GRAEL: an agent-based evolutionary computing approach for natural language grammar development

Guy De Pauw
CNTS - Language Technology Group
UIA - University of Antwerp
Antwerp - Belgium
guy.depauw@ua.ac.be

Abstract

This paper describes an agent-based evolutionary computing technique called GRAEL (Grammar Evolution), that is able to perform different natural language grammar optimization and induction tasks. Two different instantiations of the GRAEL-environment are described in this paper: in GRAEL-1 large annotated corpora are used to bootstrap grammatical structure in a society of agents, who engage in a series of communicative attempts, during which they redistribute grammatical information to reflect useful statistics for the task of parsing. In GRAEL-2, agents are allowed to mutate grammatical information, effectively implementing grammar rule discovery in a practical context. A combination of both GRAEL-1 and GRAEL-2 can be shown to provide an interesting all-round optimization for corpus-induced grammars.

1 Introduction

Evolutionary computing has seen many interesting applications on a broad range of research domains over the years. Its ability to overcome the problem of local maxima in finding a solution to a particular problem, by recombining and mutating individuals in a society of possible solutions, has made it an attractive technique for problems involving large, complicated and non-linearly divisible search spaces. The evolutionary computing paradigm has however always seemed reluctant to deal with natural language syntax, probably because it is essentially a recursive, non-propositional system, dealing with complex issues such as long-distance dependencies and constraints. This has made it difficult to incorporate it in typically propositional evolutionary systems such as genetic algorithms.

A limited amount of GA-related syntactic research has focused on linguistic data [Smith and Witten, 1996; Wyard, 1991; Antonisse, 1991; Araujo, 2002], but none of these systems are suited to a generic (treebank) grammar optimization task, mainly because the grammatical formalism and evolutionary processes underlying these systems are designed to fit a particular task, such as information retrieval [Losee, 1995]. Other work on syntax in the evolutionary computing paradigm has either been involved in studying the origins of grammar in a computational context [Batali, 2002; Kirby, 2001] or the co-ordinated co-evolution of grammatical principles [Briscoe, 1998]. Yet so far, little or no progress has been achieved in evaluating evolutionary computing as a tool for the induction or optimization of data-driven parsing techniques.

The GRAEL[1] environment provides a suitable framework for the induction and optimization of any type of grammar for natural language in an evolutionary setting. In this paper we hope to provide an overview of GRAEL as a grammar optimization and induction technique. We will first outline the basic architecture of the GRAEL environment in Section 2 on the basis of a toy example. Next, we introduce GRAEL-1 (Section 3) as a grammar optimization technique that can enhance corpus-induce grammars. By adding an element of mutation in GRAEL-2 we implement a method to extend the coverage of a corpus-induced grammar. We will also describe a combination of both GRAEL-1 and GRAEL-2 which can be shown to provide an interesting all-round optimization technique for corpus-induced grammars.

2 GRAEL - Grammar Evolution

A typical GRAEL society contains a population of agents in a virtual environment. Each of these agents holds a number of structures that allows it to produce sentences as well as induce a probabilistic grammar to analyze other agents' sentences. During an extended series of error-driven inter-agent interactions, these grammars are updated over time. While the evolutionary computing approach of GRAEL is able to define the quality of the grammars that are developed over time, the agent-based aspect of GRAEL ensures that the grammar optimization is grounded in the practical task of parsing itself. From an engineering point of view, GRAEL provides a general framework for grammar optimization and induction, but from a more theoretical point of view, GRAEL can also help us to understand the dynamics of grammar emergence and evolution over time.

In the data-driven GRAEL experiments described in this paper, the grammatical knowledge of the agents in the society is bootstrapped by using an annotated natural language corpus [Marcus *et al.*, 1993]. At the onset of such a data-driven

[1] **GRA**mmar **E**vo**L**ution

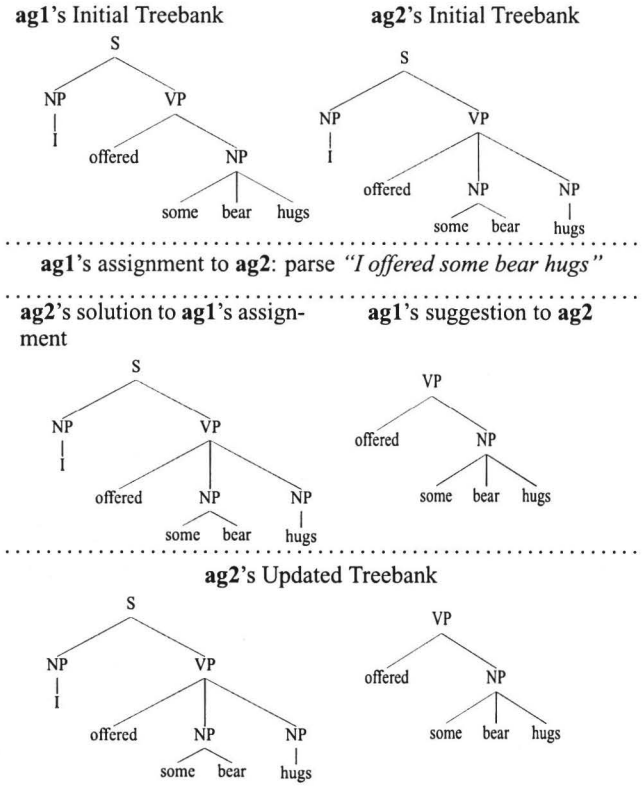

Figure 1: A GRAEL language game

GRAEL society, the syntactic structures of a treebank are randomly distributed over the agents, so that each agent holds a number of tree-structures in memory. These structures enable them to generate sentences, as well as provide grammars that allow them to analyze other agents' sentences.

The actual interaction between agents is implemented in *language games*: an agent (**ag1**) presents a sentence to another agent (**ag2**). If **ag2** is able to correctly parse **ag1**'s sentence, the communication is successful. If on the other hand, **ag2** is lacking the proper grammatical information to parse the sentence correctly, **ag1** shares the necessary information for **ag2** to arrive at the proper solution.

Figure 1 displays a toy example of such a language game. In this example, a "treebank" of two structures has been distributed over a society of two agents. The two agents engage in a language game, in which **ag1** presents the sentence "I offered some bear hugs" to **ag2** for parsing. At this point in time, **ag2**'s grammar does not contain the proper grammatical information to interpret this sentence the way **ag1** intended and so **ag2** will return an incorrect parse, even though it is consistent with its own grammar.

Consequently, **ag1** will try and help **ag2** out by revealing the minimal correct substructure of the correct parse that should enable **ag2** to arrive at a better solution. This information is incorporated in **ag2**'s grammar, who will try to parse the sentence again with the updated knowledge. When **ag2** is able to provide the correct analysis (or is not able to after a certain number of attempts) either **ag1**'s next sentence will be parsed, or two other agents in the GRAEL society will be randomly selected to play a language game.

These interactions, which introduce a concept of error-driven knowledge sharing, extend the agents' grammars fast, so that the datasets can grow very large in a short period of time. A generation-based GRAEL society can be used to allow the society to purge itself of bad agents and build new generations of good parser agents, who contain a fortuitous distribution of grammatical knowledge. This involves the use of fitness functions that can distinguish good agents from bad ones. For a full overview of all the evolutionary parameters in the GRAEL environment, many of which have a significant impact on processing, we would like to refer to [De Pauw, 2002]. In this paper, we will describe the most relevant subset of experiments that allows us to evaluate GRAEL as a grammar induction and optimization technique.

3 GRAEL-1: Probabilistic Grammar Optimization

Historically, most syntactic parsers for natural language have made use of hand-written grammars, consisting of a laboriously crafted set of grammar rules. But in recent years, a lot of research efforts employ annotated corpora to automatically induce grammars [Bod, 1998; Collins, 1999; De Pauw, 2000]. Yet, data-analysis of the output generated by these parsers still brings to light fundamental limitations to these corpus-based methods [Klein and Manning, 2001]. Even though generally providing a much broader coverage than hand-built grammars, corpus-induced grammars may still not hold enough grammatical information to provide parses for a large number of sentences, as some rules that are needed to generate the correct tree-structures are not induced from the original corpus (cf. Section 4).

But even if there were such a thing as a full-coverage corpus-induced grammar, performance would still be limited by the probabilistic weights attributed to its rules. A typical data-driven parser provides a huge collection of possible parses for any given sentence. Fortunately, we can also induce useful statistics from the annotated corpus that provides a way to order these parse forests to express a preference for a particular parse. These statistics go a long way in providing well ordered parse forests, but in many other cases, it can be observed that the ranking of the parse forest is sometimes counter-intuitive in that correct constructs are often overtaken by obviously erroneous, but highly frequent structures.

This can easily be explained by the inherent nature of corpus-based grammars: the initial probabilistic values attached to the grammar rules induced from the annotated data are equal to their relative frequency in the corpus. It might be the case however that, even though they are directly induced from the annotated corpus, the probabilities of these rules are not suited to the disambiguation task as yet. It may therefore be useful to have the parser use the grammar to *practice* the task of parsing and adjust the probabilistic weights of particular structures according to these test cases. We then need to consider the initial grammar as basic raw material in need of optimization, as it is merely a reflection of the original data set and is not yet optimized for the task of parsing (unseen)

sentences itself.

Typical methods of probabilistic grammar optimization include, among others, bagging and boosting [Henderson and Brill, 2000; Collins, 2000], re-estimation of the constituents probabilities [Goodman, 1998; Charniak, 2000] and including extra information sources [Belz, 2001; Collins, 1999]. But we propose an agent-based evolutionary computing method to resolve this issue. Grammar optimization using a GRAEL environment is in this vein related to the aforementioned bagging approach to grammar optimization, albeit with some notable differences. By distributing the knowledge over a group of agents and having them interact with each other, we basically create a multiple-route model for probabilistic grammar optimization. Grammatical structures extracted from the training corpus, will be present in different quantities and variations throughout the GRAEL society. While the agents interact with each other and in effect practice the task on each other's grammar, a varied range of probabilistic grammars are optimized in a situation that directly relates to the task at hand. The evolutionary aspects of the system make sure that, while marginally useful grammatical information is down-toned, common constructs are enforced, providing a better balanced model for statistical parsing.

The way GRAEL accomplishes a re-distribution of the original probabilistic values is by using the default GRAEL architecture described in Section 2. This type of error-driven learning makes sure that mistakes are being dealt with by transferring difficult grammatical constructs, thereby increasing their probabilistic value in the other agent's grammar. This probabilistic adjustment will be taken into account during subsequent parsing attempts by this agent, hopefully triggering the correct grammatical structure in the future.

3.1 Experimental Setup

The overall setup of the GRAEL experiments is displayed in Figure 2. Baseline accuracy is measured by directly inducing a grammar from the training set to power Parser 1, which disambiguates the test set. This grammar takes on the form of a PMPG as outlined in [De Pauw, 2000] (cf. infra). The same training set is also randomly and equally distributed over a number of agents in the GRAEL society, who will consequently engage in a number of *language games*. At some point, established by the halting procedure (cf. infra), the society is halted and the fittest agent is selected from the society. This agent effectively constitutes a redistributed and probabilistically optimized grammar, which can be used to power Parser 2. GRAEL-1 accuracy is achieved by having this parser disambiguate the same test set.

We used two data sets from the Penn Treebank [Marcus *et al.*, 1993]. The main batch of experiments was conducted on the small, homogeneous ATIS-corpus, which consists of a collection of annotated sentences recorded by a spoken-dialogue system. The larger Wall Street Journal Corpus (henceforth WSJ), a collection of annotated newspaper articles, was used to test the system on a larger scale corpus. The common division between training set (Section 02-21) and test set (Section 23) was used. Semantically oriented flags and numeric flags indicating internal relations were removed to allow for more streamlined syntactic processing.

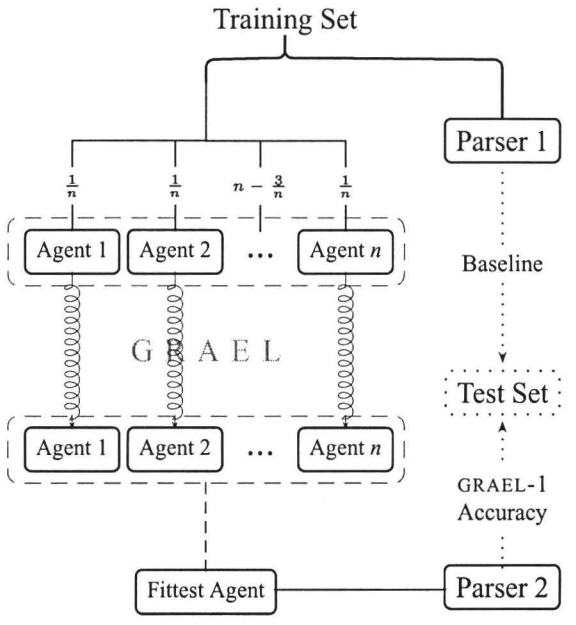

Figure 2: Comparing parsers: baseline parser and GRAEL

For syntactic processing, the agents use the parsing system PMPG described in [De Pauw, 2000], which integrates a CKY parser [Chappelier and Rajman, 1998] and a parse forest ranking scheme that employs probabilistic information as well as a memory-based operator to maximize for each parse the number of nodes that can be retrieved from memory. A PMPG takes the form of a simple PCFG-type grammar, enriched with numerical indices that encode contextual information previously observed in a treebank. This memory-based instantiation of Data-Oriented Parsing [Bod, 1998] ensures that larger syntactic structures are used as the basis for parsing, with a minimal loss of computational efficiency over regular PCFGs. The PMPG approach can therefore be considered to introduce a psycho-linguistically relevant memory-based operator in the parsing process.

The full experimental run varied society sizes, generation methods, fitness functions and halting procedures [De Pauw, 2002]. The subset of experiments described in this paper employed the sexual procreation method to introduce new generations by combining the grammars of two fit agents in the society to create new generations of parser agents. The fitness of an agent is defined by recording a weighted average of the F-score during inter-agent communication (also see Figure 3) and the F-score of the agent's parser on a held-out validation set. This information was also used to try and halt the society at a global maximum and select the fittest agent from the society. For computational reasons, the experiments on the WSJ-corpus were limited to two different population sizes and used an approximation of GRAEL that can deal with large datasets in a reasonable amount of time. The test set was not used in any way during actual GRAEL-processing in agreement with blind-testing procedures.

	ATIS		WSJ	
	Exact Match	$F_{\beta=1}$-score	Exact Match	$F_{\beta=1}$-score
Baseline	70.7	89.3	16.0	80.5
GRAEL (5)	72.4	90.9	—	—
GRAEL (10)	77.6	92.1	—	—
GRAEL (20)	77.6	92.1	—	—
GRAEL (50)	75.9	92.2	22.2	80.7
GRAEL (100)	75.9	92.0	22.8	81.1
bagging (15)	75.9	91.8	20.0	81.4
boosting (15)	75.9	91.7	21.0	81.0

Table 1: Baseline vs GRAEL-1 results

3.2 Results

Table 1 displays the exact match accuracy and F-scores for the baseline model, a standard PMPG parser using a grammar induced from the training set (cf. Figure 2). It also displays scores of the GRAEL system for different population sizes. We notice a significant gain for all GRAEL models over the baseline model on the ATIS corpus, but increasing population size over 20 agents seems to decrease exact match accuracy on the ATIS corpus. Likewise, the small society of 5 agents achieves only a very limited improvement over the baseline method. Data analysis showed that the best moment to halt the society and select the fittest agent from the society, is a relatively brief period right before actual convergence sets and grammars throughout the society are starting to resemble each other more closely. The size of the the society seems to be the determining factor controlling the duration of this period.

Preliminary tests on a subset of the WSJ corpus had shown that society sizes of 20 agents and less to be unsuitable for a large-scale corpus, again ending up in a harmful premature convergence. The gain achieved by the GRAEL society is less spectacular than on the ATIS corpus, but it is still statistically significant. Larger society sizes and full GRAEL processing on the WSJ corpus should achieve a more significant gain.

The experiments do show however, that GRAEL-1 is indeed an interesting method for probabilistic grammar redistribution and optimization. Data analysis shows that many of the counter-intuitive parse forest orderings that were apparent in the baseline model, are being resolved after GRAEL-1 processing. It is also interesting to point out that we are achieving an error reduction rate of more than 26% over the baseline method, without introducing any new grammatical information in the society, but solely by redistributing what is already there. These experimental results indicate that annotated data can indeed be considered as raw material that can be optimized for the practical use of parsing unseen data.

[De Pauw, 2002] also describes experiments that directly compare GRAEL to the similar methods of bagging and boosting [Henderson and Brill, 2000], which are summarized in the bottom two lines of Table 1. Bagging and boosting were shown to obtain significantly lower accuracy figures on almost all accounts. Only the F-score for the bagging experiment exceeded that of the optimal GRAEL configuration, but this can be attributed to the fact that an approximation of GRAEL was used for full processing on the extensive WSJ dataset.

4 GRAEL-2: Grammar Rule Discovery

The functionality of GRAEL-1 can be extended by only applying minor alterations to the GRAEL system. With GRAEL-2 we wish to provide a grammar rule discovery method which can deal with the problem of *grammar sparseness*. Handwritten and corpus-induced grammars alike have to deal with the fundamental issue of coverage. [Collins, 1999] for example reports that when using sections 2-21 of the WSJ-corpus as a training set and section 23 as a test set, 17.1% of the sentences in the test set require a rule not seen in the training set. Even for a large corpus such as the WSJ, sparse grammar is indeed a serious accuracy bottleneck.

It would therefore be useful to have a method that can take a corpus-induced grammar and extend it by generating new rules. But doing so in a blind manner, would provide huge, over-generating grammars, containing many nonsensical rules. The GRAEL-2 system described in this section, involves a *distributed* approach to this type of grammar rule discovery. The original (sparse) grammar is distributed among a group of agents, who can randomly *mutate* the grammatical structures they hold. The new grammatical information they create is tried and tested by interacting with each other. The neo-darwinist aspect of this evolutionary system will make sure that any useful mutated grammatical information is retained throughout the population, while noise is filtered out over time. This method provides a way to create new grammatical structures previously unavailable in the corpus, while at the same time evaluating them in a practical context, without the need for an external information source.

To accomplish this, we need to implement some minor alterations to the GRAEL-1 system. The most important adjustment occurs during the language games. We refer back to the toy example of the language game in Figure 1 to the point where **ag1** suggests the minimal correct substructure to **ag2**. In GRAEL-1 this step introduced a form of error-driven learning, making sure that the probabilistic value of this grammatical structure is increased. The functionality of GRAEL-2 however is different: in this step, we assume that there is a noisy channel between **ag1** and **ag2** which may cause **ag2** to misunderstand **ag1**'s structure. Small mutations on different levels of the substructure may occur, such as the deletion, addition and replacement of nodes. This effectively introduces previously unseen grammatical data in the GRAEL society, which

	ATIS		WSJ	
	$F_{\beta=1}$	Ex. Match	$F_{\beta=1}$	Ex. Match
Baseline	69.8	0	80.5	16
grael-1	73.8	0	81.4	22.8
grael-2	83.0	7.2	76.5	19.3
grael2+1	85.7	11.3	81.6	23.4

Table 2: Baseline vs GRAEL-1 vs GRAEL-2 vs GRAEL2+1 Results

will consequently be optimized over time.

Preliminary experiments however showed that this does not work as such, since the newly created structures were largely being ignored in favor of the gold-standard corpus structures. We therefore implemented another alteration to the GRAEL-1 system. Instead of just presenting the tree-structure originally assigned by the training set, we now require **ag1** to parse the string-only sentence using the grammar acquired during language games, replacing the tree-structure from the training set, with a possibly different tree-structure that incorporates some of the mutated information. This alteration makes sure that the mutated grammatical structures are actively being used, so that their usefulness as grammatical constructs can be measured in a practical context.

Experimental Setup and Results

The setup for the GRAEL-2 experiments is the same as for the GRAEL-1 experiments (cf. Figure 2). To test the grammar-rule discovery capabilities of GRAEL-2 we have compiled a special worst-case scenario test set for the ATIS corpus, consisting of the 97 sentences in the ATIS that require a grammar rule that cannot be induced from the training set. For the WSJ-experiment the normal test set was used. A 20-agent and a 100-agent society were respectively used for the ATIS and WSJ experiments.

The baseline and GRAEL-1 methods for the ATIS experiments trivially have an exact match accuracy of 0%, which also has a negative effect on the F-score (Table 2). GRAEL-2 is indeed able to improve on this significantly, proving that it is indeed an effective grammar rule discovery method. Data-analysis shows however that it has lost the beneficial probabilistic optimization effect of GRAEL-1.

We therefore performed another experiment, in which we turned the GRAEL-2 society into a GRAEL-1 society after the former's halting point. In other words: we take a society of agents using mutated information and consequently apply GRAEL-1's probabilistic redistribution properties on these grammars. Figure 3 shows the course of the GRAEL2+1 experiment. In this figure we see the F-scores recorded during inter-agent communicative attempts. After an almost linear increase during GRAEL-2 processing, the society is halted after an extended period of convergence. Next, GRAEL-1 processing resumes, which negatively affects F-scores for a brief period of time, until the society reconverges. Even though the F-scores do not seem to improve over those observed before the transition to GRAEL-1, Table 2 shows that the fittest agent selected from the society after the transition performs better on the held-out test set. The results on the WSJ-corpus are also

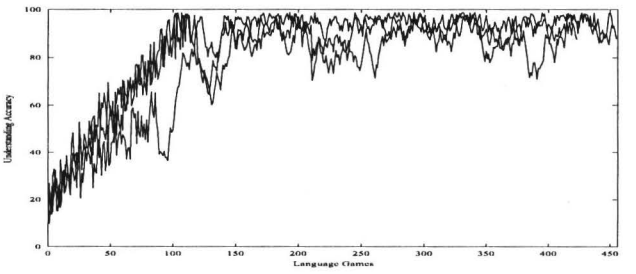

Figure 3: GRAEL2+1 Experiment - F-scores during language games

interesting in this respect. GRAEL-1 outperforms GRAEL-2 on this data set, but the combination of the two seems to be quite beneficial for parsing accuracy.

Evaluating a grammar rule discovery method poses an empirical problem in that it can never be clear what grammar rules are missing until we actually need them. The test set we compiled to perform the GRAEL-2 ATIS-experiments goes a long way in providing a touchstone to see how well GRAEL-2 performs as a supervised grammar induction method. And results indicate it performs quite well: mutated information becomes available that is able to create parses for difficult constructions, while the number of structures that constitute noise is limited and is attributed a small enough portion of the probability mass as not to stand in the way of actual useful mutated structures.

5 Concluding Remarks

This paper has presented one of the first research efforts that introduces agent-based evolutionary computing as a machine learning method for data-driven grammar optimization and induction. In recent years, many researchers have employed ensemble methods to overcome any negative bias their training data might impose on their classifiers. It is indeed important to view (annotated) data, not as an optimally distributed set of examples but as raw material that needs to be pre-processed before it can be used by a machine learning classifier. The bagging and boosting approach for instance, tries to create resamplings of the original data, to overcome the local maxima the data might restrict the classifier to, but we believe GRAEL adds an extra dimension to the task: by splicing the data and incorporating it in a society of communicating agents, we allow for the parallel development of several grammars at once, enhanced in a practical context that mirrors the goal itself: parsing unseen data.

We described two instantiations of the GRAEL environment. The basic GRAEL-1 system aims to provide a beneficial re-distribution of the probability mass of a probabilistic grammar. By using a form of error-driven learning in the course of language games between agents, probabilistic values are adjusted in a practical context. This optimizes the grammars for the task of parsing data, rather than reflecting the probability mass of the initial data set. This method favorably compares to established grammar optimization methods like bagging and boosting.

By adding an element of mutation to the concept of

GRAEL-1 we were able to extend its functionality and experiment on GRAEL-2 as a grammar rule discovery method. The results showed that GRAEL-2 could produce a broader-coverage grammar, but that GRAEL-1's ability to optimize the distribution of the probability mass of a grammar was counteracted. A grammar obtained from a GRAEL-2 society is therefore unsuited to be directly applied to parsing. But combining it again with a GRAEL-1 society however, goes a long way in resolving this issue, providing a grammar that has a broader coverage, as well as a better tuned probability mass distribution over the structures contained therein. We therefore believe the combination of GRAEL-2 and GRAEL-1 to be an interesting optimization toolkit for any given grammar, and corpus-induced grammars in particular. It can achieve a significant optimization over the baseline method, without using an external information source, simply on the basis of knowledge transfer and mutation in a practical and evolutionary context.

Acknowledgments

The research described in this paper was financed by the FWO (Fund for Scientific Research).

References

[Antonisse, 1991] H. James Antonisse. A grammar-based genetic algorithm. In Gregory J. E. Rawlings, editor, *Foundations of genetic algorithms*, pages 193–204. Morgan Kaufmann, San Mateo, 1991.

[Araujo, 2002] Lourdes Araujo. A parallel evolutionary algorithm for stochastic natural language parsing. In *Proceedings of The Seventh International Conference on Parallel Problem Solving From Nature*, pages 700–709, Granada, Spain, 2002.

[Batali, 2002] J. Batali. The negotiation and acquisition of recursive grammars as a result of competition among exemplars. In Ted Briscoe, editor, *Linguistic Evolution through Language Acquisition: Formal and Computational Models*, chapter 5. Cambridge University Press, 2002.

[Belz, 2001] A. Belz. Optimisation of corpus-derived probabilistic grammars. In *Proceedings of Corpus Linguistics 2001*, pages 46–57, Lancaster University, UK, 2001.

[Bod, 1998] R. Bod. *Beyond Grammar—An Experience-Based Theory of Language*. Cambridge University Press, Cambridge, England, 1998.

[Briscoe, 1998] E. Briscoe. Language as a complex adaptive system: co-evolution of language and of the language acquisition device. In *Proceedings of the 8th Meeting of Comp. Linguistics in the Netherlands*, pages pp. 3–40., Amsterdam, 1998. Rodopi.

[Chappelier and Rajman, 1998] J.-C. Chappelier and M. Rajman. A generalized cyk algorithm for parsing stochastic cfg. In *Proceedings of Tabulation in Parsing and Deduction (TAPD'98)*, pages 133–137, Paris (FRANCE), 1998.

[Charniak, 2000] Eugene Charniak. A maximum-entropy-inspired parser. In *Proceedings of NAACL'00*, pages 132–139, Seattle, USA, 2000.

[Collins, 1999] M. Collins. *Head-driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania, Pennsylvania, USA, 1999.

[Collins, 2000] Michael Collins. Discriminative reranking for natural language parsing. In *Proc. 17th International Conf. on Machine Learning*, pages 175–182. Morgan Kaufmann, San Francisco, CA, 2000.

[De Pauw, 2000] G. De Pauw. Aspects of pattern-matching in dop. In *Proceedings of the 18th International Conference on Computational Linguistics*, pages 236–242, 2000.

[De Pauw, 2002] G. De Pauw. *An Agent-Based Evolutionary Computing Approach to Memory-Based Syntactic Parsing of Natural Language.* PhD thesis, University of Antwerp, Antwerp, Belgium, 2002.

[Goodman, 1998] J. Goodman. *Parsing Inside-Out*. PhD thesis, Harvard University, Massachusetts, USA, 1998.

[Henderson and Brill, 2000] J. Henderson and E. Brill. Bagging and boosting a treebank parser. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-2000)*, pages 34–41, 2000.

[Kirby, 2001] S. Kirby. Spontaneous evolution of linguistic structure: an iterated learning model of the emergence of regularity and irregularity. *IEEE Transactions on Evolutionary Computation*, 5(2):102–110, 2001.

[Klein and Manning, 2001] Dan Klein and Christopher D. Manning. Parsing with treebank grammars: Empirical bounds, theoretical models, and the structure of the penn treebank. In *Prodeedings of ACL-EACL 2001*, pages 330–337, 2001.

[Losee, 1995] R.M. Losee. Learning syntactic rules and tags with genetic algorithms for information retrieval and filtering: An empirical basis for grammatical rules. *Information Processing and Management*, 32(2):185–197, 1995.

[Marcus *et al.*, 1993] M. P. Marcus, B. Santorini, and M.A. Marcinkiewicz. Building a large annotated corpus of english: the penn treebank. *Computational linguistics*, 19:313–330, 1993. Reprinted in Susan Armstrong, ed. 1994, *Using large corpora*, Cambridge, MA: MIT Press, 273–290.

[Smith and Witten, 1996] T. C. Smith and I. H. Witten. Learning language using genetic algorithms. In Stefan Wermter, Ellen Riloff, and Gabriele Scheler, editors, *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*, volume 1040 of *LNAI*, pages 132–145. Springer Verlag, Berlin, 1996.

[Wyard, 1991] P. Wyard. Context-free grammar induction using genetic algorithms. In R. Belew and L.B. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 514–518, San Mateo, 1991. ICGA, Morgan Kaufmann.