Good Enough Approach to Language Comprehension. *Language and Linguistics Compass* 1(12):71–83.
- Goldberg, A. E. 1995. *Construction grammar*. Wiley Online Library.
- Hinaut, X.; Lance, F.; Droin, C.; Petit, M.; Pointeau, G.; and Dominey, P. F. 2015. Corticostriatal response selection in sentence production: Insights from neural network simulation with reservoir computing. *Brain and Language* 150:54–68.
- Knoeferle, P., and Crocker, M. W. 2008. The coordinated processing of scene and utterance: evidence from eye tracking. In *Advances in Cognitive Science*.
- Kuchinsky, S. E. 2009. *From Seeing to Saying: Perceiving, Planning, Producing*. ERIC.
- Lee, J. 2012. *Linking eyes to mouth: a schema-based computational model for describing visual scenes*. Ph.D. Dissertation, University of Southern California.
- O'Regan, J. K. 1992. Solving the "real" mysteries of visual perception: The world as an outside memory. *Canadian Journal of Psychology/Revue Canadienne de Psychologie* 46(3):461–488.
- Pylyshyn, Z. W. 2001. Visual indexes, preconceptual objects, and situated vision. *Cognition* 80(12):127–158.
- Steels, L. 2011. *Design patterns in fluid construction grammar*, volume 11. John Benjamins.
- Wellens, P., and Steels, L. 2011. Organizing constructions in networks. *Design Patterns in Fluid Construction Grammar*. John Benjamins, Amsterdam.