Constituency and Recursion in Language

Morten H. Christiansen and Nick Chater

Introduction

Upon reflection, most people would agree that the words in a sentence are not merely arranged like beads on a string. Rather, the words group together to form coherent building blocks within a sentence. Consider the sentence, *The girl liked a boy*. Intuitively, the chunks *the girl* and *liked a boy* constitute the basic components of this sentence (compared to a simple listing of the individual words or alternative groupings, such as *the girl liked* and *a boy*). Linguistically, these chunks comprise the two major *constituents* of a sentence: a subject noun phrase (NP), *the girl*, and a verb phrase (VP), *liked a boy*. Such *phrasal* constituents may contain two types of syntactic elements: other phrasal constituents (e.g., the NP *a boy* in the above VP) or *lexical* constituents (e.g., the determiner *the* and the noun *girl* in the NP *the girl*). Both types of constituent are typically defined *distributionally* using the so-called replacement test: If a novel word or phrase has the same distribution as a word or phrase of a known constituent type—that is if the former can be *replaced* by the latter—then they are the same type of constituent. Thus, the lexical constituents *the* and *a* both belong to the lexical category of determiners because they occur in similar contexts and therefore can replace each other (e.g., *A girl liked the boy*). Likewise, *the girl* and *a boy* belong to the same phrasal category, NP, because they can be swapped around, as in *A boy liked the girl* (note, however, that there may be semantic constraints on constituent replacements. For example, replacing the animate subject NP *the girl* with the inanimate NP *the chair* yields the semantically anomalous sentence, *The chair liked a boy*).

In linguistics, grammar rules and/or principles determine how constituents can be put together to form sentences. For instance, we can use the following phrase structure rules to describe the relationship between the constituents in the example sentences above:

$$\begin{aligned} S &\rightarrow NP\ VP \\ NP &\rightarrow (det)\ N \\ VP &\rightarrow V(NP) \end{aligned}$$

Using these rules we obtain the following relationships between the lexical and phrasal constituents:

$$[_S[_{NP}[_{det}\ \text{The}\ ][_N\ \text{girl}\ ]][_{VP}[_V\ \text{liked}\ ][_{NP}\ [_{det}\ \text{a}\ ][_N\ \text{boy}\ ]]]]$$

To capture the full generativity of human language, *recursion* needs to be introduced into the grammar. We can incorporate recursion into the above rule set by introducing a new rule that adds a potential prepositional phrase (PP) to the NP:

$$\begin{aligned} NP &\rightarrow (det)N(PP) \\ PP &\rightarrow prep\ NP \end{aligned}$$

These rules are recursive because the expansion of the right-hand sides of each can involve a call to the other. For example, the complex NP *the flowers in the vase* has the simple NP *the vase* recursively embedded within it. This process can be applied arbitrarily often, creating, for instance, the complex NP with three embedded NPs:

$$[_{NP}\ \text{the flowers}\ [_{PP}\ \text{in}\ [_{NP}\ \text{the vase}\ [_{PP}\ \text{on}\ [_{NP}\ \text{the table}\ [_{PP}\ \text{by}\ [_{NP}\ \text{the window}]]]]]]]$$

Recursive rules can thus generate constructions of arbitrary complexity.

Constituency and recursion are some of the most fundamental concepts in linguistics. As we saw above, both are defined in terms of relations between symbols. Symbolic models of language processing therefore incorporate these properties by fiat. In this article, we discuss how constituency and recursion may fit into a connectionist framework and the possible implications for linguistics and psycholinguistics.

Constituency

Connectionist models of language processing can address constituency in three increasingly radical ways. First, some connectionist models are *implementations* of symbolic language processing models in "neural" hardware. Many early connectionist models of syntax used this approach; an example is Fanty's (1986) network implementation of a context-free grammar. This kind of model contains explicit representations of the constituent structure of a sentence in just the same way as a nonconnectionist implementation of the same model would. Connectionist implementations of this kind may be important; they have the potential to provide feasibility proofs that traditional symbolic models of language processing are compatible with a "brain-style" computational architecture. But these models add nothing new with respect to the treatment of constituency.

The remaining two classes of connectionist models *learn* to process constituent structure, rather than having this ability hardwired. One approach is to have a network learn from input "tagged" with information about constituent structure. For example, Kim, Srinivas, and Trueswell (in 2002) train a network to map a combination of orthographic and co-occurrence-based "semantic" information about a word onto a structured representation encoding the minimal syntactic environment for that word. With an input vocabulary consisting of 20,000 words, this model has an impressive coverage and can account for certain results from the psycholinguistic literature concerning ambiguity resolution in sentence processing. But because constituent structure has been "compiled" into the output representations that the network was trained to produce, this kind of model does not offer any fresh insight into how linguistic constituency might operate, based on connectionist principles.

The third class of connectionist models addresses the more ambitious problem of learning the constituent structure of a language from untagged linguistic input. Such models have the potential to develop a new or unexpected notion of constituency, and hence may have substantial implications for theories of constituency in linguistics and psycholinguistics.

To understand how the more radical connectionist models address constituency, we need to frame the problem more generally. We can divide the problem of finding constituent structure in linguistic input of a language into two interrelated parts: segmenting the sentence into chunks that correspond, to some extent, to linguistic constituents, and categorizing these units appropriately. The first problem is an aspect of the general problem of *segmenting* speech into appropriate units (e.g., phonemes, words) and more generally is an aspect of perceptual grouping. The second problem is an aspect of the general problem of classifying linguistic units—for instance, recognizing different classes of phonemes or establishing the parts of speech of individual lexical items. The segmentation and classification problems need not be solved sequentially. Indeed, there may be mutual influence between the decision to segment a particular chunk of language and the decision that it can be classified in a particular way. Nonetheless, it is useful to keep the two aspects of the analysis of constituency conceptually separate.

It is also important to stress the difference between the problem of assigning constituent structure to novel sentences where the language is known and the problem of acquiring the constituent structure of an unknown language. Statistical symbolic parsers are able to make some inroads into the first problem (Charniak, 1993). For highly stylized language input, and given a prestored grammar, they can apply grammatical knowledge to establish one or more possible constituent structures for novel sentences. But symbolic methods are much less advanced in acquiring the constituent structure of language, because this requires solving the hard problem of learning a grammar from a set of sentences generated by that grammar. It is therefore in relation to the acquisition of constituency that connectionist methods, with their well-developed learning methods, have attracted the most interest.

We begin by considering models that focus on the problem of classifying, rather than segmenting, the linguistic input. One connectionist model (Finch and Chater, 1993) learns the part of speech of individual words by clustering words together on the basis of the immediate linguistic contexts in which they occur. The rationale is based on the replacement test mentioned earlier: if two words are observed to occur in highly similar immediate contexts in a corpus, they probably belong to the same syntactic category. Finch and Chater used a single-layer network with Hebbian learning to store co-occurrences between “target” words and their near neighbors. This allowed each target word to be associated with a vector representing the contexts in which it typically occurred. A competitive learning network classified these vectors, thus grouping together words with similar syntactic categories. This method is able to operate over unrestricted natural language, in contrast to most symbolic and connectionist models. From a linguistic perspective, the model slices lexical categories too finely, producing, for example, many word classes that correspond to nouns or verbs. On the other hand, the words within a class tend to be semantically related, which is useful from a cognitive perspective. The same method can be extended to classify sequences of words as NPs, VPs, etc. An initial classification of words is used to recode the input as a sequence of lexical constituents. Then, short sequences of lexical constituents are classified by their context, as before. The resulting groups of “phrases” (e.g., determiner-adjective-noun) are readily interpretable as NPs, and so on, but again, these groupings are too linguistically restrictive (i.e., only a small number of NPs are included in any particular cluster). Moreover, this phrasal level classification has not yet been implemented in a connectionist network.

A different attack on the problem of constituency involves training simple recurrent networks (SRNs) on linguistic input (Elman, 1990). An SRN involves a crucial modification to a feedforward network: the current set of hidden unit values is “copied back” to a set of additional input units, and paired with the *next* input to the network. The current hidden unit values can thus directly affect the next hidden unit values, providing the network with a memory for past inputs. This enables it to tackle sentence processing, where the input is revealed gradually over time rather than being presented at once.

Segmentation into constituents can be achieved in two ways by an SRN trained to *predict* the next input. One way is based on the assumption that predictability is higher within a constituent than across constituent boundaries, and hence that high prediction error indicates a boundary. This method has been advocated as potentially applicable at a range of linguistic levels (Elman, 1990), but in practice it has been successfully applied only on corpora of unrestricted natural language input in finding word boundaries (Cairns et al., 1997). Even here, the prediction strategy is a very partial cue to segmentation. If the network is provided with information about naturally occurring pauses between utterances (or parts of utterances), an alternative method is to assume that constituent boundaries occur where the network has an unusually high expectation of an *utterance boundary*. The rationale is that pauses tend to occur at constituent boundaries, and hence the prediction of a possible utterance boundary suggests that a constituent boundary may have occurred. This approach seems highly applicable to segmenting sentences into phrases, but it, too, has primarily been used for finding word boundaries in real corpora of language, when combined with other cues (Christiansen, Allen, and Seidenberg, 1998).

So far we have considered how SRNs might find constituents. But how well do they classify constituents? At the word level, cluster analysis of hidden unit activations shows that, to some extent, the hidden unit patterns associated with different word classes group naturally into syntactic categories, for SRNs trained on simple artificial grammars (Elman, 1990). These results are important because they show that even though the SRN may not learn to classify constituents explicitly, it is nevertheless able to *use* this information to process constituents appropriately.

Another way of assessing how SRNs have learned constituency is to see if they can generalize to predicting novel sentences of a language. The logic is that to predict successfully, the SRN must exploit linguistic regularities that are defined across constituents, and hence develop a notion of constituency to do so. However, Hadley (1994) points out that this type of evidence is not compelling if the novel sentences are extremely similar to the network's training sentences. He suggests that, to show substantial evidence for generalization across constituents, the network should be able to handle novel sentences in which words appears in sentence locations where they have not previously occurred (see SYSTEMACITY OF GENERALIZATIONS IN CONNECTIONIST NETWORKS). For example, a novel sentence might involve a particular noun in object position, where it has previously occurred only in subject position. To generalize effectively, the network must presumably develop some abstract category of nouns. Christiansen and Chater (1994) demonstrated that an SRN can show this kind of generalization.

Despite this demonstration, though, connectionist models do not mirror classical constituency precisely. That is, they do not derive rigid classes of words and phrases that are interchangeable across contexts. Rather, they divide words and phrases into clusters without precisely defined boundaries, and they treat words and phrases differently, depending on the linguistic contexts in which they occur. This *context-sensitive* constituency can be viewed either as the undoing of connectionist approaches to language or as their radical contribution.

The potential problem with context-sensitive constituency is the productivity of language. To take Chomsky's famous example, how do we know that the statement *colorless green ideas sleep furiously* is syntactically correct, except by reference to a context-*in*sensitive representation of the relevant word classes? This seems necessary, because each word occurs in a context in which it has rarely been encountered before. But Allen and Seidenberg (1999) argue that this problem may not be fatal for context-sensitive notions of constituency. They trained a network to mutually associate two input sequences, a sequence of word forms and a corresponding sequence of word meanings. The network was able to learn a small artificial language successfully: it was able to regenerate the word forms from the meanings, and vice versa. Allen and Seidenberg then tested whether the network could recreate a sequence of word forms presented to it, by passing information from form to meaning and back. Ungrammatical sentences were recreated less accurately than grammatical sentences, and the network was thus able to distinguish grammatical from ungrammatical sentences. Importantly, this was true for sentences in which words appeared in novel combinations, as specified by Hadley's criterion and as exemplified by Chomsky's famous sentence. Thus, the context-sensitivity of connectionist constituency may not rule out the possibility of highly creative and novel use of language, because abstract relations may be encoded at a semantic level as well as at the level of word forms.

If the apparent linguistic limitations of context-sensitive constituency can be overcome, then the potential psychological contribution of this notion is enormous. First, context sensitivity seems to be the norm throughout human classification. Second, much data on sentence processing seem most naturally to be explained by assuming that constituents are represented in a fuzzy and context-bound manner. The resulting opportunities for connectionist modeling of language processing are extremely promising. Thus, connectionist research may provide a more psychologically adequate notion of constituency than is currently available in linguistics.

Recursion

As with constituency, connectionist models have dealt with recursion in three increasingly radical ways. The least radical approach is to hardwire recursion into the network (e.g., as in Fanty's (1986) implementation of phrase structure rules) or to add an external symbolic ("first-in-last-out") stack to the model (e.g., as in Kwasny and Faisal's (1990) deterministic connectionist parser). In both cases, recursive generativity is achieved entirely through standard symbolic means, and although this is a perfectly reasonable approach to recursion, it adds nothing new to symbolic accounts of natural language recursion. The more radical connectionist approaches to recursion aim for networks to *learn* to deal with recursive structure. One approach is to construct a modular system of networks, each of which is trained to acquire different aspects of syntactic processing. For example, Miikkulainen's (1996) system consists of three different networks: one trained to map words onto case-role assignments, another trained to function as a stack, and a third trained to segment the input into constituent-like units. Although the model displays complex recursive abilities, the basis for these abilities and their generalization to novel sentence structures derive from the configuration of the stack network combined with the modular architecture of the system, rather than being discovered by the model. The most radical connectionist approaches to recursion attempt to learn recursive abilities with minimal prior knowledge built into the system. In this type of model, the network is most often required to discover both the constituent structure of the input and how these constituents can be recursively assembled into sentences. As with the similar approach to constituency described in the previous section, such models may provide new insights into the notion of recursion in human language processing.

Before discussing these modeling efforts, we need to assess to what extent recursion is observed in human language behavior. It is useful to distinguish *simple* and *complex* recursion. Simple recursion consists in recursively adding new material to the left (e.g., the adjective phrases (AP) in *the gray cat* → *the fat gray cat* → *the ugly fat gray cat* or the right (e.g., the PPs in *the flowers in the vase* → *the flowers in the vase on the table* → *the flowers in the vase on the table by the window*) of existing phrase material. In complex recursion, new material is added in more complicated ways, such as through center-embedding of sentences (*The chef admired the musicians* → *The chef whom the waiter appreciated admired the musicians*). Psycholinguistic evidence shows that people find simple recursion relatively easy to process, whereas complex recursion is almost impossible to process with more than one level of recursion. For instance, the following sentence with two levels of simple (right-branching) recursion, *The busboy offended the waiter who appreciated the chef who admired the musicians*, is much easier to comprehend than the comparable sentence with two levels of complex recursion, *The chef whom the waiter whom the busboy offended appreciated admired the musicians*. Because recursion is built into the symbolic models, there are no *intrinsic* limitations on how many levels of recursion can be processed. Instead, such models must invoke *extrinsic* constraints to accommodate the human performance asymmetry on simple and complex constructions. The radical connectionist approach models human performance directly without the need for extrinsic performance constraints.

The SRN model developed by Elman (1991) was perhaps the first connectionist attempt to simulate human behavior on recursive constructions. This network was trained on sentences generated by a small context-free grammar incorporating center-embedding and a single kind of right-branching recursive structure. In related work, Christiansen and Chater (1994) trained SRNs on a recursive artificial language incorporating four kinds of right-branching structures, a left-branching structure, and center-embedding. The behavior of these networks was qualitatively comparable with human performance in that the SRN predictions for right-branching structures were more accurate than on sentences of the same length involving center-embedding, and performance degraded appropriately as the depth of center-embedding increased. Weckerly and Elman (1992) further corroborated these results, suggesting that semantic bias (incorporated via co-occurrence restrictions on the verbs) can facilitate network performance in center-embedded constructions, similar to the semantic facilitation effects found in human processing. Using abstract artificial languages, Christiansen and Chater (1999) showed that the SRN's general pattern of performance is relatively invariant across network size and training corpus, and concluded that the human-like pattern of performance derived from intrinsic constraints inherent to the SRN architecture.

Connectionist models of recursive syntax typically use "toy" fragments of grammar and small vocabularies. Aside from raising concerns over scaling-up, this makes it difficult to provide detailed fits with empirical data. Nonetheless, some attempts have recently been made to fit existing data and derive new empirical predictions from the models. For example, the Christiansen and Chater (1999) SRN model fits grammaticality ratings data from several behavioral experiments, including an account of the relative processing difficulty associated with the processing of center-embeddings (with the following relationship between nouns and verbs: $N_1N_2N_3V_3V_2V_1$) versus cross-dependencies (with the following relationship between nouns and verbs: $N_1N_2N_3V_1V_2V_3$). Human data have shown that sentences with two center-embeddings (in German) are significantly harder to process than comparable sentences with two cross-dependencies (in Dutch). The simulation results demonstrated that the SRNs exhibited the same kind of qualitative processing difficulties as humans on these two types of complex recursive constructions.

Just as the radical connectionist approach to constituency deviates from classical constituency, the above approach to recursion deviates from the classical notion of recursion. The radical models of recursion do not acquire "true" recursion because they are unable to process infinitely complex recursive constructions. However, the classical notion of recursion may be ill-suited for capturing human recursive abilities. Indeed, the psycholinguistic data suggest that people's performance may be better construed as being only quasi-recursive. The semantic facilitation of recursive processing, mentioned earlier, further suggests that human recursive performance may be partially context-sensitive. For example, the semantically biased *The bees that the hive that the farmer built housed stung the children* is easier to comprehend than the neutral *The chef that the waiter that the busboy offended appreciated admired the musicians*, even though both sentences contain two center-embeddings. This dovetails with the context-sensitive notion of constituency and suggests that context sensitivity may be a more pervasive feature of language processing than is typically assumed by symbolic approaches.

Discussion

This article has outlined several ways in which constituency and recursion may be accommodated within a connectionist framework, ranging from direct implementation of symbolic systems to the acquisition of constituency and recursion from untagged input. We have focused on the radical approach, because this approach has the greatest potential to affect psycholinguistics and linguistic theory. However, much of this research is still preliminary. More work is needed to decide whether the promising but limited initial results can eventually be scaled up to deal with the complexities of real language input, or whether a radical connectionist approach is beset by fundamental limitations. Another challenge is to find ways—theoretically and practically—to interface models that have been proposed at different levels of linguistic analyses, such as models of morphology with models of sentence processing.

Nevertheless, the connectionist models described in this article have already influenced the study of language processing. First, connectionism has helped promote a general change toward replacing "box-and-arrow" diagrams with explicit computational models. Second, connectionism has reinvigorated the interest in computational models of learning, including learning properties, such as recursion and constituent structure, that were previously assumed to be innate. Finally, connectionism has helped increase interest in the statistical aspects of language learning and processing.

Connectionism has thus already had a considerable impact on the psychology of language. But the final extent of this influence depends on the degree to which practical connectionist models can be developed and extended to deal with complex aspects of language processing in a psychologically realistic way. If realistic connectionist models of language processing can be provided, then the possibility of a radical rethinking not just of the nature of language processing, but of the structure of language itself, may be required.

Roadmap: Linguistics and Speech Processing
Background: Language Processing
Related Reading: Language Acquisition; Recurrent Networks, Learning Algorithms

References

Allen, J., and Seidenberg, M. S., 1999, The emergence of grammaticality in connectionist networks, in *The Emergence of Language* (B. MacWhinney, Ed.), Mahwah, NJ: Erlbaum, pp. 115–151.

Cairns, P., Shillcock, R. C., Chater, N., and Levy, J., 1997, Bootstrapping word boundaries: A bottom-up corpus-based approach to speech segmentation, *Cogn. Psychol.*, 33:111–153.

Charniak, E., 1993, *Statistical Language Learning*, Cambridge, MA: MIT Press. ◆

Christiansen, M. H., Allen, J., and Seidenberg, M. S., 1998, Learning to segment speech using multiple cues: A connectionist model, *Lang. Cogn. Proc.*, 13:221–268.

Christiansen, M. H., and Chater, N., 1994, Generalization and connectionist language learning, *Mind Lang.*, 9:273–287.

Christiansen, M. H., and Chater, N., 1999, Toward a connectionist model of recursion in human linguistic performance, *Cogn. Sci.*, 23:157–205. ◆

Elman, J. L., 1990, Finding structure in time, *Cogn. Sci.*, 14:179–211. ◆

Elman, J. L., 1991, Distributed representation, simple recurrent networks, and grammatical structure, *Machine Learn.*, 7:195–225.

Fanty, M. A., 1986, Context-free parsing with connectionist networks, in *Neural Networks for Computing* (J. S. Denker, Ed.), New York: American Institute of Physics, pp. 140–145.

Finch, S., and Chater, N., 1993, Learning syntactic categories: A statistical approach, in *Neurodynamics and Psychology* (M. Oaksford and G. D. A. Brown, Eds.), New York: Academic Press, pp. 295–321.

Hadley, R. F., 1994, Systematicity in connectionist language learning. *Mind Lang.*, 9:247–272.

Kim, A. E., Srinivas, B., and Trueswell, J. C., 2002, The convergence of lexicalist perspectives in psycholinguistics and computational linguistics, in *Sentence Processing and the Lexicon: Formal, Computational and Experimental Perspectives* (P. Merlo and S. Stevenson, Eds.), Amsterdam: John Benjamins Publishing, pp. 109–135.

Kwasny, S. C., and Faisal, K. A., 1990, Connectionism and determinism in a syntactic parser, *Connect. Sci.*, 2:63–82.

Miikkulainen, R., 1996, Subsymbolic case-role analysis of sentences with embedded clauses, *Cogn. Sci.*, 20:47–73.

Weckerly, J., and Elman, J. L., 1992, A PDP approach to processing center-embedded sentences, in *Proceedings of the Fourteenth Annual Meeting of the Cognitive Science Society*, Hillsdale, NJ: Erlbaum, pp. 414–419.