A BROAD-COVERAGE, REPRESENTATIONALLY MINIMAL LFG PARSER: CHUNKS AND F-STRUCTURES ARE SUFFICIENT

Gerold Schneider

Institute of Computational Linguistics, University of Zurich
English Department of the University of Zurich
Proceedings of the LFG05 Conference

University of Bergen

Miriam Butt and Tracy Holloway King (Editors)

2005

CSLI Publications

http://csli-publications.stanford.edu/

Abstract

A major reason why LFG employs c-structure is because it is context-free. According to Tree-Adjoining Grammar (TAG), the only context-sensitive operation that is needed to express natural language is Adjoining, from which LFG functional uncertainty has been shown to follow. Functional uncertainty, which is expressed on the level of f-structure, would then be the only extension needed to an otherwise context-free processing of natural language. We suggest that if f-structures can be derived context-freely, full-fledged c-structures are not strictly needed in LFG, and that chunks and dependencies may be sufficient for a formal grammar theory. In order to substantiate this claim, we combine a projection of f-structures from chunks model with statistical techniques and present a parser that outputs LFG f-structure like representations. The parser is representationally minimal, deep-linguistic, robust, and fast, and has been evaluated and applied. The parser addresses context-sensitive constructions by treating the vast majority of long-distance dependencies by approximation with finite-state patterns, by post-processing, and by LFG functional uncertainty.

1 Introduction

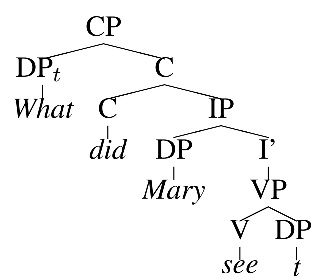
In this paper we argue that full-flegded c-structures can be obviated for the syntactic analysis of natural language. We present and evaluate a broad-coverage statistical parser, Pro3Gres, that substantiates this claim. By reducing grammar complexity (Frank, 2002; Frank, 2004), by reducing parsing complexity to mostly context-free parsing and finite-state based chunking (Cahill et al., 2004; Schneider, 2003; Schneider, 2004), by bridging the gap between language engineering and formal grammar (Kaplan et al., 2004) by aiming for a representationally minimal theory (Jurafsky, 1996) we argue that chunks and dependencies (Abney, 1995; Frank, 2003) may be sufficient for a gormal grammar theory.

Two major factors that make broad-coverage parsing hard are (1) long-distance dependencies, as they break c-structure context-freeness, and (2) natural language ambiguity, which leads to immense search spaces during the parsing operation. We discuss long-distance dependencies in section 2 and ambiguity resolution in section 3.

1.1 Long-distance Dependencies

Long-distance dependencies as f-structure level mechanism The original LFG treatment of long-distance dependencies (Kaplan and Bresnan, 1982) used empty c-structure constituents, traces. For example, the relation between the DP node dominating *what* and the DP node dominating its trace *t* ensures that the wh-phrase *what* is both the FOCUS and OBJ of the sentence:

c-structure:



f-structure:

$$\begin{bmatrix} \text{PRED} & \text{`see}\left\langle \text{SUBJ , OBJ} \right\rangle\text{'} \\ \text{TENSE} & \text{past} \\ \text{FOCUS} & \begin{bmatrix} \text{PRED} & \text{`what'} \end{bmatrix} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{`Mary'} \end{bmatrix} \\ \text{OBJ} & \end{bmatrix}$$

Subsequently, Kaplan and Zaenen (1989) proposed that long-distance dependencies are best expressed in functional and not phrasal terms. *Functional uncertainty* expresses long-distance dependency on the level of f-structure and obviates the need for trace-like devices in the theory of grammar, which has been described as descriptively more adequate and theoretically less redundant (Dalrymple, Kaplan, and King, 2001). A rule like the one in example 1 establishes two roles for the NP daughter of CP: it is the FOCUS, and it plays the grammatical role defined by the functional uncertainty path COMP* OBJ:

$$
(1) \quad \text{CP} \longrightarrow \underset{\substack{(\uparrow \text{FOCUS}) = \downarrow \\ (\uparrow \text{COMP* OBJ}) = \downarrow}}{\text{DP}} \quad \underset{\uparrow = \downarrow}{\text{C'}}
$$

Constituency and Dependency Considerations of theoretical redundancy and linguistic accuracy can also give rise to questions concerning the necessity for c-structure. The grammar theory of Dependency Grammar (DG) is based on functional, grammar role dependencies in the spirit of LFG f-structure. Bröker, Hahn, and Schacht (1994) refers to DG as an LFG that only knows f-structure. Tesnière (1959)'s original DG concept aims at being a proto-semantic, monostratal, language-independent theory rather than merely a syntactic theory. In LFG terms, he challenged the need for c-structure. His view is to parse surface text (*ordre linéaire*) directly to f-structure (*ordre structurale*) in which word order plays no primary role, but may of course help disambiguate as a secondary role, for example by preferring projectivity. A theory that does not constrain dependency directions and allows non-projectivity (which is equivalent to using structure-sharing or movement) can express the same structures as constituency (Covington, 1994; Miller, 1999).

Discussions on headedness (Zwicky, 1985; Hudson, 1987), the prevalence of Chomskyan configurationalism and the desire to distinguish between different levels of analysis led to multistratal versions of DG (Mel'čuk, 1988) on the one hand, and influenced important DG based formal grammars, notably LFG and HPSG, on the other hand. LFG is an answer to the question of whether constituency or dependency should be exclusive – by respecting both: on the one hand the constituency-based context-free c-structure, on the other hand a non-configurational f-structure which expresses functional dependencies between constituents.

Parsing Complexity Dependency Grammar in its original conception allows non-projectivity which makes it computationally hard to process. Parsing algorithms able to treat completely unrestricted long-distance dependencies are NP-complete (Neuhaus and Bröker, 1997). In order to make broad-coverage DG parsing tractable, context-sensitivity needs to be maximally restricted. We discuss in section 2 how this can be done by using finite-state long-distance dependency approximations and functional uncertainty. Completely context-free traceless parsing only requires parsing algorithms with O(n^3) complexity (Eisner, 1997), for example CYK (Younger, 1967). From a language-engineering perspective, context-freeness is a major appeal of c-structure. LFG constrains context-sensitivity by using a context-free c-structure backbone and then mapping to non-configurational f-structure. We follow arguments from Tree-Adjoining Grammar (TAG) (Joshi, 1985) to show that functional uncertainty is the only context-sensitive device needed to achieve the expressiveness exhibited by natural language. LFG functional uncertainty has been shown to follow as a corollary from TAG Adjoining (Joshi and Vijay-Shanker, 1989).

Context-free parsing was already recognised as potential candidate for broad-coverage application. When coupled with a probabilistic disambiguation, it turned out to be very successful (Collins, 1999; Charniak, 2000). But these parsers typically produce context-free data as output, trees that do not express long-distance dependencies. Although grammatical function and empty node annotation expressing long-distance dependencies are provided in Treebanks such as the Penn Treebank (Marcus, Santorini, and Marcinkiewicz, 1993), these probabilistic Treebank trained parsers fully or largely ignore them (Collins (1999) Model 2 uses some of the functional labels, and Model 3 some long-distance dependencies). This entails two problems: first, the training cannot profit from valuable annotation data. Second, the extraction of long-distance dependencies (LDD) and the mapping to shallow semantic representations is not always possible from the output of these parsers. This limitation is aggravated by a lack of co-indexation information and parsing errors across an LDD.

Typical formal grammar parser complexity is much higher than the O(n^3) for context-free grammar. The complexity of some formal grammars is still unknown. For Tree-Adjoining Grammars (TAG) it is O(n^7) or O(n^8) depending on the implementation (Eisner, 2000). Sarkar, Xia, and Joshi (2000) state that the theoretical bound of worst time complexity for Head-Driven Phrase Structure Grammar (HPSG) parsing is exponential. From a language engineering perspective, deep-linguistic formal grammars as a whole proved computationally too costly until recently; research thus successfully focused on finite-state based approaches such as chunking or cascaded shallow parsing. Abney (1995) suggests a chunks & dependency model, but his chunks and cascaded parsing model (Abney, 1996) proved more successful.

We discuss in section 2 that most LDDs can be be expressed in a context-free way (Schneider, 2003), and the remaining ones, if we follow TAG argumentation, by functional uncertainty. The vast majority of traces in the Penn Treebank can be treated as local dependencies by (1) using and modeling dedicated finite-state patterns across several levels of constituency subtrees partly leading to dedicated but fully local dependency syntactic relations and by (2) lexicalized post-processing rules. We also discuss that (3) some non-local dependencies are artifacts of configurational grammatical representations. The remaining long-distance dependencies can (4) be modelled with mild context-sensitivity by LFG functional uncertainty.

1.2 Ambiguity resolution

A Probabilistic Beam Search Approach Many approaches including ours profit from statistical data to prune unlikely partial analyses at parse-time, for example with a beam search. Parser performance decreases only marginally while time behaviour improves by at least an order of magnitude if reasonable pruning is used (Brants and Crocker, 2000) and allows us to explain psycholinguistic phenomena (Jurafsky, 1996; Crocker and Brants, 2000). A beam search approach also closes the gap between deterministic parsing (Nivre, 2004) and full parsing. Section 3 introduces our probability model and compares it to (Collins, 1999).

Shallow Chunking and F-Structure Parsing Some approaches (Kaplan et al., 2004; Schneider, 2004) include POS tagging preprocessing to reduce parsing ambiguity. Some systems include chunking preprocessing (Schneider, 2004) as is often used in probabilistic context-free parsing (Collins, 1999). The parser stays as shallow as is possible for each task, using finite-state based techniques for base phrase recognition. Parsing only takes place between the chunks of heads. Such chunks & dependency models can be attributed to Abney (1995). A chunk largely corresponds to a *nucleus* (Tesnière, 1959).

1.3 Related approaches

Recently, thanks to advances in exploiting and integrating statistics, the first deep-linguistic formal grammar based parsers have achieved the coverage and robustness needed to parse large corpora: Riezler et al. (2002) show how a hand-crafted LFG grammar can scale to the Penn Treebank with Maximum Entropy probability models. Hockenmaier and Steedman (2002) acquire a wide-coverage CCG grammar from the Penn Treebank automatically, Burke et al. (2004) an LFG grammar. Miyao, Ninomiya, and Tsujii (2004) semi-automatically acquire a broad-coverage HPSG grammar from the Penn Treebank and describe its efficiency (Tsuruoka and Tsujii, 2004).

Kaplan et al. (2004) compare speed and accuracy of a successful probabilistic context-free parser (Collins, 1999) to a robust LFG system based on (Riezler et al., 2002). They show that the gap between probabilistic context-free parsing and deep-linguistic full LFG parsing can be closed. On a random test set of 560 sentences from the Penn Treebank (4/5th of the PARC700 corpus[1]) their full LFG grammar gives an overall improvement in F-score of 5% over (Collins, 1999) at a parsing time cost factor of 5. They also show that a limited LFG grammar (so called core system) still achieves a considerably higher f-score at a parsing time cost factor of only 1.5: about 200 seconds for Collins (1999) and about 300 seconds for the LFG core system. A conclusion that can be drawn from their and our results is that research in simplifying, restricting and limiting formal grammar expressiveness is bridging the gap between between probabilistic parsing and formal grammar-based parsing.

Another important reason why deep-linguistic formal grammar parsing has become feasible and relatively fast is because long-distance dependencies are being approximated by deterministic or context-free approaches. Johnson (2002) shows that simple pattern-based approaches to obtaining LDDs from context-free probabilistic

[1] `www2.parc.com/istl/groups/nltt/fsbank/`

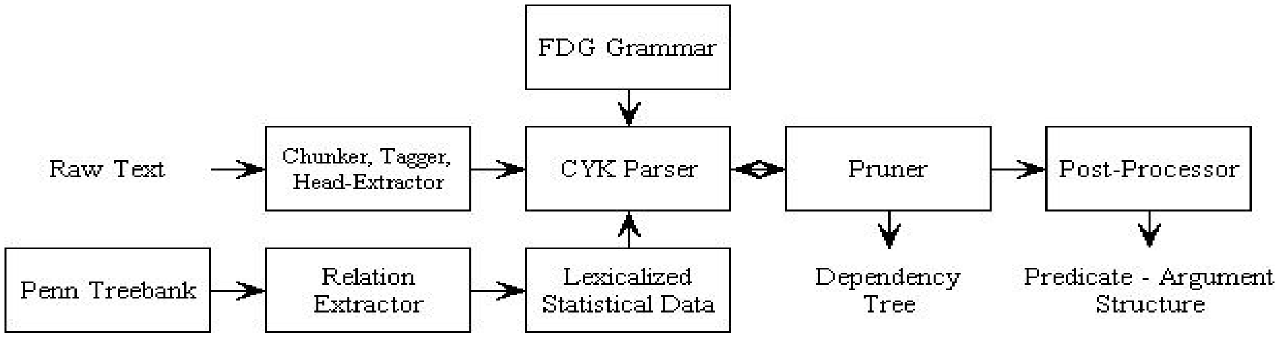


Figure 1: Pro3Gres flowchart

Relation	Label	Example	Relation	Label	Example
verb–subject	subj	*he sleeps*	verb–prep. phrase	pobj	*slept in bed*
verb–first object	obj	*sees it*	noun–prep. phrase	modpp	*draft of paper*
verb–second object	obj2	*gave (her) kisses*	noun–participle	modpart	*report written*
verb–adjunct	adj	*ate yesterday*	verb–complementizer	compl	*to eat apples*
verb–subord. clause	sentobj	*saw (they) came*	noun–preposition	prep	*to the house*

Table 1: The most important dependency types used by the parser

parsers such as Collins (1999) are not successful. Jijkoun (2003) has used similar patterns, but containing LDD information, on the Penn Treebank in order to convert it to a Dependency format. We use a similar approach, assigning dedicated dependency labels to dependencies involving LDDs and statistical post-processing so that deep-linguistic parsing can mostly stay context-free (Schneider, 2003). Burke et al. (2004; Cahill et al. (2004) use a similar approach in LFG.

Frank (2003) suggests a (albeit non-probabililistic) chunks & dependencies model for LFG. Chunks can be freely combined subject to adjacecy and projectivity (contiguity) constraints, which leads to a context-free parsing algorithm. Except for the added book-keeping functional annotations, her parsing algorithm is akin to CYK, which we use.

1.4 Our Parser

We present Pro3Gres, a parser that has been implemented to substantiate our claims. It has a highly modular architecture, shown in figure 1. It has been designed to keep search spaces and parsing complexity low while only taking minimal linguistic compromises (Schneider, 2004) and to be robust for broad-coverage parsing (Schneider, Dowdall, and Rinaldi, 2004). In order to keep parsing complexity as low as possible, aggressive use of shallow techniques and of context-free parsing is made. For low-level syntactic tasks, we use the shallow techniques of tagging and chunking, thus combining shallow and full parsing. We reduce the majority of context-sensitive tasks to context-free tasks by the use of patterns that are deep-linguistic because they are non-local, but shallow because they are fixed. For the few remaining context-sensitive tasks, mild context-sensitivity is sufficient.

We report evaluations of Pro3Gres on the 500 sentence Carroll corpus (Carroll, Minnen, and Briscoe, 1999). Special attention is given to long-distance dependencies and a linguistic analysis of errors. Comparisons to other parsers show that its performance is competitive.

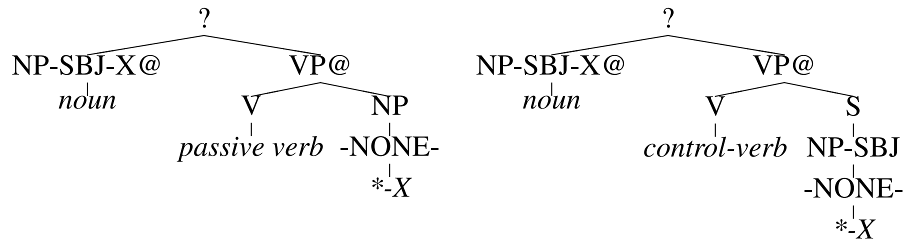


Figure 2: The extaction patterns for passive subjects (left) and subject control (right)

2 Long-distance dependencies

Treating long-distance dependencies is very costly (Neuhaus and Bröker, 1997), as they are context-sensitive. Most statistical Treebank trained parsers thus fully or largely ignore them. Johnson (2002) presents a pattern-matching algorithm for post-processing the Treebank output of such parsers to add empty nodes expressing long-distance dependencies to their parse trees. Encouraging results are reported for perfect parses, but performance drops considerably when using parser output trees.

We have applied structural patterns to the Treebank, where like in perfect parses precision and recall are high, and where in addition functional labels and empty nodes are available, so that patterns similar to Johnson's but relying on functional labels and empty nodes reach precision close to 100%. Unlike in Johnson, patterns for local dependencies are also used; non-local patterns simply stretch across more subtree-levels. We use the extracted lexical counts as lexical frequency training material. Every dependency relation has a group of structural extraction patterns associated with it. This amounts to a partial mapping of the Penn Treebank to Functional DG (Hajič, 1998), similar to the mapping described in Jijkoun (2003). Table 1 gives an overview of the most important dependencies.

The $subj$ relation, for example, has the head of an arbitrarily nested NP with the functional tag *SBJ* as dependent, and the head of an arbitrarily nested VP as head for all active verbs. In passive verbs, however, a movement involving an empty constituent is assumed, which corresponds to the extraction pattern in figure 2 (left), where VP@ is an arbitrarily nested VP, and NP-SBJ-X@ the arbitrarily nested surface subject and X the co-indexed, moved element. Representing passive as movement, however, does not suggest long-distance movement. A close investigation confirms that passive movement is fixed, always local to a verbal domain, inside one clause. It can thus be represented by a single, local dependency.

Similar local restrictions can be formulated for other relations involving empty nodes in the Treebank, for example control structures, which have the extraction pattern shown in figure 2 (right), which are across two (possibly cascaded) clauses.

Grammatical role labels, empty node labels and tree configurations spanning several local subtrees are used as an integral part of some of the patterns. This leads to flatter trees, as typical for DG, which has the advantages that it helps to alleviate sparse data by mapping several nested structures that express the same dependency relation onto one dependency, that fewer decisions are needed at parse-time, which may reduce complexity and the risk of errors (Johnson, 2002), and that the costly overhead for dealing with unbounded dependencies can be largely avoided.

Let us consider the quantitative coverage of these patterns in detail. The ten most frequent types of empty nodes cover more than 60,000 of the approximately 64,000 empty nodes of sections 2-21 of the Penn Treebank. Table 2, reproduced from Johnson (2002) [line numbers and counts from the whole Treebank added], gives an overview.

Empty units, empty complementizers and empty relative pronouns [lines 4,5,9,10] pose no problem for DG as they are optional, non-head material. For example, a complementizer is an optional dependent of the subordinated verb.

Moved clauses [line 6] are mostly PPs or clausal complements of verbs of utterance. Only verbs of utterance

	Antecedent	POS	Label	Count	Description	Example
1	NP	NP	*	22,734	NP trace	*Sam* was seen *
2		NP	*	12,172	NP PRO	* to sleep is nice
3	WHNP	NP	*T*	10,659	WH trace	the woman *who* you saw *T*
(4)			*U*	9,202	Empty units	$ 25 *U*
(5)			0	7,057	Empty complementizers	Sam said 0 Sasha snores
(6)	S	S	*T*	5,035	Moved clauses	*Sam had to go*, Sasha said *T*
7	WHADVP	ADVP	*T*	3,181	WH-trace	Sam explained *how* to leave *T*
(8)		SBAR		2,513	Empty clauses	*Sam had to go*, said Sasha (SBAR)
(9)		WHNP	0	2,139	Empty relative pronouns	the woman 0 we saw
(10)		WHADVP	0	726	Empty relative pronouns	the reason 0 to leave

Table 2: The distribution of the 10 most frequent types of empty node and their antecedents in the Penn Treebank (adapted from Johnson2002). Bracketted lines designate long-distance dependencies that are local in DG

Type	Count	prob-modeled	Treatment
passive subject	6,803	YES	local relation
indexed gerund	4,430	NO	Tesnière translation
control, raise, semi-aux	6,020	YES	post-parsing processing
others / not covered	5,481		
TOTAL	22,734		

Table 3: Coverage of the patterns for the most frequent NP traces [row 1]

allow subject-verb inversion in affirmative clauses [line 8]. In a dependency framework, none of them involve non-local dependencies or empty nodes, [line 6] and [line 8] are covered by rules that allow an inversion of the dependency direction under well-defined conditions.

NP Traces A closer look at NP traces ([line 1] of table 2) reveals that the majority of them are recognized by the grammar, and except for the indexed gerunds, they participate in the probability model. In control, raising and semi-auxiliary constructions, the non-surface semantic arguments, i.e. the subject-verb relation in the subordinate clause, are created based on lexical probabilities at the post-parsing stage, where minimal predicate-argument structures are output. In LFG terms, the probabilistic information on how likely a subordinate verb is to subcategorize for a control subject or object if they are unrealized is furnished by the matrix verb.

Unlike in control, raising and semi-auxiliary constructions, the antecedent of an indexed gerund cannot be established easily. The parser does not try to decide whether the target gerund is an indexed or non-indexed gerund nor does it try to find the identity of the lacking participant in the latter case. This is an important reason why recall values for the subject and object relations are lower than the precision values.

NP PRO As for the 12,172 NP PRO [line 2] in the Treebank, 5,656 are recognized by the $modpart$ pattern (which covers reduced relative clauses), which means they are covered in the probability model. The dedicated $modpart$ relation typically expresses the object function for past participles and the subject function for present participles.[2] A further 3,095 are recognized as non-indexed gerunds. Infinitives and gerunds may act as subjects, which are covered by translations (Tesnière, 1959), although these rules do not participate in the probability model. Many of the structures that are not covered by the extraction patterns and the probability model are still parsed correctly, for example adverbial clauses as unspecified subordinate clauses. Non-indexed adverbial phrases of the verb account for 1,598 NP PRO, non-indexed adverbial phrases of the noun for 268. As the NP is non-indexed, the identity of the lacking argument in the adverbial is unknown anyway, thus no semantic information is lost.

[2] The possible functional ambiguity is not annotated in the Treebank, hence the reduced relative clause is an unindexed empty NP

WH Traces Only 113 of the 10,659 WHNP antecedents in the Penn Treebank [line 3] are actually question pronouns. The vast majority, over 9,000, are relative pronouns. For them, an inversion of the direction of the relation they have to the verb is allowed if the relative pronoun precedes the subject.

Only non-subject WH-question pronouns and support verbs need to be treated as "real" non-local dependencies. In question sentences, before the main parsing is started, the support verb is attached to any lonely participle chunk in the sentence, and the WH-pronoun pre-parses with any verb, as we discuss in the following section.

2.1 Localising Long-Distance Dependencies

LDDs are traditionally grouped into two classes (see e.g. (Pollard and Sag, 1994, p. 157)). In the first class, there is an overt constituent in a nonargument position that can be thought of as strongly associated with (or filling) the gap or trace. An argument is fronted to a non-argument position. In this class we find topicalisations, WH-questions and relative clauses. In the second class there is no overt filler in a nonargument position, instead there is a constituent in an argument position that is interpreted as coreferential with the trace. Functionally speaking, a constituent that is realized once appears more than once as a semantic argument of a predicate. In the second class we find control and raising and *it*-cleft constructions.

For the second class, context-free parsing is sufficient, because the coreference of the argument positions is resolved at the post-processing stage by means of a statistical method. For control and raising, if a subordinate clause is subjectless and is in the infinitive, a decision based on the lexical probability of the superordinate verb or adjective to introduce subject or object control constructs a coreference. Parsing can stay context-free because there is no dependence between syntactic ambiguity and control or relative clause antecedent resolution.

We have discussed that most LDDs of the first class, with the notable exception of non-subject WH questions, can be treated locally in Dependency Grammar. We now discuss the mild context-sensitive approach that Tree-Adjoining Grammar (Joshi, 1985) uses for such WH questions. It has been suggested that mild context-sensitivity is expressive enough for natural language processing (Frank, 2002).

2.1.1 TAG Adjoining and mild context-sensitivity

The TAG formalism (Joshi, 1985; Joshi and Kroch, 1985) has developed a mathematically restrictive formulation of phrase structure grammar. In contrast to the string-rewriting systems of the Chomsky hierarchy, TAG is a system of tree-rewriting. Structural representations are built up from pieces of phrase structure, so-called *elementary trees*, which are taken as atomic. These trees can be combined by using one of two operations: *Substitution* and *Adjoining*.

Substitution Substitution involves the rewriting of a non-terminal node at the frontier of one elementary tree as another elementary tree with the requirement that the rewritten node must have the same label as as the root of the elementary tree that rewrites it. Substitution can be understood as a traditional rewriting operation. Substitution accomplishes effects similar to those of the Merge operation form (Chomsky, 1995): it inserts XPs into the argument positions of syntactic predicates. Crucially, it is a context-free operation: context-free elementary trees combined by substitution only yield context-free structures. An example of Substitution is in fig. 3

Elementary trees are context-free by definition. "Every syntactic dependency is expressed locally within a single elementary tree" (Frank, 2002, p. 22)

Adjoining The Adjoining operation rewrites a non-terminal node anywhere within an elementary tree as another elementary tree. Unlike substitution, which rewrites or expands trees only along the frontier, Adjoining

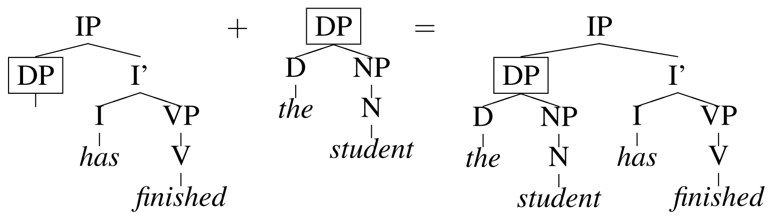


Figure 3: An example of the Substitution operation. The rewritten node is boxed

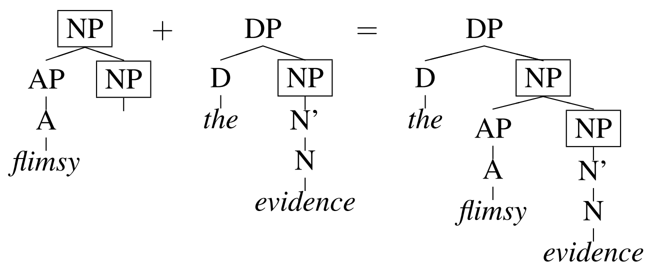


Figure 4: An example of the Adjoining operation. The foot node is boxed

uses a special class of recursive trees, so-called *auxiliary trees*. The root of an auxiliary tree is labeled identically to some node along its frontier, the *foot node*.

Given an auxiliary tree A with foot node X, Adjoining rewrites as A a node N that that is labeled as X in an elementary tree T, and attaches the node that was under N in T at the foot node of the auxiliary tree. Adjoining thus works by rewriting some node of an elementary tree as a recursive piece of structure (the auxiliary tree). An example is in figure 4. Trees that have undergone Adjoining can be subject to subsequent Adjoining operations.

Adjoining on the one hand makes Chomsky adjunction possible. In this case, the recursion of the foot node in the auxiliary tree is across one level only, the recursive nodes are immediate mothers/daughters of each other, as in 4. On the other hand, TAG also allows the use of auxiliary trees in which the recursion stretches across several nodes. In this fashion, auxiliary trees that contain terminal nodes between the two recursive nodes can be inserted into elementary trees and thus stretch out local dependencies. An example is in figure 5.

TAG treats this sentence as follows: First, the dependency between the WH-element and its base position is established locally, within a single elmentary tree, according to TAG principles. The effect of dislocating the WH-element into a higher clause is accomplished by means of Adjoining in fig. 5. Further embedding of instances can be derived analogously by further Adjoining operations.

Such stretching by Adjoining with recursive auxiliary trees is the one and only way in which context-sensitive constructions can be generated in TAG. This fact is known as the nonlocal dependency corollary: "Nonlocal dependencies always reduce to local ones once recursive structure is factored out." (Frank, 2002, p. 27). Current research in TAG reveals that the severely restricted type of context-sensitivity generated by Adjoining, so-called *mild context-sensitivity*, accurately characterizes the non-locality present in natural language (Frank, 2002).

2.1.2 The Nature of Elementary and Auxiliary Trees

While the basic operations over elementary and auxiliary trees have been outlined now, nothing has been said about the nature of these trees. We will follow Frank (2004) and "assume that elementary trees are built around a single lexical element, that is, a semantically contentful word like a noun, verb or adjective" (Frank, 2004, p. 11).

This means that elementary trees are similar to DG nuclei or chunks (if we allowed attributive adjectives to be part of elementary trees). Elementary trees are assumed to provide argument slots and are closely related to predicate-argument structure:

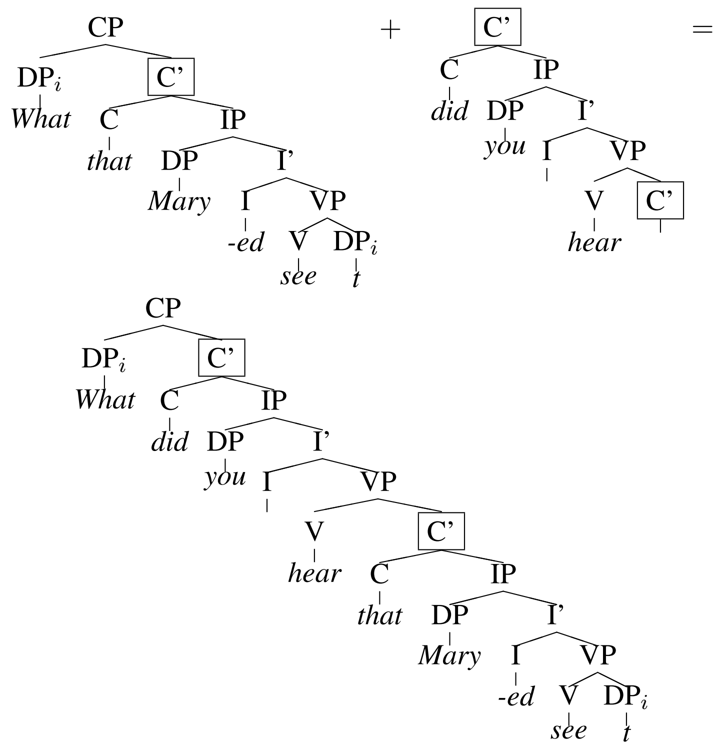


Figure 5: Adjoining for WH-questions. The deep recursion of the auxiliary trees introduces mild context-sensitivity. The foot node is boxed.

> A great deal of work in syntactic theory has assigned a privileged status to the syntactic analogue of predicate argument structure. Such a domain, which we call a *thematic domain*, consists of a single lexical predicate along with the structural context in which it takes its arguments. This notion takes a variety of forms and names, but the same idea seems to underlie kernel sentences in Harris (1957) and Chomsky (1955; Chomsky (1957), cyclic domains in Chomsky (1965), strata in Relational Grammar (Perlmutter, 1983), f-structure nuclei in LFG (Bresnan, 1982) and governing categories in Government-Binding Theory (Chomsky, 1981). (Frank, 2002, p. 38)

DG parses directly for a predicate argument structure and DG structures have been described as the f-structure part of LFG (Bröker, Hahn, and Schacht, 1994). DG and TAG thus take a very similar stance on the inherent aims and structures of syntactic theory. Following work by Grimshaw (1991), elementary trees are assumed to include extended projections. "Grimshaw (1991) characterizes the linkage between between lexical and functional projections via a notion she labels *extended projection*. In essence, the extended projection of a lexical head includes the projections of all those functional heads that embed it (up through but not including the next lexical head)." (Frank, 2002, p. 43). Auxiliary trees are defined as elementary trees that show the recursive characteristics described.

TAG uses transformations to generate elementary trees. Grimshaw (1991) and Frank (2002) discuss that in head-movement the base position and the ultimate landing site lie within a single extended projection. This entails that head-movement generally is not unbounded. We have discussed in 2.1 for English how finite-state patterns can be used to cover them. Elementary trees, which include extended projections, are much larger than the production rules that are used in phrase-structure (PSG) frameworks. Therefore, many dependencies (for example head-movement) that stretch across more than a mother-daughter node relation and are thus non-local for PSG remain local in TAG, as they only involve a single elementary tree. The extended projections of a TAG elementary tree (Grimshaw, 1991) are also called *extended domain of locality* (EDOL) (Carroll et al., 1999). Much of the reduction in TAG grammar complexity is owed to EDOL. Features do not need to be percolated,

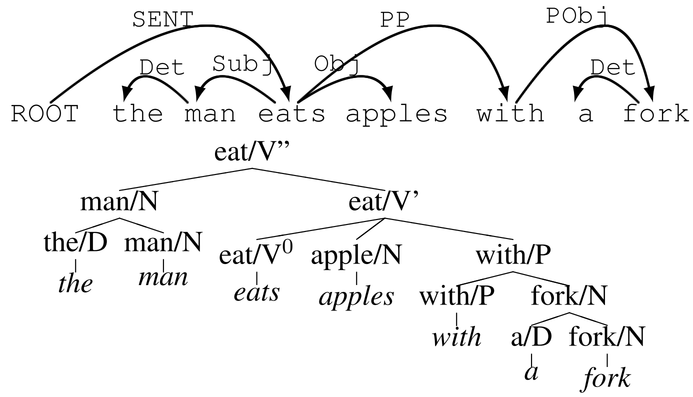


Figure 6: A DG representation and a principled conversion of DG to X-bar representation

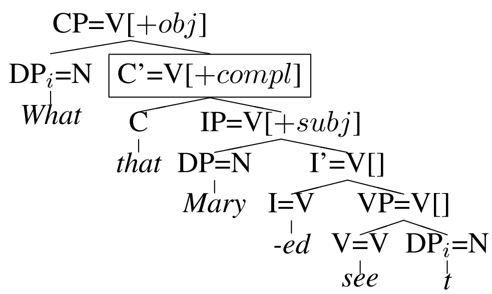


Figure 7: A DG to TAG tree mapped representation. DG relation labels are in square brackets.

and parsing algorithms of lower complexity can be applied. EDOL has a practical benefit for broad-coverage parsing, reducing search spaces and the number of unifications needed in a unification-based grammar.

2.1.3 TAG Adjoining in DG

DG shares EDOL with TAG, because it only knows content word projections (*nuclei*). At the same time, because DG grammar rules are binary, grammar size, which is a parameter in parsing complexity, stays low.

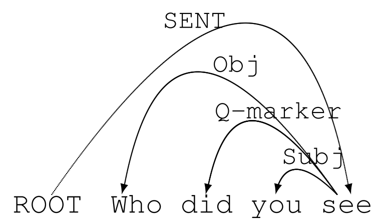
In LFG f-structure, HPSG and Functional DG, where functional projections appear below the content-word head as what HPSG has termed markers, the elementary tree of a word W that falls into a content word class and the maximal projection of W coincide. All bar-levels are isomorphic to the head word W in DG (Miller, 1999). The important difference between Ws at different bar-levels is that they have attached more or less dependents. Different projections of W can be seen as different stages of derivation. A possible conversion from DG to X-bar for example distinguishes between a projection or derivation state of V with all dependents except subject attached (V’, internal arguments), and a projection or derivation state of V with all dependents attached (V”, including the external argument). Such a conversion, and the equivalence of DG and X-bar is described in (Covington, 1994) and illustrated in fig. 6. A DG to TAG tree mapped representation following from that is shown in fig. 7. Unlike in TAG, the equivalent of elementary trees are also constructed without transformations in DG. The verb has local access to the fronted object in the elementary tree, i.e. in a non-embedded WH-question, just like in LFG f-structure, where all arguments appear flat under the verb predicate.

Because functional words are attached as markers, all DG equivalents of functional projections (the combination of a head word and a function word) are also all projections of the head word. The only possible foot node N in DG is therefore a projection W of the head word W. Adjoining inserts a recursive structure at some projection N which is called the foot node. The head of the inserted structure is N, and the part of the elementary tree that appeared below N occurs below the recursive N. Since the foot node N of the inserted auxiliary tree appears above the N of the original elementary tree, Adjunction inserts new governors into an existing structure and thus breaks the context-freeness. In a nutshell, the DG difference between Substitution and Adjoining is: Substitution inserts dependents, Adjoining inserts governors.

In DG, Adjoining inserts an auxiliary tree into some projection or derivation stage of W. Adjoining to maximal projections (in which all dependents are attached) is pointless, because then Adjoining A to B is equivalent to Substituting B to A. The point is that the auxiliary tree is inserted at a derivation stage in which not all dependents have been attached, at a partial projection.

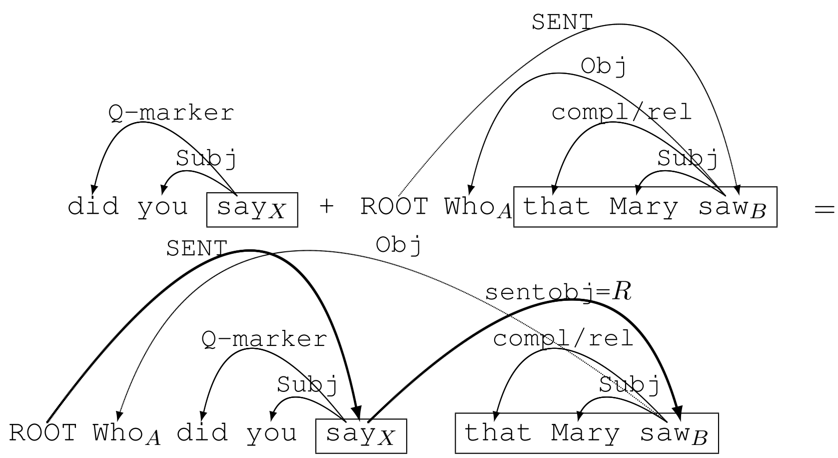
While in the example of 6 derivation order coincides with the internal/external argument ordering, that may not necessarily be so. If a standard CYK algorithm is employed for

(2) *Who did you see ?*



the subject is attached before the object, as can be seen in fig. 7. At the stage where all dependents except for the object are attached, Adjoining can occur.

(3) *Who did you see ?*



Adjoining can be described as follows in Functional DG: Given a local relation (of a type falling inside a TAG elementary tree, hence non-clausal) from B to A, if there is a maximal projection equivalent to a TAG auxiliary tree X, and if there can be a relation (or a relation chain) from X to both A and B such that

- the relation type R from X to B is across elementary trees, hence clausal,
- the possible relation from X to A would be of the same type as the one from B to A
- the governor of B is also licenced to be governor of X, and has the same relation type

then X can adjoin into the structure formed by A and B. Adjoining inserts X between A and B, constructs a relation R from X to B, and the governor of B becomes the governor of X – this is the mildly context-sensitive relation. As a result, the possible surface relation from X to A remains unrealized, delegated to the head of a lower clause (Nivre and Nilsson, 2005).

If we apply the principled conversion suggested in Covington (1994) for the conversion between a labelled DG relation and a constituent tree we can conclude that if every projective DG relation corresponds to a TAG elementary tree and every trigger for a non-projective DG relation corresponds to a TAG auxiliary tree, then DG and TAG are equivalent.

2.1.4 TAG Adjoining in LFG

LFG uses functional uncertainty for mild context-sensitivity (Kaplan and Zaenen, 1989; Dalrymple, Kaplan, and King, 2001). Functional uncertainty allows LDDs to extend across an unlimited, recursive path. Subordinate clauses appear as a COMP or XCOMP (the latter for control) dependent in f-structure, hence the recursion, expressed by the Kleene star, is COMP* or XCOMP*, but this is equivalent to TAG recursion on C-bar or DG recursion on V.

Modelling the recursion on the functional level, as in LFG or the suggested DG approach leads to a representationally minimal theory (Jurafsky, 1996).

2.1.5 Implementation

An implementation for the treatment of such embedded WH-dependencies exists in Pro3Gres. TAG Adjoining recursively inserts local trees into the middle of other trees. Due to this characteristic, only LDDs from the beginning of one elementary tree to the end of the originally same (elementary) tree can be generated.

In non-subject WH-questions, the WH-pronoun appears at the front of the sentence rather than in its usual post-verbal position. The implemented approach is based on pre-parsing: in WH-pronoun question sentences, before the main parsing is started, the WH-pronoun pre-parses with each verb, which may constitute the end of the originally same (elementary) tree.

We have thus implemented a simple version of TAG Adjoining or equivalently LFG functional uncertainty by using mild context-sensitivity in order to fulfill the goal of reducing grammatical complexity and expressiveness.

3 Probability Model

Pro3Gres is a probabilistic parser that parses between heads of chunks and thus profits from a combination of finite-state techniques and parsing. The chunks & dependencies model has been suggested by (Abney, 1995). It is described as psycholinguistically adequate (Crocker and Corley, 2002), especially when combined with a statistical model by (Jurafsky, 1996). (Frank, 2003) presents a (albeit non-probabilistic) chunks & dependencies model for LFG. Chunks can be freely combined subject to adjacency and projectivity (contiguity) constraints, which leads to a context-free parsing algorithm. Except for the added book-keeping functional annotations, her parsing algorithm is akin to CYK, which we use. Unlike (Frank, 2003), Pro3Gres is probabilistic. This is an important asset for a robust, broad-coverage and practically applicable parser. The statistical model that we suggest cannot be said to be probabilistic in the sense that it captures the probability of *generating* a sentence (Charniak, 1996; Collins, 1999), but rests on the psycholinguistically adequate assumption that parsing is a decision process. The probabilities of possible decisions at an ambiguous point in the derivation are assumed to add up to 1 (Crocker and Brants, 2000). In this sense, its probability estimation is closer to *discriminative* models (Johnson, 2001).

We will explain Pro3Gres' main probability model by way of comparing it to (Collins, 1996). Both Collins (1996) and Pro3Gres are mainly dependency-based statistical parsers parsing over heads of chunks, a close relation can therefore be expected. The Collins (1996) MLE and the main Pro3Gres MLE can be juxtaposed as follows:

(4) Collins (1996) MLE estimation: $P(R|\langle a, atag\rangle, \langle b, btag\rangle, dist) \cong \frac{\#(R,\langle a,atag\rangle,\langle b,btag\rangle,dist)}{\#(\langle a,atag\rangle,\langle b,btag\rangle,dist)}$

(5) Main Pro3Gres MLE estimation: $P(R, dist|a, b) \cong p(R|a, b) \cdot p(dist|R) \cong \frac{\#(R,a,b)}{\#(a,b)} \cdot \frac{\#(R,dist)}{\#R}$

The following differences are observed:

- Pro3Gres does not use tag information. The first reason for this is because the licensing, hand-written grammar is based on Penn tags.

- The second reason for not using tag information is because Pro3Gres backs off to semantic WordNet classes (Fellbaum, 1998) which has the advantage that it is more fine-grained[3].
- Pro3Gres uses real distances, measured in chunks, instead of a vector of features. While the type of relation R is lexicalized, i.e. conditioned on the lexical items, the distance is assumed to be dependent only on R. This is based on the observation that some relations typically have very short distances (e.g. verb-object), others can be quite long (e.g. Verb-PP attachment). This observation greatly reduces the sparse data problem. (Chung and Rim, 2003) have made similar observations for Korean.
- The co-occurrence count in the MLE denominator is not the sentence-context, but the sum of competing relations. For example, the $object$ and the $adjunct$ relation are in competition, as they are licensed by the same tag sequence ($VB{*}\ NN{*}$). Pro3Gres models attachment probabilities as decision probabilities, which is in accordance with the view that parsing is a decision process.
- Relations (R) have a Functional Dependency Grammar definition, including long-distance dependencies.

4 Evaluation

In traditional constituency approaches, parser evaluation is done in terms of the correspondence of the bracketting between the gold standard and the parser output. Lin (1995) suggested evaluating on the linguistically more meaningful level of syntactic relations. For the current evaluation, a hand-compiled gold standard following this suggestion is used (Carroll, Minnen, and Briscoe, 1999). It contains the grammatical relation data of 500 sentences from the Susanne corpus[4].

	Percentage Values for			
	Subject	Object	noun-PP	verb-PP
Precision	91	89	73	74
Recall	81	83	67	83
	Comparison to Lin (on the whole Susanne corpus)			
	Subject	Object	PP-attachment	
Precision	89	88	78	
Recall	78	72	72	
	Comparison to Buchholz (Buchholz, 2002), according to Preiss			
	Subject	Object		
Precision	86	88		
Recall	73	77		
	Comparison to Charniak (Charniak, 2000), according to Preiss			
	Subject	Object		
Precision	82	84		
Recall	70	76		

Table 4: Results of evaluating the parser output on subject, object and PP-attachment relations and a partial comparison

[3]For the semantic backoff of verbs, a version in which verbs use a Levin class (Levin, 1993) backoff has been tested. But Wordnet backoff performs better, possibly due to the fact that Levin coverage is lower

[4]The 500 sentences are a random sample of all those sentences from the Susanne corpus which their system was able to parse

	RASP		Pro3Gres			
Relation	Precision	Recall	Precision		Recall	
	%	%	%	#	%	#
ncmod	78	73	75.0	1590 of 2119	70.6	1690 of 2391
arg_mod	84	41	76.1	16 of 21	51.2	21 of 41
ncsubj	85	88	92.6	825 of 891	81.1	775 of 956
dobj	86	84	88.7	425 of 479	84.5	317 of 375
obj2	39	84	90.0	9 of 10	56.3	9 of 16
iobj	42	65	74.8	80 of 107	56.1	88 of 157

Table 5: Comparison of evaluation results to RASP

	LDD relations results for	
WH-Subject Precision	57/62	92%
WH-Subject Recall	45/50	90%
WH-Object Precision	6/10	60%
WH-Object Recall	6/7	86%
Anaphora of the rel. clause subject Precision	41/46	89%
Anaphora of the rel. clause subject Recall	40/63	63%
Passive subject Recall	132/160	83%
Precision for subject-control subjects	40/50	80%
Precision for object-control subjects	5/5	100%
Precision of $modpart$ relation	34/46	74%
Precision for topicalized verb-attached PPs	25/35	71%

Table 6: Available results for relations traditionally considered to involve LDDs

Comparing these results to Lin (1998) and Preiss (2003) as far as is possible shows that the performance of the parser is state-of-the-art (see table 4). Carroll, Minnen, and Briscoe (2003) have evaluated their own parser (RASP) using this evaluation scheme. Their reported performance is compared to the Pro3Gres in table 5. We have used a simple post-processor to recover chunk-internal relations and do an argument/adjunct distinction for PPs. It appears that Pro3Gres performs better on chunk-external, RASP better on chunk-internal relations.

The new local relations corresponding to LDDs in the Penn Treebank have been selectively evaluated as far as the annotations permit, shown in table 6. For NP traces and NP PRO, the annotation does not directly provide all the necessary data. Passivity is not currently expressed in the predicate-argument parser output, thus only recall values can be delivered. Since Carroll, Minnen, and Briscoe (2003)'s annotation does not directly express control, reduced relative clauses or the dependency direction, only reliable precision values are available in those cases. As for gerunds, neither Carroll nor the parser output retains tagging information, which makes a selective evaluation of them impossible. The fact that performance for the new local relations corresponding to LDDs is not generally lower than in the dependencies corresponding to local constituency, although they correspond to a sequence of decisions in a traditional statistical parser, indicates that our LDD approach improves parsing performance. Absolute values are given due to the low counts of these relatively rare relations.

Table 7 shows that about half of the PP-attachment errors are real attachment errors. The second most frequent error is deficient tagging or chunking – the price to pay for shallowness.

Error Classification of PP-Attachment Errors of the first 100 evaluation corpus sentences					
Attachment Error	Head Extraction Error	Chunking or Tagging Error	compl/prep Error	Grammar Mistake or incomplete Parse	Grammar Assumption
Noun-PP Attachment Precision					
22	1	8	0	3	3
Verb-PP Attachment Precision					
12	1	5	1	1	2
Noun-PP Attachment Recall					
25	1	14	0	12	5
Verb-PP Attachment Recall (on PP arguments only)					
2	0	1	0	0	0
Percentages					
51%	3%	24%	1%	13%	12%

Table 7: Analysis of PP-Attachment Errors

5 Conclusions

We have presented a fast, lexicalized broad-coverage parser delivering simple f-structures as output. An evaluation at the grammatical relation level shows that its performance is state-of-the-art.

We have shown that the parser stays as shallow as is possible for each task, combining shallow and deep-linguistic methods by integrating chunking and by expressing long-distance dependencies in a mostly context-free way, thus offering on the one hand a parsing complexity as low as for a probabilistic parser, but on the other hand a deep-lingusitic analysis as with a type of formal grammars.

We have discussed that the vast majority of long-distance dependencies can be modelled locally in a functional representation. We have discussed the nature of the remaining truly context-sensitive cases, namely mild context-sensitivity as recursion over syntactic structures in TAG or equivalently, but representationally minimal, recursion over f-structures in LFG or DG. Unlike in TAG elementary trees, movement is obviated.

Following these theoretical considerations, the LFG suggestion by Frank (2003), as well as our broad-coverage evidence (Schneider, Dowdall, and Rinaldi, 2004; Rinaldi et al., 2004a; Rinaldi et al., 2004b; Weeds et al., 2005), we suggest that c-structures or other configurational “surface” representations may be obviated for the syntactic analysis of natural language. By reducing grammar complexity (Frank, 2002; Frank, 2004), by reducing parsing complexity to mostly context-free parsing and finite-state based chunking (Schneider, 2003; Schneider, 2004), by bridging the gap between language engineering and formal grammar (Kaplan et al., 2004) and by aiming for a representationally minimal theory (Jurafsky, 1996) we conclude that chunks and dependencies (Abney, 1995; Frank, 2003) may be sufficient for a formal grammar theory.

References

Abney, Steven. 1995. Chunks and dependencies: Bringing processing evidence to bear on syntax. In Jennifer Cole, Georgia Green, and Jerry Morgan, editors, *Computational Linguistics and the Foundations of Linguistic Theory*, pages 145–164. CSLI.

Abney, Steven. 1996. Partial parsing via finite-state cascades. In John Carroll, editor, *Proc. of the Workshop on Robust Parsing at the 8th Summer School on Logic, Language and Information*, number 435 in CSRP, pages 8–15. University of Sussex, Brighton.

Brants, Thorsten and Matthew Crocker. 2000. Probabilistic parsing and psychological plausibility.

In *Proceedings of 18th International Conference on Computational Linguistics COLING-2000*, Saarbrücken/Luxembourg/Nancy.

Bresnan, Joan, editor. 1982. *The Mental Representation of Grammatical Relations*. The MIT Press, Cambridge, Massachusetts.

Bröker, Norbert, Udo Hahn, and Susanne Schacht. 1994. Concurrent lexicalized dependency parsing: The ParseTalk model. In *Coling 94*, pages 379–385.

Buchholz, Sabine. 2002. *Memory-Based Grammatical Relation Finding*. Ph.D. thesis, University of Tilburg, Tilburg, Netherlands.

Burke, M., A. Cahill, R. O'Donovan, J. van Genabith, and A. Way. 2004. Treebank-based acquisistion of wide-coverage, probabilistic LFG resources: Project overview, results and evaluation. In *The First International Joint Conference on Natural Language Processing (IJCNLP-04), Workshop "Beyond shallow analyses - Formalisms and statistical modeling for deep analyses"*, Sanya City, China.

Cahill, Aoife, Michael Burke, Ruth O'Donovan, Josef van Genabith, and Andy Way. 2004. Long-distance dependency resolution in automatically acquired wide-coverage PCFG-based LFG approximations. In *Proceedings of ACL-2004*, Barcelona, Spain.

Carroll, John, Guido Minnen, and Edward Briscoe. 2003. Parser evaluation: using a grammatical relation annotation scheme. In Anne Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*. Kluwer, Dordrecht, pages 299–316.

Carroll, John, Guido Minnen, and Ted Briscoe. 1999. Corpus annotation for parser evaluation. In *Proceedings of the EACL-99 Post-Conference Workshop on Linguistically Interpreted Corpora*, Bergen, Norway.

Carroll, John, Nicolas Nicolov, Olga Shaumyan, Martine Smets, and David Weir. 1999. Parsing with an extended domain of locality. In *Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Charniak, Eugene. 1996. Tree-bank grammar. Technical Report Technical Report CS-96-02, Department of Computer Science, Brown University.

Charniak, Eugene. 2000. A maximum-entropy-inspired parser. In *Proceedings of the North American Chapter of the ACL*, pages 132–139.

Chomsky, Noam. 1955. *The logical structure of linguistic theory*. Distributed by University of Indiana Linguistics Club.

Chomsky, Noam. 1957. *Syntactic Structures*. Mouton, The Hague.

Chomsky, Noam. 1965. *Aspects of the Theory of Syntax*. MIT Press, Cambridge, MA.

Chomsky, Noam. 1981. *Lectures on Government and Binding*. Foris Publications, Foris Publications.

Chomsky, Noam. 1995. *The Minimalist Program*. The MIT Press, Cambridge, Massachusetts.

Chung, Hoojung and Hae-Chang Rim. 2003. A new probabilistic dependency parsing model for head-final, free word order languages. *IEICE Transaction on Information & System*, E86-D, No. 11:2490–2493.

Collins, Michael. 1996. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*, pages 184–191, Philadelphia.

Collins, Michael. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA.

Covington, Michael A. 1994. An empirically motivated reinterpretation of Dependency Grammar. Technical Report AI1994-01, University of Georgia, Athens, Georgia.

Crocker, Matthew and Thorsten Brants. 2000. Wide coverage probabilistic sentence processing. *Journal of Psycholinguistic Research*, 29(6):647–669.

Crocker, Matthew and Steffan Corley. 2002. Modular architectures and statistical mechanisms: The case from lexical category disambiguation. In Paola Merlo and Suzanne Stevenson, editors, *The Lexical Basis of Sentence Processing: Formal, Computational and Experimental Issues*. John Benjamins, Amsterdam.

Dalrymple, Mary, Ronald Kaplan, and Tracy Halloway King. 2001. Weak crossover and the absence of traces. In Mriram Butt and Tracy Holloway King, editors, *Proceedings of the LFG01 Conference*, Hong Kong. CSLI.

Eisner, Jason. 1997. Bilexical grammars and a cubic-time probabilistic parser. In *Proceedings of the 5th International Workshop on Parsing Technologies*, pages 54–65, MIT, Cambridge, MA, September.

Eisner, Jason. 2000. Bilexical grammars and their cubic-time parsing algorithms. In Harry Bunt and Anton Nijholt, editors, *Advances in Probabilistic and Other Parsing Technologies*. Kluwer.

Fellbaum, Christiane, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

Frank, Anette. 2003. Projecting F-structures from chunks. In Mriram Butt and Traci Holloway King, editors, *Proceedings of the LFG03 Conference*, Albany, NY. CSLI.

Frank, Robert. 2002. *Phrase Structure Composition and Syntactic Dependencies*. MIT Press, Cambridge, MA.

Frank, Robert. 2004. Restricting grammatical complexity. *Cognitive Science*, 28(5).

Grimshaw, Jane. 1991. Extended projection. manuscript.

Hajič, Jan. 1998. Building a syntactically annotated corpus: The Prague Dependency Treebank. In Eva Hajičová, editor, *Issues of Valency and Meaning. Studies in Honor of Jarmila Panevová*. Karolinum, Charles University Press, Prague, pages 106–132.

Harris, Zellig. 1957. Co-occurrence and transformation in linguistic structure. *Language*, pages 283–340.

Hockenmaier, Julia and Mark Steedman. 2002. Generative models for statistical parsing with combinatory categorial grammar. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia.

Hudson, RIchard. 1987. *Journal of Linguistics*, 23:109 – 132.

Jijkoun, Valentin. 2003. Finding non-local dependencies: beyond pattern matching. In *Proceedings of the ACL 03 Student Workshop*, Budapest.

Johnson, Mark. 2001. Joint and conditional estimation of tagging and parsing models. In *Proceedings of the 39th Meeting of the ACL*, pages 314–321, Toulouse, France.

Johnson, Mark. 2002. A simple pattern-matching algorithm for recovering empty nodes and their antecedents. In *Proceedings of the 40th Meeting of the ACL*, University of Pennsylvania, Philadelphia.

Joshi, Aravind. 1985. How much context-sensitivty is required to provide reasonable syntactic descriptions: Tree Adjoining Grammars. In David Dowty, Lauri Karttunen, and Arnold Zwicky, editors, *Natural Language Parsing: Psychological, computational, and theoretical perspectives*. CUP, Cambridge, pages 206–250.

Joshi, Aravind and Anthony Kroch. 1985. The linguistic relevance of Tree Adjoining Grammar. Technical Report MS-CS-85-16, Department of Computer and Information Sciences, University of Pennsylvania.

Joshi, Aravind K. and K. Vijay-Shanker. 1989. Treatment of long-distance dependencies in LFG and TAG: Functional uncertainty in LFG is a corollary in TAG. In *Proceedings of ACL '89*.

Jurafsky, Daniel. 1996. A probabilistic model of lexical and syntactic access and disambiguation. *Cognitive Science*.

Kaplan, Ron, Stefan Riezler, Tracy H. King, John T. Maxwell III, Alex Vasserman, and Richard Crouch. 2004. Speed and accuracy in shallow and deep stochastic parsing. In *Proceedings of HLT/NAACL 2004*, Boston, MA.

Kaplan, Ronald and Annie Zaenen. 1989. Long-distance dependencies, constituent structure, and functional uncertainty. In Mark Baltin and Anthony Kroch, editors, *Alternative Concepts of Phrase Structrue*. Chicago University Press, pages 17 – 42.

Kaplan, Ronald M. and Joan Bresnan. 1982. Lexical-Functional Grammr: A formal system for grammatical representation. In Joan Bresnan, editor, *The Mental Representation of Grammatical Relations*. MIT Press, Cambridge, MA, pages 173–281.

Levin, Beth C. 1993. *English Verb Classes and Alternations: a Preliminary Investigation*. University of Chicago Press, Chicago, IL.

Lin, Dekang. 1995. A dependency-based method for evaluating broad-coverage parsers. In *Proceedings of IJCAI-95*, Montreal.

Lin, Dekang. 1998. Dependency-based evaluation of MINIPAR. In *Workshop on the Evaluation of Parsing Systems*, Granada, Spain.

Marcus, Mitch, Beatrice Santorini, and M.A. Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19:313–330.

Mel'čuk, Igor. 1988. *Dependency Syntax: theory and practice*. State University of New York Press, New York.

Miller, Philip H. 1999. *Strong Generative Capacity*. CSLI, Stanford, CA.

Miyao, Yusuke, Takashi Ninomiya, and Jun'ichi Tsujii. 2004. Corpus-oriented grammar development for acquiring a Head-driven Phrase Structure Grammar from the Penn Treebank. In *Proceedings of IJCNLP-04*.

Neuhaus, Peter and Norbert Bröker. 1997. The complexity of recognition of linguistically adequate dependency grammars. In *Proceedings of the 35th ACL and 8th EACL*, pages 337–343, Madrid, Spain.

Nivre, Joakim. 2004. Inductive dependency parsing. In *Proceedings of Promote IT*, Karlstad University.

Nivre, Joakim and Jens Nilsson. 2005. Pseudo-projective dependency parsing. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 99–106, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Perlmutter, David. 1983. *Studies in Relational Grammar I*. Chicago Unicersity Press, Chicago.

Pollard, Carl and Ivan Sag. 1994. *Head-Driven Phrase Structure Grammar*. Chicago University Press, Chicago, Illinois.

Preiss, Judita. 2003. Using grammatical relations to compare parsers. In *Proc. of EACL 03*, Budapest, Hungary.

Riezler, Stefan, Tracy H. King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell, and Mark Johnson. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques. In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, Philadephia, PA.

Rinaldi, Fabio, James Dowdall, Gerold Schneider, and Andreas Persidis. 2004a. Answering Questions in the genomics domain. In *ACL-2004 workshop on Question Answering in restricted domains*, Barcelona, Spain.

Rinaldi, Fabio, Gerold Schneider, James Dowdall, Christos Andronis, Andreas Persidis, and Ourania Konstanti. 2004b. Mining relations in the GENIA corpus. In *Proceedings of the Second workshop on Data Mining and Text Mining for Bioinformatics*, Pisa, Italy.

Sarkar, Anoop, Fei Xia, and Aravind Joshi. 2000. Some experiments on indicators of parsing complexity for lexicalized grammars. In *Proceedings of COLING '00*.

Schneider, Gerold. 2003. Extracting and using trace-free Functional Dependencies from the Penn Treebank to reduce parsing complexity. In *Proceedings of Treebanks and Linguistic Theories (TLT) 2003*, Växjö, Sweden.

Schneider, Gerold. 2004. Combining shallow and deep processing for a robust, fast, deep-linguistic dependency parser. In *ESSLLI 2004 Workshop on Combining Shallow and Deep Processing for NLP (ComShaDeP 2004)*, Nancy, France, August 2004.

Schneider, Gerold, James Dowdall, and Fabio Rinaldi. 2004. A robust and deep-linguistic theory applied to large-scale parsing. In *Coling 2004 Workshop on Robust Methods in the Analysis of Natural Language Data (ROMAND 2004)*, Geneva, Switzerland, August 2004.

Tesnière, Lucien. 1959. *Eléments de Syntaxe Structurale*. Librairie Klincksieck, Paris.

Tsuruoka, Yoshimasa, Yusuke Miyao and Jun'ichi Tsujii. 2004. Towards efficient probabilistic HPSG parsing: integrating semantic and syntactic preference to guide the parsing. In *Proceedings of IJCNLP-04 Workshop: Beyond shallow analyses - Formalisms and statistical modeling for deep analyses*.

Weeds, Julie, James Dowdall, Gerold Schneider, Bill Keller, and David Weir. 2005. Using distributional similarity to organise BioMedical terminology. *Terminology*.

Younger, D. H. 1967. Recognition and parsing of context-free languages in time n^3. *Information and Control*, 10:189-208.

Zwicky, Arnold. 1985. Heads. *Journal of Linguistics*, 21:1–30.