Darwinised Data-Oriented Parsing – Statistical NLP with added Sex and Death

Dave Cochran

Cognitive Systems Group,
School of Computer Science,
University of St. Andrews

davec@cs.st-andrews.ac.uk

Abstract

We present the Darwinised Data-Oriented Parsing algorithm, an incremental, dynamic form of Data-Oriented Parsing, in which exemplars are used as replicators, subject to a selection pressure towards generalisability.¹

1 Introduction

Data-Oriented Parsing (DOP) is a state-of-the-art approach to both supervised and unsupervised parsing (Bod 1992, 1998, 2006a, 2006b, 2007a, 2007b, Zollman and Sima'an 2005), which has mostly been developed within a technologically-oriented computer science context. Recent work has highlighted some interesting cognitive properties of the Data-Oriented approach (Borensztajn, Zuidema & Bod 2008, Bod 2008). However, these studies have mostly focused on the static properties of the DOP probability model. Here, we present the first attempt at a dynamic, incremental Data-Oriented model which can address the time course of language learning, rather than just the outcome; Darwinised DOP.

2 Data-Oriented Parsing

2.1 Supervised DOP

Data-Oriented Parsing (DOP) is a paradigm in Natural Language Processing in which linguistic knowledge is represented as fragmentable, re-combinable exemplars of concrete previous experience, usually in the form of *trees*. What crucially distinguishes DOP from other approaches is the fact that fragments of arbitrary size are

used, ranging, in the case of the usual tree-structures, from depth-1 context-free rewrite rules to entire trees, and all points in between; this gives it the power to pick up on whatever statistical patterns are present in the data, to a considerable extent bypassing of the researcher's theoretical prejudices. Moreover, it allows these regularities to be exploited *without being represented*. DOP was first proposed by Scha (1990), and implemented and developed by Bod (1992, 1998).

The simplest manifestation of DOP is DOP1, as described in Bod (1998 p12-23 and 40-50), though more sophisticated versions exist. The parser uses a large corpus of natural language strings annotated with labeled, ordered tree-structures, divided into a training corpus and a smaller corpus against which the parser is tested. The parser uses every possible subtree (of unlimited depth) of all the available trees, constrained only by the following wellformedness criteria (Fig. 1).

- Every subtree must be of at least depth 1.
- Sister relationships must be preserved: that is, either all or none of the daughters of a given node may be extracted, but not only some.

Figure 1: Well-formed subtrees of a parse of "John likes Mary"

¹ The author thanks Rens Bod, Mark-Jan Nederhof and three anonymous reviewers for exceedingly helpful comments, suggestions and references.

Fig 2: A sequence of substitutions comprising a derivation of “John likes Mary”. $\circ$ is the operator for leftmost non-terminal leaf-node substitution.

The parser is given test corpus strings and builds up new parse-trees for these using the fragments available to it from the training corpus (Fig. 2), starting with a fragment with an S-node at the top, and then, for each nonterminal leaf-node, working rightwards, substituting in additional subtrees, the topmost node of which must carry the same label as the node to be substituted. (see figure 2.2).

In DOP research it is necessary to distinguish between *parses* and *derivations*. A parse is the tree structure expressed over a string; a derivation is the particular sequence of subtree substitutions by which it was constructed. When parsing with probabilistic context-free grammars (PCFG’s, see Manning and Schütze 1999, pp.381-405; note that a PCFG is equivalent to a DOP grammar in which subtree depth has been restricted to 1), there is a one-to-one mapping between parses and derivations, because all non-terminal nodes (nodes which have daughters in the completed parse) *must* be substitution sites. In DOP, subtrees can be of any depth, and so in any given derivation, any subset of the non-terminal nodes could have been substitution sites, while the remainder will not have been. As such, if a parse contains N many non-terminal nodes, it will have 2^N many derivations.

For each subtree t , its probability $P(t)$ is its total frequency $|t|$ of occurrence in the training corpus over the summed corpus frequency of subtrees with the same root node;²

$$P(t) = \frac{|t|}{\sum_{\{t': r(t)=r(t')\}} |t'|} \quad (1)$$

...where $r(t)$ and $r(t')$ are the node-labels on the root-nodes of subtrees t and t' .

² Note that although, beside the node-label on the substitution site, the input to be parsed is also a constraint on the selection of subtrees for substitutions, these constraints are not factored in to the calculation of probabilities. That is to say, the probability used is that of the subtree, not the substitution.

The probability of a derivation is the product of the probabilities of its subtrees (note that $\circ$ is the notation for the substitution operation; thus $t_1 \circ \dots \circ t_n$ is the sequence of substitutions, which together comprise the derivation);

$$P(t_1 \circ \dots \circ t_n) = \prod_i P(t_i) \quad (2)$$

And the probability of a parse T is the sum of the probabilities of its possible derivations D ;

$$P(T) = \sum_{\{D: D \text{ derives } T\}} P(D) \quad (3)$$

The output of the parser is, in theory, the most probable parse. In practice, there are issues of computational complexity that prevent this from being calculated directly; instead, a Monte-Carlo sample is taken. Furthermore, although the number of subtrees for any given tree increases exponentially with tree-size, it is possible to reduce DOP trees to a stochastically equivalent PCFG which expands linearly with tree-size; for details see Goodman 1996, 2003)

Bod (ibid p.54) reports accuracies of 85% on the ATIS³ corpus for DOP1. However, better results are achieved with later, more sophisticated versions of DOP – the current state of the art is DOP* (Zollman and Sima'an 2005), which selects parses on the basis of the shortest derivation, and only uses probabilities to tie-break if there is more than one shortest derivation; this approach overcomes the problems of statistical consistency and bias which Johnson (1998) pointed out as afflicting DOP1.

2.2 Unsupervised DOP

Unsupervised DOP (UDOP, Bod 2006b, 2007a, 2007b) extends the DOP approach to bootstrap language without recourse to a training corpus of manually annotated language data as a representation of “prior experience”. UDOP expands on DOP’s maximalist, all-subtrees, approach by using all subtrees of all possible (binary) trees to provide the parser with a resource of subtrees for the construction of derivations (which may be stored within a chart in quadratic space: Bod 1998, pp.40-8). Not only is this shown to achieve state-of-the-art results compared to other unsupervised parsing methods (Bod 2006b, 2007a), it is also shown to outperform state-of-the-art *supervised* parsing techniques when evaluated as a language model for a practical application (Machine Translation), rather than using the rather academical and the-

³ Air Transport Information System – part of the Penn Treebank.

ory-laden standard of agreement with the judgements of manual annotations on a corpus (Bod 2007b).

In the UDOP* implementation of UDOP, all subtrees of all possible trees are extracted from the training corpus, and from this exemplar-base, the shortest derivation (in the fashion of DOP*) for each string is calculated (again, the complexity of the task may be reined in using Goodman’s PCFG-reduction, 1996, 2003). The set of trees that results from this is then converted again to subtrees and used as a Stochastic Tree Substitution Grammar, which is then used to parse the test strings. In UML-DOP, another implementation, this last step is iterated over the training data until there is negligible reduction in cross-entropy. Bod (2007a) shows that these methods applied to miniature “toy” corpora can be used to explain linguistically interesting phenomena, such as long-distance agreement and “movement”, as emerging out of simpler structures without themselves being found in either experience or “hardwired” grammar. However, the batch learning methods noted above are cognitively implausible. This is not a criticism of the approach in itself; Batterman (2005) shows that idealisations, however unrealistic in themselves, are necessary in scientific modeling in order to explain universalities which more “realistic” models would miss. However, UDOP cannot itself model the time-course of developmental learning processes, which are by nature incremental. It is to make good this deficit that we have developed the Darwinised DOP (DDOP) approach.

2.3 Darwinised DOP

Darwinised Data-Oriented Parsing is a new unsupervised parsing algorithm which allows the time-course of pattern-learning to be modeled. Unlike previous DOP algorithms, it begins with a completely empty training set; it is fed strings one by one, and its own outputs are entered into the training set. In so doing, it exploits a hitherto underexploited property of exemplar-based systems; when an exemplar (subtree) is reused in producing the system’s eventual output, this output contains a new copy of the reused exemplar which is inserted back into the exemplar-base upon which the algorithm operates; thus, exemplars become *replicators*; packets of information coupled to mechanisms by which new copies of themselves are generated. Furthermore, exemplars which are able to be used more often – those that are more highly *generalisable* – are

likely to make more new copies of themselves; thus we find a *selection pressure* favouring generalisability.⁴ It is also worth noting that because the subsequent sampling of subtrees from a stored tree can and most likely will cross-cut the substitution-sites of the original derivations by which the stored tree was created, replication of subtrees is *recombinant*; exemplars do not just reproduce, they reproduce sexually.

Trees have a limited lifespan, and are erased from memory after K many parses have been generated following their creation. This serves two functions; firstly, if the dataset is small, the training data may be iterated through several times, but because DDOP uses the DOP* shortest derivation method, it must be prevented from seeing strings for which it already has a complete parse in memory, or else it will simply return the parse it gave before, which of course can be generated in a single-step derivation. Secondly, death is an essential component of evolving systems. Without death, maladaptive and primitive forms are allowed to remain in the system, still reproducing, albeit at a slower rate than newer, more highly evolved replicators.

As mentioned above, DDOP uses DOP*’s shortest-derivation method of parsing, but because it begins with an empty exemplar-base, it needs to have a backoff behaviour, in case it encounters a situation, at some point in a derivation, where no subtree can be found in the exemplar-base which would allow the derivation to continue. When a backoff subtree is generated, the following information only is used:

- The node-label l of the substitution site. This does not affect the probability model, but determines the node-label of the root of the new subtree. In fact, this feature is redundant in all the versions of the model tested so far, as the number of available node labels has been limited to one.
- The substring $w_x \dots w_x$, of the total string being parsed, which represents the largest possible substring capable of eventually

⁴ This notion of exemplars as evolving replicators is not without precedent; see Batali 1994; however, Batali’s model concerned the evolution of a shared linguistic code in a population of agents, rather than exploiting this property for the individual learning of a preexisting language. We are also indebted to Kirby’s (1999, for instance) insight that evolving linguistic systems will favour generalisability, though again Kirby’s models concern evolution over glosogenetic, generational time-scales.

being daughters of the node at the substitution site.

We have tested two versions of DDOP, Non-Folding DDOP (NF-DDOP) and Folding DDOP (F-DDOP), each of which use different backoff procedures: RANDOM and FLAT, respectively. The other difference between the two versions of DDOP concerns the interpretation of flat structures in parses (here taken to mean context-free rewrite rules with an arity of three or greater) – internal nodes with three or more immediate daughters. In NF-DDOP, the RANDOM backoff procedure generates subtrees as a random sample of the complete set of all subtrees of all possible parses of substring $w_i \dots w_{i'}$ with root node t . A description of how this is calculated can be found at <http://www.cs.st-andrews.ac.uk/backoff.pdf>. No distinction is made between flat and deep structures. F-DDOP, by contrast, takes flat structures to be an indicator uncertainty regarding actual, lower-arity structure, and therefore as a shorthand for a set of possible lower-arity structures, or “foldings”, wherein the high-arity context free rewrite rules (subtree of depth one) is replaced with a subtree of equal or greater depth, with the same root and frontier. In order to reduce computational load, this set of allowable foldings is limited to subtrees of depth one or two, in the case of the depth two foldings further limited to foldings containing no more than one internal node⁵. By way of example, figure 3 shows all the allowable foldings of a 4-ary subtree.

Derivations proceed one substitution at a time. DOP* subtrees extracted from the training data are always used provided there exists at least one which fits the string being parsed. If at any point no such subtree can be found, then a backoff subtree is generated at that step only. A Monte Carlo sample of N derivations is generated (in the simulations reported here, $N = 1500$). In order to introduce an element of mutation to the process (crucial in any evolving system), a single derivation consisting only of backoff subtrees is occasionally added to the sample, at a probability given by $p(AllBackoff)$, which is a parameter set before the run commences, where $0 \leq p(AllBackoff) \leq 1$. Following the procedure of

Figure 3: 3(a-f) show the allowable foldings of the 4-ary subtree in 3(a).

DOP*, the shortest derivation in the sample is selected as the output parse. If there is more than one parse with the shortest derivation, DOP1 probabilities are applied to tie-break, following equations 1-3 above. The chosen parse is added to the exemplar-base. Note that because, if an all-backoff derivation is included in the sample, it is assessed for derivation-length and probability just like all the rest, this mutation procedure can only actually introduce novel structures if the random derivation is shorter than all the others in the sample, because only then does it avoid the probability-based tiebreak, which precisely penalizes novelty and favours well-known structures.

Initially, because the exemplar-base is empty, the backoff behaviour is the only behaviour, which, over time, gives way to mostly using corpus trees. Importantly, whenever the outputted tree is derived from subtrees taken from memory, the output tree contains new copies of all those subtrees. In this way, subtrees replicate and more generalisable subtrees are selected.

3 Tests & Results

3.1 Test 1: Six-line toy corpus

The first test each of the versions of DDOP were subjected to was a very simple toy corpus, consisting of six three-word sentences;

Corpus	Oracle
<i>The dog barks.</i>	[<i>The dog</i>] barks.
<i>Watch the dog.</i>	Watch [<i>the dog</i> .]
<i>The dog eats.</i>	[<i>The dog</i>] eats.
<i>The cat barks.</i>	[<i>The cat</i>] barks.
<i>Watch the cat.</i>	Watch [<i>the cat</i> .]
<i>The cat eats.</i>	[<i>The cat</i>] eats.

Table 1: six-line corpus

⁵ Note that although this limitation excludes possible parses which may be linguistically pertinent from the immediate folding event (in the case of structures of arity greater than 3), any flat structures which remain can, when re-used, be folded again, allowing the entire space of possible folding to be explored eventually.

This initial task was simply to recognise that “the cat” and “the dog” are constituents. The advantage of initially testing the models on this very simple toy corpus is that it is very easy to analyse what the model is doing. The toy corpus contains two sentence types, V-NP and NP-V (V and N for short), each of which may be parse as left-branching, flat or right branching (L, F and R, respectively). After the first three parses on its run, the parser always has parses of $n-1$ of the n sentences in memory; thus each step in an iteration through the data, each stored parse exemplifies one of the 3 possible parse-structures, giving 3^{n-1} possible memory-states for each step, and $n(3^{n-1})$ possible memory states overall. Of the memory-states possible at any step, nine represent a consistent assignment of parse-structures (L/F/R) to sentence-types (V/N), and of these only VR-NL represents a successful outcome. Inconsistent states are never stable; in the absence of mutation they are inaccessible and if accessed as a result of mutation the parser will return to a consistent state within one iteration. Barring multiple mutations it is possible to calculate which consistent state that will be. We call the n states of a consistent assignment (one state for each step), plus all the inconsistent states that lead predictably into it, a “territory” within the state-space. If the states of a consistent assignment always predictably lead into states in the same territory at the next step, the territory will form a cyclical trajectory through state-space, and will be stable barring a disruptive mutation; if they do not, they will lead into the territory of another consistent assignment. A mutation while the parser is on a cyclical trajectory will have one of the following outcomes:

- Parser moves into an inconsistent state of the same territory, returns to cycle.
- Parser moves into an inconsistent state of another territory, changes cycle.

For NF-DDOP, the model was first probed using a state-space model of all 108 of the possible memory-states of the parser based on a four-line version of the above corpus, minus the lines “the dog eats” and “the cat eats”. This analysis found that eight of the nine consistent assignments are stable cycles, with VR-NR being the only exception. VR-NR states always run off to the “true” VR-NL assignment. This means that a mutation in the VR-NL cycle has a greater likelihood of being non-disruptive (returning to VR-NL) than a mutation in any other cycle, and a disruptive

mutation in another cycle is more likely to result in a shift to VR-NL than to any other cycle. However, with seven other cyclical territories to compete with, the parser still spends the majority of its time in states other than VR-NL; VR-NL is more robust than the other cyclical modes, but no modes are wholly robust. This finding was first calculated *a priori* based on the state-space model, then confirmed empirically. Tests with the full 6-line corpus found further destabilization of VR-NL (Chart 1 below).

Chart 1: Performance of NF-DDOP on the six-line test corpus

Chart 2: F1-Scores for F-DDOP on the six-line mini-corpus

Chart 3: F1 Scores for F-DDOP on the six-line mini-corpus using the RANDOM backoff routine.

Each data point averages over the 20 parses, sampled at 10-parse intervals.

In contrast, for F-DDOP, VR-NL is the only accessible territory, and all other consistent assignments run off to it, with the result that VR-NL is completely robust, and the parser’s performance on the six-line toy corpus holds fast to F1-Scores of 100%, as shown in Chart 2 below.

This is because any state containing flat parse-structures runs off to VR-NL, and states contain-

Chart 4: NF-DDOP performance on the 20-line toy corpus.

Chart 5: F-DDOP performance on the 20-line toy corpus

ing VL and/or NR are inaccessible, because the FLATback-off always yields VF and NF. This is made clearer by comparison with a hybrid DDOP, in which F-DDOP is combined with the RANDOM back-off routine. Here we find that VL-NL is accessible and cyclical, reducing the robustness of VR-NL.

3.2 Test 2: 20-line toy corpus

The second round of tests used a slightly larger and more complex corpus – this time of 20 sentences, varying between 3 and 11 words in length. Again, notable differences were found between NF-DDOP and F-DDOP. Charts 4 and 5 show the performance of the model on this corpus.

In both cases, the performance of the model was not especially great, with only NF-DDOP showing an overall trend towards improvement over the course of the run; however, the difference in mean F1-score is small; 41.6 for NF-DDOP compared to 37.7 for F-DDOP. Most puzzling of all was the wild instability of F-

Chart 6: frequency distribution of differences between F1-score at adjacent datapoints in the tests of NF-DDOP and F-DDOP on the 20-line toy corpus.

Chart 7: F-DDOP performance with gradually increasing attention span

DDOP on this test, especially when contrasted with its robustness in the simpler test. Chart 6 shows a histogram of the changes in F1-score between adjacent datapoints in the preceding two tests. Note that the values for F-DDOP are rather less sharply peaked and have rather fatter tails. Clearly, further research is required to make sense of this.

More interestingly, a second test was done on F-DDOP in which the model was subject to limitations on attention, which were loosened as time went on: specifically, for the first 700 iterations through the data, it ignored all but the 3-word sentences, then continued in 700-iteration blocks, each with an attention span one word greater than the previous, until it could see the whole corpus. Not only was the overall performance much better, it also showed much greater stability, as can be seen in Chart 7above

4 Discussion

UDOP has the unrestricted ability to sample the entire range of possible subtrees of possible parses of the entirety of whatever dataset it is presented with, and has a limitless scope to revisit and reconsider past judgements on the basis of new information. As such, we would not expect DDOP to exceed it in power; having access only to a limited subset of the UDOP subtree-set, and strictly limited powers to revisit and reanalyze past judgements, it is fair to assume that at best the performance of UDOP represents a theoretical upper limit of the power of DDOP. As

such, the questions we must attend to are as follows; how far and in what way can we limit the sampling and reanalyzing power of unsupervised DOP systems, and still approach the convergence properties of UDOP in the limit of learning? And if this is possible, can the restrictions adduced be illuminatingly related to empirical work in Developmental and Evolutionary Linguistics?

DDOP in its original form underperformed because it was an unsystematic search through too large a parameter space for too small an error minimum. The population of exemplars is not best conceptualized as an assembly of individuals in a species, who, in competing with their conspecifics, enhance the genetic health of the whole; rather, they are best seen as an assembly of species (though, like bacteria, promiscuously engaged in lateral gene transfer), competing for space in a shifting landscape of ecological niches defined by statistical patterns in the language data. From the viewpoint of an individual exemplar, the ecology of this landscape is as much defined by its competitors and potential mates as by the language-data itself. The problem is the selfishness of replicators described by Dawkins (1976, Williams 1966) in the *Selfish Gene*. Evolution is a dumb, blind process, and selection concerns only the individual replicator in its immediate fitness landscape.

However two modifications to the original DDOP algorithm, which can both be independently motivated in terms of cognitive realism, have been shown to usefully constrain the availability of niches so as to produce more favourable results.

The first, the introduction of the Folding procedure changes the nature of the algorithm's initial assumptions regarding unknown structure; rather than assigning an arbitrary structure to everything on first sight, no initial structure is assumed, allowing the learner to pick recurring motifs out of the data as they are spotted. (see Saffran, Aslin and Newport 1996 on the rapidity with which infants can pick out motifs). The analogue of this behaviour, in terms of evolutionary biology, is phenotypic plasticity, whereby the genotype of a species provides for multiple possible developmental pathways, to multiple phenotypes, modulated in selectionally advantageous ways by environmental conditions. (West-Eberhard 2003). We may consider the case of higher-arity structures in F-DDOP to be the only instance in DDOP of a separation between genotype and phenotype; in the case of the ternary structures in the six-line corpus, the underlying

flat structure is analogous to the genotype, while the flat, left-branching and right-branching subtrees extracted from it in subsequent derivations may be understood as alternative phenotypes. This sort of adaptive plasticity allows "populations to move into new adaptive zones without abandoning old ones ... [a]lternative phenotypes enable condition sensitive evolutionary experimentation within populations" (West-Eberhard 2003, p.392). These variations in phenotype arise out of an interaction between "genotype" and environment (DeWitt and Scheiner 2004b) – understood as both the incoming stream of language inputs and the accumulated store of other exemplars – in a manner that responds adaptively to the frequencies of linguistic patterns in the environment.

Secondly, we introduced the assumption of an initial limitation on the length of sentences the parser is exposed to, which expands with maturation. It is known that adults' speech to infants tends to be characterized by shorter, simpler utterances than normal adult speech (Cameron-Faulkner, Lieven and Tomasello 2003); furthermore, children's speech is characterized by a gradual increase in Mean Length of Utterance (MLU: calculated as morphemes-per-utterance) with maturation (Brown 1973: p270-5), and it has been shown that MLU is a better predictor of comprehension of syntactically complex adult utterances than chronological age (de Villiers and de Villiers, 1973; see also Elman 1993 for further computational work on the payoff of "starting small"). This not only improved overall performance substantially, but also achieved a much-needed gain in stability.

5 Conclusion

We have seen that, by building in additional cognitively realistic assumptions, the overall performance of DDOP, both in terms of average parse quality, and diachronic stability is considerably enhanced; this in itself should be taken as a *prima facie* indication of DDOP's promise as a platform for developmental cognitive modeling. We also note with interest a possible resonance between the current model and neural models of development as evolution at the ontogenetic scale: Edelman's (1987) "Neural Darwinism", or more saliently Fernando, Karishma and Szathmáry's (2008) work on neural-developmental evolution with true replication. Future research in DDOP will investigate the role of the global properties of the exemplar base in determining

the evolutionary dynamics in relation to which exemplars compete and die, and the success or failure of Data-Oriented models of learning and cognition.

References

- Batali, J. 1994. "Innate biases and critical periods: Combining evolution and learning in the acquisition of syntax". In R. Brooks and P. Maes, editors, *Artificial Life IV*. 160-171. Cambridge, MA: MIT Press.
- Batterman, R., 2005. "Critical Phenomena and Breaking Drops: Infinite Idealizations in Physics", *Studies In History and Philosophy of Modern Physics*, 36:225-244.
- Bod, R., 1992. A Computational Model of Language Performance: Data-Oriented Parsing. In *Proc. COLING 1992*, 855--859. Stroudsburg: Association for Computational Linguistics.
- Bod, R. (1998). *Beyond Grammar: An Experience-Based Theory of Language*. Stanford: CSLI Publications.
- Bod, R., 2006a. Exemplar-Based Syntax: How to Get Productivity from Examples. *The Linguistic Review* 23, 291-320.
- Bod, R., 2006b. An All-Subtrees Approach to Unsupervised Parsing. In *Proc. ACL 2006*, 865-872. Stroudsburg: Association for Computational Linguistics.
- Bod, R., 2007a. Is the End of Supervised Parsing in Sight? In *Proc. ACL 2007*, 400-407. Stroudsburg: Association for Computational Linguistics.
- Bod, R., 2007b. A Linguistic Investigation into UDOP. In *Proc. Workshop on Cognitive Aspects of Computational Language Acquisition ACL 2007*, 1-8. Stroudsburg: Association for Computational Linguistics.
- Bod, R., 2008. From Exemplar to Grammar: A Probabilistic Analogy-based Model of Language Learning. Accepted for publication in *Cognitive Science*.
- Bod, R., R. Scha and K. Sima'an, (eds.), 2003. *Data-Oriented Parsing*. Stanford, CA: Centre for the Study of Language and Information.
- Borensztajn, G., Zuidema, W., & Bod, R., 2008. Children's grammars grow more abstract with age: evidence from an automatic procedure for identifying the productive units of language. In *Proc. CogSci 2008*, 47-51. Austin: Cognitive Science Society.
- Brown, R., 1973. *A First Language: The Early Stages*. Cambridge, Mass.: Harvard University Press.
- Cameron-Faulkner, T., E. Lieven and M. Tomasello, 2003. A construction-based analysis of child directed speech. *Cognitive Science* 23, 843-873.
- Dawkins, R., 1976. *The Selfish Gene*, Oxford: Oxford University Press.
- De Villiers, J. and P. de Villiers, 1973. Development of the Use of Word Order in Comprehension, *Journal of Psycholinguistic Research*, 2; 331-341.
- DeWitt, T. and S. Scheiner, 2004a. *Phenotypic Plasticity: Functional and Conceptual Approaches*. Oxford: Oxford University Press.
- DeWitt, T. and S. Scheiner, 2004b. "Phenotypic Variation from Single Genotypes: A Primer", in DeWitt and Scheiner 2004a, 1-9.
- Edelman, G. 1987. *The Theory of Neuronal Group Selection*. New York NY: Basic Books.
- Elman, J. 1993. "Learning and development in neural networks: The importance of starting small". *Cognition*, 48(1):71-99.
- Fernando C, K. Karishma and E. Szathmáry. 2008. "Copying and Evolution of Neuronal Topology". *PLoS ONE* 3(11): e3775.
- Goodman, J., 1996. Efficient algorithms for parsing the DOP model. In *Proc. EMNLP 1996*, 143-152. Stroudsburg: Association for Computational Linguistics
- Goodman, J., 2003. "Efficient parsing of DOP with PCFG-reductions". In Bod, Scha and Sima'an 2003, 125-146.
- Johnson, M. 1998. "The DOP Estimation Method is Biased and Inconsistent", *Computational Linguistics*, 28:71-76.
- Kirby, S. 1999. "Learning, bottlenecks, and infinity: a working model of the evolution of syntactic communication", *Proc. AISB'99 Symposium on Imitation in Animals and Artifacts*, 55-63. Society of the Study of Artificial Intelligence and the Simulation of Behaviour.
- Manning, C. and H. Schütze, 1999. *Foundations of Statistical Natural Language Processing*. Cambridge, Mass; The MIT Press.
- Saffran, J., R. Aslin, and E. Newport. 1996. "Statistical learning by 8-month-old infants". *Science*, 274:1926-1928.
- Scha, R., 1990. "Taaltheorie en Taaltechnologie: Competence en Performance", in Q. de Kort and G. Leerdam (eds.), *Computertoepassingen in de Neerlandistiek*, Almere: Landelijke Vereniging van Neerlandici.
- West-Eberhard, M., 2003. *Developmental Plasticity and Evolution*, Oxford: Oxford University Press.
- Williams, G., 1966. *Adaptation and Natural Selection*. Princeton: Princeton University Press.
- Zollmann, A., and K. Sima'an, 2005. A consistent and efficient estimator for data-oriented parsing.

ing *Journal of Automata, Languages and Combinatorics* 10, 367-388.