A Robust and Hybrid Deep-Linguistic Theory Applied to Large-Scale Parsing

Gerold Schneider, James Dowdall, Fabio Rinaldi
Institute of Computational Linguistics, University of Zurich
{gschneid,rinaldi}@ifi.unizh.ch, j.m.dowdall@sussex.ac.uk

Abstract

Modern statistical parsers are robust and quite fast, but their output is relatively shallow when compared to formal grammar parsers. We suggest to extend statistical approaches to a more deep-linguistic analysis while at the same time keeping the speed and low complexity of a statistical parser. The resulting parsing architecture suggested, implemented and evaluated here is highly robust and hybrid on a number of levels, combining statistical and rule-based approaches, constituency and dependency grammar, shallow and deep processing, full and near-full parsing. With its parsing speed of about 300,000 words per hour and state-of-the-art performance the parser is reliable for a number of large-scale applications discussed in the article.

1 Introduction

Robustness in Computational Linguistics has been recently recognized as a central issue for the design of reliable, large-scale Natural Language Processing (NLP) systems. While the highest possible linguistic coverage is desirable, speed and robustness are equally important in practical applications.

Formal Grammar Parser have carefully crafted grammars written by professional linguists. In addition to expressing local relations, i.e. relations between a mother and a direct daughter node, a number of non-local relations, i.e. relations involving more than two generations, are also modeled. An example of a non-local relation is the *subject control* relation in the sentence *John wants to leave*, where *John* is not only the explicit subject of *want*, but equally the implicit subject of *leave*. A parser that fails to recognize control subjects misses important information, quantitatively about 3 % of all subjects.

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) for an exception, and (Burke et al., 2004; Hockenmaier and Steedman, 2002) for approaches extracting formal grammars from the Treebank), 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. Note that statistical parsers can equally suffer from this problem, see e.g. (Kaplan et al., 2004).

Parsing complexity Typical formal grammar parser complexity is much higher than the O(n^3) for CFG (Eisner, 1997). 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 et al., 2000) state that the theoretical bound of worst time complexity for Head-Driven Phrase Structure Grammar (HPSG) parsing is exponential. 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 really what is expected of a syntactic analyzer if it should be of practical use, since for a human there is usually only one "correct" interpretation. A clear indication of preference, by means of ranking the analyses in a preference order is needed.

Pruning In order to keep search spaces manageable it is in fact necessary to discard unconvincing alternatives already during the parsing process. In a statistical parser, the ranking of intermediate structures occurs naturally, while a rule-based system has to rely on ad hoc heuristics. With a beam search in a parse-time pruning system, which means that the total number of alternatives kept is constant from a certain search complexity onwards, real-world parsing time can be reduced to near-linear. If one were to assume a constantly full beam, or uses an oracle (Nivre, 2004) it is linear in practice.

A number of robust statistical parsers that offer solutions to these problems have now become available (Charniak, 2000; Collins, 1999; Henderson, 2003), but they typically produce CFG constituency data as output, trees that do not express long-distance dependencies.

Although grammatical function and empty 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[1], 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) provocatively refers to them as "half-grammars".

The paper is organized as follows. We first explore a deep-linguistic grammar theory for English that is inherently designed to be robust by extending the low processing complexity and the robustness of statistical approaches to a more deep-linguistic level, by making careful use of underspecification, grammar compression techniques and using a grammar that directly delivers simple predicate-argument structures. This allow us to use a context-free grammar at parse-time while successfully treating long-distance dependencies using low-complexity approaches before and after parsing. Our approach is to use finite-state approximations of long-distance dependencies, as they are described in (Schneider, 2003a) for Dependency Grammar (DG) and (Cahill et al., 2004) for Lexical Functional Grammar (LFG). (Dienes and Dubey, 2003) show that finite-state preprocessing modules can successfully deal with LDDs. Our approach is similar in also amounting to a preprocessing recognition of LDDs.

Then we show that the implementation (Pro3Gres) profits from hybridness and is fast and robust enough to do large-scale parsing of totally unrestricted texts and give an overview of its applications. To conclude, two evaluations are given.

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

2 A Robust Deep-Linguistic Theory

Generally, a linguistic analysis model aims at complete and correct analysis, which means that the mapping between the text data and its syntactic and semantic analysis is sound (the model extracts correct readings) and complete (the model deals with all language phenomena). In practice, however, both goals cannot be totally reached. The main obstacle for soundness is the all-pervasive characteristic of natural language to be ambiguous, where ambiguities can often only be resolved with world knowledge.

Statistical disambiguation such as (Collins and Brooks, 1995) for PP-attachment or (Collins, 1997; Charniak, 2000) for generative parsing greatly improve disambiguation, but as they model by imitation instead of by understanding, complete soundness has to remain elusive.

As for completeness, already early "naïve" statistical approaches have shown that the problem of grammar size is not solved but even aggravated by a naive probabilistic parser implementation, in which e.g. all CFG rules permitted in the Penn Treebank are extracted. From his 300,000 words training part of the Penn Treebank (Charniak, 1996) obtains more than 10,000 CFG rules, of which only about 3,000 occur more than once. It is therefore necessary to either discard infrequent rules, do manual editing, use a different rule format such as individual dependencies (Collins, 1996) or gain full linguistic control and insight by using a hand-written grammar – each of which sacrifices total completeness.

2.1 Near-full Parsing

The approach we have chosen is to use a manually-developed wide-coverage tag sequence grammar (Abney, 1995; Briscoe and Carroll, 2002), and to exclude or restrict rare, marked and error-prone phenomena. For example, while it is generally possible for nouns to be modified by more than one PP, only nouns seen in the Treebank with several PPs are allowed to have several PPs. Or, while it is generally possible for a subject to occur to the immediate right of a verb (*said she*), this is only allowed for verbs seen with a subject to the right in the training corpus, typically verbs of utterance, and only in a comma-delimited or sentence-final context. This entails that the parser profits from a lean grammar but finds a complete structure spanning the entire sentence in the majority of real-world sentences and needs to resorts to collecting partial parses in the remaining minority. Starting from the most probable longest span, recursively the most probable longest span to left and right is searched.

Near-full parsing only leads to a very small loss. If an analysis consists of two partial parses, on the dependency relation level only the one, usually high-level relation between the heads of the two partial parses remains unexpressed. The risk of returning "garden path", locally correct but globally wrong, analyses diminishes with increasing span length.

2.2 Functional Dependency Grammar

We follow the broad architecture suggested by (Abney, 1995) which naturally integrates chunking and dependency parsing and has proven to be practical, fast and robust (Collins, 1996; Basili and Zanzotto, 2002). Tagging and chunking are very robust, finite-state approaches, parsing then only occurs between heads of chunks.[2] The perspicuous rules of a handwritten dependency grammar build up the possible syntactic structures, which are ranked and pruned by calculating lexical attachment probabilities for the majortiy of the dependency relations used in the grammar. The grammar contains around 1000 rules containing the dependent's and the head's tag, the direction of the dependency, lexical information for closed class words, and context restrictions[3]. Context restrictions express e.g. that only a verb which has an object in its context is allowed to attach a secondary object.

Our approach can be seen as an extension of (Collins and Brooks, 1995) from PP-attachment to most dependency relations. Training data is a partial mapping of the Penn Treebank to deep-linguistic dependency structures, similar to (Basili et al., 1998).

Robustness also depends on the grammar formalism. While many formalisms fail to project when subcategorized arguments cannot be found, in a grammar like DG, in which maximal projections and terminal nodes are isomorphic, projection can never fail.

In classical DG, only content words can be heads, and there is no distinction between syntactic and semantic dependency – semantic dependency is used as far as possible. These assumptions entail that there are no functional and no empty nodes, which means that low complexity $O(n^3)$ algorithms such as CYK, which is used here, can be employed.

The classical dependency grammar distinction between *ordre linéaire* and *ordre structural*, basically an immediate dominance / linear precedence distinction (ID/LP) also has the advantage that a number of phenomena classically assumed to involve long-distance dependencies, fronted or inversed constituents, can be treated locally. They only need rules that allow an inversion of the "canonical" dependency direction under well-defined conditions. As for fronted elements, since DG does not distinguish between external and internal arguments, front positions are always locally available to the verb.

2.3 Underspecification and Disambiguation

The cheapest approach to dealing with the all-pervasive NL ambiguity is to underspecifiy everything, which leads to a sound and complete mapping, but one that is content-free and absurd. But in few, carefully selected areas where distinctions do not matter for the task at hand, where the disambiguation task is particularly unreliable, or where inter-annotator agreement is very low, underspecification can serve as a tool to greatly facilitate linguistic analysis. For example, intra-base NP ambiguities, such as quantifier scope ambiguities do not matter for a parser like ours aiming at predicate-argument structure, and are thus not attempted to analyze. There is one part-of-speech distinction where inter-annotator agreement is quite low and the performance of taggers generally very poor: the distinction between verbal particles and prepositions. We currently leave the distinction underspecified, but a statistical disambiguator is being developed.

Conversely, the Penn Treebank annotation is sometimes not specific enough. The parser distinguishes between the reading of the tag *IN* as a complementizer or as a preposition, and disambiguates commas as far as it can, between

[2]Practical experiments using a toy NP and verb-group grammar have shown that parsing between heads of chunks only is about four times faster than parsing between every word, i.e. without chunking.

[3]the number of rules is high because of tag combinatorics leading to many almost identical rules. A subject relations is e.g. possible between the 6 verb tags and the 4 noun tags

apposition, subordination and conjunction.

Some typical tagging errors can be robustly corrected by the hand-written grammar. For example, the distinction between verb past tense *VBD* and participle *VBN* is unreliable, but can usually be disambiguated in the parsing process by leaving this tag distinction underspecified for a number of constructions.

2.4 Long-distance Dependencies

Long-distance dependencies exponentially increase parsing complexity (Neuhaus and Bröker, 1997). We therefore use an approach that preprocesses, post-processes and partly underspecifies them, allowing us to use a context-free grammar at parse time.

In detail, (1) before the parsing we model dedicated patterns across several levels of constituency subtrees partly leading to dedicated, compressed and fully local dependency relations, (2) we use statistical lexicalized post-processing, and (3) we rely on traditional Dependency Grammar assumptions (section 2.2).

2.4.1 Pre-processing

(Johnson, 2002) presents a pattern-matching algorithm for post-processing the output of statistical parsers to add empty nodes to their parse trees. While encouraging results are reported for perfect parses, performance drops considerably when using trees produced by a statistical parser. “If the parser makes a single parsing error anywhere in the tree fragment matched by the pattern, the pattern will no longer match. This is not unlikely since the statistical model used by the parser does not model these larger tree fragments. It suggests that one might improve performance by integrating parsing, empty node recovery and antecedent finding in a single system ... ” (Johnson, 2002).

We have applied structural patterns to the Penn 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 – like (Jijkoun, 2003) – relying on functional labels and empty nodes reach precision close to 100%. Unlike in Johnson, also patterns for local dependencies are 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

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 1: The most important dependency types used by the parser

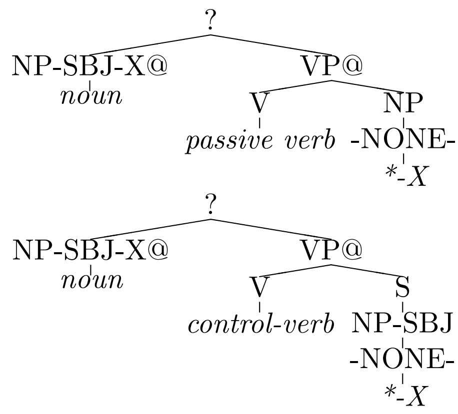

Figure 1: The extaction patterns for passive subjects (top) and subject control (bottom)

DG (Hajič, 1998), (Tapanainen and Järvinen, 1997). 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 1, where VP@ is an arbitrarily nested VP, and NP-SBJ-X@ the arbitrarily nested surface subject and X the co-indexed, moved element. Movements are generally supposed to be of arbitrary length, but a closer investigation reveals that this type of movement is fixed.

The same argument can be made for other relations, for example control structures, which have the extraction pattern shown in figure 1. Grammatical role labels, empty node labels and tree configurations spanning several local subtrees are used as integral part of some of the patterns. This leads to much flatter trees, as typical for DG, which has the advantages that (1) it helps to alleviate sparse data by mapping nested structures that express the same dependency relation, (2) the costly overhead for dealing with unbounded dependencies can be largely avoided, (3) 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, which greatly reduces complexity and the risk of errors (Johnson, 2002). Collapsing deeply nested structures into a single dependency relation is less complex but has the same effect as carefully selecting what goes in to the parse history in history-based approaches. "Much of the interesting work is determining what goes into [the history] H(c)" (Charniak, 2000).

(Schneider, 2003a) shows that the vast majority of LDDs can be treated in this way, essentially compressing non-local subtrees into dedicated relations even before grammar writing starts. The compressed trees correspond to a simple LFG f-structure. The trees obtained from parsing can be decompressed into traditional constituency trees including empty nodes and co-indexation, or into shallow semantic structures such as Minimal Logical Forms (MLF) (Rinaldi et al., 2004b; Schneider et al., 2000; Schwitter et al., 1999). This approach leaves LDDs underspecified, but recoverable, and makes no claims as to whether empty nodes at an automonous syntactic level exist or not.

2.4.2 Post-Processing

After parsing, shared constituents can be extracted again. The parser explicitly does this for control, raising and semi-auxiliary relations, because the grammar does not distinguish between subordinating clauses with and without control. A probability model based on the verb semantics is invoked if a subordinate clause without overt subject is seen, in order to decide whether the matrix clause subject or object is shared.

2.4.3 What do we lose?

Among the 10 most frequent types of empty nodes, which cover more than 60,000 of the 64,000 empty nodes in the Penn treebank, there are only two problematic LDD types: WH Traces and indexed gerunds.

WH traces Only 113 of the 10,659 WHNP antecedents in the Penn Treebank 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. This method succeeds in most cases, but linguistic non-standard assumptions need to be made for stranded prepositions.

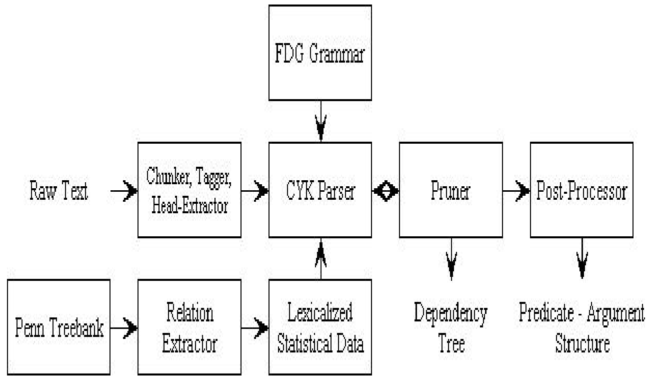

Figure 2: Pro3Gres flowchart

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.

Indexed Gerunds Unlike in control, raising and semi-auxiliary constructions, the antecedent of an indexed gerund cannot be established easily. The fact that almost half of the gerunds are non-indexed in the Penn Treebank indicates that information about the unexpressed participant is rather semantic than syntactic in nature, much like in pronoun resolution. Currently, 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.

3 Robustness "in the small"

In addition to a robust deep-linguistic design (robustness "in the large", section 2), the implemented parser, Pro3Gres, uses a number of practical robust approaches "in the small" at each processing level, such as relying on finite-state tagging and chunking or collecting partial parses if no complete analysis can be found, or using incrementally more aggressive pruning techniques in very long sentences. During the parsing process, only a certain number of alternatives for each possible span are kept. Experiments have shown that using a fixed number or a number dependent on the parsing complexity in terms of global chart entries lead to very similar results. Using reasonable beam sizes increases parsing speed by an order of magnitude while hardly affecting parser performance. For the fixed number model, performance starts to collapse only when less than 4 alternatives per span are kept.

When a certain complexity has been reached (currently 1000 chart entries), only reductions above a certain probability threshold are permissible. The threshold starts very low, but is a function of the total number of chart entries. This entails that even sentences with hundreds of words can be parsed quickly, but it is not aimed at finding complete parses for them, rather a *graceful degradation* of performance (Menzel, 1995) is intended.

4 A hybrid approach on many levels

Pro3Gres profits from being hybrid on many levels. Hybridness means that the most robust approach can be chosen for each task and each processing level.

statistical vs. rule-based the most obvious way in which Pro3Gres is a hybrid (Schneider, 2003b). Unlike formal grammars to which post-hoc statistical disambiguators can be added, Pro3Gres has been designed to be hybrid, carefully distinguishing between tasks that can best be solved by finite-state methods, rule-based methods and statistical methods. While e.g. grammar writing is easy for a linguist, and a naive Treebank grammar suffers from similar complexity problems as a comprehensive formal grammar, the scope of application and the amount of ambiguity a rule creates is often beyond our imagination and best handled by a statistical system.

shallow vs. deep the designing philosophy for Pro3Gres has been to stay as shallow as possible to obtain reliable results at each level.

Treebank constituency vs. DG the observation that a DG that expresses grammatical relations is more informative, but also more intuitive to interpret for a non-expert, and that Functional DG can avoid a number of LDD types has made DG the formalism of our choice. For lexicalizing the grammar, a partial mapping from the largest manually annotated corpus available, the Penn Treebank, was necessary, exhibiting a number of mapping challenges.

history-based vs. mapping-based Pro3Gres is not a parse-history-based approach. Instead of manually selecting what goes into the history, as is usually done (see (Henderson, 2003) for an exception), we manually select how to linguistically meaningfully map Treebank structures onto dependency relations by the use of mapping patterns adapted from (Johnson, 2002).

probabilistic vs. statistical Pro3Gres is not a probabilistic system in the sense of a PCFG. From a practical viewpoint, knowing the probability of a certain rule expansion per se is of little interest. Pro3Gres models decision probabilities, the probability of a parse is understood to be the product of all the decision probabilities taken during the derivation.

local subtress vs. DOP psycholinguistic experiments and Data-Oriented Parsing (DOP) (Bod et al., 2003) suggest that people store subtrees of various sizes, from two-word fragments to entire sentences. But (Goodman, 2003) suggests that the large number of subtrees can be reduced to a compact grammar that makes DOP parsing computationally tractable. In Pro3Gres, a subset of non-local fragments which, based on linguistic intuition are especially important, are used.

generative vs. structure-generating DG generally, although generative in the sense that connected complete structures are generated, is not generative in the sense that it is always guaranteed to terminate if used for random generation of language. Since a complete or partial hierarchical structure that follows CFG assumptions due to the employed grammar is built up for each sentence. Pro3Gres' constraint to allow each complement dependency type only once per verb can be seen as a way of rendering it generative in practice.

syntax vs. semantics instead of using a back-off to tags (Collins, 1999), semantic classes, Wordnet for nouns and Levin classes for verbs, are used, in the hope that they better manage better to express selectional restrictions than tags. Practical experiments have shown, however, that, in accordance to (Gildea, 2001) on head-lexicalisation, there is almost no increase in performance.

5 Applications and Evaluation

Pro3Gres is currently being applied in a Question Answering system specifically targeted at

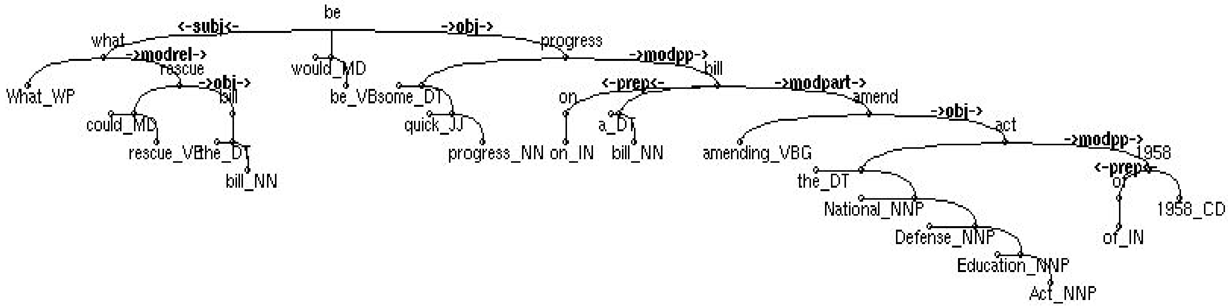

Figure 3: Dependency Tree output of the SWI Prolog graphical implementation of the parser

technical domains (Rinaldi et al., 2004b). One of the main advantages of a dependency-based 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; Rinaldi et al., 2002).

The original version of the QA system used the Link Grammar (LG) parser (Sleator and Temperley, 1993), which however had a number of significant shortcomings. In particular the set of the dependency relations used in LG is very idiosyncratic, which makes any syntactic-semantic mapping created for LG necessarily unportable and difficult to extend and maintain.

A recent line of research concerns applications for the Semantic Web. The documents available in the World Wide Web are mostly written in natural language. As such, they are understandable only to humans. One of the directions of Semantic Web research is about adding a layer to the documents that somehow formalizes their content, making it understandable also to software agents. Such Semantic Web annotations can be seen as a way to mark explicitly the meaning of certain parts of the documents. The dependency relations provided by a parser such as Pro3Gres, combined with domain specific axioms, allow the creation of (some of) the semantic annotations, as described in (Rinaldi et al., 2003; Kaljurand et al., 2004).

The modified QA system (using Pro3Gres) is being exploited in the area of 'Life Sciences', for applications concerning Knowledge Discovery over Medline abstracts (Rinaldi et al., 2004a; Dowdall et al., 2004). We illustrate some of the differences between general-purpose parsing and the parsing of highly technical texts like Medline and give two evaluations.

5.1 General unrestricted texts

We first report an evaluation on sentences from an open domain, which gives a good impression of the performance of the parser on general, unrestricted text.

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; Carroll et al., 1999) suggest evaluating on the linguistically more meaningful level of syntactic relations. Two evaluations on the syntactic relation level are reported in the following. 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, shown in table 2, is, according to (Preiss, 2003), similar to a large selection of statistical parsers and a grammatical relation finder. Relations involving long-distance dependencies form part of these relations. In order to measure specifically their performance, 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).

5.2 Parsing highly technical language

While measuring general parsing performance is fundamental in the development of any parsing system there is a danger of fostering domain dependence in concentrating on a single domain.

In order to answer how the parser performs over domains markedly different to the training corpus , 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.

Average sentence length is 27 words, the lan-

	Percentage Values for some relations, general, on Carroll corpus					only LDD-involving				
	Subject	Object	noun-PP	verb-PP	subord. cl.	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

Table 2: Results of evaluating the parser output on Carroll's test suite on subject, object, PP-attachment and clause subordination relations, and a selective evaluation of 5 relations involving long-distance dependencies (LDD)

	Percentage Values for some relations, general, on the GENIA corpus				
	Subject	Object	noun-PP	verb-PP	subord. clause
Precision	90	94	83	82	71
Recall	86	95	82	84	75

Table 3: Results of evaluating 100 random sentences from the terminology-annotated GENIA corpus, on subject, object, PP-attachment and clause subordination relations

guage is very technical and extremely domain-specific. But the most striking characteristic of this domain is the frequency of MultiWord Terms (MWT) which are known to cause serious problems for NLP systems (Sag et al., 2002), (Dowdall et al., 2003). The token to chunk ratio: NPs = 2.3 , VPs = 1.3 (number of tokens divided by the number of chunks) is unusually high.

The GENIA corpus does not include any syntactic annotation (making standard evaluation more difficult) but approx. 100, 000 MWTs are annotated and assigned a semantic type from the GENIA ontology.

This novel parsing application is designed to determine how parsing performance interacts with MWT recognition as well as the applicability and possible improvements to the probablistic model over this domain, to test the hypothesis if terminology is the key to a successful parsing system. We do not discard this information, thus simulating a situation in which a near-perfect terminology-recognition tool is at one's disposal. MWT are regarded as chunks, the parsing thus takes place between between the heads of MWT, words and chunks.

100 random sentences from the GENIA corpus have been manually annotated for this evaluation and compared to the parser output. Despite the extreme complexity and technical language, parsing performance under these conditions is considerably better than on the Carroll corpus when using automated chunking, as table 3 reveals.

It is worth noting that 10 of the 17 subject precision errors (out of 171 subjects) are "hard" cases involving long-distance dependencies (1 control, 4 relative pronouns) and 5 verb group chunking errors. Equally interesting, 2 of the 4 object recall errors (out of 79 objects) are due to 1 mistagging and 1 mischunking.

In practice, MWT extraction is still not automated to the level of chunking or Name Entity recognition simulated in this experiment (for a comprehensive review of the state-of-the-art see (Castellv et al., 2001)). This is, in a large part, due to the lack of definitive orthographic, morphological and syntactic characteristics to differentiante between MWTs and canonical phrases. So MWT extraction remains a semi-automated task performed in cycles with the result of each cycle requiring manual validation. The return for this time consuming activity are the characteristics of MWTs which can be use to fine tune the algorithms during the next extraction cycle.

6 Conclusion

We have suggested a robust, deep-linguistic grammar theory delivering grammatical relation structures as output, which are closer to predicate-argument structures than pure constituency structures, and more informative if non-local dependencies are involved. We have presented an implementation of the theory that is used for large-scale parsing. An evaluation at the grammatical relation level shows that its performance is state-of-the-art.

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.

Roberto Basili and Fabio Massimo Zanzotto. 2002.

Parsing engineering and empirical robustness. *Journal of Natural Language Engineering*, 8/2-3.

Roberto Basili, Maria Teresa Pazienza, and Fabio Massimo Zanzotto. 1998. Evaluating a robust parser for Italian language. In *Proceedings of Evaluations of Parsing Systems Workshop, held jointly with 1st LREC*, Granada,Spain.

Rens Bod, Remko Scha, and Khalil Sima'an, editors. 2003. *Data-Oriented Parsing.* Center for the Study of Language and Information, Studies in Computational Linguistics (CSLI-SCL). Chicago University Press.

Ted Briscoe and John Carroll. 2002. Robust accurate statistical annotation of general text. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, pages 1499–1504, Las Palmas, Gran Canaria.

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, Hainan Island, 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.

M. Teresa Cabré Castellv, Rosa Estopá, and Jordi Vivaldi Palatresi, 2001. *Recent Advances in Computational Terminology*, chapter Automatic term detection: A review of current systems, pages 53–87. John Benjamins.

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

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

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. 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.

Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proc. of the 35th Annual Meeting of the ACL*, pages 16–23, Madrid, Spain.

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

Pter Dienes and Amit Dubey. 2003. Antecedent recovery: Experiments with a trace tagger. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Sapporo, Japan.

James Dowdall, Fabio Rinaldi, Fidelia Ibekwe-Sanjuan, and Eric SanJuan. 2003. Complex structuring of term variants for question answering. In *Proceedings of the ACL workshop on MultiWord Expressions: Analysis, Acquisition and Treatment*, Sapporo, Japan, July.

James Dowdall, Fabio Rinaldi, Andreas Persidis, Kaarel Kaljurand, Gerold Schneider, and Michael Hess. 2004. Terminology expansion and relation identification between genes and pathways. In *Workshop on Terminology, Ontology and Knowledge Representation.* Universite Jean Moulin (Lyon 3).

Jason Eisner. 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.

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

Daniel Gildea. 2001. Corpus variation and parser performance. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 167–202, Pittsburgh, PA.

Joshua Goodman. 2003. Efficient parsing of DOP with PCFG-reductions. In Bod et al. (Bod et al., 2003).

Jan Hajič. 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*, pages 106–132. Karolinum, Charles University Press, Prague.

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.

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

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.

Kaarel Kaljurand, Fabio Rinaldi, James Dowdall, and Michael Hess. 2004. Exploiting language resources for semantic web annotations. In *Proceedings of LREC 2004, Lisbon, May 24-30.* accepted for publication.

Ron Kaplan, 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.

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.

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

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

Wolfgang Menzel. 1995. Robust processing of natural language. *Lecture Notes in Computer Science*, 981:19–34.

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, Michael Hess, Diego Mollá, and Rolf Schwitter. 2002. Towards Answer Extraction: an application to Technical Domains. In *ECAI2002, European Conference on Artificial Intelligence, Lyon*, pages 460–464.

Fabio Rinaldi, Kaarel Kaljurand, James Dowdall, and Michael Hess. 2003. Breaking the deadlock. In *ODBASE 2003 (International Conference on Ontologies, Databases and Applications of Semantics) Catania, Italy.*, volume 2889 of *Lecture Notes in CS*. Springer Verlag.

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.

Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword Expressions: a Pain in the Neck for NLP. In *Proceedings of the Third International Conference, CICLing 2002*, pages 1–15, Mexico City, Februrary.

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

Gerold Schneider, Diego Mollà Aliod, and Michael Hess. 2000. Inkrementelle minimale logische formen fr die antwortextraktion. In *Proceedings of 34th Linguistic Colloquium, September 1999*, University of Mainz, FASK.

Gerold Schneider. 2003a. 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.

Gerold Schneider. 2003b. A low-complexity, broad-coverage probabilistic dependency parser for English. In *Proceedings of HLT-NAACL 2003 Student session*, Edmonton, Canada.

Rolf Schwitter, Diego Mollá Aliod, and Michael Hess. 1999. ExtrAns - answer extraction from technical documents by minimal logical forms and selective highlighting. In *Proceedings of The Third International Tbilisi Symposium on Language, Logic and Computation*, Batumi, Georgia.

Daniel D. Sleator and Davy Temperley. 1993. Parsing English with a link grammar. In *Proc. Third International Workshop on Parsing Technologies*, pages 277–292.

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.