

# Natural Language Processing and the Now-or-Never Bottleneck

Carlos Gómez-Rodríguez

*LyS Research Group, Departamento de Computación  
Facultade de Informática, Universidade da Coruña  
Campus de A Coruña, 15071 A Coruña, Spain.  
E-mail: cgomezr@udc.es*

This is a commentary article on: Morten H. Christiansen and Nick Chater, "The Now-or-Never Bottleneck: A Fundamental Constraint on Language". Behavioral and Brain Sciences, 39:e62, 2016. ISSN: 0140-525X, doi: [10.1017/S0140525X1500031X](https://doi.org/10.1017/S0140525X1500031X).

This is the peer-reviewed version of a contribution accepted for publication in Behavioral and Brain Sciences, by Cambridge University Press. Changes resulting from editorial input may not be reflected in this document. © Cambridge University Press.

The definitive version of this article has been published in Behavioral and Brain Sciences, 39:e74, 2016. ISSN: 0140-525X, doi: [10.1017/S0140525X15000795](https://doi.org/10.1017/S0140525X15000795).

**Abstract:** *Recent research on natural language processing tools, motivated by the need to improve their efficiency to handle web-scale data, has arrived at models that remarkably match the expected features of human language processing under the Now-or-Never bottleneck framework. This provides additional support for said framework and highlights the research potential in the interaction between applied computational linguistics and cognitive science.*

Christiansen & Chater (C&C) describe how the brain's limitations to retain language input (the Now-or-Never bottleneck) constrain and shape human language processing and acquisition.

Interestingly, there is a very strong coincidence between the characteristics of processing and learning under the Now-or-Never bottleneck and recent computational models used in the field of natural language processing (NLP), especially in syntactic parsing. C&C provide some comparison with classic cognitively-inspired models of parsing, noting that they are in contradiction with the constraints of the Now-or-Never bottleneck. However, a close look at the recent NLP and computational linguistics literature (rather than the cognitive science literature) shows a clear trend towards systems and models that fit remarkably well with C&C's framework.

It is worth noting that most NLP research is driven by purely pragmatic, engineering-oriented requirements: the primary goal is not to find models that provide plausible explanations of the properties of language and its processing by humans, but rather to design systems that can parse text and utterances as accurately and efficiently as possible for practical applications like opinion mining, machine translation or information extraction, among others.

In recent years, the need to develop faster parsers that can work on web-scale data has led to much research interest in incremental, data-driven parsers; mainly under the so-called *transition-based* (or shift-reduce) framework (Nivre, 2008). This family of parsers has been implemented in systems like MaltParser (Nivre et al., 2007), ZPar (Zhang & Clark, 2011), ClearParser (Choi & McCallum, 2013) or Stanford CoreNLP (Chen & Manning, 2014), and it is increasingly popular, as they are easy to train from annotated data and provide a very good tradeoff between speed and accuracy.

Strikingly, these parsing models present practically all the characteristics of processing and acquisition that C&C describe as originating from the Now-or-Never bottleneck in human processing:

- *Incremental Processing (Section 3.1)*: A defining feature of transition-based parsers is that they build syntactic analyses incrementally as they receive the input, from left to right. These systems can build analyses even under severe working memory constraints: although the issue of “stacking up” with right-branching languages mentioned by C&C exists for so-called arc-standard parsers (Nivre, 2004), parsers based on the arc-eager model (e.g. Nivre, 2003; Gómez-Rodríguez & Nivre, 2013) do not accumulate right-branching structures in their stack, as they build dependency links as soon as possible. In these parsers, we only need to keep a word in the stack while we wait for its head or its direct dependents, so the time that linguistic units need to be retained in memory is kept to the bare minimum.
- *Multiple Levels of Linguistic Structure (Section 3.2)*: As mentioned by C&C, the organization of linguistic representation in multiple levels is “typically assumed in the language sciences”; this includes computational linguistics and transition-based parsing models. Traditionally, each of these levels was processed sequentially in a pipeline, contrasting with the parallelism of the Chunk-and-Pass framework. However, the appearance of general incremental processing frameworks spanning various levels, from segmentation to parsing (Zhang & Clark, 2011) has led to recent research on joint processing where the processing of several levels takes place simultaneously and in parallel, passing information between levels (Bohnet & Nivre, 2012; Hatori et al., 2012). These models, which improve accuracy over pipeline models, are very close to the Chunk-and-Pass framework.
- *Predictive Language Processing (Section 3.3)*: The joint processing models just mentioned are hypothesized to provide accuracy improvements precisely because they allow for a degree of predictive processing. Contrary to pipeline approaches where information only flows in a bottom-up way, these systems allow top-down information from higher levels “to constrain the processing of the input at lower levels”, just as C&C describe.
- *Acquisition as Learning to Process (Section 4)*: Transition-based parsers learn a sequence of processing actions (transitions), rather than a grammar (Nivre, 2008; Gómez-Rodríguez et al., 2014), making the learning process simple and flexible.
- *Local learning (Section 4.2)*: This is also a general characteristic of all transition-based parsers. As they do not learn grammar rules but processing actions to take in specific situations, adding a new example to the training data will create only local changes to the inherent language model. At the implementation level, this typically corresponds to small weight changes in the underlying machine learning model - be it an SVM classifier (Nivre et al., 2007), perceptron (Zhang & Clark, 2011), neural network (Chen & Manning, 2014), etc.
- *Online Learning and Learning to Predict (Sections 4.1 and 4.3)*: As evaluation of NLP

systems usually takes place in standard, fixed corpora; recent NLP literature has not placed much emphasis on online learning. However, some systems and frameworks do use online learning models with error-driven learning, like the perceptron (Zhang & Clark, 2011). The recent surge of interest in parsing with neural networks (e.g. Chen & Manning, 2014, Dyer et al., 2015) also seems to point future research in this direction.

Putting it all together, we can see that researchers whose motivating goal was not psycholinguistic modeling, but only raw computational efficiency, have nevertheless arrived at models that conform to the description in the target article. This provides further support for the views expressed in the article.

A natural question arises as to what extent this coincidence is due to similarities between the efficiency requirements of human and automated processing; or rather to the fact that as evolution shapes natural languages to be easy to process by humans (constrained by the Now-or-Never bottleneck), computational models that mirror human processing will naturally work well on them. Relevant differences between the brain and computers, such as in short-term memory capacity, seem to suggest the latter. Either way, there is clearly much to be gained from cross-fertilization between cognitive science and computational linguistics: for example, computational linguists can find inspiration in cognitive models for designing NLP tools that work efficiently with limited resources; and cognitive scientists can use computational tools as models to test their hypotheses. Bridging the gap between these areas of research is essential to further our understanding of language.

## Acknowledgment

This work was funded by the Spanish Ministry of Economy and Competitiveness/ERDF (grant FFI2014-51978-C2-2-R) and Xunta de Galicia (grant R2014/034). I thank Ramon Ferrer-i-Cancho for helpful comments on an early version of this commentary.

## References

- Bohnet, B. & Nivre, J. (2012) A Transition-Based System for Joint Part-of-Speech Tagging and Labeled Non-Projective Dependency Parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Jeju Island, Korea, July 12-14, 2012*, pp. 1455–1465. Association for Computational Linguistics. (ISBN: 978-1-937284-43-5).
- Chen, D. & Manning, C.D. (2014) A Fast and Accurate Dependency Parser using Neural Networks. In *Proceedings of the 2014 Conference on Empirical Methods on Natural Language Processing, Doha, Qatar, October 25-29, 2014*, pp. 740-750. Association for Computational Linguistics. (ISBN: 978-1-937284-96-1).
- Choi, J.D. & McCallum, A. (2013) Transition-based Dependency Parsing with Selectional Branching. In *Proceedings of the 51<sup>st</sup> Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Sofia, Bulgaria, August 4-9, 2013*. pp. 1052–1062. Association for Computational Linguistics. Sofia, Bulgaria. (ISBN: 978-1-937284-50-3).

- Dyer, C., Ballesteros, M., Ling, W., Matthews, A. & Smith, N. (2015) Transition-Based Dependency Parsing with Stack Long Short-Term Memory. In *Proceedings of the 53<sup>rd</sup> Annual Meeting of the Association for Computational Linguistics, Beijing, China, July 26-31, 2015*. Association for Computational Linguistics, in press.
- Gómez-Rodríguez, C. & Nivre, J. (2013) Divisible Transition Systems and Multiplanar Dependency Parsing. *Computational Linguistics* 39(4):799-845.
- Gómez-Rodríguez, C., Sartorio, F. & Satta, G. (2014) A Polynomial-Time Dynamic Oracle for Non-Projective Dependency Parsing. In *Proceedings of the 2014 Conference on Empirical Methods on Natural Language Processing, Doha, Qatar, October 25-29, 2014*, pp. 917-927. Association for Computational Linguistics. (ISBN: 978-1-937284-96-1).
- Hatori, J., Matsuzaki, T., Miyao, Y. & Tsujii, J. (2012) Incremental Joint Approach to Word Segmentation, POS Tagging, and Dependency Parsing in Chinese. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Jeju Island, Korea, July 8-14, 2012*. pp. 1216-1224. Association for Computational Linguistics. (ISBN: 978-1-937284-24-4).
- Nivre, J. (2003) An efficient algorithm for projective dependency parsing. In *Proceedings of the 8<sup>th</sup> International Workshop on Parsing Technologies (IWPT 03), Nancy, France, April 23-25, 2003*, pp. 149–160. Association for Computational Linguistics.
- Nivre, J. (2004) Incrementality in deterministic dependency parsing. In *Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together*, ed. F. Keller, S. Clark, M. Crocker & M. Steedman, pp. 50–57. Association for Computational Linguistics.
- Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S. & Marsi, E. (2007) MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering* 13(2):95-135.
- Nivre, J. (2008) Algorithms for Deterministic Incremental Dependency Parsing. *Computational Linguistics* 34(4):513-553.
- Zhang, Y. & Clark, S. (2011) Syntactic Processing Using the Generalized Perceptron and Beam Search. *Computational Linguistics* 37(1):105-151.