Rebanking CCGbank for improved NP interpretation

Matthew Honnibal and James R. Curran
School of Information Technologies
University of Sydney
NSW 2006, Australia
{mhonn,james}@it.usyd.edu.au

Johan Bos
University of Groningen
The Netherlands
bos@meaningfactory.com

Abstract

Once released, treebanks tend to remain unchanged despite any shortcomings in their depth of linguistic analysis or coverage of specific phenomena. Instead, separate resources are created to address such problems. In this paper we show how to improve the quality of a treebank, by integrating resources and implementing improved analyses for specific constructions.

We demonstrate this *rebanking* process by creating an updated version of CCGbank that includes the predicate-argument structure of both verbs and nouns, base-NP brackets, verb-particle constructions, and restrictive and non-restrictive nominal modifiers; and evaluate the impact of these changes on a statistical parser.

1 Introduction

Progress in natural language processing relies on direct comparison on shared data, discouraging improvements to the evaluation data. This means that we often spend years competing to reproduce partially incorrect annotations. It also encourages us to approach related problems as discrete tasks, when a new data set that adds deeper information establishes a new incompatible evaluation.

Direct comparison has been central to progress in statistical parsing, but it has also caused problems. Treebanking is a difficult engineering task: coverage, cost, consistency and granularity are all competing concerns that must be balanced against each other when the annotation scheme is developed. The difficulty of the task means that we ought to view treebanking as an ongoing process akin to grammar development, such as the many years of work on the ERG (Flickinger, 2000).

This paper demonstrates how a treebank can be *rebanked* to incorporate novel analyses and information from existing resources. We chose to work on CCGbank (Hockenmaier and Steedman, 2007), a Combinatory Categorial Grammar (Steedman, 2000) treebank acquired from the Penn Treebank (Marcus et al., 1993). This work is equally applicable to the corpora described by Miyao et al. (2004), Shen et al. (2008) or Cahill et al. (2008).

Our first changes integrate four previously suggested improvements to CCGbank. We then describe a novel CCG analysis of NP predicate-argument structure, which we implement using NomBank (Meyers et al., 2004). Our analysis allows the distinction between core and peripheral arguments to be represented for predicate nouns.

With this distinction, an entailment recognition system could recognise that *Google's acquisition of YouTube* entailed *Google acquired YouTube*, because equivalent predicate-argument structures are built for both. Our analysis also recovers non-local dependencies mediated by nominal predicates; for instance, *Google* is the agent of *acquire* in *Google's decision to acquire YouTube*.

The rebanked corpus extends CCGbank with:

1. NP brackets from Vadas and Curran (2008);
2. Restored and normalised punctuation;
3. Propbank-derived verb subcategorisation;
4. Verb particle structure drawn from Propbank;
5. Restrictive and non-restrictive adnominals;
6. Reanalyses to promote better head-finding;
7. Nombank-derived noun subcategorisation.

Together, these changes modify 30% of the labelled dependencies in CCGbank, demonstrating how multiple resources can be brought together in a single, richly annotated corpus. We then train and evaluate a parser for these changes, to investigate their impact on the accuracy of a state-of-the-art statistical CCG parser.

Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 207–215,
Uppsala, Sweden, 11-16 July 2010.

2 Background and motivation

Formalisms like HPSG (Pollard and Sag, 1994), LFG (Kaplan and Bresnan, 1982), and CCG (Steedman, 2000) are *linguistically motivated* in the sense that they attempt to explain and predict the limited variation found in the grammars of natural languages. They also attempt to specify how grammars construct semantic representations from surface strings, which is why they are sometimes referred to as *deep* grammars. Analyses produced by these formalisms can be more detailed than those produced by skeletal phrase-structure parsers, because they produce fully specified predicate-argument structures.

Unfortunately, statistical parsers do not take advantage of this potential detail. Statistical parsers induce their grammars from corpora, and the corpora for linguistically motivated formalisms currently do not contain high quality predicate-argument annotation, because they were derived from the Penn Treebank (PTB Marcus et al., 1993). Manually written grammars for these formalisms, such as the ERG HPSG grammar (Flickinger, 2000) and the XLE LFG grammar (Butt et al., 2006) produce far more detailed and linguistically correct analyses than any English statistical parser, due to the comparatively coarse-grained annotation schemes of the corpora statistical parsers are trained on. While rule-based parsers use grammars that are carefully engineered (e.g. Oepen et al., 2004), and can be updated to reflect the best linguistic analyses, statistical parsers have so far had to take what they are given.

What we suggest in this paper is that a treebank's grammar need not last its lifetime. For a start, there have been many annotations of the PTB that add much of the extra information needed to produce very high quality analyses for a linguistically motivated grammar. There are also other transformations which can be made with no additional information. That is, sometimes the existing trees allow transformation rules to be written that improve the quality of the grammar.

Linguistic theories are constantly changing, which means that there is a substantial lag between what we (think we) understand of grammar and the annotations in our corpora. The grammar engineering process we describe, which we dub *rebanking*, is intended to reduce this gap, tightening the feedback loop between formal and computational linguistics.

2.1 Combinatory Categorial Grammar

Combinatory Categorial Grammar (CCG; Steedman, 2000) is a lexicalised grammar, which means that all grammatical dependencies are specified in the lexical entries and that the production of derivations is governed by a small set of rules.

Lexical categories are either atomic (S, NP, PP, N), or a functor consisting of a result, directional slash, and argument. For instance, *in* might head a PP-typed constituent with one NP-typed argument, written as PP/NP.

A category can have a functor as its result, so that a word can have a complex valency structure. For instance, a verb phrase is represented by the category $S\backslash NP$: it is a function from a leftward NP (a subject) to a sentence. A transitive verb requires an object to become a verb phrase, producing the category $(S\backslash NP)/NP$.

A CCG grammar consists of a small number of schematic rules, called combinators. CCG extends the basic application rules of pure categorial grammar with (generalised) composition rules and type raising. The most common rules are:

$$\begin{array}{rrcll} X/Y & Y & \Rightarrow & X & (>) \\ Y & X\backslash Y & \Rightarrow & X & (<) \\ X/Y & Y/Z & \Rightarrow & X/Z & (>\mathbf{B}) \\ Y\backslash Z & X\backslash Y & \Rightarrow & X\backslash Z & (<\mathbf{B}) \\ Y/Z & X\backslash Y & \Rightarrow & X/Z & (<\mathbf{B}_{\times}) \end{array}$$

CCGbank (Hockenmaier and Steedman, 2007) extends this compact set of combinatory rules with a set of type-changing rules, designed to strike a better balance between sparsity in the category set and ambiguity in the grammar. We mark type-changing rules **TC** in our derivations.

In wide-coverage descriptions, categories are generally modelled as typed-feature structures (Shieber, 1986), rather than atomic symbols. This allows the grammar to include a notion of headedness, and to unify under-specified features.

We occasionally must refer to these additional details, for which we employ the following notation. Features are annotated in square-brackets, e.g. $S[dcl]$. Head-finding indices are annotated on categories in subscripts, e.g. $(NP_y\backslash NP_y)/NP_z$. The index of the word the category is assigned to is left implicit. We will sometimes also annotate derivations with the heads of categories as they are being built, to help the reader keep track of what lexemes have been bound to which categories.

3 Combining CCGbank corrections

There have been a few papers describing corrections to CCGbank. We bring these corrections together for the first time, before building on them with our further changes.

3.1 Compound noun brackets

Compound noun phrases can nest inside each other, creating bracketing ambiguities:

(1) (crude oil) prices

(2) crude (oil prices)

The structure of such compound noun phrases is left underspecified in the Penn Treebank (PTB), because the annotation procedure involved stitching together partial parses produced by the Fidditch parser (Hindle, 1983), which produced flat brackets for these constructions. The bracketing decision was also a source of annotator disagreement (Bies et al., 1995).

When Hockenmaier and Steedman (2002) went to acquire a CCG treebank from the PTB, this posed a problem. There is no equivalent way to leave these structures under-specified in CCG, because derivations must be binary branching. They therefore employed a simple heuristic: assume all such structures branch to the right. Under this analysis, *crude oil* is not a constituent, producing an incorrect analysis as in (1).

Vadas and Curran (2007) addressed this by manually annotating all of the ambiguous noun phrases in the PTB, and went on to use this information to correct 20,409 dependencies (1.95%) in CCGbank (Vadas and Curran, 2008). Our changes build on this corrected corpus.

3.2 Punctuation corrections

The syntactic analysis of punctuation is notoriously difficult, and punctuation is not always treated consistently in the Penn Treebank (Bies et al., 1995). Hockenmaier (2003) determined that quotation marks were particularly problematic, and therefore removed them from CCGbank altogether. We use the process described by Tse and Curran (2008) to restore the quotation marks and shift commas so that they always attach to the constituent to their left. This allows a grammar rule to be removed, preventing a great deal of spurious ambiguity and improving the speed of the C&C parser (Clark and Curran, 2007) by 37%.

3.3 Verb predicate-argument corrections

Semantic role descriptions generally recognise a distinction between core arguments, whose role comes from a set specific to the predicate, and peripheral arguments, who have a role drawn from a small, generic set. This distinction is represented in the surface syntax in CCG, because the category of a verb must specify its argument structure. In (3) *as a director* is annotated as a complement; in (4) it is an adjunct:

(3) *He* *joined* *as a director*
NP $(S\backslash NP)/PP$ PP

(4) *He* *joined* *as a director*
NP $S\backslash NP$ $(S\backslash NP)\backslash(S\backslash NP)$

CCGbank contains noisy complement and adjunct distinctions, because they were drawn from PTB function labels which imperfectly represent the distinction. In our previous work we used Propbank (Palmer et al., 2005) to convert 1,543 complements to adjuncts and 13,256 adjuncts to complements (Honnibal and Curran, 2007). If a constituent such as *as a director* received an adjunct category, but was labelled as a core argument in Propbank, we changed it to a complement, using its head's part-of-speech tag to infer its constituent type. We performed the equivalent transformation to ensure all peripheral arguments of verbs were analysed as adjuncts.

3.4 Verb-particle constructions

Propbank also offers reliable annotation of verb-particle constructions. This was not available in the PTB, so Hockenmaier and Steedman (2007) annotated all intransitive prepositions as adjuncts:

(5) *He* *woke* *up*
NP $S\backslash NP$ $(S\backslash NP)\backslash(S\backslash NP)$

We follow Constable and Curran (2009) in exploiting the Propbank annotations to add verb-particle distinctions to CCGbank, by introducing a new atomic category PT for particles, and changing their status from adjuncts to complements:

(6) *He* *woke* *up*
NP $(S\backslash NP)/PT$ PT

This analysis could be improved by adding extra head-finding logic to the verbal category, to recognise the multi-word expression as the head.

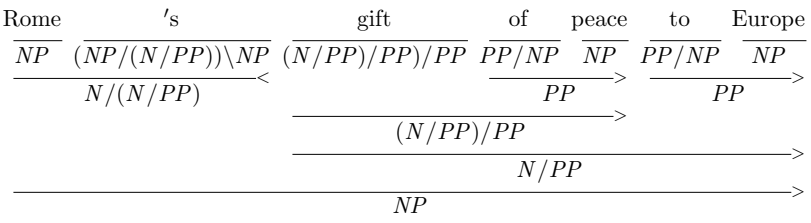


Figure 1: Deverbal noun predicate with agent, patient and beneficiary arguments.

4 Noun predicate-argument structure

Many common nouns in English can receive optional complements and adjuncts, realised by prepositional phrases, genitive determiners, compound nouns, relative clauses, and for some nouns, complementised clauses. For example, deverbal nouns generally have argument structures similar to the verbs they are derived from:

(7) Rome's destruction of Carthage

(8) Rome destroyed Carthage

The semantic roles of *Rome* and *Carthage* are the same in (7) and (8), but the noun cannot case-mark them directly, so *of* and the genitive clitic are pressed into service. The semantic role depends on both the predicate and subcategorisation frame:

(9) Carthage's$_{\mathrm{p}}$ destruction$_{\mathrm{Pred.}}$

(10) Rome's$_{\mathrm{a}}$ destruction$_{\mathrm{Pred.}}$ of Carthage$_{\mathrm{p}}$

(11) Rome's$_{\mathrm{a}}$ gift$_{\mathrm{Pred.}}$

(12) Rome's$_{\mathrm{a}}$ gift$_{\mathrm{Pred.}}$ of peace$_{\mathrm{p}}$ to Europe$_{\mathrm{b}}$

In (9), the genitive introduces the patient, but when the patient is supplied by the PP, it instead introduces the agent. The mapping differs for *gift*, where the genitive introduces the agent.

Peripheral arguments, which supply generically available modifiers of time, place, cause, quality etc, can be realised by pre- and post-modifiers:

(13) The portrait *in the Louvre*

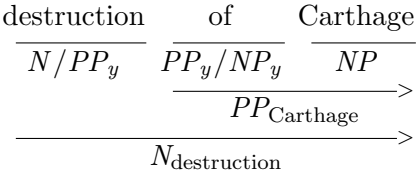
(14) The *fine* portrait

(15) *The Louvre's* portraits

These are distinct from core arguments because their interpretation does not depend on the predicate. The ambiguity can be seen in an NP such as *The nobleman's portrait*, where the genitive could mark possession (peripheral), or it could introduce the patient (core). The distinction between core and peripheral arguments is particularly difficult for compound nouns, as pre-modification is very productive in English.

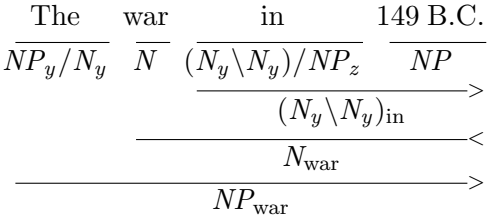
4.1 CCG analysis

We designed our analysis for transparency between the syntax and the predicate-argument structure, by stipulating that all and only the core arguments should be syntactic arguments of the predicate's category. This is fairly straightforward for arguments introduced by prepositions:

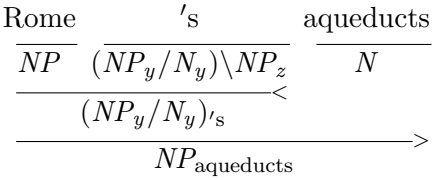


In our analysis, the head of *of Carthage* is *Carthage*, as *of* is assumed to be a semantically transparent case-marker. We apply this analysis to prepositional phrases that provide arguments to verbs as well — a departure from CCGbank.

Prepositional phrases that introduce peripheral arguments are analysed as syntactic adjuncts:



Adjunct prepositional phrases remain headed by the preposition, as it is the preposition's semantics that determines whether they function as temporal, causal, spatial etc. arguments. We follow Hockenmaier and Steedman (2007) in our analysis of genitives which realise peripheral arguments, such as the literal possessive:



Arguments introduced by possessives are a little trickier, because the genitive also functions as a determiner. We achieve this by having the noun subcategorise for the argument, which we type PP, and having the possessive subcategorise for the unsaturated noun to ultimately produce an NP:

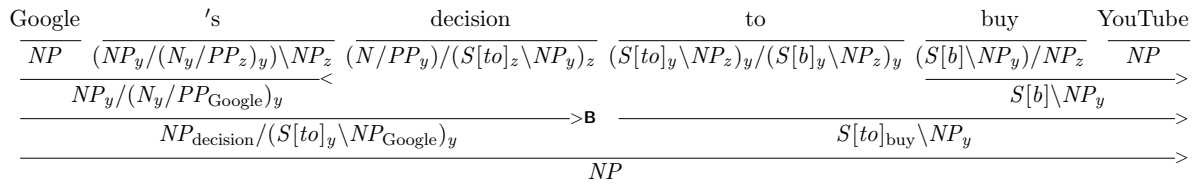
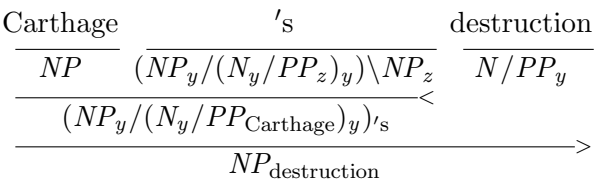


Figure 2: The coindexing on *decision*'s category allows the hard-to-reach agent of *buy* to be recovered. A non-normal form derivation is shown so that instantiated variables can be seen.



In this analysis, we regard the genitive clitic as a case-marker that performs a movement operation roughly analogous to WH-extraction. Its category is therefore similar to the one used in object extraction, $(N\backslash N)/(S/NP)$. Figure 1 shows an example with multiple core arguments.

This analysis allows recovery of verbal arguments of nominalised raising and control verbs, a construction which both Gildea and Hockenmaier (2003) and Boxwell and White (2008) identify as a problem case when aligning Propbank and CCGbank. Our analysis accommodates this construction effortlessly, as shown in Figure 2. The category assigned to *decision* can coindex the missing NP argument of *buy* with its own PP argument. When that argument is supplied by the genitive, it is also supplied to the verb, *buy*, filling its dependency with its agent, *Google*. This argument would be quite difficult to recover using a shallow syntactic analysis, as the path would be quite long. There are 494 such verb arguments mediated by nominal predicates in Sections 02-21.

These analyses allow us to draw complement/adjunct distinctions for nominal predicates, so that the surface syntax takes us very close to a full predicate-argument analysis. The only information we are not specifying in the syntactic analysis are the role labels assigned to each of the syntactic arguments. We could go further and express these labels in the syntax, producing categories like $(N/PP\{0\}_y)/PP\{1\}_z$ and $(N/PP\{1\}_y)/PP\{0\}_z$, but we expect that this would cause sparse data problems given the limited size of the corpus. This experiment would be an interesting subject of future work.

The only local core arguments that we do not annotate as syntactic complements are compound nouns, such as *decision makers*. We avoided these arguments because of the productivity of noun-noun compounding in English, which makes these argument structures very difficult to recover.

We currently do not have an analysis that allows support verbs to supply noun arguments, so we do not recover any of the long-range dependency structures described by Meyers et al. (2004).

4.2 Implementation and statistics

Our analysis requires semantic role labels for each argument of the nominal predicates in the Penn Treebank — precisely what NomBank (Meyers et al., 2004) provides. We can therefore draw our distinctions using the process described in our previous work, Honnibal and Curran (2007).

NomBank follows the same format as Propbank, so the procedure is exactly the same. First, we align CCGbank and the Penn Treebank, and produce a version of NomBank that refers to CCGbank nodes. We then assume that any prepositional phrase or genitive determiner annotated as a core argument in NomBank should be analysed as a complement, while peripheral arguments and adnominals that receive no semantic role label at all are analysed as adjuncts.

We converted 34,345 adnominal prepositional phrases to complements, leaving 18,919 as adjuncts. The most common preposition converted was *of*, which was labelled as a core argument 99.1% of the 19,283 times it occurred as an adnominal. The most common adjunct preposition was *in*, which realised a peripheral argument in 59.1% of its 7,725 occurrences.

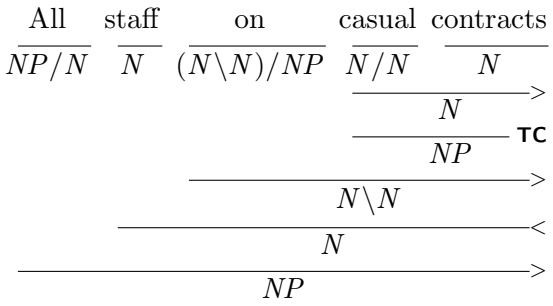
The frequent prepositions were more skewed towards core arguments. 73% of the occurrences of the 5 most frequent prepositions (*of*, *in*, *for*, *on* and *to*) realised peripheral arguments, compared with 53% for other prepositions.

Core arguments were also more common than peripheral arguments for possessives. There are 20,250 possessives in the corpus, of which 75% were converted to complements. The percentage was similar for both personal pronouns (such as *his*) and genitive phrases (such as *the boy's*).

5 Adding restrictivity distinctions

Adnominals can have either a restrictive or a non-restrictive (appositional) interpretation, determining the potential reference of the noun phrase it modifies. This ambiguity manifests itself in whether prepositional phrases, relative clauses and other adnominals are analysed as modifiers of either N or NP, yielding a restrictive or non-restrictive interpretation respectively.

In CCGbank, all adnominals attach to NPs, producing non-restrictive interpretations. We therefore move restrictive adnominals to N nodes:



This corrects the previous interpretation, which stated that there were no permanent staff.

5.1 Implementation and statistics

The Wall Street Journal's style guide mandates that this attachment ambiguity be managed by bracketing non-restrictive relatives with commas (Martin, 2002, p. 82), as in *casual staff, who have no health insurance, support it*. We thus use punctuation to make the attachment decision.

All $NP\backslash NP$ modifiers that are not preceded by punctuation were moved to the lowest N node possible and relabelled $N\backslash N$. We select the lowest (i.e. closest to leaf) N node because some adjectives, such as *present* or *former*, require scope over the qualified noun, making it safer to attach the adnominal first.

Some adnominals in CCGbank are created by the $S\backslash NP \rightarrow NP\backslash NP$ unary type-changing rule, which transforms reduced relative clauses. We introduce a $S\backslash NP \rightarrow N\backslash N$ in its place, and add a binary rule cued by punctuation to handle the relatively rare non-restrictive reduced relative clauses.

The rebanked corpus contains 34,134 $N\backslash N$ restrictive modifiers, and 9,784 non-restrictive modifiers. Most (61%) of the non-restrictive modifiers were relative clauses.

6 Reanalysing partitive constructions

True partitive constructions consist of a quantifier (16), a cardinal (17) or demonstrative (18) applied to an NP via *of*. There are similar constructions headed by common nouns, as in (19):

(16) Some of us

(17) Four of our members

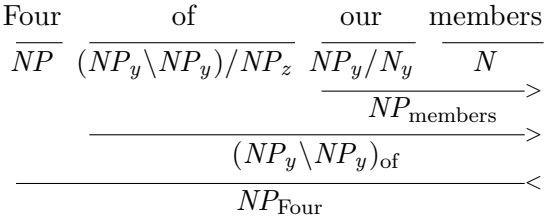
(18) Those of us who smoke

(19) A glass of wine

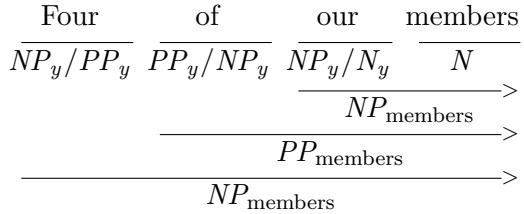
We regard the common noun partitives as headed by the initial noun, such as *glass*, because this noun usually controls the number agreement. We therefore analyse these cases as nouns with prepositional arguments. In (19), *glass* would be assigned the category N/PP.

True partitive constructions are different, however: they are always headed by the head of the NP supplied by *of*. The construction is quite common, because it provides a way to quantify or apply two different determiners.

Partitive constructions are not given special treatment in the PTB, and were analysed as noun phrases with a PP modifier in CCGbank:



This analysis does not yield the correct semantics, and may even hurt parser performance, because the head of the phrase is incorrectly assigned. We correct this with the following analysis, which takes the head from the NP argument of the PP:



The cardinal is given the category NP/PP, in analogy with the standard determiner category which is a function from a noun to a noun phrase (NP/N).

Corpus	L. DEPS	U. DEPS	CATS
+NP brackets	97.2	97.7	98.5
+Quotes	97.2	97.7	98.5
+Propbank	93.0	94.9	96.7
+Particles	92.5	94.8	96.2
+Restrictivity	79.5	94.4	90.6
+Part. Gen.	76.1	90.1	90.4
+NP Pred-Arg	70.6	83.3	84.8

Table 1: Effect of the changes on CCGbank, by percentage of dependencies and categories left unchanged in Section 00.

Corpus	CATS	Cats ≥ 10	CATS/WORD
CCGbank	1286	425	8.6
+NP brackets	1298	429	8.9
+Quotes	1300	431	8.8
+Propbank	1342	433	8.9
+Particles	1405	458	9.1
+Restrictivity	1447	471	9.3
+Part. Gen.	1455	474	9.5
+NP Pred-Arg	1574	511	10.1

Table 2: Effect of the changes on the size of the lexicon.

6.1 Implementation and Statistics

We detect this construction by identifying NPs post-modified by an *of* PP. The NP's head must either have the POS tag CD, or be one of the following words, determined through manual inspection of Sections 02-21:

> *all, another, average, both, each, another, any, anything, both, certain, each, either, enough, few, little, most, much, neither, nothing, other, part, plenty, several, some, something, that, those.*

Having identified the construction, we simply relabel the NP to NP/PP, and the $NP\backslash NP$ adnominal to PP. We identified and reanalysed 3,010 partitive genitives in CCGbank.

7 Similarity to CCGbank

Table 1 shows the percentage of labelled dependencies (L. Deps), unlabelled dependencies (U. Deps) and lexical categories (Cats) that remained the same after each set of changes.

A labelled dependency is a 4-tuple consisting of the head, the argument, the lexical category of the head, and the argument slot that the dependency fills. For instance, the subject fills slot 1 and the object fills slot 2 on the transitive verb category $(S\backslash NP)/NP$. There are more changes to labelled dependencies than lexical categories because one lexical category change alters all of the dependencies headed by a predicate, as they all depend on its lexical category. Unlabelled dependencies consist of only the head and argument.

The biggest changes were those described in Sections 4 and 5. After the addition of nominal predicate-argument structure, over 50% of the labelled dependencies were changed. Many of these changes involved changing an adjunct to a complement, which affects the unlabelled dependencies because the head and argument are inverted.

8 Lexicon statistics

Our changes make the grammar sensitive to new distinctions, which increases the number of lexical categories required. Table 2 shows the number of lexical categories (Cats), the number of lexical categories that occur at least 10 times in Sections 02-21 (Cats ≥ 10), and the average number of categories available for assignment to each token in Section 00 (Cats/Word). We followed Clark and Curran's (2007) process to determine the set of categories a word could receive, which includes a part-of-speech back-off for infrequent words.

The lexicon steadily grew with each set of changes, because each added information to the corpus. The addition of quotes only added two categories (LQU and RQU), and the addition of the quote tokens slightly decreased the average categories per word. The Propbank and verb-particle changes both introduced rare categories for complicated, infrequent argument structures.

The NP predicate-argument structure modifications added the most information. Head nouns were previously guaranteed the category N in CCGbank; possessive clitics always received the category $(NP/N)\backslash NP$; and possessive personal pronouns were always NP/N. Our changes introduce new categories for these frequent tokens, which meant a substantial increase in the number of possible categories per word.

9 Parsing Evaluation

Some of the changes we have made correct problems that have caused the performance of a statistical CCG parser to be over-estimated. Other changes introduce new distinctions, which a parser may or may not find difficult to reproduce. To investigate these issues, we trained and evaluated the C&C CCG parser on our rebanked corpora.

The experiments were set up as follows. We used the highest scoring configuration described by Clark and Curran (2007), the hybrid dependency model, using gold-standard POS tags. We followed Clark and Curran in excluding sentences that could not be parsed from the evaluation. All models obtained similar coverage, between 99.0 and 99.3%. The parser was evaluated using depen-

Corpus	WSJ 00			WSJ 23		
	LF	UF	CAT	LF	UF	CAT
CCGbank	87.2	92.9	94.1	87.7	93.0	94.4
+NP brackets	86.9	92.8	93.8	87.3	92.8	93.9
+Quotes	86.8	92.7	93.9	87.1	92.6	94.0
+Propbank	86.7	92.6	94.0	87.0	92.6	94.0
+Particles	86.4	92.5	93.8	86.8	92.6	93.8
All Rebanking	84.2	91.2	91.9	84.7	91.3	92.2

Table 3: Parser evaluation on the rebanked corpora.

Corpus	Rebanked		CCGbank	
	LF	UF	LF	UF
+NP brackets	86.45	92.36	86.52	92.35
+Quotes	86.57	92.40	86.52	92.35
+Propbank	87.76	92.96	87.74	92.99
+Particles	87.50	92.77	87.67	92.93
All Rebanking	87.23	92.71	88.02	93.51

Table 4: Comparison of parsers trained on CCGbank and the rebanked corpora, using dependencies that occur in both.

dencies generated from the gold-standard derivations (Boxwell, p.c., 2010).

Table 3 shows the accuracy of the parser on Sections 00 and 23. The parser scored slightly lower as the NP brackets, Quotes, Propbank and Particles corrections were added. This apparent decline in performance is at least partially an artefact of the evaluation. CCGbank contains some dependencies that are trivial to recover, because Hockenmaier and Steedman (2007) was forced to adopt a strictly right-branching analysis for NP brackets.

There was a larger drop in accuracy on the fully rebanked corpus, which included our analyses of restrictivity, partitive constructions and noun predicate-argument structure. This might also be explained by the evaluation, as the rebanked corpus includes much more fine-grained distinctions. The labelled dependencies evaluation is particularly sensitive to this, as a single category change affects multiple dependencies. This can be seen in the smaller gap in category accuracy.

We investigated whether the differences in performance were due to the different evaluation data by comparing the parsers' performance against the original parser on the dependencies they agreed upon, to allow direct comparison. To do this, we extracted the CCGbank intersection of each corpus's Section 00 dependencies.

Table 4 compares the labelled and unlabelled recall of the rebanked parsers we trained against the CCGbank parser on these intersections. Note that each row refers to a different intersection, so results are not comparable between rows. This comparison shows that the declines in accuracy seen in Table 3 were largely confined to the corrected dependencies. The parser's performance remained fairly stable on the dependencies left unchanged.

The rebanked parser performed 0.8% worse than the CCGbank parser on the intersection dependencies, suggesting that the fine-grained distinctions we introduced did cause some sparse data problems. However, we did not change any of the parser's maximum entropy features or hyperparameters, which are tuned for CCGbank.

10 Conclusion

Research in natural language understanding is driven by the datasets that we have available. The most cited computational linguistics work to date is the Penn Treebank (Marcus et al., 1993)[1]. Propbank (Palmer et al., 2005) has also been very influential since its release, and NomBank has been used for semantic dependency parsing in the CoNLL 2008 and 2009 shared tasks.

This paper has described how these resources can be jointly exploited using a linguistically motivated theory of syntax and semantics. The semantic annotations provided by Propbank and NomBank allowed us to build a corpus that takes much greater advantage of the semantic transparency of a deep grammar, using careful analyses and phenomenon-specific conversion rules.

The major areas of CCGbank's grammar left to be improved are the analysis of comparatives, and the analysis of named entities. English comparatives are diverse and difficult to analyse. Even the XTAG grammar (Doran et al., 1994), which deals with the major constructions of English in enviable detail, does not offer a full analysis of these phenomena. Named entities are also difficult to analyse, as many entity types obey their own specific grammars. This is another example of a phenomenon that could be analysed much better in CCGbank using an existing resource, the BBN named entity corpus.

Our rebanking has substantially improved CCGbank, by increasing the granularity and linguistic fidelity of its analyses. We achieved this by exploiting existing resources and crafting novel analyses. The process we have demonstrated can be used to train a parser that returns dependencies that abstract away as much surface syntactic variation as possible — including, now, even whether the predicate and arguments are expressed in a noun phrase or a full clause.

[1] http://clair.si.umich.edu/clair/anthology/rankings.cgi

Acknowledgments

James Curran was supported by Australian Research Council Discovery grant DP1097291 and the Capital Markets Cooperative Research Centre.

The parsing evaluation for this paper would have been much more difficult without the assistance of Stephen Boxwell, who helped generate the gold-standard dependencies with his software.

We are also grateful to the members of the CCG technicians mailing list for their help crafting the analyses, particularly Michael White, Mark Steedman and Dennis Mehay.

References

Ann Bies, Mark Ferguson, Karen Katz, and Robert MacIntyre. 1995. Bracketing guidelines for Treebank II style Penn Treebank project. Technical report, MS-CIS-95-06, University of Pennsylvania, Philadelphia, PA, USA.

Stephen Boxwell and Michael White. 2008. Projecting propbank roles onto the CCGbank. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, pages 3112–3117. European Language Resources Association (ELRA), Marrakech, Morocco.

Miriam Butt, Mary Dalrymple, and Tracy H. King, editors. 2006. *Lexical Semantics in LFG*. CSLI Publications, Stanford, CA.

Aoife Cahill, Michael Burke, Ruth O'Donovan, Stefan Riezler, Josef van Genabith, and Andy Way. 2008. Wide-coverage deep statistical parsing using automatic dependency structure annotation. *Computational Linguistics*, 34(1):81–124.

Stephen Clark and James R. Curran. 2007. Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, 33(4):493–552.

James Constable and James Curran. 2009. Integrating verb-particle constructions into CCG parsing. In *Proceedings of the Australasian Language Technology Association Workshop 2009*, pages 114–118. Sydney, Australia.

Christy Doran, Dania Egedi, Beth Ann Hockey, B. Srinivas, and Martin Zaidel. 1994. Xtag system: a wide coverage grammar for english. In *Proceedings of the 15th conference on Computational linguistics*, pages 922–928. ACL, Morristown, NJ, USA.

Dan Flickinger. 2000. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6(1):15–28.

Daniel Gildea and Julia Hockenmaier. 2003. Identifying semantic roles using combinatory categorial grammar. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 57–64. ACL, Morristown, NJ, USA.

Donald Hindle. 1983. User manual for fidditch, a deterministic parser. Technical Memorandum 7590-142, Naval Research Laboratory.

Julia Hockenmaier. 2003. *Data and Models for Statistical Parsing with Combinatory Categorial Grammar*. Ph.D. thesis, University of Edinburgh, Edinburgh, UK.

Julia Hockenmaier and Mark Steedman. 2002. Acquiring compact lexicalized grammars from a cleaner treebank. In *Proceedings of the Third Conference on Language Resources and Evaluation Conference*, pages 1974–1981. Las Palmas, Spain.

Julia Hockenmaier and Mark Steedman. 2007. CCGbank: a corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396.

Matthew Honnibal and James R. Curran. 2007. Improving the complement/adjunct distinction in CCGBank. In *Proceedings of the Conference of the Pacific Association for Computational Linguistics*, pages 210–217. Melbourne, Australia.

Ronald M. Kaplan and Joan Bresnan. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In Joan Bresnan, editor, *The mental representation of grammatical relations*, pages 173–281. MIT Press, Cambridge, MA, USA.

Mitchell Marcus, Beatrice Santorini, and Mary Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Paul Martin. 2002. *The Wall Street Journal Guide to Business Style and Usage*. Free Press, New York.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. The NomBank project: An interim report. In *Frontiers in Corpus Annotation: Proceedings of the Workshop*, pages 24–31. Boston, MA, USA.

Yusuke Miyao, Takashi Ninomiya, and Jun'ichi Tsujii. 2004. Corpus-oriented grammar development for acquiring a head-driven phrase structure grammar from the Penn Treebank. In *Proceedings of the First International Joint Conference on Natural Language Processing (IJCNLP-04)*, pages 684–693. Hainan Island, China.

Stepan Oepen, Daniel Flickenger, Kristina Toutanova, and Christopher D. Manning. 2004. LinGO Redwoods. a rich and dynamic treebank for HPSG. *Research on Language and Computation*, 2(4):575–596.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

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

Libin Shen, Lucas Champollion, and Aravind K. Joshi. 2008. LTAG-spinal and the treebank: A new resource for incremental, dependency and semantic parsing. *Language Resources and Evaluation*, 42(1):1–19.

Stuart M. Shieber. 1986. *An Introduction to Unification-Based Approaches to Grammar*, volume 4 of *CSLI Lecture Notes*. CSLI Publications, Stanford, CA.

Mark Steedman. 2000. *The Syntactic Process*. The MIT Press, Cambridge, MA, USA.

Daniel Tse and James R. Curran. 2008. Punctuation normalisation for cleaner treebanks and parsers. In *Proceedings of the Australian Language Technology Workshop*, volume 6, pages 151–159. ALTW, Hobart, Australia.

David Vadas and James Curran. 2007. Adding noun phrase structure to the Penn Treebank. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 240–247. ACL, Prague, Czech Republic.

David Vadas and James R. Curran. 2008. Parsing noun phrase structure with CCG. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 335–343. ACL, Columbus, Ohio, USA.