Fast, Deep-Linguistic Statistical Dependency Parsing

Gerold Schneider, Fabio Rinaldi, James Dowdall

Institute of Computational Linguistics, University of Zurich
{gschneid,rinaldi}@ifi.unizh.ch, j.m.dowdall@sussex.ac.uk

Abstract

We present and evaluate an implemented statistical minimal parsing strategy exploiting DG charateristics to permit fast, robust, deep-linguistic analysis of unrestricted text, and compare its probability model to (Collins, 1999) and an adaptation, (Dubey and Keller, 2003). We show that DG allows for the expression of the majority of English LDDs in a context-free way and offers simple yet powerful statistical models.

1 Introduction

We present a fast, deep-linguistic statistical parser that profits from DG characteristics and that uses am minimal parsing strategy. First, we rely on finite-state based approaches as long as possible, secondly where parsing is necessary we keep it context-free as long as possible[1]. For low-level syntactic tasks, tagging and base-NP chunking is used, parsing only takes place between heads of chunks. Robust, successful parsers (Abney, 1995; Collins, 1999) have shown that this division of labour is particularly attractive for DG.

Deep-linguistic, Formal Grammar parsers have carefully crafted grammars written by professional linguists. But unrestricted real-world texts still pose a problem to NLP systems that are based on Formal Grammars. Few hand-crafted, deep linguistic grammars achieve the coverage and robustness needed to parse large corpora (see (Riezler et al., 2002), (Burke et al., 2004) and (Hockenmaier and Steedman, 2002) for exceptions), and speed remains a serious challenge. The typical problems can be grouped as follows.

Grammar complexity Fully comprehensive grammars are difficult to maintain and considerably increase parsing complexity.

Parsing complexity Typical formal grammar parser complexity is much higher than the $O(n^3)$ for CFG. The complexity of some formal grammars is still unknown.[2] Parsing algorithms able to treat completely unrestricted long-distance dependencies are NP-complete (Neuhaus and Bröker, 1997).

Ranking Returning all syntactically possible analyses for a sentence is not what is expected of a syntactic analyzer. A clear indication of preference is needed.

Pruning In order to keep search spaces manageable it is necessary to discard unconvincing alternatives already during the parsing process.

A number of robust statistical parsers that offer solutions to these problems have become available (Charniak, 2000; Collins, 1999; Henderson, 2003). In a statistical parser, the ranking of intermediate structures occurs naturally and based on empirical grounds, while most rule-based systems rely on ad hoc heuristics. With an aggressive beam for parse-time pruning (so in our parser), real-world parsing time can be reduced to near-linear. If one were to assume a constantly full fixed beam, or uses an oracle (Nivre, 2004) it is linear in practice[3].

Also worst-case complexity for exhaustive parsing is low, as these parsers are CFG-based (Eisner, 2000)[4]. But they typically produce CFG constituency data as output, trees that do not express long-distance dependencies. Although grammatical function and empty

[1] Non-subject WH-question pronouns and support verbs cannot be treated context-free with our approach. We use a simple pre-parsing step to analyze them

[2] For Tree-Adjoining Grammars (TAG) it is $O(n^7)$ or $O(n^8)$ depending on the implementation (Eisner, 2000). (Sarkar et al., 2000) state that the theoretical bound of worst time complexity for Head-Driven Phrase Structure Grammar (HPSG) parsing is exponential.

[3] In practical terms, beam or oracle approach have very similar effects

[4] Parsing complexity of the original Collins Models is $O(n^5)$, but theoretically $O(n^3)$ would be possible

	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 1: The distribution of the 10 most frequent types of empty nodes and their antecedents in the Penn Treebank (adapted from (Johnson, 2002)). Bracketed line numbers only involve LDDs as grammar artifact

nodes annotation expressing long-distance dependencies are provided in Treebanks such as the Penn Treebank (Marcus et al., 1993), most statistical Treebank trained parsers fully or largely ignore them[5], which 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. In fact, some syntactic relations cannot be recovered on configurational grounds only. For these reasons, (Johnson, 2002) refers to them as "half-grammars".

An approach that relies heavily on DG characteristics is explored in this paper. It uses a hand-written DG grammar and a lexicalized probability model. It combines the low complexity of a CFG parser, the pruning and ranking advantages of statistical parsers and the ability to express the majority of LDDs of Formal Grammars. After presenting the DG benefits, we define our DG and introduce our statistical model. Then, we give an evaluation.

2 The Benefit of DG Characteristics

In addition to some obvious benefits, such as the integration of chunking and parsing (Abney, 1995), where a chunk largely corresponds to a *nucleus* (Tesnière, 1959), or that in an endocentric theory projection can never fail, we present eight characteristics in more detail, which in their combination allow us to treat the majority of English long-distance dependencies (LDD) in our DG parser *Pro3Gres* in a context-fee way.

[5](Collins, 1999) Model 2 uses some of the functional labels, and Model 3 some long-distance dependencies

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 1, reproduced from (Johnson, 2002) [line numbers and counts from the whole Treebank added], gives an overview.

2.1 No Empty Nodes

The fact that traditional DG does not know empty nodes allows a DG parser to use the efficient $0(n^3)$ CYK algorithm.

2.2 Only Content Words are Nuclei

Only content words can be nuclei in a traditional DG. This means that 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.

2.3 No External Argument, ID/LP

Moved clauses [line 6] are mostly PPs or clausal complements of verbs of utterance. Only verbs of utterance allow subject-verb inversion in affirmative clauses [line 8]. Our hand-written grammar provides rules with appropriate restrictions for them, allowing an inversion of the "canonical" dependency direction under well-defined conditions, distinguishing between *ordre linéaire* (linear precedence(LP)) and *ordre structural* (immediate dominance(ID)). Fronted positions are available locally to the verb in a theory that does not posit a distinction between internal and external arguments.

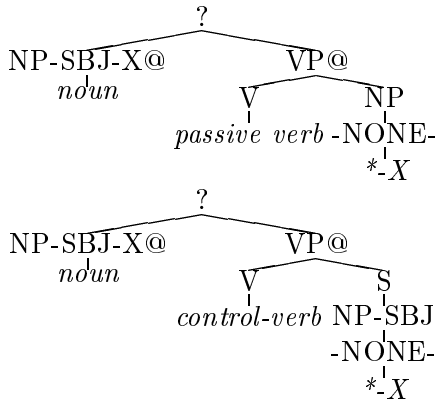
2.4 Exploiting Functional DG Labels

The fact that dependencies are often labeled is a main difference between DG and constituency. We exploit this by using dedicated labels to model a range of constituency LDDs, relations

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

Table 2: Important Pro3Gres Dependency types

spanning several constituency levels, including empty nodes and functional Penn Treebank labels, by a purely local DG relation[6]. The selective mapping patterns for MLE counts of passive subjects and control subjects from the Penn Treebank, the most frequent NP traces [line 1], are e.g. (@ stands for arbitrary nestedness):



Our approach employs finite-state approximations of long-distance dependencies, described in (Schneider, 2003) for DG and (Cahill et al., 2004) for Lexical Functional Grammar (LFG)It leaves empty nodes underspecified but largely recoverable. Table 2 gives an overview of important dependencies.

2.5 Monostratalism and Functionalism

While multistratal DGs exist and several dependency levels can be distinguished (Mel'čuk, 1988) we follow a conservative view close to the original (Tesnière, 1959), which basically parses directly for a simple LFG f-structure without needing a c-structure detour.

[6]In addition to taking less decisions due to the gained high-level shallowness, it is ensured that the lexical information that matters is available in one central place, allowing the parser to take one well-informed decision instead of several brittle decisions plagued by sparseness. Collapsing deeply nested structures into a single dependency relation is less complex but has a similar effect as selecting what goes in to the parse history in history-based approaches.

2.6 Graphs

DG theory often conceives of DG structures as graphs instead of trees (Hudson, 1984). A statistical lexicalized post-processing module in Pro3Gres transforms selected subtrees into graphs, e.g. in order to express control.

2.7 Transformation to Semantic Layer

Pro3Gres is currently being applied in a Question Answering system specifically targeted at technical domains (Rinaldi et al., 2004b). One of the main advantages of a DG parser such as Pro3Gres over other parsing approaches is that a mapping from the syntactic layer to a semantic layer (meaning representation) is partly simplified (Mollá et al., 2000).

2.8 Tesnière's Translations

The possible functional changes of a word called translations (Tesnière, 1959) are an exception to endocentricity. They are an important contribution to a traceless theory. Gerunds (*after winning/VBG the race*) or infinitives [line 2] may function as nouns, obviating the need for an empty subject. In nounless NPs such as *the poor*, adjectives function as nouns, obviating the need for an empty noun head. Participles may function as adjectives (*Western industrialized/VBN countries*), again obviating the need for an empty subject.

3 The Statistical Dependency Model

Most successful deep-linguistic Dependency Parsers (Lin, 1998; Tapanainen and Järvinen, 1997) do not have a statistical base. But one DG advantage is precisely that it offers simple but powerful statistical Maximum Likelihood Estimation (MLE) models. We now define our DG and the probability model.

The rules of a context-free, unlabeled DG are equivalent to binary-branching CFG rewrite rules in which the head and the mother node are isomorphic. When converting DG structures to CFG, the order of application of these rules is not necessarily known, but in a labeled DG, the set of rules can specify the order (Covington, 1994). Fig. 1 shows such two structures, equivalent except for the absence of functional labels in CFG. *Subj* (but not *PP*) has been used in this example conversion to specify the application order, hence we get a repetition of the *eat/V* node, mirroring a traditional CFG S and VP distinction.

In a binary CFG, any two constituents A and B which are adjacent during parsing are candi-

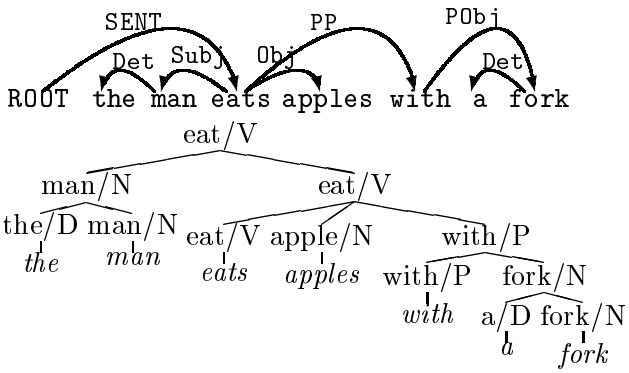


Figure 1: DG and CFG representation

dates for the RHS of a rewrite rule. As terminal types we use word tags.

$$X \rightarrow AB, e.g. NP \rightarrow \_DT\_NN \quad (1)$$

In DG, one of these is isomorphic to the LHS, i.e. the head. This grammar is also a Bare Phrase Structure grammar known from Minimalism (Chomsky, 1995).

$$B \rightarrow AB, e.g. \_NN \rightarrow \_DT\_NN \quad (2)$$

$$A \rightarrow AB, e.g. \_VB \rightarrow \_VB\_PP \quad (3)$$

Labeled DG rules additionally use a syntactic relation label R. A non-lexicalized model would be:

$$p(R|A \rightarrow AB) \cong \frac{\#(R, A \rightarrow AB)}{\#(A \rightarrow AB)} \quad (4)$$

Research on PCFG and PP-attachment has shown the importance of probabilizing on lexical heads (a and b).

$$p(R|A \rightarrow AB, a, b) \cong \frac{\#(R, A \rightarrow AB, a, b)}{\#(A \rightarrow AB, a, b)} \quad (5)$$

All that A $\rightarrow$ AB expresses is that the dependency relation is towards the *right*.

$$p(R|right, a, b) \cong \frac{\#(R, right, a, b)}{\#(right, a, b)} \quad (6)$$

e.g. for the Verb-PP attachment relation *pobj* (following (Collins and Brooks, 1995) including the description noun[7])

$$p(pobj|right, verb, prep, desc.noun) \cong \frac{\#(pobj, right, verb, prep, desc.noun)}{\#(right, verb, prep, desc.noun)}$$

The distance (measured in chunks) between a head and a dependent is a limiting factor for the probability of a dependency between them.

$$p(R, dist|right, a, b) \cong \frac{\#(R, dist, right, a, b)}{\#(right, a, b)} \quad (7)$$

[7]PP is considered to be an exocentric category, since both the preposition and the description noun can be seen as head; in LFG they appear as double-head

Many relations are only allowed towards one direction, the left/right factor is absent for them. Typical distances mainly depend on the relation. Objects usually immediately follow the verb, while a PP attached to the verb may easily follow only at the second or third position, after the object and other PPs etc. By application of the chain rule and assuming that distance is independent of the lexical heads we get:

$$p(R, dist|a, b) \cong \frac{\#(R, a, b)}{\#(a, b)} \cdot \frac{\#(R, dist)}{\#R} \quad (8)$$

We now explore Pro3Gres' main probability model by comparing it to (Collins, 1999), and an adaptation of it, (Dubey and Keller, 2003).

3.1 Relation of Pro3Gres to Collins Model 1

We will first consider the non-generative Model 1 (Collins, 1999). Both (Collins, 1999) Model 1 and Pro3Gres are mainly dependency-based statistical parsers over heads of chunks, a close relation can thus be expected. The (Collins, 1999) Model 1 MLE estimation is: $P(R|\langle a, atag\rangle, \langle b, btag\rangle, dist) \cong$

$$\frac{\#(R, \langle a, atag\rangle, \langle b, btag\rangle, dist)}{\#(\langle a, atag\rangle, \langle b, btag\rangle, dist)} \quad (9)$$

Differences in comparison to (8) are:

- Pro3Gres does not use tag information. This is because, first, the licensing handwritten 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) for nouns and to Levin classes (Levin, 1993) for verbs instead of to tags, which has the advantage of being more fine-grained.
- Pro3Gres uses real distances, measured in chunks, instead of a feature vector. Distance is assumed to be dependent only on R, which reduces the sparse data problem. (Chung and Rim, 2003) made similar observations for Korean.
- The co-occurrence count in the MLE denominator is not the sentence-context, but the sum of counts of competing relations. E.g. the *object* and *adjunct* relation are in competition, as they are licensed by the same tag sequence $VB*\ NN*$. Pro3Gres models attachment (thus decision) probabilities, viewing parsing as a decision process.
- Relations (R) have a Functional DG definition, including LDDs.

3.2 Relation to Collins Model 2

(Collins, 1999) Model 2 extends the parser to include a complement/adjunct distinction for NPs and subordinated clauses, and it includes a subcategorisation frame model.

For the subcategorisation-dependent generation of dependencies in Model 2, first the probabilities of the possible subcat frames are calculated and the selected subcat frame is added as a condition. Once a subcategorized constituent has been found, it is removed from the subcat frame, ensuring that non-subcategorized constituents cannot be attached as complement, which is one of the two major function of a subcat frame. The other major function of a subcat frame is to find all the subcategorized constituents. In order to ensure this, the probability when a rewrite rule can stop expanding is calculated. Importantly, the probability of a rewrite rule with a non-empty subcat frame to stop expanding is low, the probability of a rewrite rule with an empty subcat frame to stop expanding is high.

Pro3Gres includes a complement/adjunct distinction for NPs. The examples given in support of the subcategorisation frame model in (Collins, 1999) Model 2 are dealt with by the hand-written grammar in Pro3Gres.

Every complement relation type, namely *subj*, *obj*, *obj*2, *sentobj*, can only occur once per verb, which ensures one of the two major functions of a subcat frame, that non-subcategorized constituents cannot be attached as complements. This amounts to keeping separate subcat frames for each relation type, where the selection of the appropriate frame and removing the found constituent coincide, which has the advantage of a reduced search space: no hypothesized, but unfound subcat frame elements need to be managed. As for the second major function of subcat frames – to ensure that if possible all subcategorized constituents are found – the same principle applies: selection of subcat frame and removing of found constituents coincide; lexical information on the verb argument candidate is available at frame selection time already. This implies that Collins Model 2 takes an unnecessary detour.

As for the probability of stopping the expansion of a rule – since DG rules are always binary – it is always 0 before and 1 after the attachment. But what is needed in place of interrelations of constituents of the same rewrite rule is proper cooperation of the different subcat types. For example, the grammar rules only allow a noun to be *obj*2 once *obj* has been found, or a verb is required to have a subject unless it is non-finite or a participle, or all objects need to be closer to the verb than a subordinate clause.

3.3 Relation to Dubey & Keller 03

(Dubey and Keller, 2003) address the question whether models such as Collins also improve performance on freer word order languages, in their case German. German is considerably more inflectional which means that discarding functional information is more harmful, and which explains why the NEGRA annotation has been conceived to be quite flat (Skut et al., 1997). (Dubey and Keller, 2003) observe that models such as Collins when applied directly perform worse than an unlexicalized PCFG baseline. The fact that learning curves converge early indicates that this is not mainly a sparse data effect. They suggest a linguistically motivated change, which is shown to outperform the baseline.

The (Collins, 1999) Model 2 rule generation model for $P \rightarrow L_m...L_1HR_1...R_n$, is

$$P(RHS|LHS) = P_h(H|P, t(P), l(P)) \cdot \prod_{i=0}^{m} P_l(L_i, t(L_i), l(L_i)|P, H, t(H), l(H), d(i)) \cdot \prod_{i=0}^{n} P_r(R_i, t(R_i), l(R_i)|P, H, t(H), l(H), d(i))$$

P_h	P of head	t(H)	tag of H head word
LHS	left-hand side	RHS	right-hand side
$P_{l:1..m}$	P(words left of head)	$P_{r:1..n}$	P(words right of head)
H	LHS Head Category	P	RHS Mother Category
L	left Constit. Cat.	R	right Constit. Cat.
l(H)	head word of H	d	distance measure

Dubey & Keller suggest the following change in order to respect the NEGRA flatness: P_h is left unchanged, but P_l and P_r are conditioned on the preceding sister instead of on the head:

$$P(RHS|LHS) = P_h(H|P, t(P), l(P)) \cdot \prod_{i=0}^{m} P_l(L_i, t(L_i), l(L_i)|P, L_{i-1}, t(L_{i-1}), l(L_{i-1}), d(i)) \cdot \prod_{i=0}^{n} P_r(R_i, t(R_i), l(R_i)|P, R_{i-1}, t(R_{i-1}), l(R_{i-1}), d(i))$$

Their new model performs considerably better and also outperforms the unlexicalized baseline. The authors state that "[u]sing sister-head relationships is a way of counteracting the flatness of the grammar productions; it implicitly adds binary branching to the grammar." (ibid.). DG is binary branching by definition; adding binary branching implicitly converts the CFG rules into an ad-hoc DG.

Whether the combination ((Chomsky, 1995) *merge*) of two binary constituents directly projects to a "real" CFG rule LHS or an implicit intermediate constituent does not matter.

Observations

- What counts is each individual Functional DG dependency, no matter whether it is expressed as a sister-head or a head-head dependency, or stretches across several CFG levels (control, *modpart* etc.)
- Not adjacency (i,i-1) but headedness counts. Instead of conditioning on the preceding (i-1) sister, conditioning on the real DG head is linguistically more motivated[8].
- Not adjacency (i,i-1) but the type of GR counts: the question why Dubey & Keller did not use the NEGRA GR labels has to arise when discussing a strongly inflectional language such as German.
- The use of a generative model, calculating the probability of a rule and ultimately the probability of producing a sentence given the grammar only has theoretical advantages. For practical purposes, modeling parsetime decision probabilities is as valid.

With these observations in mind, we can compare Pro3Gres to (Dubey and Keller, 2003). As for the Base-NP Model, Pro3Gres only respects the best tagging & chunking result reported to it – a major source of errors (see section 4). In DG, projection (although not expansion) is deterministic. H and P are usually isomorphic, if not Tesnière-translations are rule-based. Since in DG, only lexical nodes are categories, P=t(P). P_h is thus l(h), the prior, we ignore it for maximizing. In analogy, also category (L/R) and their tags are identical. The revised formula is

$$P(RHS|LHS) \cong l(h) \cdot \prod_{i=0}^{m} P_l(t(L_i), l(L_i)|P, t(L_{i-1}), l(L_{i-1}), d(i)) \cdot \prod_{i=0}^{n} P_r(t(R_i), l(R_i)|P, t(R_{i-1}), l(R_{i-1}), d(i))$$

If a DG rule is head-right, P is L_i or R_i, if it is head-left, P is L_{i-1} or R_{i-1}, respectively. Headedness and not direction matters. L_i/R_i is replaced by H_i and $L/R_{i-1/i+1}$ by H'. H' is understood to be the DG dependent, although, as mentioned, H' could also be the DG head in this implicit ad-hoc DG.

$$P(RHS|LHS) \cong l(h) \cdot \prod_{i=0}^{n+m} P_{l,r}(t(H_i), l(H_i)|t(H_i), t(H_i'), l(H_i'), d(i))$$

$P(t(H_i)|t(H_i), t(H_i'))$ is a *projection or attachment* grammar model modeling the unlexicalized probability of t(H) and t(H') participating in a binary rule with t(H) as head – the *merge* probability in Bare Phrase Structure (Chomsky, 1995); an unlabeled version of (4). $P(t(H_i), l(H_i)|t(H_i), t(H_i'), l(H_i'))$ is a lexicalized version of the same *projection or attachment* grammar model; $P(t(H_i), l(H_i)|t(H_i), t(H_i'), l(H_i', d(i)))$ in addition conditions on the distance[9]. Pro3Gres expresses the unlexicalized rules by licensing grammar rules for relation R. Tags are not used in Pro3Gres' model, because semantic backoffs and tag-based licensing rules are used.

$$P(d(i)|l(H_i), l(H_i')) \quad (10)$$

The Pro3Gres main MLE estimation (8) ($l(H) = a, l(H') = b$) differs from (10) by using labeled DG, and thus from the Dubey & Keller Model by using a consistent functional DG.

4 Evaluation

(Lin, 1995; Carroll et al., 1999) suggest evaluating on the linguistically meaningful level of dependency relations. Two such evaluations are reported now.

First, a general-purpose evaluation using a hand-compiled gold standard corpus (Carroll et al., 1999), which contains the grammatical relation data of 500 random sentences from the Susanne corpus. The performance (table 3), according to (Preiss, 2003), is similar to a large selection of statistical parsers and a grammatical relation finder. Relations involving LDDs form part of these relations. A selection of them is also given: WH-Subject (WHS), WH-Object (WHO), passive Subject (PSubj), control Subject (CSubj), and the anaphor of the relative clause pronoun (RclSubjA).

[8] In primarily right-branching languages such as English or German (i-1) actually amounts to being the head in the majority of, but not all cases. In a more functional DG perspective such as the one taken in Pro3Gres, these languages turn out to be less right-branching, however, with prepositions or determiners analyzed as markers to the nominal head or complementizers or relative pronouns as markers to the verbal head of the subclause.

[9] Since normalized probabilities are used

$$P(t(H_i), l(H_i)|t(H_i), t(H_i'), l(H_i', d(i))) = P(t(H_i), d(i)|t(H_i), t(H_i'), l(H_i), l(H_i'))$$

CARROLL	Percentages for some relations, general, on Carroll testset					only LDD-involving				
	Subject	Object	noun-PP	verb-PP	subord. clause	WHS	WHO	PSubj	CSubj	RclSubjA
Precision	91	89	73	74	68	92	60	n/a	80	89
Recall	81	83	67	83	n/a	90	86	83	n/a	63
GENIA	Percentages for some relations, general, on GENIA corpus									
	Subject	Object	noun-PP	verb-PP	subord. clause					
Precision	90	94	83	82	71					
Recall	86	95	82	84	75					

Table 3: Evaluation on Carroll's test suite on subj, obj, PP-attachment and clause subord. relations and a selection of 5 LDD relations, and on the terminology-annotated GENIA corpus

Secondly, to answer how the parser performs over domains markedly different to the training corpus, to test whether terminology is the key to a successful parsing system, and to assess the impact of chunking errors, the parser has been applied to the GENIA corpus (Kim et al., 2003), 2000 MEDLINE abstracts of more than 400,000 words describing the results of Biomedical research, which is annotated for multi-word terms and thus contains near-perfect chunking. 100 random sentences from the GENIA corpus have been manually annotated and compared to the parser output (Rinaldi et al., 2004a).

5 Conclusions

We have discussed how DG allows the expression of the majority of LDDs in a context-free way and shown that DG allows for simple but powerful statistical models. An evaluation shows that the performance of its implementation is state-of-the-art[10]. Its parsing speed of about 300,000 words per hour is very good for a deep-linguistic parser and makes it fast enough for unlimited application.

References

Steven Abney. 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.

M. Burke, 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.

Aoife Cahill, 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.

John Carroll, 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.

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

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

Hoojung Chung 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.

Michael Collins and James Brooks. 1995. Prepositional attachment through a backed-off model. In *Proceedings of the Third Workshop on Very Large Corpora*, Cambridge, MA.

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

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

Amit Dubey and Frank Keller. 2003. Probabilistic parsing for German using sister-head dependencies. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo.

Jason Eisner. 2000. Bilexical grammars and their cubic-time parsing algorithms. In Harry

[10] We are currently starting evaluation on the PARC 700 corpus

Bunt and Anton Nijholt, editors, *Advances in Probabilistic and Other Parsing Technologies*. Kluwer.

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

James Henderson. 2003. Inducing history representations for broad coverage statistical parsing. In *Proceedings of HLT-NAACL 2003*, Edmonton, Canada.

Julia Hockenmaier 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.

Richard Hudson. 1984. *Word Grammar*. Basil Blackwell, Oxford.

Mark Johnson. 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.

J.D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. 2003. Genia corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(1):i180–i182.

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

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

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

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

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

Diego Mollá, Gerold Schneider, Rolf Schwitter, and Michael Hess. 2000. Answer Extraction using a Dependency Grammar in ExtrAns. *Traitement Automatique de Langues (T.A.L.), Special Issue on Dependency Grammar*, 41(1):127–156.

Peter Neuhaus 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.

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

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

Stefan Riezler, 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.

Fabio Rinaldi, 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, 21–26 July.

Fabio Rinaldi, Michael Hess, James Dowdall, Diego Mollá, and Rolf Schwitter. 2004b. Question answering in terminology-rich technical domains. In Mark Maybury, editor, *New Directions in Question Answering*. MIT/AAAI Press.

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

Gerold Schneider. 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.

Wojciech Skut, Brigitte Krenn, Thorsten Brants, and Hans Uszkoreit. 1997. An annotation scheme for free word order languages. In *Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP-97)*, Washington, DC.

Pasi Tapanainen and Timo Järvinen. 1997. A non-projective dependency parser. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, pages 64–71. Association for Computational Linguistics.

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