Deep dependencies from context-free statistical parsers: correcting the surface dependency approximation

Roger Levy
Department of Linguistics
Stanford University
rog@stanford.edu

Christopher D. Manning
Departments of Computer Science and Linguistics
Stanford University
manning@cs.stanford.edu

Abstract

We present a linguistically-motivated algorithm for reconstructing nonlocal dependency in broad-coverage context-free parse trees derived from treebanks. We use an algorithm based on loglinear classifiers to augment and reshape context-free trees so as to reintroduce underlying nonlocal dependencies lost in the context-free approximation. We find that our algorithm compares favorably with prior work on English using an existing evaluation metric, and also introduce and argue for a new dependency-based evaluation metric. By this new evaluation metric our algorithm achieves 60% error reduction on gold-standard input trees and 5% error reduction on state-of-the-art machine-parsed input trees, when compared with the best previous work. We also present the first results on non-local dependency reconstruction for a language other than English, comparing performance on English and German. Our new evaluation metric quantitatively corroborates the intuition that in a language with freer word order, the surface dependencies in context-free parse trees are a poorer approximation to underlying dependency structure.

1 Introduction

While parsers are been used for other purposes, the primary motivation for syntactic parsing is as an aid to semantic interpretation, in pursuit of broader goals of natural language understanding. Proponents of traditional 'deep' or 'precise' approaches to syntax, such as GB, CCG, HPSG, LFG, or TAG, have argued that sophisticated grammatical formalisms are essential to resolving various hidden relationships such as the source phrase of moved *wh*-phrases in questions and relativizations, or the controller of clauses without an overt subject. Knowledge of these hidden relationships is in turn essential to semantic interpretation of the kind practiced in the semantic parsing (Gildea and Jurafsky, 2002) and QA (Pasca and Harabagiu, 2001) literatures. However, work in statistical parsing has for the most part put these needs aside, being content to recover surface context-free (CF) phrase structure trees. This perhaps reflects the fact that context-free phrase structure grammar (CFG) is in some sense at the the heart of the majority of both formal and computational syntactic research. Although, upon introducing it, Chomsky (1956) rejected CFG as an adequate framework for natural language description, the majority of work in the last half century has used context-free structural descriptions and related methodologies in one form or another as an important component of syntactic analysis. CFGs seem adequate to weakly generate almost all common natural language structures, and also facilitate a *transparent* predicate-argument and/or semantic interpretation for the more basic ones (Gazdar et al., 1985). Nevertheless, despite their success in providing surface phrase structure analyses, if statistical parsers and the representations they produce do not provide a useful stepping stone to recovering the hidden relationships, they will ultimately come to be seen as a dead end, and work will necessarily return to using richer formalisms.

In this paper we attempt to establish to what degree current statistical parsers are a useful step in analysis by examining the performance of further statistical classifiers on non-local dependency recovery from CF parse trees. The natural isomorphism from CF trees to dependency trees induces only *local* dependencies, derived from the head-sister relation in a CF local tree. However, if the output of a context-free parser can be algorithmically augmented to accurately identify and incorporate nonlocal dependencies, then we can say that the context-free parsing model is a *safe approximation* to the true task of dependency reconstruction. We investigate the safeness of this approximation, devising an algorithm to reconstruct non-local dependencies from context-free parse trees using loglinear classifiers, tested on treebanks of not only English but also German, a language with much freer word order and correspondingly more discontinuity than English. This algorithm can be used as an intermediate step between the surface output trees of modern statistical parsers and semantic interpretation systems for a variety of tasks.[1]

[1]Many linguistic and technical intricacies are involved in the interpretation and use of non-local annotation structure found in treebanks. A more complete exposition of the work presented here can be found in Levy (2004).

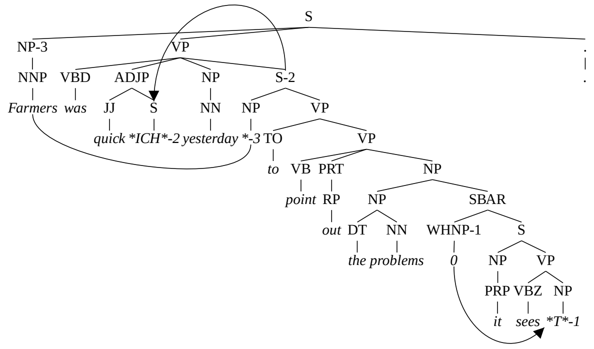


Figure 1: Example of empty and nonlocal annotations from the Penn Treebank of English, including null complementizers (*0*), relativization (**T*-1*), right-extraposition (**ICH*-2*), and syntactic control (**-3*).

1.1 Previous Work

Previous work on nonlocal dependency has focused entirely on English, despite the disparity in type and frequency of various non-local dependency constructions for varying languages (Kruijff, 2002). Collins (1999)'s Model 3 investigated GPSG-style trace threading for resolving nonlocal relative pronoun dependencies. Johnson (2002) was the first post-processing approach to non-local dependency recovery, using a simple pattern-matching algorithm on context-free trees. Dienes and Dubey (2003a,b) and Dienes (2003) approached the problem by pre-identifying empty categories using an HMM on unparsed strings and threaded the identified empties into the category structure of a context-free parser, finding that this method compared favorably with both Collins' and Johnson's. Traditional LFG parsing, in both non-stochastic (Kaplan and Maxwell, 1993) and stochastic (Riezler et al., 2002; Kaplan et al., 2004) incarnations, also divides the labor of local and nonlocal dependency identification into two phases, starting with context-free parses and continuing by augmentation with functional information.

2 Datasets

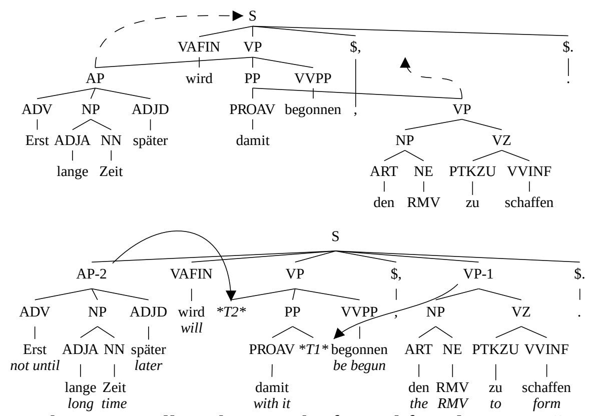
The datasets used for this study consist of the Wall Street Journal section of the Penn Treebank of English (WSJ) and the context-free version of the NEGRA (version 2) corpus of German (Skut et al., 1997b). Full-size experiments on WSJ described in Section 4 used the standard sections 2-21 for training, 24 for development, and trees whose yield is under 100 words from section 23 for testing. Experiments described in Section 4.3 used the same development and test sets but files 200-959 of WSJ as a smaller training set; for NEGRA we followed Dubey and Keller (2003) in using the first 18,602 sentences for training, the last 1,000 for development, and the previous 1,000 for testing. Consistent with prior work and with common practice in statistical parsing, we stripped categories of all functional tags prior to training and testing (though in several cases this seems to have been a limiting move; see Section 5).

Nonlocal dependency annotation in Penn Treebanks can be divided into three major types: *unindexed empty* elements, *dislocations*, and *control*. The first type consists primarily of null complementizers, as exemplified in Figure 1 by the null relative pronoun *0* (c.f. *aspects* ***that*** *it sees*), and do not participate in (though they may mediate) nonlocal dependency. The second type consists of a *dislocated element* coindexed with an *origin site* of semantic interpretation, as in the association in Figure 1 of WHNP-1 with the direct object position of *sees* (a relativization), and the association of S-2 with the ADJP *quick* (a right dislocation). This type encompasses the classic cases of nonlocal dependency: topicalization, relativization, *wh*- movement, and right dislocation, as well as expletives and other instances of non-canonical argument positioning. The third type involves *control loci* in syntactic argument positions, sometimes coindexed with overt *controllers*, as in the association of the NP *Farmers* with the empty subject position of the S-2 node. (An example of a control locus with no controller would be $[_{\mathrm{S}}$ NP-* $[_{\mathrm{VP}}$ *Eating ice cream*]] *is fun*.) Controllers are to be interpreted as syntactic (and possibly semantic) arguments both in their overt position and in the position of loci they control. This type encompasses raising, control, passivization, and unexpressed subjects of *to*- infinitive and gerund verbs, among other constructions.[2]

NEGRA's original annotation is as dependency trees with phrasal nodes, crossing branches, and no empty elements. However, the distribution includes a context-free version produced algorithmically by recursively remapping discontinuous parts of nodes upward into higher phrases and marking their sites of origin.[3] The resulting "traces" correspond roughly to a subclass of the second class of Penn Treebank empties discussed above, and include *wh*- movement, topicalization, right extrapositions from NP, expletives, and scrambling of sub-

[2]Four of the annotation errors in WSJ lead to uninterpretable dislocation and sharing patterns, including failure to annotate dislocations corresponding to marked origin sites, and mislabelings of control loci as origin sites of dislocation that lead to cyclic dislocations (which are explicitly *prohibited* in WSJ annotation guidelines). We corrected these errors manually before model testing and training.

[3]For a detailed description of the algorithm for creating the context-free version of NEGRA, see Skut et al. (1997a).


S
VAFIN VP $, $.
AP wird PP VVPP , VP .
ADV NP ADJD PROAV begonnen NP VZ
Erst ADJA NN später damit ART NE PTKZU VVINF
lange Zeit den RMV zu schaffen
S
AP-2 VAFIN VP $, VP-1 $.
ADV NP ADJD wird *T2* PP VVPP , NP VZ .
will
Erst ADJA NN später PROAV *T1* begonnen ART NE PTKZU VVINF
not until later be begun
lange Zeit damit den RMV zu schaffen
long time with it the RMV to form

"The RMV will not begin to be formed for a long time."

Figure 2: Nonlocal dependencies via right-extraposition (**T1**) and topicalization (**T2**) in the NEGRA corpus of German, before (top) and after (bottom) transformation to context-free form. Dashed lines show where nodes go as a result of remapping into context-free form.

jects after other complements. The positioning of NEGRA's "traces" inside the mother node is completely algorithmic; a dislocated constituent C has its trace at the edge of the original mother closest to C's overt position. Given a context-free NEGRA tree shorn of its trace/antecedent notation, however, it is far from trivial to determine which nodes are dislocated, and where they come from. Figure 2 shows an annotated sentence from the NEGRA corpus with discontinuities due to right extraposition (**T1**) and topicalization (**T2**), before and after transformation into context-free form with traces.

3 Algorithm

Corresponding to the three types of empty-element annotation found in the Penn Treebank, our algorithm divides the process of CF tree enhancement into three phases. Each phase involves the identification of a certain subset of tree nodes to be operated on, followed by the application of the appropriate operation to the node. Operations may involve the insertion of a category at some position among a node's daughters; the marking of certain nodes as dislocated; or the relocation of dislocated nodes to other positions within the tree. The content and ordering of phases is consistent with the syntactic theory upon which treebank annotation is based. For example, WSJ annotates relative clauses lacking overt relative pronouns, such as the SBAR in Figure 1, with a trace in the relativization site whose antecedent is an empty relative pronoun. This requires that empty relative pronoun insertion precede dislocated element identification. Likewise, dislocated elements can serve as controllers of control loci, based on their *originating site*, so it is sensible to return dislocated nodes to their originating sites before identifying control loci and their controllers. For WSJ, the three phases are:

1. (a) Determine nodes at which to insert null COMPlementizers[4] (IDENTNULL)
   (b) For each COMP insertion node, determine position of each insertion and insert COMP (INSERTNULL)
2. (a) Classify each tree node as +/- DISLOCATED (IDENTMOVED)
   (b) For each DISLOCATED node, choose an ORIGIN node (RELOCMOVED)
   (c) For each pair $\langle$DISLOCATED,*origin*$\rangle$, choose a position of insertion and insert *dislocated* (INSERTRELOC)
3. (a) Classify each node as +/- control LOCUS (IDENTLOCUS)
   (b) For each LOCUS, determine position of insertion and insert LOCUS (INSERTLOCUS)
   (c) For each LOCUS, determine CONTROLLER (if any) (FINDCONTROLLER)

Note in particular that phase 2 involves the classification of overt tree nodes as dislocated, followed by the identification of an origin site (annotated in the treebank as an empty node) for each dislocated element; whereas phase 3 involves the identification of (empty) control loci *first*, and of controllers later. This approach contrasts with Johnson (2002), who treats empty/antecedent identification as a joint task, and with Dienes and Dubey (2003a,b), who always identify empties first and determine antecedents later. Our motivation is that it should generally be easier to determine whether an overt element is dislocated than whether a given position is the origin of some yet unknown dislocated element (particularly in the absence of a sophisticated model of argument expression); but control loci are highly predictable from local context, such as the subjectless non-finite S in Figure 1's S-2.[5] Indeed this difference seems to be implicit in the nonlocal feature templates used by Dienes and Dubey (2003a,b) in their empty element tagger, in particular lookback for *wh-* words preceding a candidate verb.

As described in Section 2, NEGRA's nonlocal annotation schema is much simpler, involving no

[4]The WSJ contains a number of SBARs headed by empty complementizers with trace S's. These SBARs are introduced in our algorithm as projections of identified empty complementizers as daughters of non-SBAR categories.

[5]Additionally, whereas dislocated nodes are always overt, control loci may be controlled by other (null) control loci, meaning that identifying controllers before control loci would still entail looking for nulls.

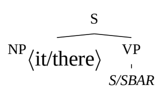
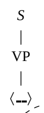
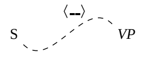
IDENTMOVED		Expletive dislocation
IDENTLOCUS		VP-internal context to determine null subjecthood
INSERTNULLS		Possible null complementizer (records syntactic path from every S in sentence)

Figure 3: Different classifiers' specialized tree-matching fragments and their purposes

uncoindexed empties or control loci. Correspondingly, our NEGRA algorithm includes only phase 2 of the WSJ algorithm, step (c) of which is trivial for NEGRA due to the deterministic positioning of trace insertion in the treebank.

In each case we use a loglinear model for node classification, with a combination of quadratic regularization and thresholding by individual feature count to prevent overfitting. In the second and third parts of phases 2 and 3, when determining an originating site or controller for a given node N, or an insertion position for a node N′ in N, we use a competition-based setting, using a binary classification (yes/no for association with N) on each node in the tree, and during testing choosing the node with the highest score for positive association with N.[6] All other phases of classification involve independent decisions at each node. In phase 3, we include a special zero node to indicate a control locus with no antecedent.

3.1 Feature templates

Each subphase of our dependency reconstruction algorithm involves the training of a separate model and the development of a separate feature set. We found that it was important to include both a variety of general feature templates and a number of manually designed, specialized features to resolve specific problems observed for individual classifiers. We developed all feature templates exclusively on the training and development sets specified in Section 2.

Table 1 shows which general feature templates we used in each classifier. The features are coded as follows. The prefixes {∅,M,G,D,R} indicate that the feature value is calculated with respect to the node in question, its mother, grandmother, daughter, or *relative* node respectively.[7] {CAT,POS,TAG,WORD} stand for syntactic category, position (of daughter) in mother, head tag, and head word respectively. For example, when determining whether an infinitival VP is extraposed, such as S-2 in Figure 1, the plausibility of the VP head being a deep dependent of the head verb is captured with the MHD×HD template. (FIRST/LAST)CAT and (L/RSIS)CAT are templates used for choosing the position to insert insert relocated nodes, respectively recording whether a node of a given category is the first/last daughter, and the syntactic category of a node's left/right sisters. PATH is the syntactic path between relative and base node, defined as the list of the syntactic categories on the (inclusive) node path linking the relative node to the node in question, paired with whether the step on the path was upward or downward. For example, in Figure 2 the syntactic path from VP-1 to PP is [↑-VP,↑-S,↓-VP,↓-PP]. This is a crucial feature for the relativized classifiers RELOCATEMOVED and FINDCONTROLLER; in an abstract sense it mediates the gap-threading information incorporated into GPSG-

Feature type	IdentNull	InsertNull	IdentMoved	RelocMoved	InsertReloc	IdentLocus	InsertLocus	FindController
TAG						✓		✓
HD								✓
CAT×MCAT	⊗							✓
CAT×MCAT×GCAT			✓			✓	✓	
CAT×HD×MCAT×MHD			⊗					
CAT×TAG×MCAT×MTAG			⊗					
CAT×TAG	✓					✓		
CAT×HD						⊗		
(FIRST/LAST)CAT		✓			✓			
(L/RSIS)CAT		✓			✓			
DPOS×CAT		✓						
PATH				✓				✓
CAT×RCAT				✓				
TAG×RCAT				✓				
CAT×TAG×RCAT				✓				
CAT×RCAT×DPOS					✓			
HD×RHD				⊗				
CAT×HD×RHD				✓				
CAT×DCAT	✓			✓		✓		✓
MHD×HD			⊗					
# Special	9	0	11	0	0	12	0	3

Table 1: Shared feature templates. See text for template descriptions. # Special is the number of special templates used for the classifier. ⊗ denotes that all subsets of the template conjunction were included.

[6]The choice of a unique origin site makes our algorithm unable to deal with right-node raising or parasitic gaps. Cases of right-node raising could be automatically transformed into single-origin dislocations by making use of a theory of coordination such as Maxwell and Manning (1996), while parasitic gaps could be handled with the introduction of a secondary classifier. Both phenomena are low-frequency, however, and we ignore them here.

[7]The relative node is DISLOCATED in RELOCMOVED and LOCUS in FINDCONTROLLER.

	Gold trees		Parser output		
	Jn	Pres	Jn	DD	Pres
NP-*	62.4	75.3	55.6	(69.5)	61.1
WH-t	85.1	67.6	80.0	(82.0)	63.3
0	89.3	99.6	77.1	(48.8)	87.0
SBAR	74.8	74.7	71.0	73.8	71.0
S-t	90	93.3	87	84.5	83.6

Table 2: Comparison with previous work using Johnson's PARSEVAL metric. Jn is Johnson (2002); DD is Dienes and Dubey (2003b); Pres is the present work.

style (Gazdar et al., 1985) parsers, and in concrete terms it closely matches the information derived from Johnson (2002)'s connected local tree set patterns. Gildea and Jurafsky (2002) is to our knowledge the first use of such a feature for classification tasks on syntactic trees; they found it important for the related task of semantic role identification.

We expressed specialized hand-coded feature templates as tree-matching patterns that capture a fragment of the content of the pattern in the feature value. Representative examples appear in Figure 3. The italicized node is the node for which a given feature is recorded; underscores __ indicate variables that can match any category; and the angle-bracketed parts of the tree fragment, together with an index for the pattern, determine the feature value.[8]

4 Evaluation

4.1 Comparison with previous work

Our algorithm's performance can be compared with the work of Johnson (2002) and Dienes and Dubey (2003a) on WSJ. Valid comparisons exist for the insertion of uncoindexed empty nodes (COMP and ARB-SUBJ), identification of control and raising loci (CONTROLLOCUS), and pairings of dislocated and controller/raised nodes with their origins (DISLOC,CONTROLLER). In Table 2 we present comparative results, using the PARSEVAL-based evaluation metric introduced by Johnson (2002) – a correct empty category inference requires the *string position* of the empty category, combined with the left and right boundaries plus syntactic category of the antecedent, if any, for purposes of comparison.[9,10] Note that this evaluation metric does not require correct attachment of the empty category into the parse tree. In Figure 1, for example, WHNP-1 could be erroneously remapped to the right edge of any S or VP node in the sentence without resulting in error according to this metric. We therefore abandon this metric in further evaluations as it is not clear whether it adequately approximates performance in predicate-argument structure recovery.[11]

	P_{CF}	P	$A \circ P$	$J \circ P$	D	G	$A \circ G$	$J \circ G$
Overall	91.2	87.6	**90.5**	90.0	88.3	95.7	**99.4**	98.5
NP	91.6	89.9	**91.4**	91.2	89.4	97.9	**99.8**	99.6
S	93.3	83.4	**91.2**	89.9	89.2	89.0	**98.0**	96.0
VP	91.2	87.3	**90.2**	89.6	88.0	95.2	**99.0**	97.7
ADJP	73.1	72.8	**72.9**	72.8	72.5	99.7	**99.6**	98.8
SBAR	94.4	66.7	**89.3**	84.9	85.0	72.6	**99.4**	94.1
ADVP	70.1	69.7	69.5	**69.7**	67.7	99.4	99.4	**99.7**

Table 3: Typed dependency F1 performance when composed with statistical parser. P_{CF} is parser output evaluated by context-free (shallow) dependencies; all others are evaluated on deep dependencies. P is parser, G is string-to-context-free-gold-tree mapping, A is present remapping algorithm, J is Johnson 2002, D is the COMBINED model of Dienes 2003.

4.2 Composition with a context-free parser

If we think of a statistical parser as a function from strings to CF trees, and the nonlocal dependency recovery algorithm A presented in this paper as a function from trees to trees, we can naturally *compose* our algorithm with a parser P to form a function $A \circ P$ from strings to trees whose dependency interpretation is, hopefully, an improvement over the trees from P.

To test this idea quantitatively we evaluate performance with respect to recovery of *typed dependency relations* between words. A dependency relation, commonly employed for evaluation in the statistical parsing literature, is defined at a node N of a lexicalized parse tree as a pair $\langle w_i, w_j \rangle$ where w_i is the lexical head of N and w_j is the lexical head of some non-head daughter of N. Dependency relations may further be typed according to information at or near the relevant tree node; Collins (1999), for example, reports dependency scores typed on the syntactic categories of the mother, head daughter, and dependent daughter, plus on whether the dependent precedes or follows the head. We present here dependency evaluations where the gold-standard dependency set is defined by the *remapped tree*, typed

[8]A complete description of feature templates can be found at *http://nlp.stanford.edu/˜rog/acl2004/templates/index.html*

[9]For purposes of comparability with Johnson (2002) we used Charniak's 2000 parser as P.

[10]Our algorithm was evaluated on a more stringent standard for NP-* than in previous work: control loci-related mappings were done after dislocated nodes were actually relocated by the algorithm, so an incorrect dislocation remapping can render incorrect the indices of a correct NP-* labeled bracketing. Additionally, our algorithm does not distinguish the syntactic category of null insertions, whereas previous work has; as a result, the null complementizer class 0 and WH-t dislocation class are aggregates of classes used in previous work.

[11]Collins (1999) reports 93.8%/90.1% precision/recall in his Model 3 for accurate identification of relativization site in non-infinitival relative clauses. This figure is difficult to compare directly with other figures in this section; a tree search indicates that non-infinitival subjects make up at most 85.4% of the WHNP dislocations in WSJ.

	Performance on gold trees							Performance on parsed trees					
	ID			Rel	Combo			ID			Combo		
	P	R	F1	Acc	P	R	F1	P	R	F1	P	R	F1
WSJ(full)	92.0	82.9	87.2	95.0	89.6	80.1	84.6	34.5	47.6	40.0	17.8	24.3	20.5
WSJ(sm)	92.3	79.5	85.5	93.3	90.4	77.2	83.2	38.0	47.3	42.1	19.7	24.3	21.7
NEGRA	73.9	64.6	69.0	85.1	63.3	55.4	59.1	48.3	39.7	43.6	20.9	17.2	18.9

Table 4: Cross-linguistic comparison of dislocated node identification and remapping. ID is correct identification of nodes as +/– dislocated; Rel is relocation of node to correct mother given gold-standard data on which nodes are dislocated (only applicable for gold trees); Combo is both correct identification and remapping.

by syntactic category of the mother node.[12] In Figure 1, for example, *to* would be an ADJP dependent of *quick* rather than a VP dependent of *was*; and *Farmers* would be an S dependent both of *to* in *to point out* ... and of *was*. We use the head-finding rules of Collins (1999) to lexicalize trees, and assume that null complementizers do not participate in dependency relations. To further compare the results of our algorithm with previous work, we obtained the output trees produced by Johnson (2002) and Dienes (2003) and evaluated them on typed dependency performance. Table 3 shows the results of this evaluation. For comparison, we include shallow dependency accuracy for Charniak's parser under P_{CF}.

4.3 Cross-linguistic comparison

In order to compare the results of nonlocal dependency reconstruction between languages, we must identify equivalence classes of nonlocal dependency annotation between treebanks. NEGRA's nonlocal dependency annotation is quite different from WSJ, as described in Section 2, ignoring controlled and arbitrary unexpressed subjects. The natural basis of comparison is therefore the set of all nonlocal NEGRA annotations against all WSJ dislocations, excluding relativizations (defined simply as dislocated *wh*- constituents under SBAR).[13]

Table 4 shows the performance comparison between WSJ and NEGRA of IDENTDISLOC and RELOCMOVED, on sentences of 40 tokens or less. For this evaluation metric we use syntactic category and left & right edges of (1) dislocated nodes (ID); and (2) originating mother node to which dislocated node is mapped (Rel). Combo requires both (1) and (2) to be correct. NEGRA is smaller than WSJ ($\sim$350,000 words vs. 1 million), so for fair comparison we tested WSJ using the smaller training set described in Section 2, comparable in size to NEGRA's. Since the positioning of traces within NEGRA nodes is trivial, we evaluate remapping and combination performances requiring only proper selection of the originating mother node; thus we carry the algorithm out on both treebanks through step (2b). This is adequate for purposes of our typed dependency evaluation in Section 4.2, since typed dependencies do not depend on positional information. State-of-the-art statistical parsing is far better on WSJ (Charniak, 2000) than on NEGRA (Dubey and Keller, 2003), so for comparison of parser-composed dependency performance we used vanilla PCFG models for both WSJ and NEGRA trained on comparably-sized datasets; in addition to making similar types of independence assumptions, these models performed relatively comparably on labeled bracketing measures for our development sets (73.2% performance for WSJ versus 70.9% for NEGRA).

Table 5 compares the testset performance of algorithms on the two treebanks on the typed dependency measure introduced in Section 4.2.[14]

5 Discussion

The WSJ results shown in Tables 2 and 3 suggest that discriminative models incorporating both nonlocal and local lexical and syntactic information can achieve good results on the task of non-local dependency identification. On the PARSEVAL metric, our algorithm performed particularly well on null complementizer and control locus insertion, and on S node relocation. In particular, Johnson noted that the proper insertion of control loci was a difficult issue involving lexical as well as structural sensitivity. We found the loglinear paradigm a good one in which to model this feature combination; when run in isolation on gold-standard development trees, our model reached 96.4% F1 on control locus insertion, reducing error over the Johnson model's 89.3%

[12]Unfortunately, 46 WSJ dislocation annotations in this testset involve dislocated nodes dominating their origin sites. It is not entirely clear how to interpret the intended semantics of these examples, so we ignore them in evaluation.

[13]The interpretation of comparative results must be modulated by the fact that more total time was spent on feature engineering for WSJ than for NEGRA, and the first author, who engineered the NEGRA feature set, is not a native speaker of German.

[14]Many head-dependent relations in NEGRA are explicitly marked, but for those that are not we used a Collins (1999)-style head-finding algorithm independently developed for German PCFG parsing.

	P_{CF}	P	$A \circ P$	G	$A \circ G$
WSJ(full)	76.3	75.4	75.7	98.7	99.7
WSJ(sm)	76.3	75.4	75.7	98.7	99.6
NEGRA	62.0	59.3	61.0	90.9	93.6

Table 5: Typed dependency F1 performance when composed with statistical parser. Remapped dependencies involve only non-relativization dislocations and exclude control loci.

by nearly two-thirds. The performance of our algorithm is also evident in the substantial contribution to typed dependency accuracy seen in Table 3. For gold-standard input trees, our algorithm reduces error by over 80% from the surface-dependency baseline, and over 60% compared with Johnson's results. For parsed input trees, our algorithm reduces dependency error by 23% over the baseline, and by 5% compared with Johnson's results. Note that the dependency figures of Dienes lag behind even the parsed results for Johnson's model; this may well be due to the fact that Dienes built his model as an extension of Collins (1999), which lags behind Charniak (2000) by about 1.3-1.5%.

Manual investigation of errors on English gold-standard data revealed two major issues that suggest further potential for improvement in performance without further increase in algorithmic complexity or training set size. First, we noted that annotation inconsistency accounted for a large number of errors, particularly false positives. VPs from which an S has been extracted (*[$_S$Shut up,] he [$_{VP}$ said* t*]*) are inconsistently given an empty SBAR daughter, suggesting the cross-model low-70's performance on null SBAR insertion models (see Table 2) may be a ceiling. Control loci were often under-annotated; the first five development-set false positive control loci we checked were all due to annotation error. And *why*-WHADVPs under SBAR, which are always dislocations, were not so annotated 20% of the time. Second, both control locus insertion and dislocated NP remapping must be sensitive to the presence of argument NPs under classified nodes. But temporal NPs, indistinguishable by gross category, also appear under such nodes, creating a major confound. We used customized features to compensate to some extent, but temporal annotation already exists in WSJ and could be used. We note that Klein and Manning (2003) independently found retention of temporal NP marking useful for PCFG parsing.

As can be seen in Table 3, the absolute improvement in dependency recovery is smaller for both our and Johnson's postprocessing algorithms when applied to parsed input trees than when applied to gold-standard input trees. It seems that this degradation is *not* primarily due to noise in parse tree outputs reducing recall of nonlocal dependency identification: precision/recall splits were largely the same between gold and parsed data, and manual inspection revealed that incorrect nonlocal dependency choices often arose from syntactically reasonable yet incorrect input from the parser. For example, the gold-standard parse *right-wing whites ... will* [$_{VP}$ *step up* [$_{NP}$ *their threats* [$_S$ [$_{VP}$ * *to take matters into their own hands*]]]] has an unindexed control locus because Treebank annotation specifies that infinitival VPs inside NPs are not assigned controllers. Charniak's parser, however, attaches the infinitival VP into the higher *step up* ... VP. Infinitival VPs inside VPs generally *do* receive controllers for their null subjects, and our algorithm accordingly yet mistakenly assigns *right-wing-whites* as the antecedent.

The English/German comparison shown in Tables 4 and 5 is suggestive, but caution is necessary in its interpretation due to the fact that differences in both language structure and treebank annotation may be involved. Results in the G column of Table 5, showing the accuracy of the context-free dependency approximation from gold-standard parse trees, quantitatively corroborates the intuition that nonlocal dependency is more prominent in German than in English.

Manual investigation of errors made on German gold-standard data revealed two major sources of error beyond sparsity. The first was a widespread ambiguity of S and VP nodes within S and VP nodes; many true dislocations of all sorts are expressed at the S and VP levels in CFG parse trees, such as VP-1 of Figure 2, but many adverbial and subordinate phrases of S or VP category are genuine dependents of the main clausal verb. We were able to find a number of features to distinguish some cases, such as the presence of certain unambiguous relative-clause introducing complementizers beginning an S node, but much ambiguity remained. The second was the ambiguity that some matrix S-initial NPs are actually dependents of the VP head (in these cases, NEGRA annotates the finite verb as the head of S and the non-finite verb as the head of VP). This is not necessarily a genuine discontinuity per se, but rather corresponds to identification of the subject NP in a clause. Obviously, having access to reliable case marking would improve performance in this area; such information is in fact included in NEGRA's morphological annotation, another argument for the utility of involving enhanced annotation in CF parsing.

As can be seen in the right half of Table 4, performance falls off considerably on vanilla PCFG-parsed data. This fall-off seems more dramatic than that seen in Sections 4.1 and 4.2, no doubt partly due to the poorer performance of the vanilla PCFG, but likely also because only non-relativization dislocations are considered in Section 4.3. These dislocations often require non-local information (such as identity of surface lexical governor) for identification and are thus especially susceptible to degradation in parsed data. Nevertheless, seemingly dismal performance here still provided a strong boost to typed dependency evaluation of parsed data, as seen in $A \circ P$ of Table 5. We suspect this indicates that dislocated terminals are being usefully identified and mapped back to their proper governors, even if the syntactic projections of these terminals and governors are not being correctly identified by the parser.

6 Further Work

Against the background of CFG as the standard approximation of dependency structure for broad-coverage parsing, there are essentially three options for the recovery of nonlocal dependency. The first option is to postprocess CF parse trees, which we have closely investigated in this paper. The second is to incorporate nonlocal dependency information into the *category* structure of CF trees. This was the approach taken by Dienes and Dubey (2003a,b) and Dienes (2003); it is also practiced in recent work on broad-coverage CCG parsing (Hockenmaier, 2003). The third would be to incorporate nonlocal dependency information into the *edge* structure parse trees, allowing discontinuous constituency to be explicitly represented in the parse chart. This approach was tentatively investigated by Plaehn (2000). As the syntactic diversity of languages for which treebanks are available grows, it will become increasingly important to compare these three approaches.

7 Acknowledgements

This work has benefited from feedback from Dan Jurafsky and three anonymous reviewers, and from presentation at the Institute of Cognitive Science, University of Colorado at Boulder. The authors are also grateful to Dan Klein and Jenny Finkel for use of maximum-entropy software they wrote. This work was supported in part by the Advanced Research and Development Activity (ARDA)'s Advanced Question Answering for Intelligence (AQUAINT) Program.

References

Charniak, E. (2000). A Maximum-Entropy-inspired parser. In *Proceedings of NAACL*.

Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on Information Theory*, 2(3):113–124.

Collins, M. (1999). *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania.

Dienes, P. (2003). *Statistical Parsing with Non-local Dependencies*. PhD thesis, Saarland University.

Dienes, P. and Dubey, A. (2003a). Antecedent recovery: Experiments with a trace tagger. In *Proceedings of EMNLP*.

Dienes, P. and Dubey, A. (2003b). Deep processing by combining shallow methods. In *Proceedings of ACL*.

Dubey, A. and Keller, F. (2003). Parsing German with sister-head dependencies. In *Proceedings of ACL*.

Gazdar, G., Klein, E., Pullum, G., and Sag, I. (1985). *Generalized Phrase Structure Grammar*. Harvard.

Gildea, D. and Jurafsky, D. (2002). Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Hockenmaier, J. (2003). *Data and models for Statistical Parsing with Combinatory Categorial Grammar*. PhD thesis, University of Edinburgh.

Johnson, M. (2002). A simple pattern-matching algorithm for recovering empty nodes and their antecedents. In *Proceedings of ACL*, volume 40.

Kaplan, R., Riezler, S., King, T. H., Maxwell, J. T., Vasserman, A., and Crouch, R. (2004). Speed and accuracy in shallow and deep stochastic parsing. In *Proceedings of NAACL*.

Kaplan, R. M. and Maxwell, J. T. (1993). The interface between phrasal and functional constraints. *Computational Linguistics*, 19(4):571–590.

Klein, D. and Manning, C. D. (2003). Accurate unlexicalized parsing. In *Proceedings of ACL*.

Kruijff, G.-J. (2002). Learning linearization rules from treebanks. Invited talk at the Formal Grammar'02/COLOGNET-ELSNET Symposium.

Levy, R. (2004). *Probabilistic Models of Syntactic Discontinuity*. PhD thesis, Stanford University. In progress.

Maxwell, J. T. and Manning, C. D. (1996). A theory of non-constituent coordination based on finite-state rules. In Butt, M. and King, T. H., editors, *Proceedings of LFG*.

Pasca, M. and Harabagiu, S. M. (2001). High performance question/answering. In *Proceedings of SIGIR*.

Plaehn, O. (2000). Computing the most probable parse for a discontinuous phrase structure grammar. In *Proceedings of IWPT*, Trento, Italy.

Riezler, S., King, T. H., Kaplan, R. M., Crouch, R. S., Maxwell, J. T., and Johnson, M. (2002). Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques. In *Proceedings of ACL*, pages 271–278.

Skut, W., Brants, T., Krenn, B., and Uszkoreit, H. (1997a). Annotating unrestricted German text. In *Fachtagung der Sektion Computerlinguistik der Deutschen Gesellschaft fr Sprachwissenschaft*, Heidelberg, Germany.

Skut, W., Krenn, B., Brants, T., and Uszkoreit, H. (1997b). An annotation scheme for free word order languages. In *Proceedings of ANLP*.