Good-enough parsing, Whenever possible interpretation: a constraint-based model of sentence comprehension

Philippe Blache

Good-enough parsing, Whenever possible interpretation: a constraint-based model of sentence comprehension

Philippe Blache
CNRS & Aix-Marseille Université (Laboratoire Parole et Langage)
blache@blri.fr

Natural language processing and psycholinguistics are progressively getting closer, and new language processing architectures, bringing together both computational and cognitive aspects, emerge. We propose in this position paper a brief overview of the basic needs paving the way towards such unified frameworks and show how constraints constitute an adequate answer.

What constraints can do: Let's start with some basic recalls about constraint programming. Constraints are not only efficient for *ruling out* unwanted solutions or ill-formed structures. They are also capable of building *approached solutions* (or ultimately instantiating values) by restricting the *search space*. For example, we can limit the definition domain of an integer variable x by means of interval constraints such as $[x > 1; x < 4]$ (note that adding a new constraint $[x > 2]$ leads to the solution $x = 3$). Moreover, constraints can be of different types (interval, boolean, numerical, etc.), a same variable being possibly involved is many of them. In this sense, constraints form a system that is in itself source of information: a problem can be described with a set of constraints, and this description leads to the solutions (which is the basis of distinction between **declarative** and procedural approaches in computer science [Colmerauer, 1986, Jaffar and Lassez, 1986]). Solving a problem consists in evaluating the constraint system which leads to instantiate values and more generally provide information about the set of variables forming the problem. In other words, the state of the **constraint system** after evaluation, for a given set of input values, constitutes a precise description of this input set. We propose in this short note some arguments in favor of considering this computational framework as an efficient cognitive model for language processing.

Needs and requirements for sentence processing: Language is comprehended by humans in real time. To be more precise, sentence interpretation is done such efficiently *in most of the cases*, which means that different types of mechanisms can be at work according to the input. It is important to note that this property is preserved even when the input is not perfectly-formed (errors, disfluencies, unstructured productions, etc.), which occurs frequently in natural situations (typically during conversations). One question is then to explain how does interpretation works under **noisy** input. A classical solution explains that non-canonical productions are repaired, the difficulty of the interpretation being dependent to the number of repairs [Gibson, 1998]. This **noisy-channel** approach [Levy, 2008a, Levy, 2008b, Johnson and Charniak, 2004] proposes in particular to introduce the notions of **uncertainty** and **inference**. This constitutes a first important requirement:

dealing with *scarce data*. Moreover, in case of ill-formed productions, it is also necessary to analyze the source of the problem. This means the capacity to parse the input with robust methods, also capable of describing precisely the error. In terms of parsing, this consists in finding an **optimal** description, that can also gives account for **violations** [Prince and Smolensky, 1993, Blache, 2016b]. Finally, another important question is that language processing is to be considered to be a **rational** process [Levy, 2008a, Anderson, 1990], taking into account **multiple sources** of information (verbal, gestural, contextual, etc.).

Cognitive aspects: Investigating the cognitive side of language processing also leads to the same type of needs. The most important aspect concerns *memory*: several studies have shown that sentence processing is not done in a strictly incremental manner. In many cases, a global recognition of entire **patterns** is at work. This is typically the case with *idioms*: starting from a recognition point (usually the second or the third word of the idiom), the processing becomes global and the rest of the idiom is not parsed anymore. This is shown both at syntactic and semantic levels: the difficulty generated by the introduction of a violation in the idiom is compensated by the global recognition of the pattern [Vespignani et al., 2010]. Moreover, the semantic processing remains shallow after the recognition point and the semantic content of the words is not even accessed [Rommers et al., 2013]. In the same perspective, [Swets et al., 2008] has shown that when reading with no precise task, the attachment of modifiers is not completely achieved, the dependencies remaining **underspecified**. These observations show that in many cases, a simple **shallow processing** is used without generating any difficulty in the global interpretation of the sentence. A complete, deep and strictly incremental processing remains however necessary when faced with complex sentences as illustrated in [Levy, 2013]: "*Because the girl that the teacher of the class admired didnt call her mother was concerned.*" In this case a deep analysis, resolving all dependencies and clause boundaries is necessary to interpret the sentence.

Another important feature has also been shown by several experiment, reinforcing the idea that sentence processing is not completely done word-by-word, but instead relies on a **delayed evaluation** mechanism. This effect is in particular observable in reading experiments when the presentation rate (the time left between the presentation of each word) is accelerated [Vagharchakian et al., 2012]. After a certain threshold, the intelligibility of the sentence *collapses*. This effect is explained by the fact that in such situations, words are not processed incrementally, but stored into a **buffer** (short-term memory). This delaying mechanism makes it possible to interpret the new words when enough cognitive capacity becomes available. When the maximal capacity of the buffer is reached, the process is blocked.

Processing architectures: Different processing architectures have been proposed integrating several of these features. In particular, the *Good-Enough Theory* [Ferreira and Patson, 2007, Traxler, 2014] integrates the fact that sentence interpretation is only done from time to time, **delaying** the integration until **enough information** becomes available. In this case, a complete interpretation is often delayed (or in some cases never done), and replaced by the identification of partial meanings, starting from which a general interpretation can be approximated. The basic principle consists there in finding "*interpretations over small numbers of adjacent words whenever possible*". This framework has been precised by the integration of two different levels of parsing that can be at work: shallow processing with partial interpretation for the average case and deep processing when faced with difficulties [Blache, 2016a].

This new architecture implements the delaying mechanism, opening the way to pattern or global recognition following the principles of the good-enough parsing, integrating a "*whenever possible*" interpretation.

How constraints implement architectures: Constraints offer an appropriate solution for the implementation of all the different requirements of the proposed architecture.

- *Whenever possible interpretation*: In constraint programming, all constraints are potentially active and assessed when their variables are instantiated. Moreover, all constraints in a system are independent from each others, which means that they can be assessed independently, at any time. As a consequence, constraint satisfaction implements implicitly **delayed evaluation**, until variables get values (or more generally until enough information become available). When no value is assigned to a variable, the constraint is not fully assessed, but makes it possible to restrict the definition domain (the **search space**) and maintain the coherence of the system, leading to **approximated** solutions.
- *Good-enough parsing*: Constraints in general, among which unification, makes it possible to work with underspecified structures. More precisely, all structures can be left partially uninstantiated, implementing directly **underspecification**, and progressively completed when necessary. In other words, the same structure at any level of specification, making it possible to use the same objects in both types of processing, **shallow or deep**.
- *Noisy-channel*: Ill-formed inputs can be parsed thanks to **constraint relaxation**. Constraint **violation**, completed with **weighting** or ranking, offers then the operational framework in the construction of an **optimal** solution. Moreover, the set of violated constraints constitutes a precise description of the source of error.
- *Prediction/activation*: Constraint systems implement **relation networks** within the set of variables, forming **constraint graphs**. The instantiation of a variable makes it possible to **activate** the associated subgraph and their nodes. An activated node correspond to a predictable object and a set of activated nodes (or variables) implements **category prediction**. In other words, when enough information becomes available, on top of describing the constrained structure, it becomes possible to predict new objects.
- *Patterns/constructions*: Following the **declarative** characteristics of constraint programming, a structure can be described by a set of constraints. In language processing, complex objects such as constructions can then correspond to set of constraints. A construction (or a pattern) is then recognize when the **constraint subsystem** is satisfied.
- *No structure*: Finally, and most importantly, the state of the constraint system after evaluation comes to a precise and in-depth description of the linguistic structure. As a consequence, no specific structure has to be built prior to the interpretation. Constraints is then the adequate answer for the implementation of non-modular approaches.

As shown above, the integration of cognitive and computational approaches raises new questions for language processing among which the need to work with partial structures, to interpret objects only when enough information becomes available, to deal with noisy inputs and to implement different level of processing, corresponding to different levels of complexity in the input. These needs form the basis of recent cognitive principles such as "*good-enough parsing*" and "*whenever possible interpretation*". We propose to consider constraints as an adequate and efficient framework for their computational modeling.

References

[Anderson, 1990] Anderson, J. (1990). *The adaptive character of human thought.* Lawrence Erlbaum.

[Blache, 2016a] Blache, P. (2016a). Light-and-deep parsing: a cognitive model of sentence processing. In Poibeau, T. and Villavicencio, A., editors, *Language, Cognition and Computational Models.* Cambridge University Press.

[Blache, 2016b] Blache, P. (2016b). Representing syntax by means of properties: a formal framework for descriptive approaches. *Journal of Language Modelling.*

[Colmerauer, 1986] Colmerauer, A. (1986). Theoretical model of prolog ii. In van Caneghen, M. and Wane, D., editors, *Logic Programming and its Applications.* Ablex Series in Artificial Intelligence.

[Ferreira and Patson, 2007] Ferreira, F. and Patson, N. D. (2007). The 'good enough' approach to language comprehension. *Language and Linguistics Compass*, 1(1).

[Gibson, 1998] Gibson, E. (1998). Linguistic complexity: locality of syntactic dependencies. *Cognition*, 68:1–76.

[Jaffar and Lassez, 1986] Jaffar, J. and Lassez, J.-L. (1986). Constraint logic programming. Technical report, Department of Computer Science, Monash University, Victoria, Australia.

[Johnson and Charniak, 2004] Johnson, M. and Charniak, E. (2004). A tag-based noisy channel model of speech repairs. *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 33–39.

[Levy, 2008a] Levy, R. (2008a). Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

[Levy, 2008b] Levy, R. (2008b). A noisy-channel model of rational human sentence comprehension under uncertain input. *Proceedings of EMNLP*, pages 234–243.

[Levy, 2013] Levy, R. (2013). Memory and surprisal in human sentence comprehension. In van Gompel, R. P. G., editor, *Sentence Processing*, pages 78–114. Psychology Press.

[Prince and Smolensky, 1993] Prince, A. and Smolensky, P. (1993). Optimality theory: Constraint interaction in generative grammars. Technical Report RUCCS TR-2. Rutgers Optimality Archive 537.

[Rommers et al., 2013] Rommers, J., Dijkstra, T., and Bastiaansen, M. (2013). Context-dependent Semantic Processing in the Human Brain: Evidence from Idiom Comprehension. *Journal of Cognitive Neuroscience*, 25(5):762–776.

[Swets et al., 2008] Swets, B., Desmet, T., Clifton, C., and Ferreira, F. (2008). Underspecification of syntactic ambiguities: Evidence from self-paced reading. *Memory and Cognition*, 36(1):201–216.

[Traxler, 2014] Traxler, M. J. (2014). Trends in syntactic parsing: anticipation, bayesian estimation, and good-enough parsing. *Trends in Cognitive Sciences*, 18(11):605–611.

[Vagharchakian et al., 2012] Vagharchakian, L., G., D.-L., Pallier, C., and Dehaene, S. (2012). A temporal bottleneck in the language comprehension network. *Journal of Neuroscience*, 32(26):9089–9102.

[Vespignani et al., 2010] Vespignani, F., Canal, P., Molinaro, N., Fonda, S., and Cacciari, C. (2010). Predictive mechanisms in idiom comprehension. *Journal of Cognitive Neuroscience*, 22(8):1682–1700.