The AAAI 2017 Spring Symposium on
Computational Construction Grammar and Natural Language Understanding
Technical Report SS-17-02

Constructional Analysis Using Constrained Spreading Activation in a FrameNet-Based Structured Connectionist Model

Ely Matos, Tiago Torrent, Vânia Almeida,
Adrieli Laviola, Ludmila Lage, Natália Marção, Tatiane Tavares
Federal University of Juiz de Fora – FrameNet Brasil, Rua José Lourenço Kelmer s/nº -
Campus Universitário, 36036-900 – Juiz de Fora, Minas Gerais, Brazil
projeto.framenetbr@ufjf.edu.br

Abstract

This paper presents CARMA (Constructional Analyzer using Relations among Multiple Attribute-Value Matrices), a system for generating interpretations of sentences in Brazilian Portuguese in terms of the frames and constructions defined in the FrameNet Brasil database. The system converts sentences into multiple Attribute-Value Matrices (AVMs), which are related to each other in a network generated on the fly through spreading activation of its nodes. We report on a pilot experiment for differentiating two argument structure constructions in Brazilian Portuguese: the Active Transitive and the Split Object constructions. Both of them share the same syntax – [NP [V NP]] –, but differ in meaning: while the first evokes the Transitive_action frame, in which an Agent performs an action on a Patient, the latter evokes the Undergoing frame, in which an Entity is affected by an Event, and the Part_whole frame, which establishes a relation between a Part and the Whole it belongs to. Current limitations and future developments of CARMA are also discussed.

Introduction

One of the main purposes of Natural Language Understanding (NLU) is to obtain the computational representation of the possible interpretations of a given sentence or text. Usually, such a representation is obtained through parsers, in which the analysis is modular: the text is sequentially processed by a tokenizer, a POS tagger, a syntactic parser and some semantic component. Each module uses the output of the previous component as input and generates its own enriched output.

The kind of text interpretation – or, more precisely, the kind of representation of text interpretation – obtained from a semantic parser depends essentially on the theory behind it. The way lexical and sentential meaning is regarded, as well as how polysemy is addressed by the theory, constitute important aspects of how the interpretation will be construed by the system.

This paper presents CARMA (Constructional Analyzer using Relations among Multiple Attribute-Value Matrices), a system for interpreting sentences based on Frame Semantics (Fillmore 1982) and Construction Grammar (Kay and Fillmore 1999). Developed by FrameNet Brasil, the system uses as analytical categories the frames, constructions and relations among them, as defined in the project's database.

CARMA builds on previous work on Embodied Construction Grammar (ECG) (Bergen and Chang 2005), which led to the development of the ECG Analyzer (Bryant 2008). Similarly to the ECG Analyzer, CARMA aims to provide cognitively plausible computational analyses of sentence meaning. This is to say that CARMA processes sentences incrementally, using a constraint-based structured connectionist network. Spreading activation techniques (Diederich 1990) are used in the network to produce evidence of all possible interpretations for a sentence. The final interpretation is the one with the highest level of activation.

The main purpose of CARMA is to integrate the lexical, grammatical, semantic and ontological import of sentences into one single representation.

Frames, Constructions and Networks

In Frame Semantics (Fillmore 1982), frames can be conceived of as representations of human knowledge in which sets of concepts are structured in a complex system that is usually thought of as a scene, script or schema. Since 1997, the theory of Frame Semantics has been applied to the development of FrameNet (Fillmore et al. 2003), a resource in which lexical items are described relative to the frames they evoke, based on corpus evidence. As an example of how such a description works in FrameNet, consider the Undergoing frame in Figure 1.

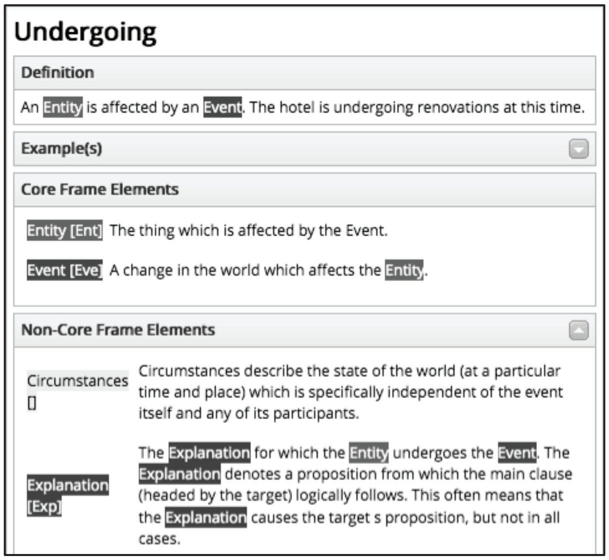

Figure 1: The Undergoing frame

The core frame elements (FEs) indicate which concepts are mandatory for the instantiation of the frame, while the non-core FEs stand for the circumstances in which the situation described by the frame may occur (see Ruppenhofer et al 2010). In the FrameNet lexicon, frames are the semantic import of lemmas, and the pairing of a lemma and a frame is a Lexical Unit (LU). In Brazilian Portuguese, the verb *passar.v 'undergo'* evokes the Undergoing frame in sentences like (1).

(1) Maria passou por uma cirurgia
Maria undergo.PST.3SG by one surgery
Maria went through surgery.

In FrameNet, once a LU is identified in the corpus, the sentence containing it can be semantically and syntactically annotated. The annotation of (1) would yield (2).

(2) [Maria$_{\text{Entity/External/NP}}$] passou$^{\text{TARGET}}$ [por uma cirurgia$_{\text{Event/Dependent/PP}}$]

Sets of annotated sentences inform the system with the valence patterns of the LU being analyzed, i.e. the grammar of the word.

As the work in FrameNet proceeded to include the annotation of full texts, researchers realized that lexical valences were not enough to account for either all grammatical properties of sentences and texts, or all of their semantic aspects (Fillmore 2008). Hence, FrameNet turned to the sister theory of Frame Semantics – Construction Grammar (Kay and Fillmore 1999; Fillmore 2013) – as a means of accounting for the semantic and syntactic properties of texts that go beyond the syntactic locality of LUs, adding a Constructicon to the existing Lexicon. Currently, there are constructicons being developed for English, Japanese, German, Swedish and Brazilian Portuguese, and, although the degree to which they are connected to their respective FrameNets varies, they all rely somehow on frames as the means for describing the semantic import of constructions.

As an example of how constructions (Cxn) are modeled in a Constructicon, take the Split_object Cxn in the Brazilian Portuguese Constructicon – (3). This is a Subject-Predicate argument structure construction whose syntax is identical to that of the Active Transitive Cxn, that is, a subject NP followed by a complement taking VP – [V NP] –, but whose semantics is correspondent to a combination of the Undergoing and Part-Whole frames.

(3) O celular trincou a tela.
The cellphone crack.3SG.PST the screen
The screen in the cellphone cracked.

Note that there is part-whole relation between the object NP and the subject NP in the construction, and that both of them are affected by the event evoked by the verb in the predicate. In FrameNet Brasil, this construction is hence mapped to the Undergoing and Part-Whole frames. Additionally, constraints apply on the Construct Elements (CEs) in the Split_Object Cxn. The Subject CE must be a NP, and is hence licensed by the Determined_NP Cxn, whereas the Predicate is licensed by the Complement-taking_VP Cxn.

Figure 2 shows a graphic representation of this construction in Brazilian Portuguese Constructicon.

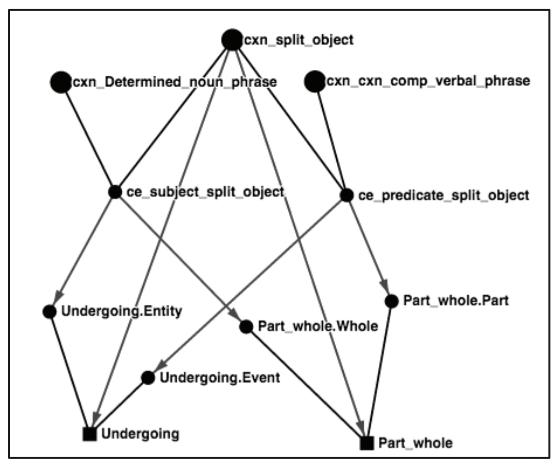

Figure 2: Graphic representation of the Split Object Construction

Both frames and constructions are connected to other frames and constructions via a series of relations: Inheritance relations link frames and constructions to other more general structures of the same kind, while Evoking relations link lexical constructions – LUs – and grammatical constructions to the frames accounting for their meanings. Hence, FrameNet Brasil can be regarded as a network model for Brazilian Portuguese.

Other kinds of knowledge structures have also been added to the model. Qualia relations (Pustejovski 1995), for instance, store the information that some parts – and not others – may be in a part-whole relation with *cellphone*. Because LUs are grouped according to the frames they evoke, if this was the only information stored in the database, there will be no way for the system to decide whether a handle would be a part of a cellphone, for example.

Nevertheless, frames are still the main connections in the FrameNet Brasil database and, once this network is built, one straightforward application of it would be a syntactic-semantic parser. As Fillmore (2012:21) once put it:

> a word [or construction] could evoke a frame, and the semantic parser's job would be to find the elements of that frame in the text, sometimes in the same sentence, in positions determined by the grammar of the word [or construction], and sometimes in neighboring sentences.

In the next sections, we describe our attempt to get the semantic parser's job done.

Network Structure

CARMA is made possible because frames, constructions and ontologies in the network can be represented by feature structures (Jurafsky and Martin 2009), i.e. data structures in which a given feature is associated to one or more values and grouped together with other features. Feature structures are able to represent how several pieces of information relate to each other. Additionally, they allow for the modeling of constraints, which can be applied to specific feature types, using feature system declarations.

Attribute-Value Matrices (AVMs), the same kind of representation used in Berkeley Construction Grammar (Kay and Fillmore 1999), are a means of representing feature structures. Because values can be represented themselves as feature structures, AVMs may contain other – nested – AVMs.

CARMA represents all elements under analysis as AVMs. Figure 3 shows that each node in CARMA can represent a matrix (M), an attribute (A) or a value (V). Hence, a matrix is activated through its attributes, which, in turn, are activated through their values. Values, thus, function as evidence for the inference of attributes and matrices. Specifically concerning the kind of constructional inference carried out by CARMA, the word forms in the sentence are the first values the system identifies. They are responsible for the activation of lexeme attributes, which, in turn, activate a lemma matrix. The lemma is then taken as a value for the lexical unit attribute, which, in turn, activates the frame matrix.

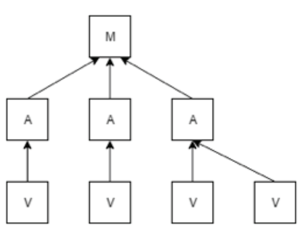

Figure 3: A standard attribute-value matrix in CARMA

Different values may activate one attribute. Moreover, constraints in the activation of the network are defined for each node: for example, a given attribute may require a specific value, or a given matrix may require the activation of all of its attributes to be activated.

Nodes and Links

CARMA combines two types of networks sharing some nodes: the Structural Network, representing the structure – constituency – of frames and constructions, that is, their FEs and CEs, respectively; and the Relational Network, representing the semantic, conceptual and ontological relations between the nodes.[1]

Attribute	Type	Description
id	string	Unique identification of the node in the network.
A	real	Level of activation of a node, computed from the input.
O	real	Output value, calculated at each activation.
Energy	integer [0..10]	Value used for avoiding infinite loops in the activation. Energy value decreases by 1 for each activation cycle. When Energy = 0, the node is no longer activated.
Input	array	Representation of the several input links of the node, optional or mandatory.
Output	array	Representation of the several output links of the node towards other nodes.
Before	array	Constraint on the order of activation of inputs.
Meets	array	Constraint on the order of activation of inputs.
Slots	array	Indication of the word forms responsible for the activation of the node.
Status	string	Status of the node at a given activation cycle. Possible statuses are: *inactive*: the node can be activated; *active*: the node is active, but hasn't fired an output; *fired*: the node has fired an output at least once; *terminal*: the node has no outputs; *blocked*: the node can be activated, but cannot fire outputs; *waiting*: the node is waiting for mandatory inputs to be activated.

Table 1: Attributes of the nodes in CARMA

Correspondingly, each network uses distinct link types. The links for the Structural Network are:

[1] The Relational Network in CARMA is not to be confused with that proposed by Lamb (1998). In the latter case, all information in the network lies in the relations themselves, while nodes only guide the activation process. In CARMA, nodes are structured and represent entities.

– Element_of: associates an Attribute to a Matrix;
– Constraint: associates a Value to an Attribute;
– Inhibitory: implements restrictions on the activation of the network.

As for the Relational Network, link types comprise:

– Is_a: classifies nodes as subtypes of other nodes and is used, for instance, for representing the Inheritance relations in the FrameNet database;
– Evokes: associates a LU or construction to a frame;
– ArgVal: associates an Argument to a Value in a Relation, for instance, it associates the cellphone and the screen in a part-whole relation node.

In turn, each node in the CARMA network is internally structured. Their main attributes are presented in Table 1. Transitions between the statuses listed in Table 1 depend both on the current status of the node and on the satisfaction of some conditions. An *inactive* node may either become *active* – if (C1) every mandatory input is activated, the constraints are respected and the activation value is above the threshold – or stay *waiting* – if (C4) the activation value is above the threshold, but some mandatory input is not activated or some constraint is not respected. Conversely, an *active* node may become *inactive*, if (C2) it runs out of energy, while a *waiting* node may turn into either *inactive*, if (C3) mandatory inputs are active, but constraints are not respected, or *active*, if (C5) all mandatory inputs are activated, the constraints are respected and the activation value is above the threshold.

Once a node is *active*, it can turn into either a *fired* node, if (C6) at least one input from a neighboring node is activated, or a *terminal* node, if (C7) it has no output. Finally, *fired* nodes may be *blocked*, if (C8) it can no longer fire.

Figure 4 summarizes the possible status transitions for any given node.

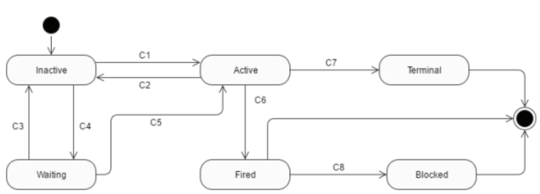

Figure 4: Status transitions of a node

CARMA uses only two types of nodes for representing AVMs and relations. And-nodes, indicated by a triangle, stand for matrices and can only be activated when all mandatory inputs are activated. For instance, constructions are and-nodes that can only be activated if every mandatory CE in it is activated. The same holds for frames and FEs. On the other hand, or-nodes, represented by a circle, stand for attributes. Usually, attributes are associated with different possible values, but only one of them must hold to activate the attribute.

Figure 5 presents an overview of the main structures and relations in CARMA.

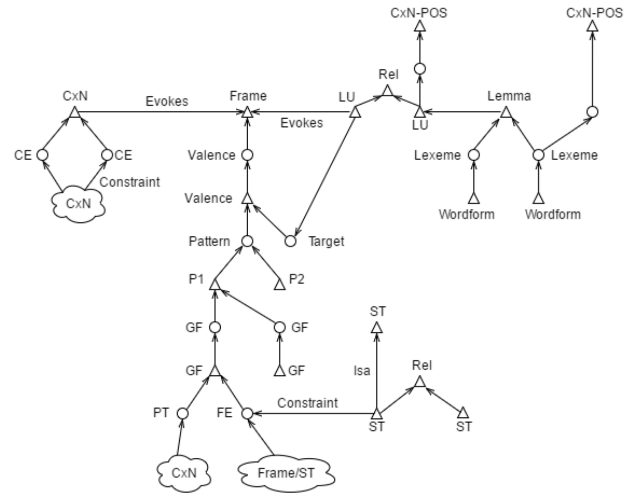

Figure 5: Main components and relations in CARMA

Constructing Sentence Understanding

CARMA applies spreading activation techniques (Diederich 1990) over its combined networks to generate a syntactic and semantic – i.e. constructional – analysis of sentences in Brazilian Portuguese. CARMA is based on a structured connectionist model (SCM) (Feldman and Ballard 1982). SCMs allow for the construction of artificial neural networks comprising higher abstraction levels. In SCMs, nodes can present the kind of structure needed for organizing knowledge in the network. In CARMA, every node has the same internal structure and takes part in the representation of AVMs, as shown in the previous section.[2]

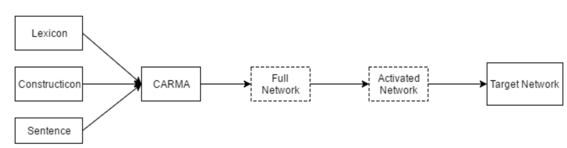

Figure 6: CARMA workflow

To generate the constructional analysis of sentences, CARMA uses as primary inputs the sentence being analyzed and the Lexicon and the Construction in the FrameNet Brasil database. Sentences are processed in three stages, as show in Figure 6: the Full Network, the Activated Network and the Target Network, each of which will be further explained in the following sections.

The Full Network

The Full Network is the first stage in the process and corresponds to the full range of possible structures extracted from the FrameNet Brasil database. Currently, those data are stored in a relational database. To reduce the processing time for each input sentence, FrameNet Brasil data are preprocessed through the compilation of the Full Net-

[2] For a comparison between SCMs and Bayesian Networks, see Feldman and Bailey (2000).

work. Such a compilation has two purposes: formatting the data for the Full Network and resolving abstract constructions.

The first purpose, that of formatting the data, is achieved by filling in two tables in the CARMA database: CARMAnode and CARMAlink. As suggested by their names, the first table processes all frames, constructions and semantic types and stores only the parts of their structures relevant for the network, while the second processes relations and stores only the data relevant for creating the links.

As for the second purpose, the network compilation also resolves abstract constructions. These constructions are the ones in charge of grouping together, in the Constructicon, instantiable constructions that share properties. For example, the general Subject_Predicate Cxn groups together the Active_Transitive, the Split_Object, the Intransitive and the Ergative constructions. Abstract constructions are also important because they store general properties of and constraints on the CEs inherited by their daughter constructions (Kay and Fillmore 1999). However, since they are not directly instantiable as constructs, abstract constructions are not used for building the CARMA networks.

When the system processes a sentence, the Full Network is built based on the two tables just described, and nodes and links specific to the sentence are added. Those nodes include word forms, lexemes, lemmas and LUs, as well as the links between them and the nodes in the compiled network.

The Activated Network

CARMA creates the Activated Network through spreading activation. This network is built from the evidence that a given node in the Full Network must be activated. When that happens, the system analyzes whether there already is some candidate node to be activated in the Activated Network. If not, a new node is created. Hence, the Full Network and the Activated Network are crucially two distinct networks.

The Full Network is formed by the nodes representing the structures of frames, constructions and semantic types. Basically, it is a network of *types* representing conceptual structures. Sentence processing, on the other hand, requires the instantiation of these types as *tokens*.[3] As an example, the Full Network may comprise a node for the Determined_NP Cxn, while one given sentence may require the instantiation of more than one determined noun phrase.

The process of building the Activated Network comprises three stages that the system executes in a loop until a stop condition occurs: node activation, inverted Inheritance computation and spreading activation of the network from the currently active nodes using best-fit binding.

The first nodes activated are those corresponding to the word forms in the sentence. After, the system cyclically identifies the nodes to be activated in the next round, updates their levels of activation, checks whether the nodes can be fired, and updates their statuses. Because the constructions processed so far do not require too many firing cycles, in the current implementation, the threshold is set to a fixed minimum value (0.001) and has no relevant role in the process.

Inverted Inheritance computation refers to the process of applying the Inheritance relations in the FrameNet Brasil database from a bottom-up perspective: word forms in the sentence lead to the activation of the types they are a subtype of all the way up to the frames and constructions. The idea is, hence, that more abstract types are enriched by the properties of the more specific types. Therefore, as the activation spreads in the network, knowledge obtained in the previous rounds also spreads. That is so because Inheritance relations are applied to tokens, not only to types. For instance, if two LUs in the sentence evoke one same frame, each LU will be associated to different tokens of the frame. Inheritance applied to tokens allow for the relations between LUs to propagate through the network and aid in the validation of constraints applied to constructions.

Finally, spreading activation allows for the binding of a given token with the other tokens that must be associated with it, using the Full Network as a map. At each round, each token is classified, i.e. associated to a type. Binding may apply to either new or previously instantiated tokens, and is guided by the Best-Fit principle, according to the constraints imposed to the network. Because tokens chosen for binding are the ones that do not violate the constraints, more than one token can be chosen, as opposed to only the one with the highest level of activation.

The application of the Best-Fit Binding principle throughout the network may lead to the generation of more than one possible interpretation for one given sentence. Because all possible interpretations are associated with a “root” node, the interpretation chosen is the one presenting the root node with the highest activation level. If necessary, inhibitory links may also be used to promote competition between interpretations.

The Best-Fit process currently implemented in CARMA is deterministic as opposed to probabilistic (Bryant 2008). The network structure and the constraints determine which tokens may (or must) be used. A probabilistic approach can be added to CARMA by varying the weight of the links, by using different non-linear transformation functions for each kind of node, or by applying different constraints to specific instances, among other methods. As we plan to implement the analysis of omissible arguments in CARMA, at least some of those methods will be implemented.

Although many stop conditions could be programmed so as to limit the spreading activation process, in the current implementation of CARMA the activation stops when there are no further nodes to be fired. This is a trivial, however suitable, condition in relatively small networks, such as the ones built from simple sentences.

Additionally, it must be stressed that the success of the analysis is directly dependent on how constructions and frames are defined in terms of their constituents, relations

[3] The type-token distinction used in CARMA is similar to the one proposed by Hudson (2007).

and constraints applying to them. Therefore, the constructions and frames modeled have a key role in CARMA.

The Target Network

Even with all the constraints applied to the spreading activation process, which were described in the previous sections, the Activated Network may comprise many nodes that, although active, are irrelevant to the interpretation. For instance, a polysemous lemma may activate several LUs, which, in turn, will activate their respective frames. However, as the analysis develops, many of those frames will no longer be activated and won't find their way up to the construction via inverted Inheritance. A solution commonly implemented for those cases is the progressive reduction of energy in the nodes for each activation round. When a node runs out of energy it is then eliminated.

To keep the process simple, CARMA ignores these nodes during the activation process. When it finishes, an additional stage is implemented for generating the interpretation of the sentence. This stage considers that all constructions relevant to the interpretation are associated to a Root node. Hence, when activation stops, the Activated Network is processed from the Root to the word forms. All constructions activated are run through recursively, considering the active nodes as inputs for the links. This process generates a specific view of the Activated Network, which is called the Target Network, or simply the interpretation. It is important to emphasize that the Target Network is not distinct from the Activated Network, but only a version of it, reduced to nodes relevant to the interpretation.

Construction Disambiguation Task

To test the performance of CARMA as a constructional parser, we designed an experiment in which the system should differentiate constructs licensed by two argument structure constructions in Brazilian Portuguese: the Active_Transitive and the Split_Object constructions, both of which share the same syntax, featuring a Subject NP followed by a verb and its Object NP. The list of sentences used in the test set is shown in Table 2.

The difference between those two constructions lies precisely in their semantics: while the Active_Transitive Cxn evokes the Transitive_action frame, in which an Agent affects a Patient, in the Split_Object Cxn (Sampaio 2010), both the Subject and the Object are affected by the Event encoded by the verb, characterizing the Undergoing frame. Moreover, the Part-Whole frame is also evoked, since the Object NP must be a part of the Subject NP. The Active_Transitive and the Split_Object constructions are exemplified in (4-5), respectively.

(4)	O	menino	quebrou	a	noz.
	The	boy	break.3SG.PST	the	walnut
	The boy cracked the walnut open.				
(5)	O	estudante	quebrou	o	pé.
	The	student	break.3SG.PST	the	foot
	The student broke his foot.				

Note that, while it is possible to say that, in (4), the boy had the intention of cracking the walnut open, the same does not normally hold for (5), since it is highly unlikely that the student would deliberately want to break his own foot. Because the verb head of the predicate, *quebrar 'break'*, is also the same in both constructs, one could not assign to the verb all the responsibility for the difference in meaning. The existence of two argument structure constructions in Brazilian Portuguese with identical syntax – at least as far as most if not all state-of-the-art parsers can tell – and such a semantic difference is precisely the reason why we chose the Split_Object Cxn for testing CARMA.

Verb	**Active_Transitive** / **Split_Object**
1. torcer *'twist'*	1a. A faxineira torceu o pano. *'The housekeeper twisted the cloth'*
	1b. O atleta torceu o tornozelo. *'The athlete twisted his ankle'*
2. arrebentar *'break'*	2a. A menina arrebentou o lacre. *'The girl broke the seal'*
	2b. O menino arrebentou o nariz. *'The boy broke his nose'*
3. machucar *'hurt'*	3a. O cachorro machucou o menino. *'The dog hurt the boy'*
	3b. O jogador machucou o joelho. *'The player hurt his knee'*
4. quebrar *'break'*	4a. O menino quebrou a noz. *'The boy cracked the walnut open'*
	4b. O estudante quebrou o pé. *'The student broke his foot'*
5. furar *'stab'*	5a. A criança furou os balões. *'The child stabbed the balloons'*
	5b. O pedreiro furou o dedo. *'The mason perforated his finger'*
6. trincar *'crack'*	6a. A faxineira trincou o copo. *'The housekeeper cracked the glass'*
	6b. O celular trincou a tela. *'The screen in the cellphone cracked'*
7. queimar *'burn'*	7a. O bandido queimou as provas. *'The criminal burnt the evidence'*
	7b. O computador queimou o HD. *'The HD in the computer burnt'*
8. soltar *'detach'*	8a. A vizinha soltou o cachorro. *'The neighbor released the dog'*
	8b. A panela soltou o cabo. *'The pan lost its handle'*
9. rasgar *'tear'*	9a. A professora rasgou o papel. *'The teacher tore the paper'*
	9b. A calça rasgou o bolso. *'The pocket in my pants ripped off'*
10. descolar *'unglue'*	10a. A criança descolou o adesivo. *'The child removed the sticker'*
	10b. O sapato descolou a sola. *'The shoe lost its sole'*

Table 2: Sentences in the test set

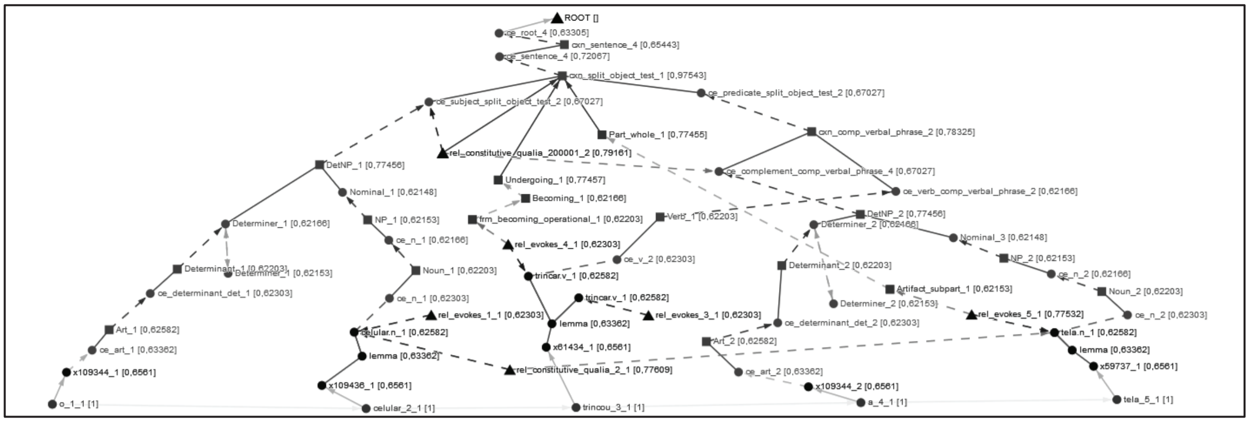

Figure 7: Target Network for O celular trincou a tela 'The screen in the cellphone cracked'

The experiment being reported here follows the same methodology defined by Almeida (2016), but uses an enhanced implementation of CARMA for the construction disambiguation task. A test set comprising twenty sentences in Brazilian Portuguese, all of them composed of five words, was used. Ten of the sentences were licensed by the Active_Transitive Cxn, while the other ten were instances of the Split_Object Cxn. Verbs in the sentence were chosen from a list of ten verbs that allow both causative and inchoative readings. Each verb was instantiated in two sentences: an Active_Transitive one and a Split_Object one.

Sentences in Table 2 were randomly mixed with other 10 Ergative sentences featuring the same verbs and 10 distractors featuring different verbs and argument structures. Native speakers of Brazilian Portuguese were then required to judge the semantics of each sentence, by choosing which alternative described better the general idea of the sentence being judged. Alternatives presented were: (a) somebody or something caused some effect on somebody or something; (b) something happened to somebody or something; (c) none.

All but one sentence in the test set scored above 0.80 for the expected constructional meaning, i.e., more than 80% of the 59 respondents chose (a) for Active_Transitive sentences and (b) for Split_Object sentences, showing that the sentences used in the test set were regarded as instances of the constructions being investigated (Almeida 2016).[4]

After, sentences in the test set were parsed by CARMA. All of the 20 sentences were correctly identified as either instances of the Active_Transitive Cxn (1a-10a) or instances of the Split_Object Cxn (1b-10b). Sentences (1a-10a) only activated the Active_Transitive, with an average activation score of 0.876, while sentences (1b-10b) activated both constructions, although they always presented higher activation scores for the Split_Object Cxn, as shown in Table 3.

The reason why the Activated Network of sentences (1b-10b) includes both constructions is the syntactic similarity of the argument structures. On the other hand, the reason why the Target Network, that is, the interpretation, favors the Split_Object construction is the semantics of the construction. In the FrameNet Brasil database, a constraint on the Split_Object Cxn models the part-whole relation between its object and its subject. To model this relation, LUs in the Artifact_subpart and Observable_body_parts frames, for example, are linked via a Constitutive_of qualia relation to LUs in the frames representing their respective wholes, e.g. Artifact and People_by_vocation. As an example, take sentence 6b, whose Target Network is presented in Figure 7.

Sentence	Active_Transitive	Split_Object
1b	0,892	0,971
2b	0,892	0,966
3b	0,876	0,966
4b	0,876	0,975
5b	0,876	0,966
6b	0,876	0,971
7b	0,876	0,975
8b	0,876	0,971
9b	0,876	0,971
10b	0,876	0,971

Table 3: Activation scores of Split_Object sentences in CARMA

What characterizes 6b as an instance of the Split_Object construction is the fact that the screen is part of the phone. CARMA, as the majority of parsers for Portuguese, is able to classify *o celular* as the Subject NP and *a tela* as the Object NP. However, this classification is not enough for telling this construct apart from the ones licensed by the Active_Transitive Cxn. To do so, CARMA applies inverted Inheritance throughout the network to inform the constructional level with the part-whole relation between the LUs in the sentence. In Figure 7, every node in the chains above *celular* and *tela* – ce_n, Noun, NP, Nominal and DetNP – "inherit" the part-whole relation between *celular* and *tela*. When the activation reaches the Subject and Ob-

[4] Sentence 3a score 0.72, probably because its subject is not human.

ject nodes, the part-whole constraint is met and the Split_object_cxn node is activated.

Limitations and Outlook

CARMA builds on decades of research in Cognitive Linguistics, and implements aspects of Frame Semantics (Fillmore 1982), Construction Grammar (Kay and Fillmore 1999) and Neural Theory of Language (Feldman 2007). It also features contributions from the Generative Lexicon Theory (Pustejovsky 1995), NeuroCognitive Linguistics (Lamb 1998) and Word Grammar (Hudson 2007). The heterogeneity of theories stands for our conception that NLU tasks can benefit from multiple theoretical constructs.

Although CARMA succeeded in the constructional disambiguation task, the analyses so far focused on simple sentences, leaving out key aspects of sentence interpretation, such as: multiple inheritance, null instantiation and contextual ground. Because this experiment was designed as a proof of concept, processing those and other aspects of constructions was not evaluated. Nevertheless, some aspects of CARMA's design already focus on scaling the system up to process longer and more complex text. Those include the use of a simple algorithm, the almost absolute lack of floating point operations, and the use of secondary memory for storing the data.

Inheritance is implemented in CARMA in two ways. First, it is used during the compilation of the network, when abstract constructions are materialized in their daughter constructions. Second, it is used as a means of enriching more general nodes with information from their more specific instances. In neither case, multiple inheritance was considered, although extensive work on Cognitive Linguistics provides evidence of its existence.

Although null instantiations are already accounted for in FrameNet as part of the valence descriptions of LUs, this information is not used in CARMA, nor is any information about the valence patterns of LUs. We aim to implement valence patterns as additional evidence for the inference of constructions.

Finally, as CARMA moves towards the interpretation of full texts, contextual ground will be needed for processing sentences. This is because information extracted from a previous sentence may be presupposed by the following sentence. We also aim to implement contextual ground processing in the next version of CARMA.

Acknowledgements

The FN-Br Constructicon project is funded by CAPES/STINT grant # 88887.125411/2016-00.

References

Almeida, V. G. 2016. Identificação Automática de Construções de Estrutura Argumental. M.A. thesis, Department of Linguistics, Federal University of Juiz de Fora, Juiz de Fora, Brazil.

Bergen, B. K. and Chang, N. 2005. Embodied Construction Grammar in Simulation-Based Language Understanding. In Östman, J-O., and Fried, M. (eds.), *Construction Grammars*, 147-190. Amsterdam: Benjamins.

Bryant, J. E. 2008. Best-Fit Constructional Analysis. Ph.D. diss., Department of Computer Science, University of California, Berkeley, CA.

Diederich, J. 1990. Spreading Activation and Connectionist Models for Natural Language Processing. *Theoretical Linguistics* 16(1):25-64.

Feldman, J. 2006. *From Molecule to Metaphor: A Neural Theory of Language*. Cambridge, MA: MIT Press.

Feldman, J.; Bailey, D. 2000. Layered Hybrid Connectionist Models for Cognitive Science. In Wermter, S., and Sun, R. (eds.), *Hybrid Neural Systems*, 14-27. Berlin: Springer-Verlag.

Feldman, J.; Ballard, D. 1982. Connectionist models and their properties. *Cognitive science*, 254:205–254.

Fillmore, C. J. 1982. Frame Semantics. In *Linguistics in The Morning Calm,* 111-137. Seoul, South Korea: Hanshin.

Fillmore, C. J. 2008. Border Conflicts: FrameNet meets Construction Grammar. In *Proceedings of XIII EURALEX*, 49-68. Barcelona, Spain: IULA.

Fillmore, C. J., Johnson, C. R., and Petruck, M. R. 2003. Background to FrameNet. *International Journal of Lexicography* 16(3):235-250.

Hudson, R. A. 2007. *Language Networks: the new Word Grammar*. New York: Oxford University Press.

Jurafsky, D. and Martin, J. H. 2009. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Upper Saddle River, NJ: Pearson.

Kay, P., and Fillmore, C. J. 1999. Grammatical Constructions and Linguistic Generalizations: The What's X Doing Y Construction. *Language* 75:1-34.

Lamb, S. 1998. *Pathways of the Brain: The Neurocognitive Basis of Language*. Amsterdam: Benjamins.

Pustejovsky, J. 1995. *The Generative Lexicon*. Cambridge, MA: MIT Press.

Ruppenhofer, J., Ellsworth, M., Petruck, M. R., Johnson, C. R., and Scheffczyk, J. 2010. *FrameNet II: Extended theory and practice*. Berkeley, CA: ICSI.

Sampaio, T. F. 2010. A Família de Construções de Argumento Cindido no Português do Brasil. Ph.D. diss., Department of Linguistics, Federal University of Juiz de Fora, Juiz de Fora, Brazil.